



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

EMBRASD

**ANDROID MOBILE APPLICATION DESIGNED FOR CHILDREN WITH A UTISM
AS A SUPPLEMENTAL TOOL FOR THERAPY**

**A Thesis Presented to the Faculty of the College of Information Technology
of Southville International School and Colleges, Las Piñas City**

**In Partial Fulfillment of the Course Requirement of the Degree
in Bachelor of Science in Information Technology**

Submitted by:

MARK LIGOT

APRIL 2017



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740
Bachelor of Science in Information Technology

i

APPROVAL SHEET

This thesis, entitled, *EmbrASD Android Mobile Application Designed for Children with Autism as a Supplemental Tool for Therapy*, prepared and submitted by Mark Ligot, in partial fulfillment of the requirements for the degree Bachelor of Science in Information Technology, has been examined and is recommended for acceptance and approval for oral defense.

Aris E. Ignacio, DIT, MCP, MCE
Thesis Adviser

Committee on Oral Defense

Alfred S. Libunao, MTA
Member

Armando C. Mendoza, MBA
Member

Fulgencio D. Ducut, MCE, MTA
Chair

Accepted and approved in partial fulfillment of the requirements for the degree Bachelor of Science in Information Technology.

Aris E. Ignacio, DIT, MCP, MCE
Dean / Department Chairperson

Marjorie S. Gutierrez-Tangog, Ed.D.
Vice-President for Academics and Research



ACKNOWLEDGMENTS

I would like to thank first and foremost God, who gave me the patience to follow through and finish this thesis. I would like to also thank my mom, Connie Almagro, for financing my college education. Also my mother-in-law, Susan Mendiola for financially supporting my son, Aaron Ligot, for his therapy sessions and my father, Arnel Ligot for helping financially the Special Education school of my son. My wife, Ana Ligot, is also instrumental in this because she was the one who took care my son Aaron and daughter, Amira Ligot, who helps taking care of Aaron. My sister also helped in doing voice over in the application. Her talents are also needed to complete this thesis. Also thank you to my cousin, Ryan Ligot, who designed the logo of EmbrASD.

Also would like to thank the therapists of my son, who are instrumental in giving me insights in how to design this application. Lastly my son because of him, I am now more aware of the challenges faced as a parent. I hope this application helps him and other parents and children with autism.



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

iii

ABSTRACT

This research aimed to provide a comprehensive and supplemental tool for therapy which is designed to be used by children with autism. The android market is lacking such an application that would address the core pathologies of autism. The goal of this research is to develop a multi-dimensional approach application that targets the core pathologies of children with autism. It is designed from the perspective of different therapists; more specifically speech therapist, occupational therapist, SPED teacher and an ABA therapist. Each of their targeted goals are comprehensively included into the design of the application.



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

Table of Contents

TITLE PAGE	
APPROVAL SHEET	i
ACKNOWLEDGMENTS	ii
ABSTRACT	iii
LIST OF FIGURES	iv
LIST OF TABLES	v
Chapter 1	1
THE PROBLEM AND ITS BACKGROUND	1
Introduction.....	1
Background of the Study	2
Problem Area	3
Research Objectives.....	5
<i>General Objective</i>	5
<i>Specific Objectives</i>	5
Significance of the Study	5
Scope and Limitations of the Study	6
REVIEW OF RELATED LITERATURE	8
Manual Approach of Therapy.....	8
State of Application for Children with Autism (Past, Present, Future)	9
Role of Application for Children with Autism.....	10
Synthesis	11
Theoretical / Conceptual Framework	12
Definition of Terms	12
Chapter 2	15
SYSTEM METHODOLOGY	15
Description of the Proposed System.....	15
<i>Physical Design</i>	16
1. Entity Relationship Diagram.....	16
2. Database Schema	16



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

ii

3. Data Dictionary	17
<i>Logical Design</i>	19
1. <i>Narrative Description of the proposed system</i>	19
2. <i>Dataflow Diagram of the proposed system</i>	19
Context Diagram	20
DFD 2.1 Level 0	20
DFD 2.2 Level 1.1	21
3. <i>Process Specifications</i>	23
System Requirements	25
1. <i>Testing</i>	25
2. <i>Resources</i>	25
2.1 <i>Developmental Resources</i>	25
2.2 <i>User Resources</i>	25
Installation	26
Training	26
Implementation	26
Chapter 3	27
Presentation and Analysis of System	27
Graphical User Interface	27
System Testing and Findings	40
Chapter 4	42
Summary, Conclusion and Recommendations	42
Summary	42
Conclusion	42
Recommendation	43
References	44
Appendices	45
A. Project Time Table	45
B. Turnitin Result	48
C. Request for Thesis Adviser	49
D. Consultation Record	50
E. Activities Checklist for Thesis Submission	51



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

iii

F. Recommendation for Oral Defense	52
G. Interview Notes Overview	53
H. Pictures during Thesis Defense	55
I. User Manual	56
J. Notes on the Author	58
K. Source Code	60



LIST OF FIGURES

Figure 1-Flow Chart	19
Figure 2-Context Diagram	20
Figure 3-DFD 2.1 Level 0	20
Figure 4-DFD 2.2 Level 1.1	21
Figure 5-DFD 2.3 Level 1.2	21
Figure 6-DFD 2.4 Level 1.3	22
Figure 7-DFD 2.5 Level 1.4	22
Figure 8-DFD 2.6 Level 1.5	23
Figure 9-Colors and Shapes	27
Figure 10-Letters and Numbers	27
Figure 11-CVC and Sight Words	28
Figure 12-Abstract Words in Simple Sentences	28
Figure 13-Reading Comprehension	29
Figure 14-Colors and Shapes Lesson Screen	30
Figure 15-Colors and Shapes Audio Recording Screen	31
Figure 16-Letters and Numbers Lesson Screen	32
Figure 17-Letters and Numbers Recording Screen	33
Figure 18-CVC and Sight Words Screen	34
Figure 19-CVC and Sight Words Recording Screen	35
Figure 20-Abstract Words in Simple Sentences Screen	36
Figure 21-Abstract Words in Simple Sentences Recording Screen	37
Figure 22-Reading Comprehension Screen	38
Figure 23-Reading Comprehension Recording Screen	39



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

v

LIST OF TABLES

Table 1-Entity Relationship Diagram	16
Table 2-AnsTable	16
Table 3-ResTable	17
Table 4-Module4and5	17
Table 5-AnsTable	17
Table 6-ResTable	18
Table 7-Module4and5	18
Table 8-3.1 Process	23
Table 9-3.2 Process	24
Table 10-3.3 Process	24
Table 11-3.4 Process	24
Table 12-2.1 Developmental Resources	25



Chapter 1

THE PROBLEM AND ITS BACKGROUND

Introduction

There are many applications in the android market that are geared towards children with autism however these applications are not really designed from the perspective of the therapists. The games and activities that are offered in these applications do not have a cohesive structure and lacks a multi-dimensional approach. Children with autism are easily distracted and will tire of the games and activities in the application. Therapists know the techniques of how to teach these children how to enhance their vocabulary, to behave properly, to follow commands and also to read and write.



Background of the Study

Background

Autism has a wide spectrum that affects children differently. These symptoms will show in the child's social interaction, communication (both verbal and nonverbal) and lastly the child's unusual behaviors and interests. Since each case of autism is unique, the application should be tailored to the children's specific needs. There must be structured, multi-dimensional and well-defined learning objectives to maximize the learning benefit of children.

The application will be designed based on the perspective of a speech pathologist, an occupational therapist and a SPED teacher. Their inputs of the structure, specific goals and learning objectives will be essential to create an application that will be a supplemental tool for therapy.

The application will record the child's responses to the activities, so that progression will be monitored. This is crucial because children will only spend a fraction of their time in therapy, therefore therapists and parents can closely monitor the child's progression.

Essentially, the application will target the core areas affected by autism such as:

- Social skills
- Language and communication
- Imitation
- Cognitive Thinking



- Communication
- Reading
- Comprehension
- Motor skills

Problem Area

The problem with existing application in the Android Market

Although there are applications currently available that target the core pathologies that are caused by autism, there isn't one that is comprehensive that targets all pathology. Also the ones available are not multi-dimensional. There are also games that are not designed for autistic children but do have therapeutic benefits for the child.

Structure Problem

Currently, games and application will usually have only a one-dimensional approach in the game design. The games should have a specific target and goal that can be measured and quantified in order for the parent and therapist to gauge the progression of the child's learning.

Specifically, the applications on the market may target one pathology and do not combine the therapeutic techniques of a SPED, ABA, Speech and Occupational Therapists. Recording the child's responses to each lesson is also lacking as well. The time spent on the application should supplement the therapy the child gets in therapy sessions. The progression of difficulty should



be designed with certain prerequisites, the order of which is colors and shapes, letters and numbers, CVC and sight words, adjectives, simple sentences and lastly reading comprehension.

Lack of variety of games and activities.

There must be a wide variety of games as well as activities for parents to do in a single application. A child may grow tired of the application and no longer use it after some time if the game becomes monotonous. As the child master each skill, the next level of difficulty will incorporate the previous skills and present new compounded challenges that would require the previous skill and the new skill that is being presented.

Not designed by therapists

Since therapists can only treat autism, the therapists should have a direct involvement in the design of the application. With the wide spectrum of pathologies that are affected by autism, there must be inputs from those therapists specifically a speech pathologist, occupational therapist, as well as a SPED teacher. Their expertise will tremendously help the learning benefit of the child because the therapists can set the learning objectives when designing the application.



Research Objectives

General Objective

The general objective of this study is to create an application for children with autism that would comprehensively address all the core pathologies that are caused by autism and would act as a supplement to therapy.

Specific Objectives

The specific objectives of the study are the following:

- Create a software with multi-dimensional design to address all core pathologies
- Design compounding learning structure that would build on previous lessons
- Software will verbalize and record audio response for child for later playback analysis

Significance of the Study

The study pertains to the creation of a comprehensive application that would be a supplemental tool for therapy for children with autism. This study will be significant to the following:

- Parents – Parents with children with autism need every available tool that would help with children autism. This study would enable parents to have another tool to help them with their children with autism. It will also help them to be actively engaged with their children's therapy



- **Therapists** – Since the application is designed to be a supplement to therapy, it would promote a continuance of the therapy even after the therapy is over. In this way, there would be improvements in subsequent therapy sessions because the goals and objectives will be actively carried out in the child's home.
- **Children with autism** – This application will first and foremost benefit all children with autism. Although improvements will vary from child to child, there will be improvements nevertheless in the child's core pathologies.

Scope and Limitations of the Study

This study will only conduct coverage on the following:

Scope:

- **Children with Autism**
 - Autistic children varying in their spectrum have a different learning approach from normal children. The Application is designed for children ages 0 to 10.
- **Android Market**
 - According to the autismspeaks.org website, the Android Market is lacking these types of application, therefore the focus is developing an app in this market so that it can reach the untapped market share of Android.



- **Speech Pathology**
 - Communication is a very important skill in learning; as such their expertise is needed for learning.
- **Occupational Therapy / ABA (Applied Behavior Analysis) Therapy**
 - The fundamental learning hindrance for autism is behavioral; therefore, behavioral modification is needed before learning.
- **Teaching Techniques of SPED Teachers**
 - The fundamental learning structure for autism is vastly different from normal children; therefore, SPED teachers are essential to design lesson structures.
- **Android API (Application Program Interface) level 12 and above**
 - Currently API level 12 (Jellybean) has a 96% market penetration. Maintaining this API will ensure maximum backward compatibility for existing Android users.

Limitations:

- **iOS Market**
 - Since the entire program is written in JAVA, it is incompatible with the iOS environment, which uses Object C. Also there is a plethora of applications currently available in the iOS Market.
- **Teens, Adults with Autism and non-Autistic persons.**



- The approach of treating teens, adults and non-autistic persons is different from children with Autism. However there can still be benefits for those who uses this application.

Once this application is completed, this will eventually be ported to the iOS market so that more parents would benefit from the application.

REVIEW OF RELATED LITERATURE

Manual Approach of Therapy

The human interaction approach of therapy will never be replaced. ABA, Occupational therapists, Speech therapists, SPED therapists and SPED teachers are crucial to address children with autism. Each therapist has a specific expertise that target specific core pathology. Though the gravity of the pathology will vastly differ from each child with autism, the therapist will adjust the approach that would be appropriate for each child. Also certain skillset such as daily life skills such as brushing teeth, putting clothes on and writing cannot be taught on a digital level. Having a highly structured curriculum that addresses the core pathologies of a child is the most effective form of therapy (Ahearn, 2010).



State of Application for Children with Autism (Past, Present, Future)

In the past, the conventional therapy methods for autism are speech therapy, Applied Behavioral Analysis (ABA), and Occupational Therapy. These therapists target specific deficiencies of children with autism. For speech, speech therapists focus first on identifying sounds each letter of the alphabet makes and eventually combines these letters to make words. Then the words will be associated with people, places, things or animals and even action words such as eating or drinking. Eventually these concepts will be put together to make a sentence (Gusto, 2016). For ABA, their focus is how learning takes place. Positive reinforcement is such one principle where a positive behavior is rewarded so that the behavior is repeated. This technique fosters basic skills such as looking, listening and imitating as well as more complex skills such as reading, conversing and understanding another one's perspective (Salvacion, 2016). Occupational therapists observe to see if the children are able to do the tasks expected of their age. Occupational therapists will help the child respond better to their environment by doing physical activities, interact with other children, and daily living skills such as brushing teeth, washing hands, combing or tie shoe laces as well as behavior modification (Aguas, 2016). SPED teachers will have a different structured approach when teaching children with autism in a classroom setting (Riofrio, 2016). There is an order of how to learn and having a SPED therapist to tailor fit the lesson on a one



on one setting would be beneficial because the learning curve for each child varies (Ignacio, 2016).

Presently, these techniques are now supplemented with technology. There are applications that are used to help children with autism learn during and even outside therapy. There are studies that show certain application such as "Alpha Writer", have helped children with autism learn to read and write (Moore & Calvert, 2000), as well as lessen tantrums with web based tutorials on how to communicate more efficiently with their children (Kobak & Stone, 2011). Another application, "Image it" uses three images to combine an idea. This application helps children to communicate an idea or a sentence, a form of an Augmented Alternative Communication or AAC application used in conjunction with speech therapy.

The studies of Moore, Calvert, Kobak and Stone proves that in the future, more applications will be developed to help supplement therapy and help children with autism. There is a need for more comprehensive, multi-dimensional approach application that would greatly improve the lives of children with autism.

Role of Application for Children with Autism

There is a lack of Andriod based applications in the market that combines the different techniques of therapies. Even the applications in the iOS do not have a comprehensive approach to all the different facets and



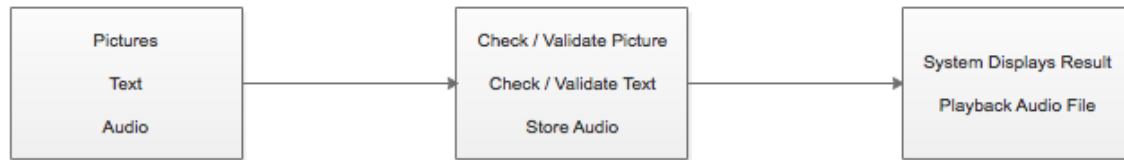
deficiencies of children with autisms, which are focused on only one or two aspects. The role of the application will be combining these techniques and making a comprehensive and interactive application for children with autism. The goal is to have a multi-dimensional approach by taking the perspective of the different therapy methods and combining them into one cohesive, structured application (William, Wright, Callaghan, & Coughlan, 2002).

Synthesis

There are numerous studies of how using applications as a medium for teaching increases the efficiency of learning for children with autism. If used as a supplemental tool for therapy, there can be tremendous progress in a child and thus the need for a multi-dimensional application wherein all the different therapeutic aspects are amalgamated into a cohesive structured application would greatly enhance the learning potential of a child with autism. Designing an application from the perspective of all different therapist will be the standard of how to construct an application for children with autism (Wong & Tam, 2001).



Theoretical / Conceptual Framework



Input

Process

Output

Figure 1.1: Conceptual Framework

The application will present the child with interactive pictures or text wherein the child can click, drag and drop the text or picture to its appropriate location within the screen. It will also record the child for every lesson. The application will then determine if the choice that was made is correct or incorrect and display the result as the output as well as store the recorded video of the child into the database

Definition of Terms

Term

Definition

Autism

a mental condition, present from early childhood, characterized by difficulty in communicating and forming relationships with other people and in using language and abstract concepts.



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

13

ABA (Applied Behavior Analysis) the application of the principles of learning and motivation from Applied Behavior Analysis, and the procedures and technology derived from those principles, to the solution of problems of social significance.

Occupational Therapy works to develop skills for handwriting, fine motor skills and daily living skills. However, the most essential role is also to assess and target the child's sensory processing disorders.

Speech Therapy specializes in the evaluation and treatment of communication disorders. This includes proper phonation, intonation, and variation of pitch as well as constructing sentences using language's grammar rules.

SPED Teacher/Therapy SPED teachers individually addresses students based on their



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

14

Android

special needs in a classroom setting
while SPED therapists are on a one
on one setting with the students.

mobile operating system developed
by Google whose source code is
under open source licenses.



Chapter 2

SYSTEM METHODOLOGY

Description of the Proposed System

The system will have five modules (lesson types) and selection cycle will be done through swipe gestures. The lesson generated are random as well as the choices. This would prevent the child to simply memorize answers. Also at the end of the lesson, the system will ask the child to verbalize the lesson and the audio of a child will be recorded.

- Colors and Shapes, Numbers and Letters, CVC and Sight Words, Abstract Words in Simple Sentences and Reading Comprehension

The child will interact with the objects by touching the objects and dragging them into the correct place specified by the program. It will have varying degree of difficulty as the child progresses in the game. At the end of each activity, the child may choose to progress to the next level or choose a different activity. Also the child will be recorded on every level to show progression.

- Progression

As the child progresses, the parent will be able to gauge how long the child plays on a certain activity by reviewing the audio clips of each completed level. The parent will then be able to gauge the child's areas of strengths and weaknesses. The results of these can be shared to therapist so then

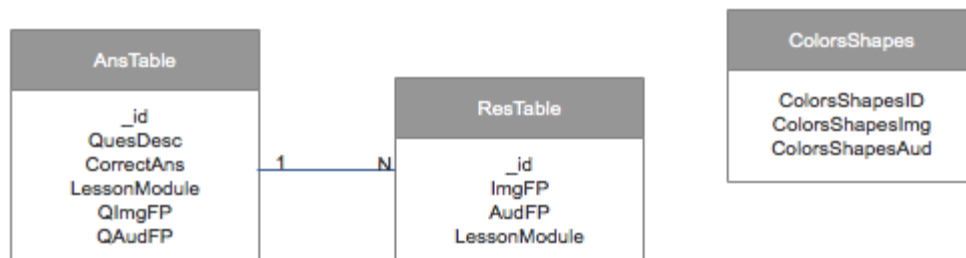


can gauge what areas of development needs to be addressed since the child's response is recorded at every lesson. Also every lesson and the selection of answer is randomized to ensure the child is not simply memorizing the answer.

Physical Design

1. Entity Relationship Diagram

Table 1-Entity Relationship Diagram



2. Database Schema

Table 2-AnsTable

AnsTable				
Field Name	Key	Data Type	Size	Integrity Control
id	PK	Int	3	Not Null
QuesDesc	N/A	VarChar	255	Not Null
CorrAns	N/A	Int	3	Not Null
LessonModule	N/A	Int	3	Not Null
QImgFP	N/A	VarChar	255	Not Null
QAudFP	N/A	VarChar	255	Not Null



Table 3-ResTable

ResTable				
Field Name	Key	Data Type	Size	Integrity Control
id	PK	Int	3	Not Null
ImgFP	N/A	VarChar	255	Not Null
AudFP	N/A	VarChar	255	Not Null
LessonModule	N/A	Int	3	Not Null

Table 4-Module4and5

Module4and5				
Field Name	Key	Data Type	Size	Integrity Control
id	PK	Int	3	Not Null
QuesDesc1	N/A	VarChar	255	Not Null
QuesDesc2	N/A	VarChar	255	Null
choiceA	N/A	VarChar	255	Not Null
choiceB	N/A	VarChar	255	Not Null
corrChoice	N/A	VarChar	255	Not Null
DescIMFP	N/A	VarChar	255	Not Null
QuesDescAUDFP	N/A	VarChar	255	Not Null
CorrAnsAUDFP	N/A	VarChar	255	Null
choiceAAUDFP	N/A	VarChar	255	Not Null
choiceBAUDFP	N/A	VarChar	255	Not Null
lessonmodule	PK	Int	3	Not Null
QuesDescAUDFP2	N/A	VarChar	255	Null
followupquesaud	N/A	VarChar	255	Null
followupquestex	N/A	VarChar	255	Null

3. Data Dictionary

Table 5-AnsTable

AnsTable	
Field Name	Description
id	ID of level
QuesDesc	Text of Question
CorrAns	ID of Correct Answer
LessonModule	Module Number
QImgFP	Image Filepath
QAudFP	Audio Filepath



Table 6-ResTable

ResTable	
Field Name	Description
id	ID of Resources
ImgFP	Image Filepath
AudFP	Audio Filepath
LessonModule	Module Number

Table 7-Module4and5

Module4and5	
Field Name	Description
id	ID of level
QuesDesc1	Text of Question Part 1
QuesDesc2	Text of Question Part 2
choiceA	Image Filepath of First Choice
choiceB	Image Filepath of Second Choice
corrChoice	Correct Choice Image Filepath
DescIMFP	Image Filepath of Question Description
QuesDescAUDFP	Audio Filepath of Question Description
CorrAnsAUDFP	Correct Answer Audio Filepath
choiceAAUDFP	Audio Filepath of First Choice
choiceBAUDFP	Audio Filepath of Second Choice
lessonmodule	Module Number
QuesDescAUDFP2	Followup Question Audio Filepath
followupquesaud	Followup Question Audio Filepath
followupquestex	Followup Question Text



Logical Design

1. Narrative Description of the proposed system

The design of this program will be based on the perspective of occupational therapists, speech pathologists, ABA specialists as well as SPED teachers. These therapists have specific goals and expertise that the therapists will accomplish during a therapy session. What this program will do is incorporate these skillsets into the games that the child will play. The child will have a selection of answer, then the application will validate the answer and display results. The child then will be recorded to verbalize the lesson that will be recorded in order to monitor the progress of the child. The program will be designed to be a supplemental tool for therapy for children with autism.

2. Dataflow Diagram of the proposed system

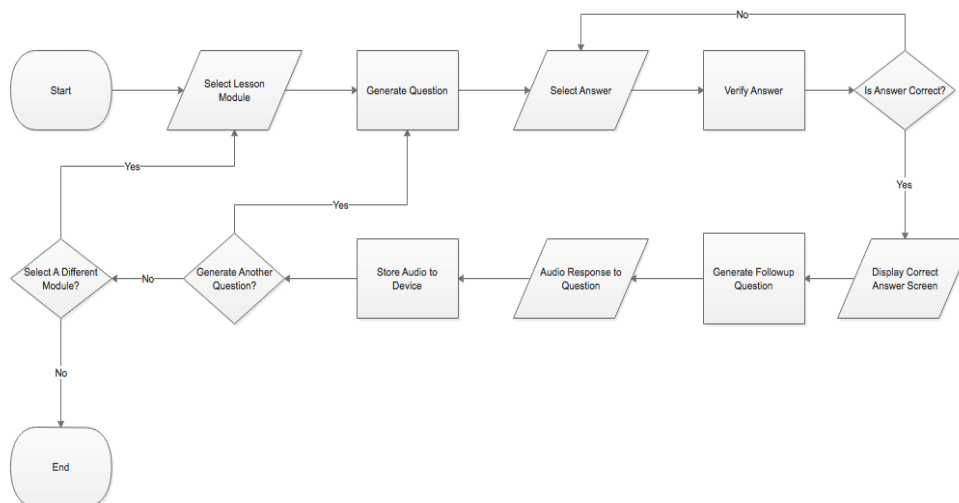


Figure 1-Flow Chart



Context Diagram

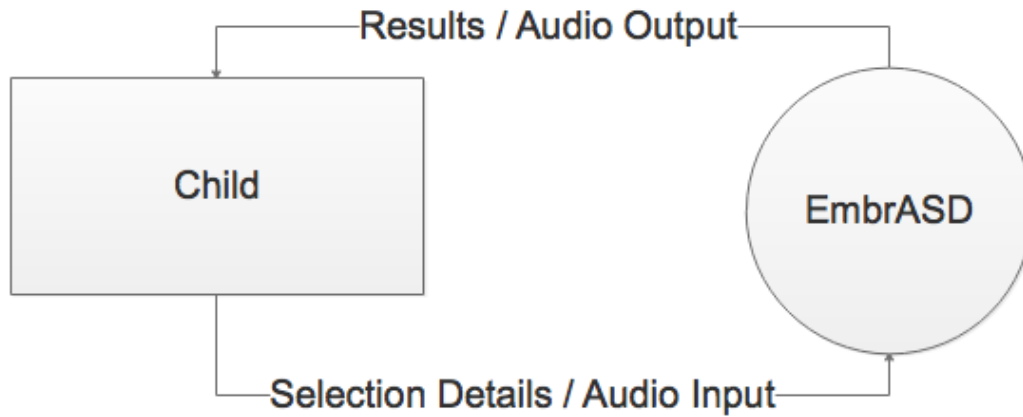
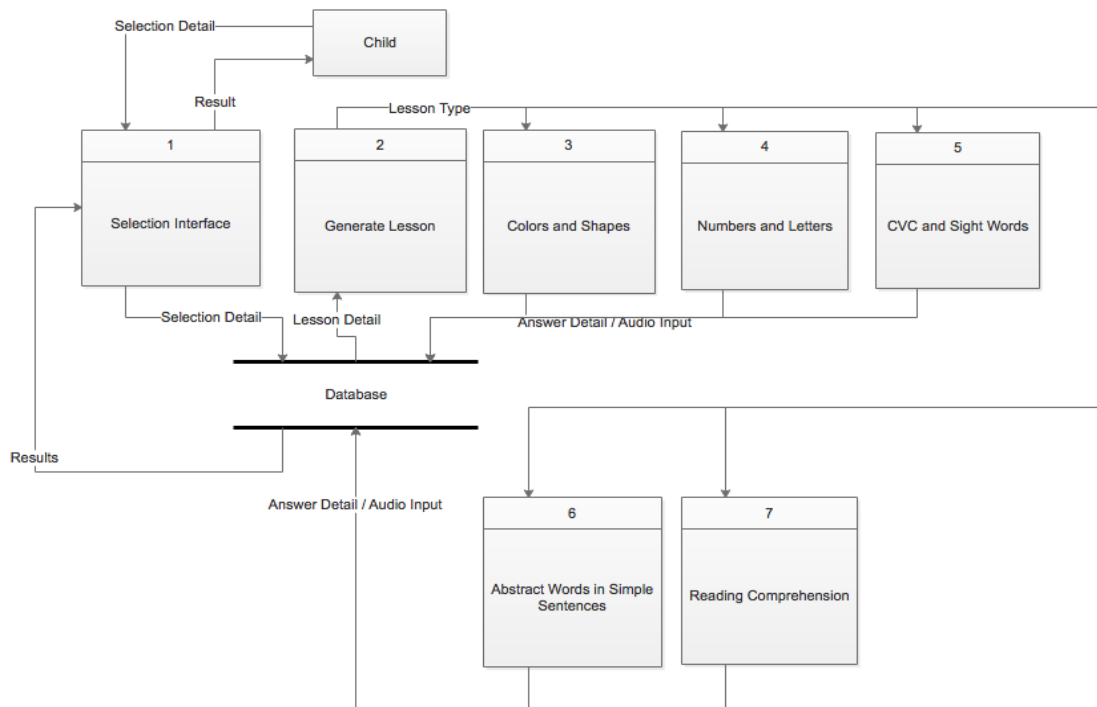


Figure 2-Context Diagram

DFD 2.1 Level 0

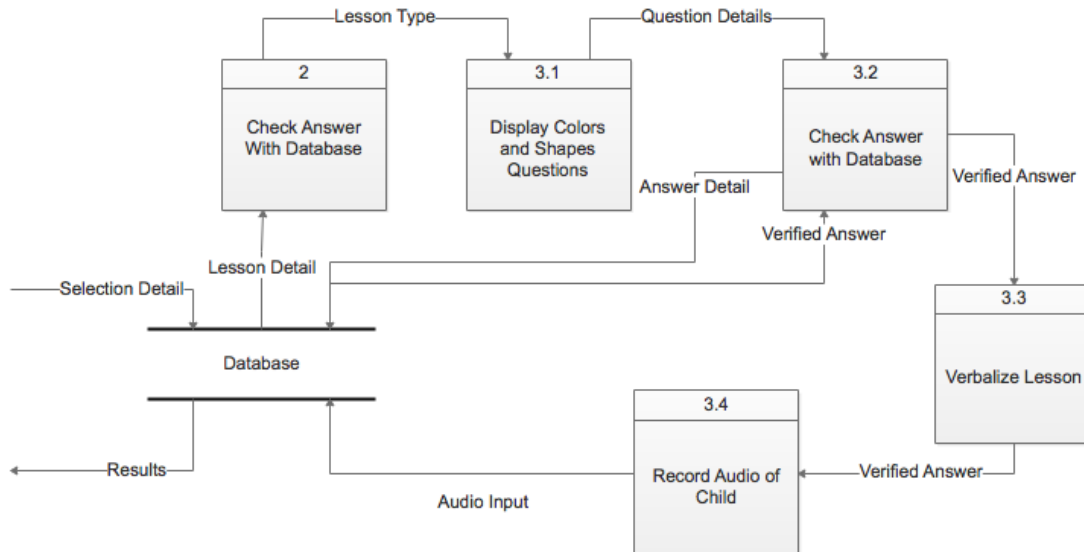
Figure 3-DFD 2.1 Level 0





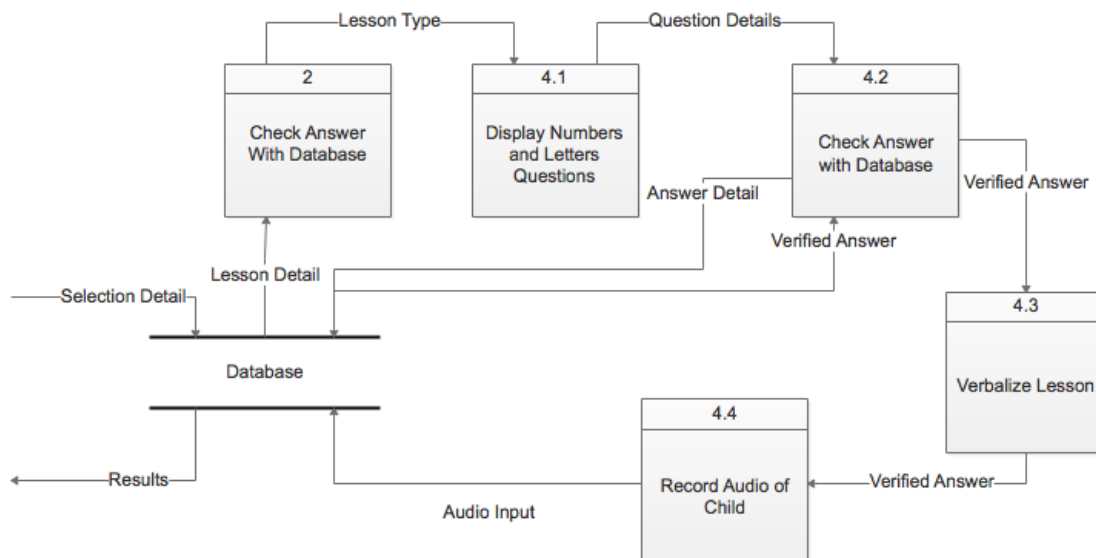
DFD 2.2 Level 1.1

Figure 4-DFD 2.2 Level 1.1



DFD 2.3 Level 1.2

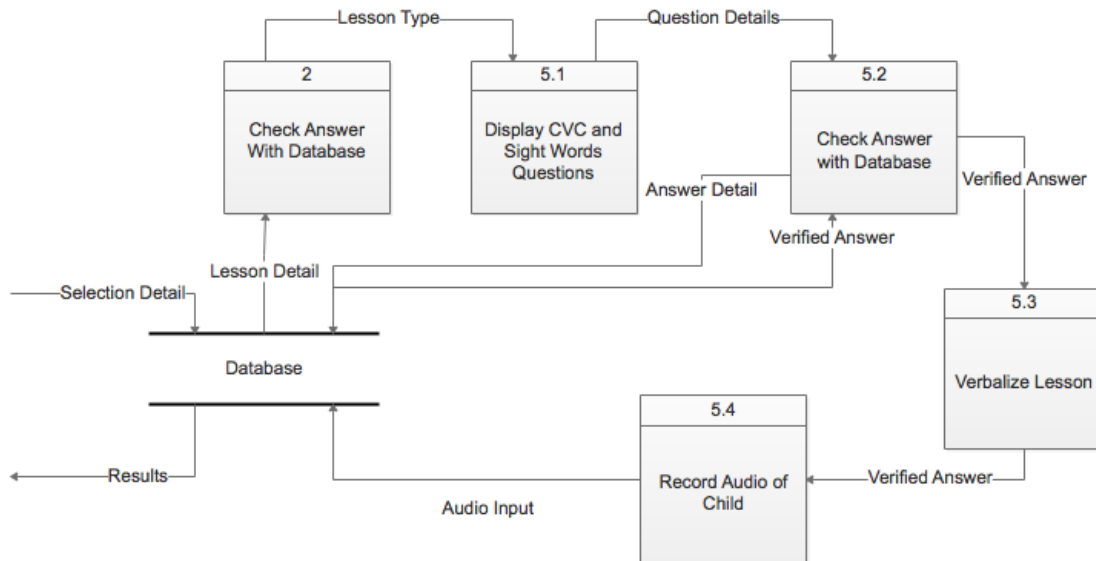
Figure 5-DFD 2.3 Level 1.2





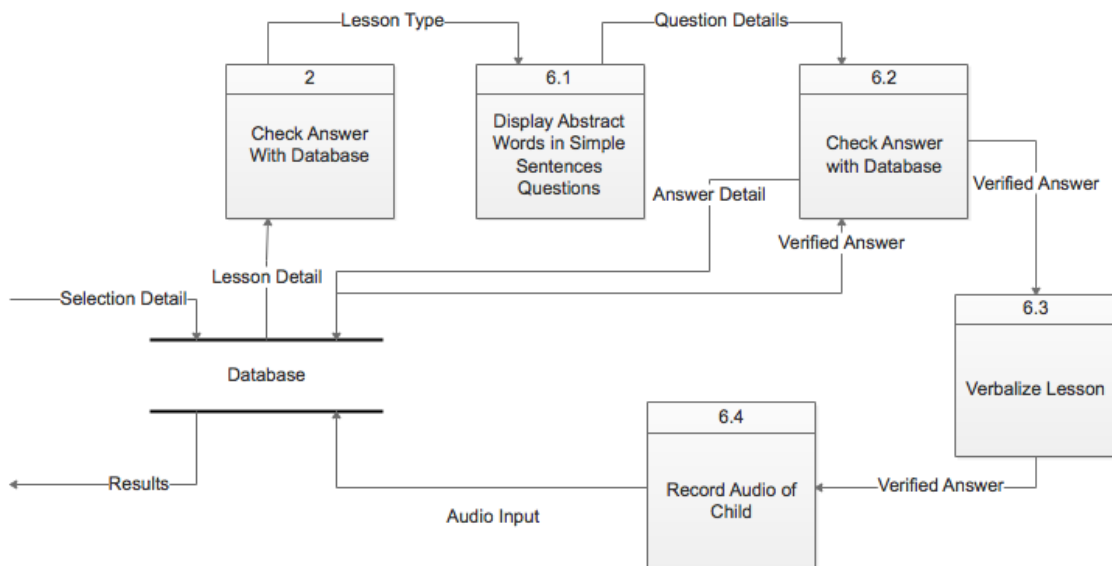
DFD 2.4 Level 1.3

Figure 6-DFD 2.4 Level 1.3



DFD 2.5 Level 1.4

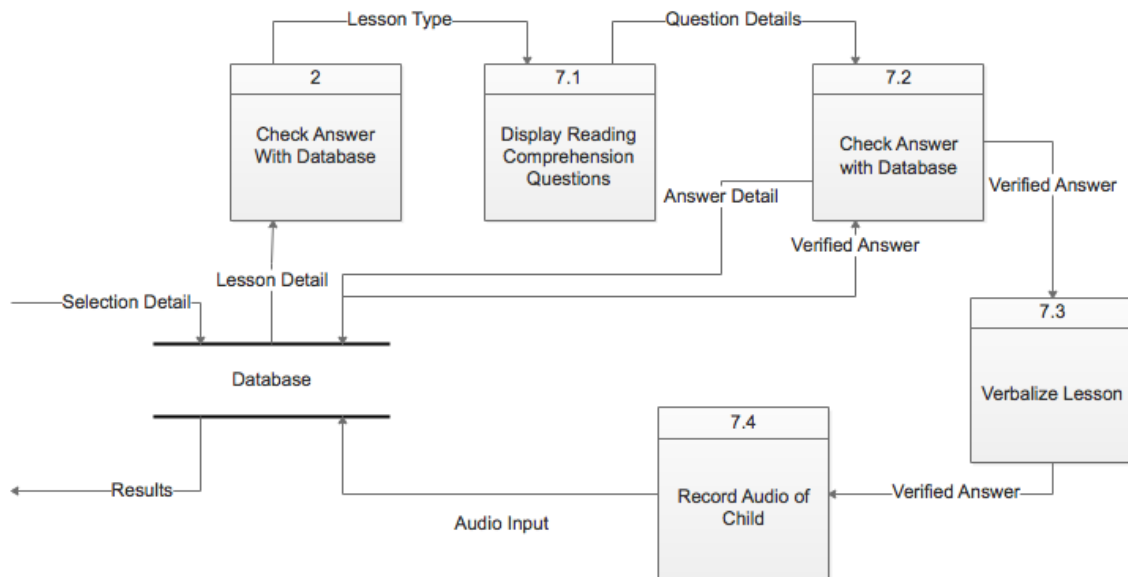
Figure 7-DFD 2.5 Level 1.4





DFD 2.6 Level 1.5

Figure 8-DFD 2.6 Level 1.5



3. Process Specifications

3.1 Process 1 Child Selects activity on Selection Interface

Table 8-3.1 Process

Input	Output
Lesson Module	Lesson Type
Process: Selection Interface is a swipe menu selection. The child taps on the box after selecting Lesson Module	



3.2 Process 2 Generate Lesson: System Generates Lesson

Table 9-3.2 Process

Input	Output
Lesson Module	Display Questions
Process: Based on the selected module, the system will generate a question in the database, which are randomized.	

3.3 Process 3 Selected Module Activity: Child drags and drops the answer on a drop spot

Table 10-3.3 Process

Input	Output
Answer Details	Accept on Drop Spot
Process: Child will drag and drop the answer that is an image format on the drop spot. The answer is verified through the database. The image will only be accepted on the drop spot if answer is correct.	

3.4 Process 4 Audio Recording: Child drags and drops the answer on the screen, and in every lesson the child's audio response is recorded.

Table 11-3.4 Process

Input	Output
Audio Input	Audio Output
Process: System will verbally instruct the child to verbalize the lesson or ask a follow up question. The audio response is recorded and saved onto the device which can be played back at a later time.	



System Requirements

1. *Testing*

The proposed system will be tested using Android Studio using the minimal O.S. so that it will cover more devices. Since the Android studio has an emulator, it will be tested using the emulated version before publishing the application on the android market. It is also tested on an actual android device, both tablet and phone.

2. *Resources*

2.1 Developmental Resources

The hardware used to create and develop the application is a Dell Laptop with a touch screen. This makes it easier to test the app on an emulated device.

Table 12-2.1 Developmental Resources

Processor	Intel i7 7 th Gen
RAM	12GB RAM
Storage	500GB SSD
Operating System	Windows 10
Development tool	Android Studio

2.2 User Resources

The potential user should have these specs to ensure compatibility and maximize the user experience.



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

26

- Minimum of A 32bit ARMv7 Processor, 256 MB of RAM, and 400 MB of storage.
- Touchscreen input device
- Minimum API level 12 (Jelly Bean O.S.), targeted API level 22 (Marshmallow and above.)

Installation

This application will be published and made available on Google Play / Android Market and will be installed on mobile devices such as Android Phones and Tablets.

Training

The user interface will instruct the user on how to operate the program using verbal and visual cues, thus training will not be required.

Implementation

There is no need to maintain this application since all necessary files will be included in the application.



Chapter 3

Presentation and Analysis of System

Graphical User Interface

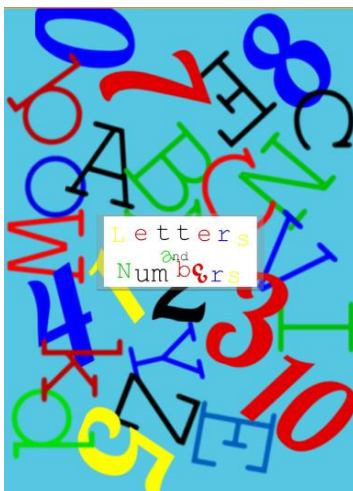
The following are the screenshots of EmbrASD; an Android application designed for children with autism as a supplemental tool for therapy.

Figure 9-Colors and Shapes



Shows the Colors and Shapes menu screen. Pressing the button “Colors and Shapes” will open the lesson type “Colors and Shapes.” Swiping left/right changes menu screen

Figure 10-Letters and Numbers

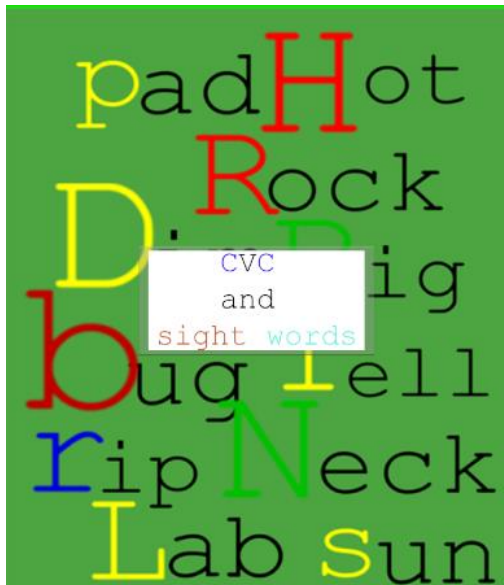


Shows the Colors and Shapes menu screen. Pressing the button “Letters and Numbers” will open the lesson type “Letters and Numbers.”

Swiping left/right changes menu screen

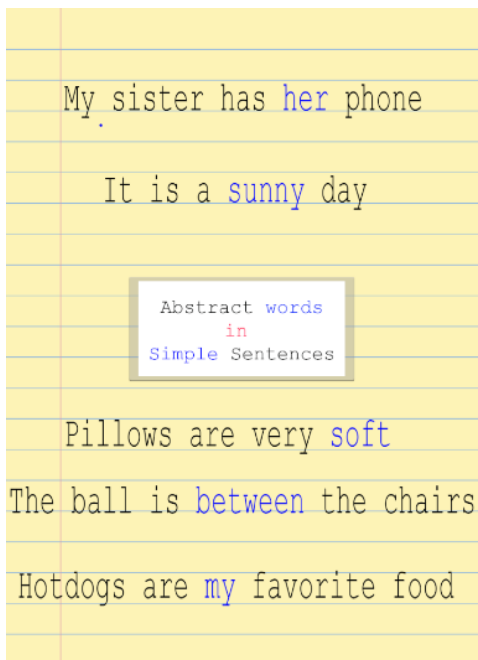


Figure 11-CVC and Sight Words



Shows the CVC and Sight Words menu screen. Pressing the button "Colors and Shapes" will open the lesson type "Colors and Shapes." Swiping left/right changes menu screen

Figure 12-Abstract Words in Simple Sentences



Shows the Abstract Words in Simple Sentence menu screen. Pressing the button "Abstract Words in Simple Sentences" will open the lesson type "Abstract Words in Simple Sentences." Swiping left/right changes menu screen



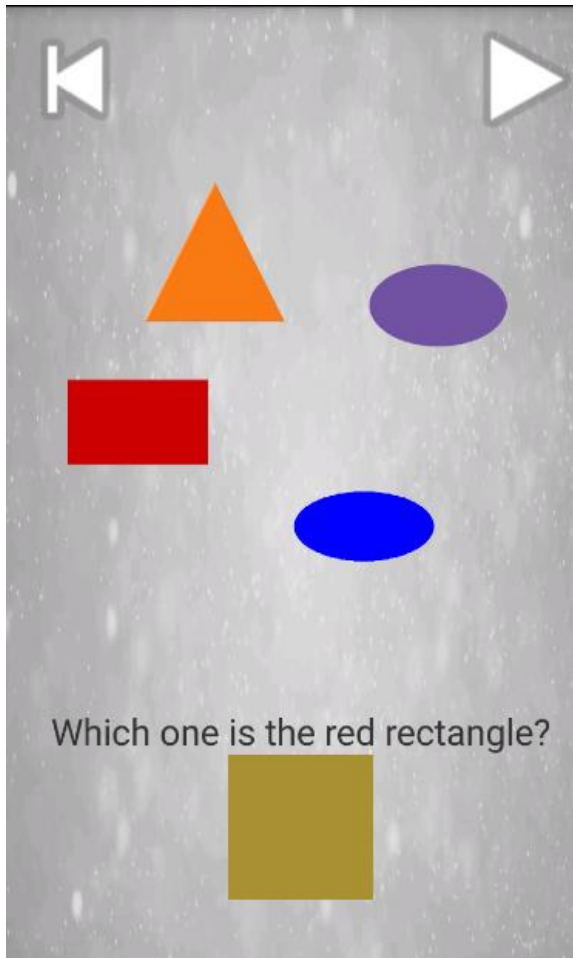
Figure 13-Reading Comprehension



Shows the Reading Comprehension menu screen. Pressing the button "Reading Comprehension" will open the lesson type "Reading Comprehension." Swiping left/right changes menu screen



Figure 14-Colors and Shapes Lesson Screen



Shows the colors and shapes lesson screen. Here, the child will get an auditory cue as well as the question typed on the bottom screen. The child will select an answer, and using a drag and drop gesture, will drop the image on the brown box on the bottom of the screen. The button on the top left will bring the screen back to the menu selection screen. The button on the top right will generate a new randomized lesson.

The background on this activity is a looping video but only if the device has an API level 16 or greater, otherwise it will just be a plain grey background.



Figure 15-Colors and Shapes Audio Recording Screen

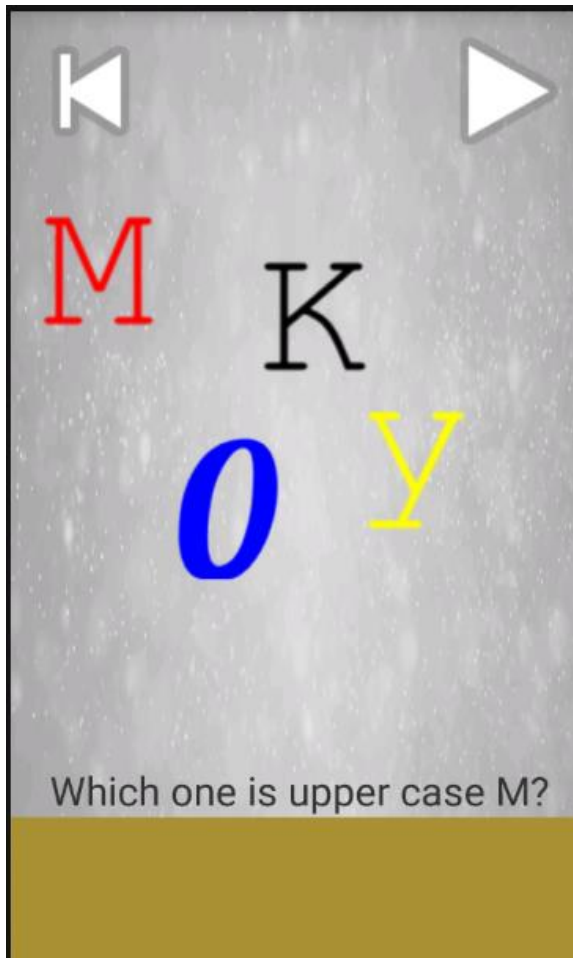


Shows the colors and shapes audio recording screen. Here, the child will get an auditory cue on what to say. The system will automatically record audio for 10 seconds. Also the screen touch input will be disabled during recording process. The button on the top left will bring the screen back to the menu selection screen. The button on the top right will generate a new randomized lesson.

The background on this activity is a looping video but only if the device has an API level 16 or greater, otherwise it will just be a plain grey background.



Figure 16-Letters and Numbers Lesson Screen



Shows the Letters and Numbers screen. Here, the child will get an auditory cue as well as the question typed on the bottom screen. The child will select an answer, and using a drag and drop gesture, will drop the image on the brown box on the bottom of the screen. The button on the top left will bring the screen back to the menu selection screen. The button on the top right will generate a new randomized lesson.

The background on this activity is a looping video but only if the device has an API level 16 or greater, otherwise it will just be a plain grey background.



Figure 17-Letters and Numbers Recording Screen

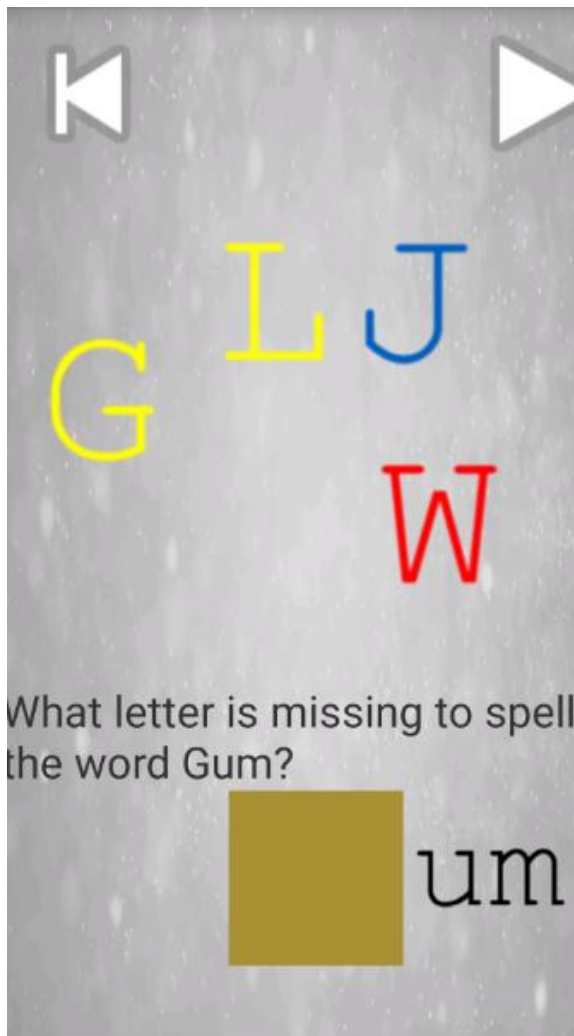


Shows the colors and shapes audio recording screen. Here, the child will get an auditory cue on what to say. The system will automatically record audio for 10 seconds. Also the screen touch input will be disabled during recording process. The button on the top left will bring the screen back to the menu selection screen. The button on the top right will generate a new randomized lesson.

The background on this activity is a looping video but only if the device has an API level 16 or greater, otherwise it will just be a plain grey background.



Figure 18-CVC and Sight Words Screen



Shows the Letters and Numbers screen. Here, the child will get an auditory cue as well as the question typed on the bottom screen. The child will select an answer, and using a drag and drop gesture, will drop the image on the brown box on the bottom of the screen. The button on the top left will bring the screen back to the menu selection screen. The button on the top right will generate a new randomized lesson.

The background on this activity is a looping video but only if the device has an API level 16 or greater, otherwise it will just be a plain grey background.



Figure 19-CVC and Sight Words Recording Screen



Shows the colors and shapes audio recording screen. Here, the child will get an auditory cue on what to say. The system will automatically record audio for 10 seconds. Also the screen touch input will be disabled during recording process. The button on the top left will bring the screen back to the menu selection screen. The button on the top right will generate a new randomized lesson. The background on this activity is a looping video but only if the device has an API level 16 or greater, otherwise it will just be a plain grey background.



Figure 20-Abstract Words in Simple Sentences Screen



Shows the Abstract Words in Simple Sentences screen. Here, the child will get an auditory cue as well as the question typed on the bottom screen. The child will select an answer, and using a drag and drop gesture, will drop the image on the brown box on the bottom of the screen. The button on the top left will bring the screen back to the menu selection screen. The button on the top right will generate a new randomized lesson.

The background on this activity is a looping video but only if the device has an API level 16 or greater, otherwise it will just be a plain grey background.



Figure 21-Abstract Words in Simple Sentences Recording Screen



Shows Abstract Words in Simple

Sentences audio recording screen.

Here, the child will get an auditory

cue on what to say. The system will

automatically record audio for 10

seconds. Also the screen touch

input will be disabled during

recording process. The button on

the top left will bring the screen

back to the menu selection screen.

The button on the top right will

generate a new randomized lesson.

The background on this activity is a

looping video but only if the device

has an API level 16 or greater,

otherwise it will just be a plain grey

background.



Figure 22-Reading Comprehension Screen



Shows Reading Comprehension screen. Here, the child will get an auditory cue as well as the question typed on the bottom screen. The child will select an answer, and using a drag and drop gesture, will drop the image on the brown box on the bottom of the screen. The button on the top left will bring the screen back to the menu selection screen. The button on the top right will generate a new randomized lesson.

The background on this activity is a looping video but only if the device has an API level 16 or greater, otherwise it will just be a plain grey background.



Figure 23-Reading Comprehension Recording Screen



Shows Reading Comprehension audio recording screen. Here, the child will get an auditory cue on what to say. The system will automatically record audio for 10 seconds. Also the screen touch input will be disabled during recording process. The button on the top left will bring the screen back to the menu selection screen. The button on the top right will generate a new randomized lesson. The background on this activity is a looping video but only if the device has an API level 16 or greater, otherwise it will just be a plain grey background.



System Testing and Findings

The following are the issues, findings and results during application testing on an phone and tablet devices using different API levels:

- **Front facing video recording**
 - **Findings:** Front facing camera recording is not uniformly accessible by codes for API level 20 and below.
 - **Solution:** Resorted to audio only recording to ensure that a wider user base will be covered.
- **VideoView Playback**
 - **Findings:** API level 16 and below does not have the capability to video clips.
 - **Solution:** System will determine the API level of the device. If device is above API 16, VideoView is enabled, otherwise it will be disabled. VideoView is used on the background of lesson screen and audio recording screen and is strictly for aesthetic function only.
- **Image File Folders**
 - **Findings:** Image files are displayed based on the device resolution capabilities. Images are obscured when tested on different screen sizes (Phone and Tablet.)



- **Solution:** Creation of two separate folders for phone and tablet resolutions were created. Images were simply copied so that both folders have identical Image resources.
- **Image repositioning not to scale when dragged and dropped**
 - **Findings:** Images not dropping in place as well as the size and position was not consistent.
 - **Solution:** Defining an absolute layout, creating a drag controller as well as added logic to where an image would remember its initial position and get the absolute position when dropped resolved the issue.
- **Having multiple instances of background music**
 - **Findings:** The background music is being reinitialize and not simply replaying the track
 - **Solution:** created a background music service that would stop the music on page navigation.



Chapter 4

Summary, Conclusion and Recommendations

Summary

Based on the study, it can be said that the researcher has successfully developed an application designed for Children with Autism that can be used as a supplemental tool for therapy. The multi-dimensional approach of targeting core pathologies of Children with Autism will greatly enhance and supplement therapy sessions for the child. Recorded audio of the child can be played back and analyzed by parents and therapists.

Conclusion

Based on the results of the study, it can be inferred that:

- The application has targeted will mitigate the core pathologies of children with Autism. Each module instructs and encourages the child to speak, think, analyze, memorize and comprehend the lesson taught.
- The application has a multi-dimensional approach on the design of lesson plans for the child. The inputs of speech therapists, occupational therapists, SPED teachers as well as ABA Therapists greatly influence the design, structure and content lesson of the application.



- The audio recording of the child each lesson can be used as a gauge where the child is excelling and what areas need improvements. This is crucial for both parents and therapists because this information will mold and adjust the lesson curriculum for the child in both SPED schooling and therapy.

Recommendation

Based on the study, the following can be recommended:

- Porting the application to an iOS environment would enable other parents or guardians to also benefit from the application.
- Future update should include video recording of child when the market penetration for API 22 is greater than 80% (Currently it is only 10%). Having a video recording can shed light on the shape of the child's mouth when speaking, as well as the child's facial expression can reveal the child's emotional state when using the application.



References

Electronic Sources:

Online Journal Articles

Ahearn, B. P.-D. (2010, January). *Wading Through the Minefield of Autism Treatments*. Retrieved from Psychology Today:

<https://www.psychologytoday.com/blog/radical-behaviorist/201001/wading-through-the-minefield-autism-treatments>

Kobak, K., & Stone, W. (2011, December). *A web-based tutorial for parents of young children with autism: results from a pilot study*. Retrieved from National Center for Biotechnology Information: <https://www.ncbi.nlm.nih.gov/pubmed/22011005>

Moore, M., & Calvert, S. (2000, August). *Brief report: vocabulary acquisition for children with autism: teacher or computer instruction*. Retrieved from National Center for Biotechnology Information: <http://www.ncbi.nlm.nih.gov/pubmed/11039862>

William, C., Wright, B., Callaghan, G., & Coughlan, B. (2002, March). *Do children with autism learn to read more readily by computer assisted instruction or traditional book methods? A pilot study*. Retrieved from NCBI: <https://www.ncbi.nlm.nih.gov/pubmed/11918110>

Wong, S., & Tam, S. (2001, December). *Effectiveness of a multimedia programme and therapist-instructed training for children with autism*. Retrieved from National Center for Biotechnology Information: <https://www.ncbi.nlm.nih.gov/pubmed/?term=Effectiveness+of+a+multimedia+programme+and+therapist-instructed+training+for+children+with+autism>

Interviews:

Aguas, A. (2016). Occupational Therapy. (M. Ligot, Interviewer)

Gusto, P. (2016). Speech Therapy. (M. Ligot, Interviewer)

Ignacio, B. (2016). SPED Therapy. (M. Ligot, Interviewer)

\
Riofrio, J. (2016). SPED Teacher. (M. Ligot, Interviewer)

Salvacion, J. M. (2016). ABA Therapy. (M. Ligot, Interviewer)



Bachelor of Science in Information Technology

Appendices

A. Project Time Table

[illegible]

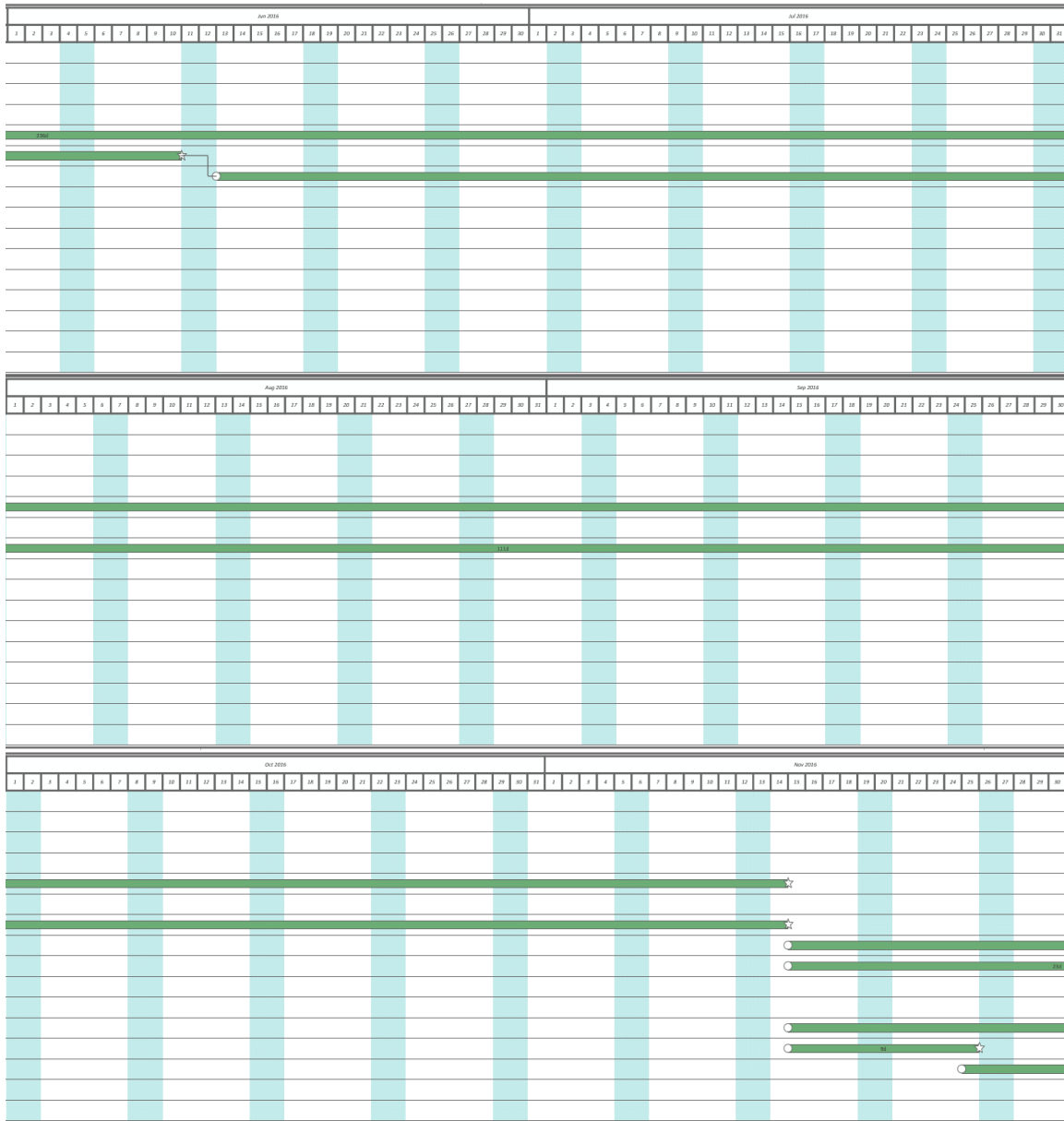


Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

46

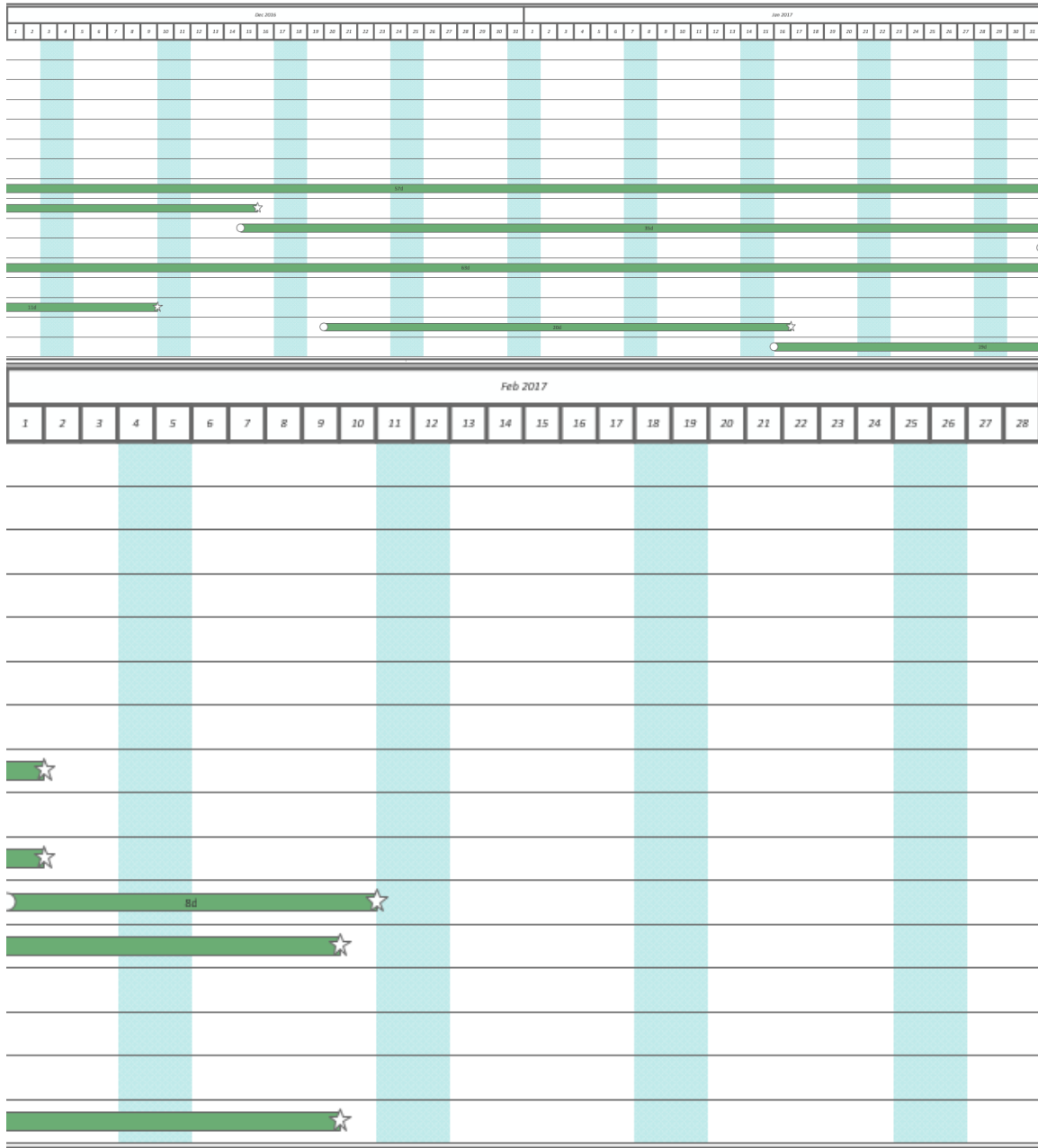




Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740
Bachelor of Science in Information Technology

47





B. Turnitin Result

 **Turnitin Originality Report**

EmbrASD Final without Source Codes by
Mark Ligt

From Year 2017 (2017)

Processed on 09-Mar-2017 11:39 PHT
ID: 781599672
Word Count: 8607

Similarity Index	Similarity by Source	
7%	Internet Sources:	2%
	Publications:	0%
	Student Papers:	6%

sources:

- 1 3% match (student papers from 04-May-2012)
[Submitted to South Bank University on 2012-05-04](#)
- 2 1% match (student papers from 27-May-2016)
[Submitted to Manchester Metropolitan University on 2016-05-27](#)
- 3 < 1% match (Internet from 27-May-2016)
<http://media.proquest.com/media/pq/classic/doc/4041869361/fmt/ai/rep/NPDF?s=yHQ0Jepw7jMrH04ws29yR%2F5H%2BY%3D>
- 4 < 1% match (student papers from 23-Oct-2013)
[Submitted to Kaplan University on 2013-10-23](#)
- 5 < 1% match (Internet from 26-Oct-2016)
<http://network.autism.org.uk/good-practice/case-studies/why-occupational-therapy-important-children-autism>
- 6 < 1% match (Internet from 12-Dec-2007)
<http://calendar.wildernesscommittee.org/week.php?date=20061203>
- 7 < 1% match (student papers from 22-Apr-2016)
[Submitted to Manchester Metropolitan University on 2016-04-22](#)
- 8 < 1% match (student papers from 30-Apr-2014)
[Submitted to University of Central England in Birmingham on 2014-04-30](#)
- 9 < 1% match (student papers from 18-Feb-2014)
[Submitted to Universiti Putra Malaysia on 2014-02-18](#)
- 10 < 1% match (student papers from 08-Jan-2016)
[Submitted to University of York on 2016-01-08](#)



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

49

C. Request for Thesis Adviser



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

50

D. Consultation Record



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

51

E. Activities Checklist for Thesis Submission



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740
Bachelor of Science in Information Technology

52

F. Recommendation for Oral Defense



SOUTHVILLE INTERNATIONAL SCHOOL AND COLLEGES

Luxembourg St. corner Tropical Ave., B.F. Homes International
Las Piñas City, Philippines 1740
Tel. No. (632) 825-6374; (632) 820-8702 to 03
E-mail: pr@southville.edu.ph

February 1, 2017

Aris E. Ignacio, DIT, MCP, MCE

Dean

College of Information Technology

Sir:

Southville International School and Colleges Scholarship Management System, a thesis project featuring different functions in for a particular client currently developed by the following Information Technology student:

Mark Ligot

|
Has been able to complete the following milestones based on the undersigned monitoring and examination and assessment:

1. Different pages based on the proposed documentation that is reflective of the different modules required (i.e. Student Information, Scholarship Application, Grade Assessment etc.),
2. Database design,
3. The main documentation,
4. Additional documents, information needed to support the main documentation of the development and,
5. Graphical user interface aligned to the needs of the consumer on the documentation.

Each of the proponents' contribution were essential in defining and improving the business model, in designing and implementing the underlying infrastructure, and extending it resulting into the implementation of a sophisticated software and documentation that will benefit the target client.

With the following objectives being completed, the proponents validated the required prerequisites for the oral presentation to commence on Thursday, the 16th of February 2017.

Sincerely,

Aris E. Ignacio, DIT, MCP, MCE

Faculty Adviser



G. Interview Notes Overview

Arnes Aguas, an occupational therapist, main focus is on behavior modification as well as teaching the child to follow basic commands. The need to learn basic commands is the building block to learn more complex tasks.

Pauleen Gusto, a speech pathologist, main focus is teaching the building blocks of communication. It must start with the basics such as the names, nouns and common words. Eventually verbs and other abstract words will be introduced to develop more complex sentence structure. Language and communication is the key to entering the mind and the world of a person with autism.

Barbie Ignacio, a SPED therapist, main focus is teaching the cognitive and critical thinking of a child with autism. The academics aspects are important because this is the key to integrating the child with autism to the mainstream school system. A one on one approach will give the child the specified goals they need based on the child's capacity.

Jherson Rufio, a SPED teacher, main focus is setting an academic curriculum in a classroom setting to children with autism. They differ from a one on one



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

54

SPED teacher because they teach the child to interact in a social environment. Their end game is to eventually integrate the child to the mainstream school system.

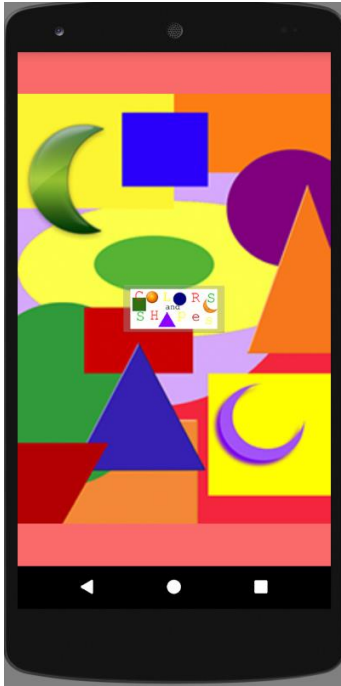
Jelly Salvacion, an ABA Therapist, is the comprehensive approach of an occupational therapist, a SPED teacher and a speech pathologist. They accomplish this by behavior modification as well. They want the child to essentially become independent, have the basic skills of reading, writing and complex analytical skills.



H. Pictures during Thesis Defense

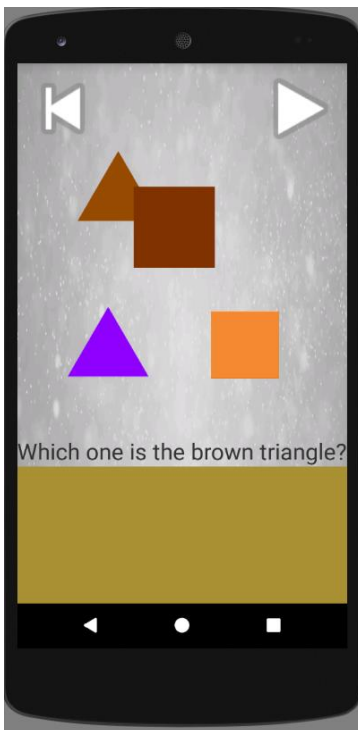


I. User Manual



At this screen, swiping right and left would navigate to the five different activities. These are Colors and Shapes, Letters and Numbers, CVC and Sight Words, Abstract words in Complete Sentences and lastly Reading Comprehension.

Pressing the button on the center of each activity page would enter that activity.

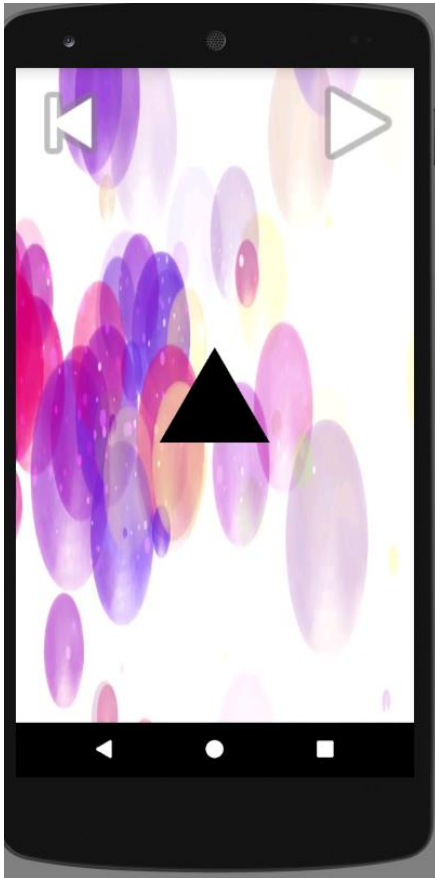


This is the activity page of the selected activity. In this example, Colors and Shapes was the selected activity.

Dragging the different images would have a voice response as to what image is being dragged. Dragging and dropping into the lower brown shaded part of the screen would mean the selection of answer to the question that is being asked verbally as well as written on the bottom of the screen.

Only the correct image would be allowed to be dropped on the lower brown shaded part of the screen.

The top left button would navigate back to the main menu. The top right button would generate another random question.



If the answer is correct, then it would go to this screen. At this time, a verbal response is recorded while in this screen for 10 seconds. Also the screen would not be responsive to touches until 10 seconds has lapsed.

There will be an audio playing that would instruct as to what verbal response is being requested. A response is not necessary as the app would still record for 10 seconds regardless.

The recorded audio will be saved into the device root directory saved as a 3GP audio file format.

The top left button would navigate back to the main menu. The top right button would generate another random question.



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

58

J. Notes on the Author

Mark Ligot, MTA
Contact No: 09164233764
E-mail: Mark.ligot@gmail.com

Computer Skills:

Microsoft Certified Software Development Fundamentals, Microsoft Certified Database Fundamentals, Advanced user of Microsoft Excel, Access, Data Management and Data Integrity, PowerPoint and Word. Proficient in the use of JD Edwards, Peoplesoft, Groupwise, Essbase and Showcase. Advanced knowledge in VB Script language.

Education:

California State University Northridge, Northridge, U.S.A. 2001-2003 Undergrad
Southville International Schools and Colleges, Las Pinas, Philippines 2013-Present
Pursuing Bachelors Degree in Information Technology

Experience:

April 2007
To 2012 Savemart

Accounts Receivable

Responsibilities included:

- Responsible for maintaining miscellaneous receivables accounts to ensure that payments received are within compliance of their contracts.
- Created multiple databases that streamlined maintaining data, ensured data integrity, and manipulating reports using complex queries with Microsoft Access, Showcase and Essbase.
- Shared all new processes with the team to maximize team efficiency.
- Extensive knowledge in VBA Script allows automation and communication between programs such as Excel, Access, Essbase, and Showcase for maintaining consistently updating information in once cohesive and user friendly database for manipulation.
- Automated mass journal entry uploads using macros created within excel.
- Created a customized comparison report that shows variance between Essbase and Line Items pulled from Oracle for automated reconciliation.
- Answered any inquiries from trade charge customers and documented changes on the customers account.

August 2006
to March 2007

Volt Information Sciences

Billing Analyst

Responsibilities included:

- Responsible for analyzing billing discrepancies that the client disputes. Determine the cause and provide proper back up to resolve the dispute.
- Worked with Payroll to re-bill invoices made in error prior to sending them to clients to minimize disputes.
- Coordinated with the vice president of accounting, branch managers and collections to determine if invoice being disputed is collectable or a write off adjustment is necessary.
- Responsible for handling high profile accounts to reconcile, research and furnished evidence required by client to get invoices paid.
- Created reports to present to management with recommendations of processes to be implemented to maximize accuracy without compromising data integrity.
- Implemented streamlined processes through work study measurement.

September 2005
to July 2006

Accounting Consultant

Worked in long term temporary positions at: **Impco Technologies, MWB Business Systems, Land America Commonwealth**
Orange County, CA
ACCOUNTS PAYABLE/ RECEIVABLE CLERK
Responsibilities included:



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

59

- Worked closely with purchase department to implement streamline orders and discrepancies are addressed in a timely manner.
- Received and reviewed invoices, matched original invoice to proof of delivery.
- Entered heavy volumes of invoices into accounting system daily.
- Assisted with weekly check runs, ensuring all utilities obligations are paid timely.
- Received checks and disbursed funds to appropriate vendor accounts.
- Ensured invoices were paid quickly as to receive discounts from vendors.
- Implemented a system to organize all Credit Memos created.
- Worked closely with the CFO in keeping track of all Credit Memos and streamlined reports.
- Calculated accruals to be entered into the general ledger.
- Verified and posted payments to customers' accounts.
- Administered customer accounts to ensure punctual payments.
- Accountable for processing cash receipts and preparing deposits on a daily basis.
- Organized files daily and set aside those needing corrections or revisions.
- Wired money to costumers' accounts and ensured sufficient funds.



K. Source Code

Android Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    android:label="@string/app_name"
    package="com.example.markligot.embrasd.dra
gview"
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL
_STORAGE" />
    <uses-permission
android:name="android.permission.READ_EXTERNAL_
_STORAGE" />
    <uses-permission
android:name="android.permission.RECORD_AUDIO"
/>
    <uses-permission
android:name="android.permission.VIBRATE">
</uses-permission>
    <application
    android:allowBackup="true"
    android:label="@string/app_name"
    android:icon="@drawable/ic_launcher"

    android:theme="@style/AppTheme.Fullscreen" >
        <service android:name=".MusicService"
/>
        <service android:name=".MusicService2"
/>
        <service android:name=".MusicService3"
/>
        <service android:name=".MusicService4"
/>
        <service
android:name=".RecordingAudioService" />
        <service
android:name=".AudiorecorderService" />
        <activity android:name=".MainActivity">
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />

                </intent-filter>
            </activity>
            <activity android:name=".SplashScreen">
                <intent-filter>
                    <action
android:name="android.intent.action.MAIN" />

                    <category
android:name="android.intent.category.LAUNCHER"
/>

                </intent-filter>
            </activity>
            <activity

    android:name="com.example.markligot.embrasd.dra
gview.DragActivityV4"
        android:label="@string/app_name" >
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />

            </intent-filter>
        </activity>

        <activity
```

```
    android:name="com.example.markligot.embrasd.dra
gview.CongratulationsActivity"
        android:label="@string/app_name" >
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />

            </intent-filter>
        </activity>
    </application>
</manifest>
```

Classes

TabPageAdapter

```
package com.example.markligot.embrasd.adapter;

import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;

import com.example.markligot.embrasd.*;

public class TabPagerAdapter extends FragmentPagerAdapter {

    public TabPagerAdapter(FragmentManager fm) {
        super(fm);
    }
    @Override
    public Fragment getItem(int index) {

        switch (index) {
            case 0:
                return new ColorsandShapesFragment();
            case 1:
                return new LettersandNumbersFragment();
            case 2:
                return new CvcandSightWordsFragment();
            case 3:
                return new AbstractFragment();
            case 4:
                return new ReadingComprehensionFragment();
        }

        return null;
    }
    @Override
    public int getCount() {
        // get item count - equal to number of tabs
        return 5;
    }
}
```

CongratulationsActivity

```
package com.example.markligot.embrasd.dragview;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.ComponentName;
import android.content.ContentValues;
import android.content.Context;
import android.content.Intent;
import android.content.ServiceConnection;
```



```
import android.content.pm.ActivityInfo;
import android.media.MediaPlayer;
import android.media.MediaRecorder;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.IBinder;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.ImageView;
import android.widget.MediaController;
import android.widget.TextView;
import android.widget.VideoView;

import com.example.markligot.embrasd.AudiorecorderService;
import com.example.markligot.embrasd.*;

import java.util.Random;
```

```
/**
 * Created by Mark Ligot on 1/15/2017.
 */
```

```
public class CongratulationsActivity extends Activity {
```

```
    public String selectedImg;
    public String selectedAud;
    public String selectedCorrAnsAud;
    public String qImgFP;
    public String selectedImgDesc;
    public String tvString;
    public String tv2String;
    public String selectedfollowupquesaud;
    public String selectedfollowupquestext;
```

```
    public Handler mHandler = new Handler();
```

```
    public final static String EXTRA_MESSAGE11 =
    "com.example.myfirstapp.MESSAGE11";
```

```
    public Context context;
```

```
    public static String lessonmodule = "0";
    public int intlessnmodule = 0;
```

```
    public int iplayed;
```

```
    public MediaPlayer mp;
    private VideoView myVideoView;
    private MediaController mediaControls;
```

```
    private boolean mIsBound = false;
    private AudiorecorderService mServ;
```

```
    private ServiceConnection Scon=new ServiceConnection(){
```

```
        public void onServiceConnected(ComponentName name,
        IBinder
            binder) {

            mServ =
            ((AudiorecorderService.ServiceBinder)binder).getService();
        }
    }
```

```
        public void onServiceDisconnected(ComponentName name) {
            mServ = null;
        }
    };
```

```
    void doBindService(){
        bindService(new Intent(this,AudiorecorderService.class),
            Scon, Context.BIND_AUTO_CREATE);
        mIsBound = true;
    }
```

```
    void doUnbindService()
    {
        if(mIsBound)
        {
            unbindService(Scon);
            mIsBound = false;
        }
    }
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATI
        ON_PORTRAIT);
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
```

```
        context = this;
```

```
        this.getWindow().addFlags(WindowManager.LayoutParams.FLAG_
        NOT_TOUCHABLE);
```

```
        if (Build.VERSION.SDK_INT <= 16) {
            Intent intent2 = getIntent();
            lessonmodule =
            intent2.getStringExtra(DragActivityV4.EXTRA_MESSAGE5);
            int intlessnmodule = Integer.parseInt(lessonmodule);
            System.out.println("lessnmod" + lessonmodule);
            if (intlessnmodule != 4 && intlessnmodule != 5) {
                setContentView(R.layout.maintestjb);
            }
            if (intlessnmodule == 4 || intlessnmodule == 5) {
                setContentView(R.layout.maintest2jb);
            }
        }
```

```
        else {
            Intent intent2 = getIntent();
            lessonmodule =
            intent2.getStringExtra(DragActivityV4.EXTRA_MESSAGE5);
            int intlessnmodule = Integer.parseInt(lessonmodule);
```

```
            if (intlessnmodule != 4 && intlessnmodule != 5) {
                setContentView(R.layout.maintest);
            }
            if (intlessnmodule == 4 || intlessnmodule == 5) {
                setContentView(R.layout.maintest2);
            }
        }
```

```
        Intent intent2 = getIntent();
        lessonmodule =
        intent2.getStringExtra(DragActivityV4.EXTRA_MESSAGE5);
        int intlessnmodule = Integer.parseInt(lessonmodule);
        System.out.println("lessnmod" + lessonmodule);
```

```
        if (intlessnmodule != 4 && intlessnmodule != 5) {
            selectedImg =
            intent2.getStringExtra(DragActivityV4.EXTRA_MESSAGE2);
            selectedAud =
```



```
intent2.getStringExtra(DragActivityV4.EXTRA_MESSAGE3);
qImgFP =
intent2.getStringExtra(DragActivityV4.EXTRA_MESSAGE4);
System.out.println("selectedaud " + selectedAud);
}
if (intlessnmodule == 4) {
    selectedImg =
    intent2.getStringExtra(DragActivityV4.EXTRA_MESSAGE2);
    selectedCorrAnsAud =
    intent2.getStringExtra(DragActivityV4.EXTRA_MESSAGE3);
    selectedImgDesc =
    intent2.getStringExtra(DragActivityV4.EXTRA_MESSAGE4);
    tvString =
    intent2.getStringExtra(DragActivityV4.EXTRA_MESSAGE6);
    tv2String =
    intent2.getStringExtra(DragActivityV4.EXTRA_MESSAGE7);
}
if (intlessnmodule == 5) {
    selectedImg =
    intent2.getStringExtra(DragActivityV4.EXTRA_MESSAGE2);
    selectedCorrAnsAud =
    intent2.getStringExtra(DragActivityV4.EXTRA_MESSAGE3);
    selectedImgDesc =
    intent2.getStringExtra(DragActivityV4.EXTRA_MESSAGE4);
    selectedfollowupquesaud =
    intent2.getStringExtra(DragActivityV4.EXTRA_MESSAGE8);
    selectedfollowupquestext =
    intent2.getStringExtra(DragActivityV4.EXTRA_MESSAGE9);
}
//qImgFP =
intent.getStringExtra(DragActivityV4.EXTRA_MESSAGE4);

if(selectedImg !=null) {
    ImageView qv = new ImageView(this);
    qv = (ImageView) findViewById(R.id.corrAnsIMG);
    int qimgid = getResources().getIdentifier(selectedImg,
"drawable", getPackageName());
    qv.setImageResource(qimgid);
    System.out.println("selectedImg " + selectedImg);
}
if(qImgFP !=null) {
    ImageView qv1 = new ImageView(this);
    qv1 = (ImageView) findViewById(R.id.qimgFP);
    int qimgid2 = getResources().getIdentifier(qImgFP,
"drawable", getPackageName());
    qv1.setImageResource(qimgid2);
    System.out.println("qimgFP " + qImgFP);
}
if(selectedImgDesc !=null) {
    ImageView qv2 = new ImageView(this);
    qv2 = (ImageView) findViewById(R.id.sellImgDesc);
    int qimgid3 = getResources().getIdentifier(selectedImgDesc,
"drawable", getPackageName());
    qv2.setImageResource(qimgid3);
    System.out.println("selectedIMGDesc " +
selectedImgDesc);
}
if (intlessnmodule != 5) {
    if (tvString != null) {
        TextView tv = (TextView) findViewById(R.id.tvString);
        tv.setTextSize(35);
        tv.setText(tvString);
        System.out.println("tvString " + tvString);
    }
    if (tv2String != null) {
        TextView tv2 = (TextView) findViewById(R.id.tv2String);
        tv2.setTextSize(35);
        tv2.setText(tv2String);
        System.out.println("tv2String " + tv2String);
    }
}
```

```
if(selectedfollowupquestext != null){
    TextView tv = (TextView) findViewById(R.id.tvString);
    tv.setTextSize(35);
    tv.setText(tvString);
    System.out.println("selectedfollowupquestext " +
selectedfollowupquestext);
}
if(selectedfollowupquestext != null){
    TextView tv2 = (TextView) findViewById(R.id.tv2String);
    tv2.setTextSize(35);
    tv2.setText(selectedfollowupquestext);
    System.out.println("selectedfollowupquestext " +
selectedfollowupquestext);
}
/**
if (selectedAud != null) {

    try {
        new TryCatchStopMediaPlayer().invoke();

        int audID = getResources().getIdentifier(selectedAud,
"raw", getPackageName());
        mp = MediaPlayer.create(this, audID);
        mp.setLooping(false);
        mp.setVolume(100, 100);
        if (mp.isPlaying()) {
            mp.stop();
            mp.release();
            mp = null;
        }
        mp.start();
    } catch (Throwable t) {
        goBlooeey(t);
    }
}
**/
if (intlessnmodule != 5) {
    int resChoice1 = 1;
    Random rResSelector = new Random();
    resChoice1 = rResSelector.nextInt((5-1)+1);
    System.out.println("reschoice1 " + resChoice1);
    if (resChoice1 == 1) {
        mp = MediaPlayer.create(this, R.raw.thatright);
    }
    if (resChoice1 == 2) {
        mp = MediaPlayer.create(this, R.raw.yourdoinggreaud);
    }
    if (resChoice1 == 3) {
        mp = MediaPlayer.create(this, R.raw.awesomeaud);
    }
    if (resChoice1 == 4) {
        mp = MediaPlayer.create(this, R.raw.goodjobaud);
    }
    if (resChoice1 == 5) {
        mp = MediaPlayer.create(this, R.raw.greatworkaud);
    }
    if (mp == null)
    {
        mp = MediaPlayer.create(this, R.raw.thatright);
    }
    mp.setLooping(false);
    mp.setVolume(100, 100);
    if (mp.isPlaying()) {
        mp.stop();
        mp.release();
        mp = null;
    }
    iplayed = 0;
    mp.start();
    mp.setOnCompleteListener(completeListener);
}
```



```
if (intlessnmodule == 5){

    int resChoice1 = 1;
    Random rResSelector = new Random();
    resChoice1 = rResSelector.nextInt((5-1)+1);
    System.out.println("reschoice1 " + resChoice1);
    if (resChoice1 == 1) {
        mp = MediaPlayer.create(this, R.raw.thatright);
    }
    if (resChoice1 == 2) {
        mp = MediaPlayer.create(this, R.raw.yourdoinggreaud);
    }
    if (resChoice1 == 3) {
        mp = MediaPlayer.create(this, R.raw.awesomeaud);
    }
    if (resChoice1 == 4) {
        mp = MediaPlayer.create(this, R.raw.goodjobaud);
    }
    if (resChoice1 == 5) {
        mp = MediaPlayer.create(this, R.raw.greatworkaud);
    }
    if (mp == null)
    {
        mp = MediaPlayer.create(this, R.raw.thatright);
    }
    mp.setLooping(false);
    mp.setVolume(100, 100);
    if (mp.isPlaying()) {
        mp.stop();
        mp.release();
        mp = null;
    }
    iplayed = 0;
    mp.start();
    mp.setOnCompletionListener(completeListener5);

}

Intent intentAudio = new Intent(this,
    AudiorecorderService.class);
startService(intentAudio);

/**
    if (mediaControls == null) {
        mediaControls = new
        MediaController(CongratulationsActivity.this);
    }
    **/

if (Build.VERSION.SDK_INT <= 16) {
}
else {
    myVideoView = (VideoView)
    findViewById(R.id.videoView);
    try {
        //myVideoView.setMediaController(mediaControls);

        myVideoView.setVideoURI(Uri.parse("android.resource://" +
        getPackageName() + "/" + R.raw.bgvid));

    } catch (Exception e) {
        Log.e("Error", e.getMessage());
        e.printStackTrace();
    }

    myVideoView.setOnPreparedListener(new
    MediaPlayer.OnPreparedListener() {
        @Override

        public void onPrepared(MediaPlayer mp) {
            mp.setLooping(true);
        }
    });

    myVideoView.start();
    myVideoView.setOnCompletionListener(new
    MediaPlayer.OnCompletionListener() {

        @Override
        public void onCompletion(MediaPlayer mp) {

            myVideoView.start();

        }
    });

    mHandler.postDelayed(new Runnable() {
        public void run() {
            doStuff();
        }, 10000);

    private void doStuff() {
        Intent intentAudio2 = new Intent(this,
        AudiorecorderService.class);
        stopService(intentAudio2);

        this.getWindow().clearFlags(WindowManager.LayoutParams.FLAG_
        NOT_TOUCHABLE);

        MediaPlayer.OnCompletionListener completeListener5 = new
        MediaPlayer.OnCompletionListener() {

            @Override
            public void onCompletion(MediaPlayer mp) {
                if (selectedfollowupquesaud != null) {

                    try {
                        System.out.println("selectedfollowupquesaud" +
                        selectedfollowupquesaud);

                        final int audID =
                        getResources().getIdentifier(selectedfollowupquesaud, "raw",
                        getPackageName());
                        mp = MediaPlayer.create(context, audID);
                        mp.setLooping(false);
                        mp.setOnCompletionListener(releaseCompletion);
                        mp.setVolume(100, 100);
                        mp.start();
                    }
                    catch (Throwable t) {
                        goBlooy(t);
                    }
                }
            }
        };

        MediaPlayer.OnCompletionListener completeListener = new
        MediaPlayer.OnCompletionListener() {

            @Override
            public void onCompletion(MediaPlayer mp) {
                mp.reset();
                mp.release();
                mp = null;

                //final int audID = getResources().getIdentifier(selectedAud,
```




```
"raw", getPackageName());
    mp = MediaPlayer.create(context, R.raw.canyousayaud);
    mp.setLooping(false);
    mp.setOnCompletionListener(completeListener2);
    mp.setVolume(100, 100);
    if (mp.isPlaying()) {
        mp.stop();
        mp.reset();
        mp.release();
        mp = null;
    }
    iplayed++;
    mp.start();
}
```

```
MediaPlayer.OnCompletionListener completeListener2 = new
MediaPlayer.OnCompletionListener() {
```

```
@Override
public void onCompletion(MediaPlayer mp) {
    mp.reset();
    mp.release();
    mp = null;
```

```
Intent intent = getIntent();
System.out.println("lessonmodule" + lessonmodule);
lessonmodule =
intent.getStringExtra(DragActivityV4.EXTRA_MESSAGE5);
System.out.println("lessonmodule" + lessonmodule);
int intlessnmodule = Integer.parseInt(lessonmodule);
if (intlessnmodule != 4 && intlessnmodule != 5) {
```

```
final int audID =
getResources().getIdentifier(selectedAud, "raw",
getPackageName());
mp = MediaPlayer.create(context, audID);
mp.setLooping(false);
mp.setOnCompletionListener(releaseCompletion);
mp.setVolume(100, 100);
if (mp.isPlaying()) {
    mp.stop();
    mp.reset();
    mp.release();
    mp = null;
}
mp.start();
```

```
}
if (intlessnmodule == 4 || intlessnmodule == 5) {
```

```
if (selectedCorrAnsAud != null) {
    final int audID =
getResources().getIdentifier(selectedCorrAnsAud, "raw",
getPackageName());
    mp = MediaPlayer.create(context, audID);
    mp.setLooping(false);
    mp.setOnCompletionListener(releaseCompletion);
    mp.setVolume(100, 100);
    if (mp.isPlaying()) {
        mp.stop();
        mp.reset();
        mp.release();
        mp = null;
    }
    mp.start();
}
}
};
```

```
MediaPlayer.OnCompletionListener releaseCompletion = new
MediaPlayer.OnCompletionListener() {
```

```
@Override
public void onCompletion(MediaPlayer mp) {
    if (mp != null) {
        mp.reset();
        mp.release();
        mp = null;
    }
}
```

```
public void onClickplayagain2(View vplayagain) {
```

```
/**
Intent intent2 = getIntent();
lessonmodule =
intent2.getStringExtra(DragActivityV4.EXTRA_MESSAGE5);
int intlessnmodule = Integer.parseInt(lessonmodule);
```

```
Intent intent = new Intent(this, DragActivityV4.class);
String lessonModule = "5";
int musicbg = 3;
intent.putExtra(EXTRA_MESSAGE, lessonModule);
intent.putExtra("intVariableName", musicbg);
startActivity(intent);
**/
```

```
Intent intent = getIntent();
int musicbg = intent.getIntExtra("intVariableName", 0);
```

```
if (musicbg == 1){
    Intent intentmusic = new Intent(CongratulationsActivity.this,
MusicService2.class);
    stopService(intentmusic);
}
if (musicbg == 2){
    Intent intentmusic = new Intent(CongratulationsActivity.this,
MusicService3.class);
    stopService(intentmusic);
}
if (musicbg == 3){
    Intent intentmusic = new Intent(CongratulationsActivity.this,
MusicService4.class);
    stopService(intentmusic);
}
```

```
/**
Intent stoprecord = new Intent(CongratulationsActivity.this,
AudiorecorderService.class);
stopService(stoprecord);
**/
```

```
Intent intentmusic = new Intent(CongratulationsActivity.this,
MusicService2.class);
stopService(intentmusic);
```

```
Intent intentmusic2 = new Intent(CongratulationsActivity.this,
MusicService3.class);
stopService(intentmusic2);
```

```
Intent intentmusic3 = new Intent(CongratulationsActivity.this,
MusicService4.class);
stopService(intentmusic3);
```

```
Intent intent3 = new Intent(this, DragActivityV4.class);
```



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740
Bachelor of Science in Information Technology

65

```
intent3.putExtra(EXTRA_MESSAGE11, lessonmodule);
intent3.putExtra("intVariableName", musicbg);

startActivity(intent3);

finish();

}

public void onClickmainmenu2(View vmenu) {

    Intent intent = getIntent();
    int musicbg = intent.getIntExtra("intVariableName",0);

    if (musicbg == 1){
        Intent intentmusic = new Intent(CongratulationsActivity.this,
        MusicService2.class);
        stopService(intentmusic);
    }
    if (musicbg == 2){
        Intent intentmusic = new Intent(CongratulationsActivity.this,
        MusicService3.class);
        stopService(intentmusic);
    }
    if (musicbg == 3){
        Intent intentmusic = new Intent(CongratulationsActivity.this,
        MusicService4.class);
        stopService(intentmusic);
    }
}

/**
 * Intent stoprecord = new Intent(CongratulationsActivity.this,
 * AudiorecorderService.class);
 * stopService(stoprecord);
 */

Intent intentmusic = new Intent(CongratulationsActivity.this,
MusicService2.class);
stopService(intentmusic);

Intent intentmusic2 = new Intent(CongratulationsActivity.this,
MusicService3.class);
stopService(intentmusic2);

Intent intentmusic3 = new Intent(CongratulationsActivity.this,
MusicService4.class);
stopService(intentmusic3);

Intent intent2 = new Intent(this, MainActivity.class);
startActivity(intent2);

finish();
}

@Override
public void onPause(){
    super.onPause();

    Intent intent2 = getIntent();
    int musicbg = intent2.getIntExtra("intVariableName",0);

    if (musicbg == 1){
        Intent intentmusic = new Intent(CongratulationsActivity.this,
        MusicService2.class);
        stopService(intentmusic);
    }
    if (musicbg == 2){
        Intent intentmusic = new Intent(CongratulationsActivity.this,
        MusicService3.class);
        stopService(intentmusic);
    }
}
```

```
if (musicbg == 3){
    Intent intentmusic = new Intent(CongratulationsActivity.this,
    MusicService4.class);
    stopService(intentmusic);
}

Intent intentmusic = new Intent(CongratulationsActivity.this,
MusicService2.class);
stopService(intentmusic);

Intent intentmusic2 = new Intent(CongratulationsActivity.this,
MusicService3.class);
stopService(intentmusic2);

Intent intentmusic3 = new Intent(CongratulationsActivity.this,
MusicService4.class);
stopService(intentmusic3);

finish();
}

public void recordAudio(String fileName) {
    final MediaRecorder recorder = new MediaRecorder();
    ContentValues values = new ContentValues(3);
    values.put(MediaStore.MediaColumns.TITLE, fileName);
    recorder.setAudioSource(MediaRecorder.AudioSource.MIC);

    recorder.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);

    recorder.setAudioEncoder(MediaRecorder.AudioEncoder.DEFAULT
);
    recorder.setOutputFile("/sdcard/sound/" + fileName);
    recorder.setMaxDuration(15000);
    try {
        recorder.prepare();
    } catch (Exception e){
        e.printStackTrace();
    }

    recorder.start();
}

private class TryCatchStopMediaPlayer {
    public void invoke() {
        try {
            if (mp != null) {
                mp.stop();
                mp.release();
                mp = null;
            }
        } catch (Exception e) {
        }
    }
}

private void goBlooey(Throwable t) {
    AlertDialog.Builder builder=new AlertDialog.Builder(this);

    builder.setTitle("Exception!")
        .setMessage(t.toString());
}
}
```

DragActivity4

```
package com.example.markligot.embrasd.dragview;
```



```
import android.app.Activity;
import android.app.AlertDialog;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.ServiceConnection;
import android.content.pm.ActivityInfo;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.IBinder;
import android.util.DisplayMetrics;
import android.util.Log;
import android.view.Menu;
import android.view.MotionEvent;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.VideoView;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.List;
import java.util.Random;

import com.example.markligot.embrasd.interfaces.CorrectAnswerListener;
import com.example.markligot.embrasd.MusicService;
import com.example.markligot.embrasd.*;

/**
 * This activity presents a screen on which
 * images can be added and moved around.
 * It also defines areas on the screen where
 * the dragged views can be dropped. Feedback is
 * provided to the user as the objects are
 * dragged over these drop zones.
 * <p>
 * <p> Like the DragActivity in the previous
 * version of the DragView example application,
 * the
 * code here is derived from the Android
 * Launcher code.
 */

public class DragActivityV4 extends Activity
    implements View.OnLongClickListener,
    View.OnClickListener, View.OnTouchListener,
    CorrectAnswerListener {
    /**
     */
    // Constants

    private static final int ENABLE_S2_MENU_ID
    = Menu.FIRST;
    private static final int DISABLE_S2_MENU_ID
    = Menu.FIRST + 1;
    private static final int ADD_OBJECT_MENU_ID
    = Menu.FIRST + 2;
    private static final int
    CHANGE_TOUCH_MODE_MENU_ID = Menu.FIRST + 3;
```

```
    public final static String EXTRA_MESSAGE2 =
    "com.example.myfirstapp.MESSAGE2";
    public final static String EXTRA_MESSAGE3 =
    "com.example.myfirstapp.MESSAGE3";
    public final static String EXTRA_MESSAGE4 =
    "com.example.myfirstapp.MESSAGE4";
    public final static String EXTRA_MESSAGE5 =
    "com.example.myfirstapp.MESSAGE5";
    public final static String EXTRA_MESSAGE6 =
    "com.example.myfirstapp.MESSAGE6";
    public final static String EXTRA_MESSAGE7 =
    "com.example.myfirstapp.MESSAGE7";
    public final static String EXTRA_MESSAGE8 =
    "com.example.myfirstapp.MESSAGE8";
    public final static String EXTRA_MESSAGE9 =
    "com.example.myfirstapp.MESSAGE9";
    public final static String EXTRA_MESSAGE10
    = "com.example.myfirstapp.MESSAGE10";

    public int correctAnswerListener;
    public String lessonmodule = "0";
    public int intlessnmodule = 0;

    public Handler mHandler = new Handler();
    public Handler mHandler2 = new Handler();

    public MediaPlayer mp;
    public String imgTag1;
    public String imgTag2;
    public String imgTag3;
    public String imgTag4;

    public ImageView newView1;
    public ImageView newView2;
    public ImageView newView3;
    public ImageView newView4;

    public String audTag1;
    public String audTag2;
    public String audTag3;
    public String audTag4;

    public String tvString;
    public String tv2String;

    public String[] ranAuds;

    public int ind = 0;
    public String[] myShuffledArray;
    public String[] quesDescription;
    public Context context;
    public MyDBEmbrASD db;
    public String selectedImg;
    public String selectedAud;
    public String selectedAud2;
    public String selectedCorrAnsAud;
    public String selectedImgDesc;
    public String selectedfollowupquesaud;
    public String selectedfollowupquestext;

    public int dmw; //get screen dimensions
    public int dmh; //get screen dimensions

    public String[] corrAnsIMPID;
    public String[] corrAnsr;
    public String[] quesDesc;
    public String[] qImgFP;
    public String[] qAudFP;
    public String[] corrAnsAUDID;
```



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740
Bachelor of Science in Information Technology

67

```
public int numQues;
public int iQues;
public int iplayed;

/**
 *
 */
// Variables

private DragController mDragController;
// Object that sends out drag-drop events while
a view is being moved.
private DragLayer mDragLayer;
// The ViewGroup that supports drag-drop.
private DropSpot mSpot2;
// The DropSpot that can be turned on and off
via the menu.

private boolean mLongClickStartsDrag =
false; // If true, it takes a long click to
start the drag operation.
public boolean showbutton = false;
// Otherwise, any touch event starts a
drag.

public static final boolean Debugging =
false;
private DragController controller;

private boolean mIsBound = false;
private MusicService mServ;
private ServiceConnection Scon =new
ServiceConnection(){

    public void
onServiceConnected(ComponentName name, IBinder
binder) {

        mServ =
((MusicService.ServiceBinder)binder).getService
();
    }

    public void
onServiceDisconnected(ComponentName name) {
        mServ = null;
    }
};

void doBindService(){
    bindService(new
Intent(this,MusicService.class),
Scon,
Context.BIND_AUTO_CREATE);
    mIsBound = true;
}

void doUnbindService()
{
    if(mIsBound)
    {
        unbindService(Scon);
        mIsBound = false;
    }
}

/**
 *
 */
// Methods

/**
 * onCreate - called when the activity is
```

```
first created.
 * <p>
 * Creates a drag controller and sets up
three views so click and long click on the
views are sent to this activity.
 * The onLongClick method starts a drag
sequence.
 */
@Override
protected void onCreate(Bundle
savedInstanceState) {

    super.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
    super.onCreate(savedInstanceState);

    requestWindowFeature(Window.FEATURE_NO_TITLE);
    mDragController = new
DragController(this);

}

private void doStuff() {

this.getWindow().clearFlags(WindowManager.LayoutParams.FLAG_NOT_TOUCHABLE);
}
/**
 * Build a menu for the activity.
 */

/**
 * Handle a click on a view.
 */

public void onClick(View v) {
    if (mLongClickStartsDrag) {
        // Tell the user that it takes a
long click to start dragging.
        toast("Press and hold to drag an
image.");
    }

}

private void goBlooey(Throwable t) {
    AlertDialog.Builder builder=new
AlertDialog.Builder(this);

    builder.setTitle("Exception!")
        .setMessage(t.toString());
}

/**
 * Handle a long click.
 *
 * @param v View
 * @return boolean - true indicates that
the event was handled
 */

public boolean onLongClick(View v) {
    if (mLongClickStartsDrag) {

        //trace ("onLongClick in view: " +
v + " touchMode: " + v.isInTouchMode ());

        // Make sure the drag was started
by a long press as opposed to a long click.
        // (Note: I got this from the
```



```
Workspace object in the Android Launcher code.
// I think it is here to ensure
that the device is still in touch mode as we
start the drag operation.)
    if (!v.isInTouchMode()) {
        toast("isInTouchMode returned
false. Try touching the view again.");
        return false;
    }
    return startDrag(v);
}

// If we get here, return false to
indicate that we have not taken care of the
event.
return false;
}

/**
 * Perform an action in response to a menu
item being clicked.
 */

/**
 * This is the starting point for a drag
operation if mLongClickStartsDrag is false.
 * It looks for the down event that gets
generated when a user touches the screen.
 * Only that initiates the drag-drop
sequence.
 */

public boolean onTouch(View v, MotionEvent
ev) {
    // If we are configured to start only
on a long click, we are not going to handle any
events here.
    if (mLongClickStartsDrag) return false;

    boolean handledHere = false;

    final int action = ev.getAction();
    if (v.getTag() == imgTag1)

        try {

            new
TryCatchStopMediaPlayer().invoke();

            int audID =
getResources().getIdentifier(audTag1, "raw",
getPackageName());
            mp= MediaPlayer.create(this,
audID);
            mp.setLooping(false);
            mp.setVolume(100, 100);

            mp.setOnCompletionListener(releaseCompletion);
            mp.start();
        }
        catch (Throwable t) {
            goBloey(t);
        }
        if (v.getTag() == imgTag2)

            try {
                new
TryCatchStopMediaPlayer().invoke();

                int audID =
getResources().getIdentifier(audTag2, "raw",
```

```
getPackageName());
                mp= MediaPlayer.create(this,
audID);
                mp.setLooping(false);
                mp.setVolume(100, 100);

                mp.setOnCompletionListener(releaseCompletion);
                mp.start();
            }
            catch (Throwable t) {
                goBloey(t);
            }
            if (imgTag3 != null) {
                if (v.getTag() == imgTag3)

                    try {
                        new
TryCatchStopMediaPlayer().invoke();

                        int audID =
getResources().getIdentifier(audTag3, "raw",
getPackageName());
                        mp =
MediaPlayer.create(this, audID);
                        mp.setLooping(false);
                        mp.setVolume(100, 100);

                        mp.setOnCompletionListener(releaseCompletion);
                        mp.start();
                    }
                    catch (Throwable t) {
                        goBloey(t);
                    }
            }
            if (imgTag4 != null) {
                if (v.getTag() == imgTag4)

                    try {
                        new
TryCatchStopMediaPlayer().invoke();
                        int audID =
getResources().getIdentifier(audTag4, "raw",
getPackageName());
                        mp =
MediaPlayer.create(this, audID);
                        mp.setLooping(false);
                        mp.setVolume(100, 100);

                        mp.setOnCompletionListener(releaseCompletion);
                        mp.start();
                    }
                    catch (Throwable t) {
                        goBloey(t);
                    }
            }

            // In the situation where a long click
is not needed to initiate a drag, simply start
on the down event.
            if (action == MotionEvent.ACTION_DOWN)
            {
                handledHere = startDrag(v);

                if (v.getTag() != selectedImg) {
                    mSpot2.setDragLayer(null);
                } else {
                    mSpot2.setDragLayer(mDragLayer);
                }
            }
            return handledHere;
        }
    }
```



```
/**
 * Start dragging a view.
 */

public boolean startDrag(View v) {
    // Let the DragController initiate a
    drag-drop sequence.
    // I use the dragInfo to pass along the
    object being dragged.
    // I'm not sure how the Launcher
    designers do this.
    Object dragInfo = v;

    mDragController.startDrag(v,
    mDragLayer, dragInfo,
    DragController.DRAG_ACTION_MOVE);
    return true;
}

/**
 * Finds all the views we need and
    configure them to send click events to the
    activity.
 */
private void setupViews() {
    DragController dragController =
    mDragController;

    mDragLayer = (DragLayer)
    findViewById(R.id.drag_layer);
    mDragLayer.setDragController(dragController);
    dragController.addDropTarget(mDragLayer);

    context = this;
    db = new MyDBEmbrASD(this);

    List<ResourceTable> resources =
    db.getAllResource(lessonmodule);
    String[] imgFPString = new
    String[resources.size()];
    String[] audFPString = new
    String[resources.size()];

    for (int i = 0; i < resources.size();
    i++) {
        imgFPString[i] =
        resources.get(i).getimgFP();
        audFPString[i] =
        resources.get(i).getaudioFP();
    }

    GetAnswerTables getAnswerTables = new
    GetAnswerTables().invoke();

    numQues = getAnswerTables.getNumQues();
    corrAnsIMPID =
    getAnswerTables.getCorrAnsIMPID();
    corrAns =
    getAnswerTables.getCorrAns();
    quesDesc =
    getAnswerTables.getQuesDesc();
    qImgFP = getAnswerTables.getqImgFP();
    qAudFP = getAnswerTables.getqAudFP();
    corrAnsAUDID =
    getAnswerTables.getCorrAnsIMPID();

    Random rQuestion = new Random();
    iQues = rQuestion.nextInt(numQues - 1)
    + 1;

    int sizeOfRes = imgFPString.length;
    int resChoice1;
    int resChoice2;
    int resChoice3;

    Random rResSelector = new Random();
    resChoice1 =
    rResSelector.nextInt((sizeOfRes - 1)+1);
    resChoice2 =
    rResSelector.nextInt((sizeOfRes - 1)+1);
    resChoice3 =
    rResSelector.nextInt((sizeOfRes - 1)+1);

    System.out.println("iQues" + iQues);
    System.out.println("resChoice1 " +
    resChoice1);
    System.out.println("resChoice2 " +
    resChoice2);
    System.out.println("resChoice3 " +
    resChoice3);

    String[] cimgFPString = new String[3];
    String[] caudFPString = new String[3];

    cimgFPString[0] =
    imgFPString[resChoice1];
    caudFPString[0] =
    audFPString[resChoice1];

    cimgFPString[1] =
    imgFPString[resChoice2];
    caudFPString[1] =
    audFPString[resChoice2];

    cimgFPString[2] =
    imgFPString[resChoice3];
    caudFPString[2] =
    audFPString[resChoice3];

    while (corrAnsIMPID[iQues].equals
    (cimgFPString[0]) || corrAnsIMPID[iQues].equals
    (cimgFPString[1]) ||
    corrAnsIMPID[iQues].equals
    (cimgFPString[2]) ||
    cimgFPString[0].equals(cimgFPString[1]) ||
    cimgFPString[0].equals(cimgFPString[2]) ||
    cimgFPString[1].equals(cimgFPString[2])){
        resChoice1 =
        rResSelector.nextInt((sizeOfRes - 1)+1);
        resChoice2 =
        rResSelector.nextInt((sizeOfRes - 1)+1);
        resChoice3 =
        rResSelector.nextInt((sizeOfRes - 1)+1);
        cimgFPString[0] =
        imgFPString[resChoice1];
        caudFPString[0] =
        audFPString[resChoice1];

        cimgFPString[1] =
        imgFPString[resChoice2];
        caudFPString[1] =
        audFPString[resChoice2];

        cimgFPString[2] =
        imgFPString[resChoice3];
        caudFPString[2] =
        audFPString[resChoice3];

        System.out.println("corransimpid "
        + corrAnsIMPID[iQues]);
    }
}
```



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

70

```
System.out.println("cimpfpstring1 "
+ cimgFPString[0]);
System.out.println("cimpfpstring2 "
+ cimgFPString[1]);
System.out.println("cimpfpstring3 "
+ cimgFPString[2]);
}
System.out.println("corransimpid " +
corrAnsIMPID[iQues]);
System.out.println("cimpfpstring0 " +
cimgFPString[0]);
System.out.println("cimpfpstring0 " +
cimgFPString[1]);
System.out.println("cimpfpstring0 " +
cimgFPString[2]);
```

```
selectedImg = corrAnsIMPID[iQues];
selectedAud = qAudFP[iQues];
int id5 =
getResources().getIdentifier(corrAnsIMPID[iQues]
, "drawable", getPackageName());
```

```
ArrayList<String> quesList = new
ArrayList<>();
quesList.add(corrAnsIMPID[iQues]);
quesList.add(cimgFPString[0]);
quesList.add(cimgFPString[1]);
quesList.add(cimgFPString[2]);
String[] ranQues = new
String[quesList.size()];
ranQues = quesList.toArray(ranQues);
for (String s : ranQues) ;
```

```
ArrayList<String> audList = new
ArrayList<>();
audList.add(corrAnsAUDID[iQues]);
audList.add(caudFPString[0]);
audList.add(caudFPString[1]);
audList.add(caudFPString[2]);
String[] ranAuds = new
String[audList.size()];
ranAuds = audList.toArray(ranAuds);
for (String s : ranAuds) ;
```

```
audTag1 = ranAuds[0];
audTag2 = ranAuds[1];
audTag3 = ranAuds[2];
audTag4 = ranAuds[3];
```

```
DisplayMetrics size = new
DisplayMetrics();
```

```
getWindowManager().getDefaultDisplay().getMetri
cs(size);
```

```
dmw = size.widthPixels;
dmh = size.heightPixels;
```

```
System.out.println("displaymetrics w" +
dmw);
```

```
System.out.println("displaymetrics h" +
dmh);
```

```
/////
newView1 = new ImageView(this);
int id1 =
getResources().getIdentifier(ranQues[0],
"drawable", getPackageName());
newView1.setImageResource(id1);
//newView1.setImageResource
(android.R.drawable.alert_dark_frame);
Random r = new Random();
```

```
int w1 = (int) (dmw * .2) + 50;
int h1 = (int) (dmh * .2) + 50;
int left1 = r.nextInt((int) (dmw * .3));
int top1 = r.nextInt((int) (dmw * .3)
));
```

```
DragLayer.LayoutParams lp1 = new
DragLayer.LayoutParams(w1, h1, left1, top1);
```

```
mDragLayer.addView(newView1, lp1);
newView1.setOnClickListener(this);
newView1.setOnLongClickListener(this);
newView1.setOnTouchListener(this);
newView1.setTag(ranQues[0]);
imgTag1 = (String) newView1.getTag();
```

```
/////
```

```
/////
newView2 = new ImageView(this);
int id2 =
getResources().getIdentifier(ranQues[1],
"drawable", getPackageName());
newView2.setImageResource(id2);
//newView2.setImageResource
(android.R.drawable.alert_dark_frame);
int w2 = (int) (dmw * .2) + 50;
int h2 = (int) (dmh * .2) + 50;
int left2 = r.nextInt((int) (dmw *
.8)+200);
int top2 = r.nextInt((int) ((dmw * .7)
));
```

```
DragLayer.LayoutParams lp2 = new
DragLayer.LayoutParams(w2, h2, left2, top2);
mDragLayer.addView(newView2, lp2);
newView2.setOnClickListener(this);
newView2.setOnLongClickListener(this);
newView2.setOnTouchListener(this);
newView2.setTag(ranQues[1]);
imgTag2 = (String) newView2.getTag();
/////
```

```
/////
newView3 = new ImageView(this);
int id3 =
getResources().getIdentifier(ranQues[2],
"drawable", getPackageName());
newView3.setImageResource(id3);
//newView3.setImageResource
(android.R.drawable.alert_dark_frame);
int w3 = (int) (dmw * .2) + 50;
int h3 = (int) (dmh * .2) + 50;
int left3 = r.nextInt((int) (dmw * .8));
int top3 = r.nextInt((int) ((dmw * .7)
));
```

```
DragLayer.LayoutParams lp3 = new
DragLayer.LayoutParams(w3, h3, left3, top3);
mDragLayer.addView(newView3, lp3);
newView3.setOnClickListener(this);
newView3.setOnLongClickListener(this);
newView3.setOnTouchListener(this);
newView3.setTag(ranQues[2]);
imgTag3 = (String) newView3.getTag();
/////
```

```
/////
newView4 = new ImageView(this);
int id4 =
getResources().getIdentifier(ranQues[3],
"drawable", getPackageName());
newView4.setImageResource(id4);
//newView4.setImageResource
```




```
(android.R.drawable.alert_dark_frame);
int w4 = (int) (dmw * .2) + 50;
int h4 = (int) (dmh * .2) + 50;
int left4 = r.nextInt((int) (dmw *
.8));
int top4 = r.nextInt((int) ((dmw * .7)
));

DragLayer.LayoutParams lp4 = new
DragLayer.LayoutParams(w4, h4, left4, top4);
mDragLayer.addView(newView4, lp4);
newView4.setOnClickListener(this);
newView4.setOnLongClickListener(this);
newView4.setOnTouchListener(this);
newView4.setTag(ranQues[3]);
imgTag4 = (String) newView4.getTag();
/////
System.out.println("imgTag1 " +
imgTag1);
System.out.println("imgTag2 " +
imgTag2);
System.out.println("imgTag3 " +
imgTag3);
System.out.println("imgTag4 " +
imgTag4);

TextView tv = (TextView)
findViewById(R.id.tv1);
tv.setTextSize(25);
tv.setText(quesDesc[iQues]);

DropSpot drop2 = (DropSpot)
mDragLayer.findViewById(R.id.drop_spot2);
drop2.setup(null, dragController,
R.color.drop_target_color2);

mSpot2 = drop2;
mSpot2.mListener = DragActivityV4.this;

// Note: It might be interesting to
allow the drop spots to be movable too.
// Unfortunately, in the current
implementation, that does not work
// because the parent view of the
DropTarget objects is not the drag layer.
// The current DragLayer.onDrop method
makes assumptions about how to reposition a
dropped view.

// Give the user a little guidance.
//String message = mLongClickStartsDrag
? "Press and hold to start dragging."
//
: "Touch a view to start dragging.";
//Toast.makeText
(getApplicationContext(), message,
Toast.LENGTH_LONG).show ();
}

private void setupViews2() {
DragController dragController =
mDragController;

mDragLayer = (DragLayer)
findViewById(R.id.drag_layer);
mDragLayer.setDragController(dragController);

dragController.addDropTarget(mDragLayer);

context = this;
db = new MyDBEmbrASD(this);

List<moduleFourGetterSetter> resources
= db.moduleFourGetterSetter(lessonmodule);

numQues = resources.size();

String[] quesNum1 = new
String[resources.size()];
String[] quesDesc1 = new
String[resources.size()];
String[] quesDesc2 = new
String[resources.size()];
String[] choiceA = new
String[resources.size()];
String[] choiceB = new
String[resources.size()];
String[] corrChoice = new
String[resources.size()];
String[] descIMGFP = new
String[resources.size()];
String[] quesDescAUDFP = new
String[resources.size()];
String[] corrAnsAUDFP = new
String[resources.size()];
String[] choiceAAUDFP = new
String[resources.size()];
String[] choiceBAUDFP = new
String[resources.size()];
String[] lessonmodule = new
String[resources.size()];
String[] QuesDescAUDFP2 = new
String[resources.size()];
String[] followupquesaud = new
String[resources.size()];
String[] followupquestext = new
String[resources.size()];

for (int i = 0; i < resources.size();
i++) {
quesNum1[i] =
resources.get(i).getQuesNum1();
quesDesc1[i] =
resources.get(i).getquesDesc1();
quesDesc2[i] =
resources.get(i).getquesDesc2();
choiceA[i] =
resources.get(i).getchoiceA();
choiceB[i] =
resources.get(i).getchoiceB();
corrChoice[i] =
resources.get(i).getcorrChoice();
descIMGFP[i] =
resources.get(i).getdescIMGFP();
quesDescAUDFP[i] =
resources.get(i).getquesDescAUDFP();
corrAnsAUDFP[i] =
resources.get(i).getcorrANSAUDFP();
choiceAAUDFP[i] =
resources.get(i).getchoiceAAUDFP();
choiceBAUDFP[i] =
resources.get(i).getchoiceBAUDFP();
lessonmodule[i] =
resources.get(i).getlessonmodule();
QuesDescAUDFP2[i] =
resources.get(i).getQuesDescAUDFP2();
}
```




Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

72

```
        followupquesaud[i] =
resources.get(i).getfollowupquesaud();
        followupquestext[i] =
resources.get(i).getfollowupquestext();
    }

    Random rQuestion = new Random();
    iQues = rQuestion.nextInt(numQues);
    selectedImg = corrChoice[iQues];
    selectedAud = quesDescAUDFP[iQues];
    selectedfollowupquesaud =
followupquesaud[iQues];
    selectedfollowupquestext =
followupquestext[iQues];

    if (QuesDescAUDFP2[iQues] != null) {
        selectedAud2 =
QuesDescAUDFP2[iQues];
        System.out.println("selectedAud2" +
selectedAud2);
    }
    selectedCorrAnsAud =
corrAnsAUDFP[iQues];
    selectedImgDesc = descIMGFP[iQues];

    ArrayList<String> quesList = new
ArrayList<>();
    quesList.add(corrChoice[iQues]);
    quesList.add(choiceB[iQues]);
    String[] ranQues = new
String[quesList.size()];
    ranQues = quesList.toArray(ranQues);

    for (String s : ranQues) ;

    ArrayList<String> audList = new
ArrayList<>();
    audList.add(choiceAAUDFP[iQues]);
    audList.add(choiceBAUDFP[iQues]);
    String[] ranAuds = new
String[audList.size()];
    ranAuds = audList.toArray(ranAuds);
    for (String s : ranAuds) ;

    audTag1 = ranAuds[0];
    audTag2 = ranAuds[1];

    DisplayMetrics size = new
DisplayMetrics();

getWindowManager().getDefaultDisplay().getMetri
cs(size);
    dmw = size.widthPixels;
    dmh = size.heightPixels;

    System.out.println("displaymetrics w" +
dmw);
    System.out.println("displaymetrics h" +
dmh);

    Random r = new Random();

    /////
    newView1 = new ImageView(this);
    int id1 =
getResources().getIdentifier(ranQues[0],
"drawable", getPackageName());
    newView1.setImageResource(id1);
    int w1 = (int) (dmw * .2) + 200;
    int h1 = (int) (dmh * .2) + 100;

    int left1 = r.nextInt((int) (dmw *
.8));
    int top1 = r.nextInt((int) ((dmw * .8)
));
    DragLayer.LayoutParams lp1 = new
DragLayer.LayoutParams(w1, h1, left1, top1);
    mDragLayer.addView(newView1, lp1);
    newView1.setOnClickListener(this);
    newView1.setOnLongClickListener(this);
    newView1.setOnTouchListener(this);
    newView1.setTag(ranQues[0]);
    imgTag1 = (String) newView1.getTag();
    /////

    newView2 = new ImageView(this);
    int id2 =
getResources().getIdentifier(ranQues[1],
"drawable", getPackageName());
    newView2.setImageResource(id2);
    int w2 = (int) (dmw * .2) + 200;
    int h2 = (int) (dmh * .2) + 100;
    int left2 = r.nextInt((int) (dmw * .4));
    int top2 = r.nextInt((int) ((dmw * .8)
));
    DragLayer.LayoutParams lp2 = new
DragLayer.LayoutParams(w2, h2, left2, top2);
    mDragLayer.addView(newView2, lp2);
    newView2.setOnClickListener(this);
    newView2.setOnLongClickListener(this);
    newView2.setOnTouchListener(this);
    newView2.setTag(ranQues[1]);
    imgTag2 = (String) newView2.getTag();
    /////

    TextView tv = (TextView)
findViewById(R.id.tv1);
    tv.setTextSize(25);
    tv.setText(quesDesc1[iQues]);

    tvString = quesDesc1[iQues];
    TextView tv2 = (TextView)
findViewById(R.id.tv2);
    tv2.setTextSize(25);
    tv2.setText(quesDesc2[iQues]);

    tv2String = quesDesc2[iQues];

    ImageView img = (ImageView)
findViewById(R.id.qimg);
    int intimg =
getResources().getIdentifier(selectedImgDesc,
"drawable", getPackageName());
    img.setImageResource(intimg);

    System.out.println("imgTag1 " +
imgTag1);
    System.out.println("imgTag2 " +
imgTag2);
    System.out.println("selectedImgDesc " +
selectedImgDesc);
    System.out.println("tvString " +
tvString);
    System.out.println("tv2String " +
tv2String);

    DropSpot drop2 = (DropSpot)
mDragLayer.findViewById(R.id.drop_spot2);
    drop2.setup(null, dragController,
```



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

73

```
R.color.drop_target_color2);

mSpot2 = drop2;
mSpot2.mListener = DragActivityV4.this;

// Note: It might be interesting to
allow the drop spots to be movable too.
// Unfortunately, in the current
implementation, that does not work
// because the parent view of the
DropTarget objects is not the drag layer.
// The current DragLayer.onDrop method
makes assumptions about how to reposition a
dropped view.

// Give the user a little guidance.
//String message = mLongClickStartsDrag
? "Press and hold to start dragging."
//
: "Touch a view to start dragging.";
//Toast.makeText
(getApplicationContext(), message,
Toast.LENGTH_LONG).show ();

}

@Override
public void onRightAnswerSelected() {

//Intent intent = getIntent();
//startActivity(intent);
//finish();
Intent intent2 = getIntent();
lessonmodule =
intent2.getStringExtra(MainActivity.EXTRA_MESSA
GE);

if (lessonmodule == null) {
    lessonmodule =
intent2.getStringExtra(CongratulationsActivity.
EXTRA_MESSAGE11);
}
int intlessnmodule =
Integer.parseInt(lessonmodule);
System.out.println("intlessnmodule " +
intlessnmodule);

if (intlessnmodule != 4 &&
intlessnmodule !=5) {
    newView1.setEnabled(false);
    newView2.setEnabled(false);
    newView3.setEnabled(false);
    newView4.setEnabled(false);
}
if (intlessnmodule == 4 ||
intlessnmodule == 5) {
    newView1.setEnabled(false);
    newView2.setEnabled(false);
}

mHandler2.postDelayed(new Runnable() {
    public void run() {
        doStuff2();
    }
}, 800);

}

private void doStuff2() {
    Intent intent2 = getIntent();
    lessonmodule =

intent2.getStringExtra(MainActivity.EXTRA_MESSA
GE);
    if (lessonmodule == null) {
        lessonmodule =
intent2.getStringExtra(CongratulationsActivity.
EXTRA_MESSAGE11);
    }
    int intlessnmodule =
Integer.parseInt(lessonmodule);
    if (intlessnmodule != 4 &&
intlessnmodule !=5) {
        Intent intent = new Intent(this,
CongratulationsActivity.class);
        intent.putExtra(EXTRA_MESSAGE2,
selectedImg);
        intent.putExtra(EXTRA_MESSAGE3,
selectedAud);
        intent.putExtra(EXTRA_MESSAGE4,
qImgFP[iQues]);
        intent.putExtra(EXTRA_MESSAGE5,
lessonmodule);
        startActivity(intent);
    }
    if (intlessnmodule == 4 ){
        Intent intent = new Intent(this,
CongratulationsActivity.class);
        intent.putExtra(EXTRA_MESSAGE2,
selectedImg);
        intent.putExtra(EXTRA_MESSAGE3,
selectedCorrAnsAud);
        intent.putExtra(EXTRA_MESSAGE4,selectedImgDesc
);
        intent.putExtra(EXTRA_MESSAGE5,lessonmodule );
        intent.putExtra(EXTRA_MESSAGE6,tvString);
        intent.putExtra(EXTRA_MESSAGE7,tv2String);
        startActivity(intent);
    }
    if (intlessnmodule == 5 ){
        Intent intent = new Intent(this,
CongratulationsActivity.class);
        intent.putExtra(EXTRA_MESSAGE2,
selectedImg);
        intent.putExtra(EXTRA_MESSAGE3,
selectedCorrAnsAud);
        intent.putExtra(EXTRA_MESSAGE4,selectedImgDesc
);
        intent.putExtra(EXTRA_MESSAGE5,lessonmodule );
        intent.putExtra(EXTRA_MESSAGE6,tvString);
        intent.putExtra(EXTRA_MESSAGE7,tv2String);
        intent.putExtra(EXTRA_MESSAGE8,selectedfollowup
quesaud);
        intent.putExtra(EXTRA_MESSAGE9,selectedfollowup
questext);
        startActivity(intent);
    }
    finish();
}
```



```
/**
 * Show a string on the screen via Toast.
 *
 * @param msg String
 * @return void
 */

public void toast(String msg) {
    Toast.makeText(getApplicationContext(),
msg, Toast.LENGTH_SHORT).show();
} // end toast

/**
 * Send a message to the debug log and
display it using Toast.
 */

public void trace(String msg) {
    if (!Debugging) return;
    Log.d("DragActivity", msg);
    toast(msg);
}

private class GetQuestionsTable {
    public String[] invoke() {
        List<ResourceTable> resources =
db.getAllResource(lessonmodule);
        String[] imgFPString = new
String[resources.size()];
        String[] temprdm = new
String[resources.size()];

        for (int i = 0; i <
resources.size(); i++) {
            imgFPString[i] =
resources.get(i).getimgFP();
            temprdm[i] = imgFPString[i];
        }

        List<String> wordList =
Arrays.asList(temprdm);
        Collections.shuffle(wordList);
        return wordList.toArray(new
String[wordList.size()]);
    }
}

public void onStart(){
    super.onStart();
    new StartBGServices().invoke();

    Intent intent = getIntent();

    lessonmodule =
intent.getStringExtra(MainActivity.EXTRA_MESSAG
E);
    if (lessonmodule == null) {
        lessonmodule =
intent.getStringExtra(CongratulationsActivity.E
XTRA_MESSAGE11);
    }
    int intlessnmodule =
Integer.parseInt(lessonmodule);
    if (Build.VERSION.SDK_INT <= 16) {
        if (intlessnmodule == 1 ||
intlessnmodule == 2) {
            setContentView(R.layout.main2jb);
        }
        if (intlessnmodule == 3) {
            setContentView(R.layout.main3jb);
        }
        if (intlessnmodule == 4) {
            setContentView(R.layout.main4jb);
        }
        if (intlessnmodule == 5) {
            setContentView(R.layout.main5jb);
        }
        else {
            if (intlessnmodule == 1 ||
intlessnmodule == 2) {
                setContentView(R.layout.main2);
                final VideoView myVideoView =
                (VideoView) findViewById(R.id.videoView);
                try {
                    myVideoView.setVideoURI(Uri.parse("android.reso
urce://" + getPackageName() + "/" +
R.raw.bgvidact));
                } catch (Exception e) {
                    Log.e("Error",
e.getMessage());
                    e.printStackTrace();
                }

                myVideoView.setOnPreparedListener(new
MediaPlayer.OnPreparedListener() {
                    @Override
                    public void
onPrepared(MediaPlayer mp) {
                        mp.setLooping(true);
                    }
                });

                myVideoView.start();

                myVideoView.setOnCompletionListener(new
MediaPlayer.OnCompletionListener() {
                    @Override
                    public void
onCompletion(MediaPlayer mp) {
                        myVideoView.start();
                    }
                });
            }
            if (intlessnmodule == 4) {
                setContentView(R.layout.main4);
                final VideoView myVideoView =
                (VideoView) findViewById(R.id.videoView);
                try {
                    myVideoView.setVideoURI(Uri.parse("android.reso
urce://" + getPackageName() + "/" +
R.raw.bgvidact));
                } catch (Exception e) {
                    Log.e("Error",
e.getMessage());
                    e.printStackTrace();
                }

                myVideoView.setOnPreparedListener(new
MediaPlayer.OnPreparedListener() {
```



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740
Bachelor of Science in Information Technology

75

```
@Override
public void
onPrepared(MediaPlayer mp) {
    mp.setLooping(true);
}

myVideoView.start();

myVideoView.setOnCompletionListener(new
MediaPlayer.OnCompletionListener() {

    @Override
    public void
    onCompletion(MediaPlayer mp) {

        myVideoView.start();

    }
});

if (intlessnmodule == 5) {
    setContentView(R.layout.main5);
    final VideoView myVideoView =
    (VideoView) findViewById(R.id.videoView);
    try {

myVideoView.setVideoURI(Uri.parse("android.reso
urce://" + getPackageName() + "/" +
R.raw.bgvidact));

    } catch (Exception e) {
        Log.e("Error",
e.getMessage());
        e.printStackTrace();
    }

myVideoView.setOnPreparedListener(new
MediaPlayer.OnPreparedListener() {
    @Override
    public void
    onPrepared(MediaPlayer mp) {
        mp.setLooping(true);
    }
});

myVideoView.start();

myVideoView.setOnCompletionListener(new
MediaPlayer.OnCompletionListener() {

    @Override
    public void
    onCompletion(MediaPlayer mp) {

        myVideoView.start();

    }
});

if (intlessnmodule == 3){
    setContentView(R.layout.main3);
    final VideoView myVideoView =
    (VideoView) findViewById(R.id.videoView);
    try {

myVideoView.setVideoURI(Uri.parse("android.reso
urce://" + getPackageName() + "/" +
R.raw.bgvidact));

    } catch (Exception e) {
        Log.e("Error",
```

```
e.getMessage());
        e.printStackTrace();
    }

myVideoView.setOnPreparedListener(new
MediaPlayer.OnPreparedListener() {
    @Override
    public void
    onPrepared(MediaPlayer mp) {
        mp.setLooping(true);
    }
});

myVideoView.start();

myVideoView.setOnCompletionListener(new
MediaPlayer.OnCompletionListener() {

    @Override
    public void
    onCompletion(MediaPlayer mp) {

        myVideoView.start();

    }
});

if (intlessnmodule == 4 ||
intlessnmodule == 5){
    setupViews2();
}
else {
    setupViews();
}

this.getWindow().addFlags(WindowManager.LayoutParams
.FLAG_NOT_TOUCHABLE);
startQuestionAudio();

if (intlessnmodule == 5) {
    mHandler.postDelayed(new Runnable()
{
        public void run() {
            doStuff();
        }
    }, 10000);
}
else {
    mHandler.postDelayed(new Runnable()
{
        public void run() {
            doStuff();
        }
    }, 3000);
}

}

private void startQuestionAudio() {
    Intent intent = getIntent();
    lessonmodule =
intent.getStringExtra(MainActivity.EXTRA_MESSAG
E);

    if (lessonmodule == null) {
        lessonmodule =
intent.getStringExtra(CongratulationsActivity.E
XTRA_MESSAGE11);
    }
    int intlessnmodule =
```



```
Integer.parseInt(lessonmodule);
System.out.println("selectedAud " +
selectedAud);
System.out.println("selectedAud2 " +
selectedAud2);

if (intlessnmodule != 4 &&
intlessnmodule !=5) {
    if (qImgFP[iQues] != null) {
        System.out.println("qIMGFP " +
qImgFP[iQues]);
        ImageView qv = new
        ImageView(this);
        qv = (ImageView)
        findViewById(R.id.qimg);
        int qimgid =
        getResources().getIdentifier(qImgFP[iQues],
        "drawable", getPackageName());
        qv.setImageResource(qimgid);
    }

    if (intlessnmodule == 1 ||
intlessnmodule == 2 || intlessnmodule == 3) {
        if (qAudFP[iQues] != null) {

            try {
                final int audID =
                getResources().getIdentifier(qAudFP[iQues],
                "raw", getPackageName());

                if(intlessnmodule != 3)
                {
                    mp =
                    MediaPlayer.create(this, R.raw.whichoneisthe);
                }
                if(intlessnmodule == 3)
                {
                    mp =
                    MediaPlayer.create(this,
                    R.raw.whatletterismissingto spelltheword);
                }
                mp.setLooping(false);

                mp.setVolume(100, 100);
                if (mp.isPlaying()) {
                    mp.stop();
                    mp.reset();
                    mp.release();
                    mp = null;
                }
                iplayed = 0;
                mp.start();

                mp.setOnCompletionListener(completeListener);

                } catch (Throwable t) {
                    goBlooeey(t);
                }
            }
        }
    }
    if (intlessnmodule == 4){

        if (selectedAud != null) {

            try {

                final int audID =
                getResources().getIdentifier(selectedAud,
```

```
"raw", getPackageName());

                mp =
                MediaPlayer.create(this, audID);
                mp.setLooping(false);

                mp.setVolume(100, 100);
                if (mp.isPlaying()) {
                    mp.stop();
                    mp.reset();
                    mp.release();
                    mp = null;
                }
                iplayed = 0;
                mp.start();

                mp.setOnCompletionListener(completeListener2);

                } catch (Throwable t) {
                    goBlooeey(t);
                }
            }
        }
    }
    if (intlessnmodule == 5){
        System.out.println("selectAud" +
selectedAud);
        if (selectedAud != null) {

            System.out.println("selectAud"
+ selectedAud);
            try {

                final int audID =
                getResources().getIdentifier(selectedAud,
                "raw", getPackageName());

                mp =
                MediaPlayer.create(this, audID);
                mp.setLooping(false);

                mp.setVolume(100, 100);
                if (mp.isPlaying()) {
                    mp.stop();
                    mp.reset();
                    mp.release();
                    mp = null;
                }
                iplayed = 0;
                mp.start();

                mp.setOnCompletionListener(completeListener3);

                } catch (Throwable t) {
                    goBlooeey(t);
                }
            }
        }
    }

    public void onResume() {
        super.onResume();
        new StartBGServices().invoke();
        if (Build.VERSION.SDK_INT <= 16) {

        }
        else{

            final VideoView myVideoView =
            (VideoView) findViewById(R.id.videoView);
            try {
```



```
myVideoView.setVideoURI(Uri.parse("android.reso
urce://" + getPackageName() + "/" +
R.raw.bgvidact));

    } catch (Exception e) {
        Log.e("Error", e.getMessage());
        e.printStackTrace();
    }

myVideoView.setOnPreparedListener(new
MediaPlayer.OnPreparedListener() {
    @Override
    public void
onPrepared(MediaPlayer mp) {
        mp.setLooping(true);
    }
});

myVideoView.start();

myVideoView.setOnCompletionListener(new
MediaPlayer.OnCompletionListener() {

    @Override
    public void
onCompletion(MediaPlayer mp) {

        myVideoView.start();

    }
});

}

@Override
public void onPause() {
    super.onPause();
    new StopBGMusicServices().invoke();
}

@Override
public void onStop() {
    super.onStop();
    new StopBGMusicServices().invoke();
}

@Override
public void onDestroy() {
    super.onDestroy();
    new StopBGMusicServices().invoke();
}

public void onClickmainmenu(View vmenu) {

    new TryCatchStopMediaPlayer().invoke();

    new StopBGMusicServices().invoke();

    finish();

}

public void onClickplayagain(View
vplayagain) {
    new TryCatchStopMediaPlayer().invoke();

    new StopBGMusicServices().invoke();

    if (mp != null) {
        //mp.stop();
        //mp.reset();
        mp.release();
        mp = null;
    }
    finish();
    startActivity(getIntent());
}

private class GetAnswerTables {
    int numQues;
    String[] quesDesc;
    String[] corrAnsr;
    String[] corrAnsIMPID;
    String[] qImgFP;
    String[] qAudFP;

    int getNumQues() {
        return numQues;
    }

    String[] getQuesDesc() {
        return quesDesc;
    }

    String[] getCorrAnsr() {
        return corrAnsr;
    }

    String[] getCorrAnsIMPID() {
        return corrAnsIMPID;
    }

    String[] getqImgFP() {
        return qImgFP;
    }

    String[] getqAudFP() {
        return qAudFP;
    }

    public GetAnswerTables invoke() {
        List<QuestionGetterSetter>
resources3 =
db.QuestionGetterSetter(lessonmodule);
        numQues = resources3.size();
        String[] quesNum = new
String[resources3.size()];
        quesDesc = new
String[resources3.size()];
        corrAnsr = new
String[resources3.size()];
        corrAnsIMPID = new
String[resources3.size()];
        qImgFP = new
String[resources3.size()];
        qAudFP = new
String[resources3.size()];
        String[] lessnModule = new
String[resources3.size()];
    }
}
```



```
for (int i = 0; i <
resources3.size(); i++) {
    quesNum[i] =
resources3.get(i).getQuesNum();
    quesDesc[i] =
resources3.get(i).getquesDesc();
    corrAnsr[i] =
resources3.get(i).getcorrAns();
    corrAnsIMPID[i] =
resources3.get(i).getcorrAnsIMPID();
    lessnModule[i] =
resources3.get(i).getlessnModule();
    qImgFP[i] =
resources3.get(i).getqImgFP();
    qAudFP[i] =
resources3.get(i).getqAudFP();
}
return this;
}

private class StopBGMusicServices {
    public void invoke() {
        Intent intent2 = getIntent();
        int musicbg =
intent2.getIntExtra("intVariableName",0);

        if (musicbg == 1){
            Intent intentmusic = new
Intent(DragActivityV4.this,
MusicService2.class);
            stopService(intentmusic);
        }
        if (musicbg == 2){
            Intent intentmusic = new
Intent(DragActivityV4.this,
MusicService3.class);
            stopService(intentmusic);
        }
        if (musicbg == 3){
            Intent intentmusic = new
Intent(DragActivityV4.this,
MusicService4.class);
            stopService(intentmusic);
        }

        Intent intentmusic = new
Intent(DragActivityV4.this,
MusicService2.class);
        stopService(intentmusic);

        Intent intentmusic2 = new
Intent(DragActivityV4.this,
MusicService3.class);
        stopService(intentmusic2);

        Intent intentmusic3 = new
Intent(DragActivityV4.this,
MusicService4.class);
        stopService(intentmusic3);
    }
}

private class StartBGServices {
    public void invoke() {
        Intent intent2 = getIntent();
        int musicbg =
intent2.getIntExtra("intVariableName",0);
        if (musicbg == 0){
            musicbg = 1;
        }
    }
}
```

```
if (musicbg == 1){
    Intent intentmusic = new
Intent(DragActivityV4.this,
MusicService2.class);
    startService(intentmusic);
}
if (musicbg == 2){
    Intent intentmusic = new
Intent(DragActivityV4.this,
MusicService3.class);
    startService(intentmusic);
}
if (musicbg == 3){
    Intent intentmusic = new
Intent(DragActivityV4.this,
MusicService4.class);
    startService(intentmusic);
}
}

private class TryCatchStopMediaPlayer {
    public void invoke() {
        try {
            if (mp != null) {
                mp.stop();
                mp.reset();
                mp.release();
                mp = null;
            }
        } catch (Exception e) {
        }
    }

    MediaPlayer.OnCompletionListener
completeListener = new
MediaPlayer.OnCompletionListener() {

        @Override
        public void onCompletion(MediaPlayer
mp) {
            mp.reset();
            mp.release();
            mp = null;
            if (iplayed == 0) {

                final int audID =
getResources().getIdentifier(qAudFP[iQues],
"raw", getPackageName());
                mp =
MediaPlayer.create(context, audID);
                mp.setLooping(false);

                mp.setOnCompletionListener(releaseCompletion);
                mp.setVolume(100, 100);
                if (mp.isPlaying()) {
                    mp.stop();
                    mp.reset();
                    mp.release();
                    mp = null;
                }
                iplayed++;
                mp.start();
                try {
                    Thread.sleep(2000);
                } catch (InterruptedException
e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
else {
```



```
        if (mp != null) {
            mp.reset();
            mp.release();
            mp = null;
        }
    }

    MediaPlayer.OnCompletionListener
    completeListener2 = new
    MediaPlayer.OnCompletionListener() {

        @Override
        public void onCompletion(MediaPlayer
    mp) {
        mp.reset();
        mp.release();
        mp = null;
    }
};

MediaPlayer.OnCompletionListener
completeListener3 = new
MediaPlayer.OnCompletionListener() {

    @Override
    public void onCompletion(MediaPlayer
    mp) {

        mp.reset();
        mp.release();
        mp = null;
        if (selectedAud2 != null) {

            try {

                System.out.println("selectedAud2" +
                selectedAud2);

                final int audID =
                getResources().getIdentifier(selectedAud2,
                "raw", getPackageName());
                mp =
                MediaPlayer.create(context, audID);
                mp.setLooping(false);

                mp.setOnCompletionListener(releaseCompletion);
                mp.setVolume(100, 100);
                mp.start();
            }
            catch (Throwable t) {
                goBlooey(t);
            }
        }
    }
};

MediaPlayer.OnCompletionListener
releaseCompletion = new
MediaPlayer.OnCompletionListener() {
    @Override
    public void onCompletion(MediaPlayer
    mp) {
        if (mp != null) {
            mp.reset();
            mp.release();
            mp = null;
        }
    }
};
```

```
    };
} // end class
```

DragController

```
/*
 * This is a modified version of a class from the Android
 * Open Source Project. The original copyright and license
 * information follows.
 *
 * Copyright (C) 2008 The Android Open Source Project
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
 * either express or implied.
 * See the License for the specific language governing permissions
 * and
 * limitations under the License.
 */
```

```
package com.example.markligot.embrasd.dragview;
```

```
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.Rect;
import android.os.IBinder;
import android.os.Vibrator;
import android.util.DisplayMetrics;
import android.util.Log;
import android.view.KeyEvent;
import android.view.MotionEvent;
import android.view.View;
import android.view.WindowManager;
import android.view.inputmethod.InputMethodManager;
```

```
import java.util.ArrayList;
```

```
/**
 * This class is used to initiate a drag within a view or across multiple
 * views.
 * When a drag starts it creates a special view (a DragView) that
 * moves around the screen
 * until the user ends the drag. As feedback to the user, this object
 * causes the device to
 * vibrate as the drag begins.
 */
```

```
public class DragController extends DragActivityV4 {
```

```
    private static final String TAG = "DragController";
```

```
    /** Indicates the drag is a move. */
```

```
    public static int DRAG_ACTION_MOVE = 0;
```

```
    /** Indicates the drag is a copy. */
```

```
    public static int DRAG_ACTION_COPY = 1;
```

```
    private static final int VIBRATE_DURATION = 35;
```

```
    private static final boolean
```




Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

80

```
PROFILE_DRAWING_DURING_DRAG = false;
```

```
private Context mContext;  
private Vibrator mVibrator;
```

```
// temporaries to avoid gc thrash  
private Rect mRectTemp = new Rect();  
private final int[] mCoordinatesTemp = new int[2];
```

```
/** Whether or not we're dragging. */  
private boolean mDragging;
```

```
/** X coordinate of the down event. */  
private float mMotionDownX;
```

```
/** Y coordinate of the down event. */  
private float mMotionDownY;
```

```
/** Info about the screen for clamping. */  
private DisplayMetrics mDisplayMetrics = new DisplayMetrics();
```

```
/** Original view that is being dragged. */  
private View mOriginator;
```

```
/** X offset from the upper-left corner of the cell to where we  
touched. */  
private float mTouchOffsetX;
```

```
/** Y offset from the upper-left corner of the cell to where we  
touched. */  
private float mTouchOffsetY;
```

```
/** Where the drag originated */  
private DragSource mDragSource;
```

```
/** The data associated with the object being dragged */  
private Object mDragInfo;
```

```
/** The view that moves around while you drag. */  
private DragView mDragView;
```

```
/** Who can receive drop events */  
private ArrayList<DropTarget> mDropTargets = new  
ArrayList<DropTarget>();
```

```
private DragListener mListener;
```

```
/** The window token used as the parent for the DragView. */  
private IBinder mWindowToken;
```

```
private View mMoveTarget;
```

```
private DropTarget mLastDropTarget;
```

```
private InputMethodManager mInputMethodManager;
```

```
public Globals sharedData;
```

```
/**  
 * Interface to receive notifications when a drag starts or stops  
 */
```

```
interface DragListener {
```

```
    /**  
     * A drag has begun  
     */  
    /** @param source An object representing where the drag  
    originated  
     * @param info The data associated with the object that is being  
    dragged  
     * @param dragAction The drag action: either {@link
```

```
DragController#DRAG_ACTION_MOVE}  
 * or {@link DragController#DRAG_ACTION_COPY}  
 */
```

```
void onDragStart(DragSource source, Object info, int  
dragAction);
```

```
/**  
 * The drag has ended  
 */  
void onDragEnd();  
}
```

```
/**  
 * Used to create a new DragLayer from XML.  
 */  
/** @param context The application's context.  
 */
```

```
public DragController(Context context) {  
    mContext = context;  
    mVibrator = (Vibrator)  
context.getSystemService(Context.VIBRATOR_SERVICE);  
}
```

```
/**  
 * Starts a drag.  
 * It creates a bitmap of the view being dragged. That bitmap is  
what you see moving.  
 * The actual view can be repositioned if that is what the onDrop  
handle chooses to do.  
 */
```

```
/** @param v The view that is being dragged  
 * @param source An object representing where the drag  
originated  
 * @param dragInfo The data associated with the object that is  
being dragged  
 * @param dragAction The drag action: either {@link  
#DRAG_ACTION_MOVE} or  
 * {@link #DRAG_ACTION_COPY}  
 */
```

```
public void startDrag(View v, DragSource source, Object dragInfo,  
int dragAction) {
```

```
    // Start dragging, but only if the source has something to drag.  
    boolean doDrag = source.allowDrag ();  
    if (!doDrag) return;
```

```
    mOriginator = v;
```

```
    Bitmap b = getViewBitmap(v);
```

```
    if (b == null) {  
        // out of memory?  
        return;  
    }
```

```
    int[] loc = mCoordinatesTemp;  
    v.getLocationOnScreen(loc);  
    int screenX = loc[0];  
    int screenY = loc[1];
```

```
    startDrag(b, screenX, screenY, 0, 0, b.getWidth(), b.getHeight(),  
source, dragInfo, dragAction);
```

```
    b.recycle();
```

```
    if (dragAction == DRAG_ACTION_MOVE) {  
        v.setVisibility(View.GONE);  
    }  
}
```

```
/**
```



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

81

```
* Starts a drag.
*
* @param b The bitmap to display as the drag image. It will be
re-scaled to the
* enlarged size.
* @param screenX The x position on screen of the left-top of the
bitmap.
* @param screenY The y position on screen of the left-top of the
bitmap.
* @param textureLeft The left edge of the region inside b to use.
* @param textureTop The top edge of the region inside b to use.
* @param textureWidth The width of the region inside b to use.
* @param textureHeight The height of the region inside b to use.
* @param source An object representing where the drag
originated
* @param dragInfo The data associated with the object that is
being dragged
* @param dragAction The drag action: either {@link
#DRAG_ACTION_MOVE} or
* {@link #DRAG_ACTION_COPY}
*/
public void startDrag(Bitmap b, int screenX, int screenY,
int textureLeft, int textureTop, int textureWidth, int
textureHeight,
DragSource source, Object dragInfo, int dragAction)
{
    if (PROFILE_DRAWING_DURING_DRAG) {
        android.os.Debug.startMethodTracing("Launcher");
    }

    // Hide soft keyboard, if visible
    if (mInputMethodManager == null) {
        mInputMethodManager = (InputMethodManager)
mContext.getSystemService(Context.INPUT_METHOD_SERVICE
);
    }

    mInputMethodManager.hideSoftInputFromWindow(mWindowTo
ken, 0);

    if (mListener != null) {
        mListener.onDragStart(source, dragInfo, dragAction);
    }

    int registrationX = ((int)mMotionDownX) - screenX;
    int registrationY = ((int)mMotionDownY) - screenY;

    mTouchOffsetX = mMotionDownX - screenX;
    mTouchOffsetY = mMotionDownY - screenY;

    mDragging = true;
    mDragSource = source;
    mDragInfo = dragInfo;

    mVibrator.vibrate(VIBRATE_DURATION);
    DragView dragView = mDragView = new
DragView(mContext, b, registrationX, registrationY,
textureLeft, textureTop, textureWidth, textureHeight);
    dragView.show(mWindowToken, (int)mMotionDownX,
(int)mMotionDownY);
}

/**
 * Draw the view into a bitmap.
 */
private Bitmap getViewBitmap(View v) {
    v.clearFocus();
    v.setPressed(false);
```

```
boolean willNotCache = v.willNotCacheDrawing();
v.setWillNotCacheDrawing(false);

// Reset the drawing cache background color to fully transparent
// for the duration of this operation
int color = v.getDrawingCacheBackgroundColor();
v.setDrawingCacheBackgroundColor(0);

if (color != 0) {
    v.destroyDrawingCache();
}
v.buildDrawingCache();
Bitmap cacheBitmap = v.getDrawingCache();
if (cacheBitmap == null) {
    Log.e(TAG, "failed getViewBitmap(" + v + ")", new
RuntimeException());
    return null;
}

Bitmap bitmap = Bitmap.createBitmap(cacheBitmap);

// Restore the view
v.destroyDrawingCache();
v.setWillNotCacheDrawing(willNotCache);
v.setDrawingCacheBackgroundColor(color);

return bitmap;
}

/**
 * Call this from a drag source view like this:
 *
 * <pre>
 * @Override
 * public boolean dispatchKeyEvent(KeyEvent event) {
 *     return mDragController.dispatchKeyEvent(this, event)
 *         || super.dispatchKeyEvent(event);
 * }
 * </pre>
 */
public boolean dispatchKeyEvent(KeyEvent event) {
    return mDragging;
}

/**
 * Stop dragging without dropping.
 */
public void cancelDrag() {
    endDrag();
}

private void endDrag() {
    if (mDragging) {
        mDragging = false;

        if (mOriginator != null) {
            mOriginator.setVisibility(View.VISIBLE);
        }
        if (mListener != null) {
            mListener.onDragEnd();
        }
        if (mDragView != null) {
            mDragView.remove();
            mDragView = null;
        }
    }
}

/**
```



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

82

```
* Call this from a drag source view.
*/
public boolean onInterceptTouchEvent(MotionEvent ev) {
    final int action = ev.getAction();

    if (action == MotionEvent.ACTION_DOWN) {
        recordScreenSize();
    }

    final int screenX = clamp((int)ev.getRawX(), 0,
mDisplayMetrics.widthPixels);
    final int screenY = clamp((int)ev.getRawY(), 0,
mDisplayMetrics.heightPixels);

    switch (action) {
        case MotionEvent.ACTION_MOVE:
            break;

        case MotionEvent.ACTION_DOWN:
            // Remember location of down touch
            mMotionDownX = screenX;
            mMotionDownY = screenY;
            mLastDropTarget = null;
            break;

        case MotionEvent.ACTION_CANCEL:
        case MotionEvent.ACTION_UP:
            if (mDragging) {
                drop(screenX, screenY);
            }
            endDrag();
            break;
    }

    return mDragging;
}

/**
 * Sets the view that should handle move events.
 */
void setMoveTarget(View view) {
    mMoveTarget = view;
}

public boolean dispatchUnhandledMove(View focused, int
direction) {
    return mMoveTarget != null &&
mMoveTarget.dispatchUnhandledMove(focused, direction);
}

/**
 * Call this from a drag source view.
 */
public boolean onTouchEvent(MotionEvent ev) {
    if (!mDragging) {
        return false;
    }

    final int action = ev.getAction();
    final int screenX = clamp((int)ev.getRawX(), 0,
mDisplayMetrics.widthPixels);
    final int screenY = clamp((int)ev.getRawY(), 0,
mDisplayMetrics.heightPixels);

    switch (action) {
        case MotionEvent.ACTION_DOWN:
            // Remember where the motion event started
            mMotionDownX = screenX;
            mMotionDownY = screenY;
            break;
        case MotionEvent.ACTION_MOVE:
```

```
// Update the drag view. Don't use the clamped pos here so
the dragging looks
// like it goes off screen a little, instead of bumping up against
the edge.
mDragView.move((int)ev.getRawX(), (int)ev.getRawY());
// Drop on someone?
final int[] coordinates = mCoordinatesTemp;
DropTarget dropTarget = findDropTarget(screenX, screenY,
coordinates);
if (dropTarget != null) {
    if (mLastDropTarget == dropTarget) {
        dropTarget.onDragOver(mDragSource, coordinates[0],
coordinates[1],
(int) mTouchOffsetX, (int) mTouchOffsetY,
mDragView, mDragInfo);
    } else {
        if (mLastDropTarget != null) {
            mLastDropTarget.onDragExit(mDragSource,
coordinates[0], coordinates[1],
(int) mTouchOffsetX, (int) mTouchOffsetY,
mDragView, mDragInfo);
        }
        dropTarget.onDragEnter(mDragSource, coordinates[0],
coordinates[1],
(int) mTouchOffsetX, (int) mTouchOffsetY,
mDragView, mDragInfo);
    }
} else {
    if (mLastDropTarget != null) {
        mLastDropTarget.onDragExit(mDragSource,
coordinates[0], coordinates[1],
(int) mTouchOffsetX, (int) mTouchOffsetY,
mDragView, mDragInfo);
    }
}

mLastDropTarget = dropTarget;

/* The original Launcher activity supports a delete region and
scrolling.
It is not needed in this example.

// Scroll, maybe, but not if we're in the delete region.
boolean inDeleteRegion = false;
if (mDeleteRegion != null) {
    inDeleteRegion = mDeleteRegion.contains(screenX,
screenY);
}
//Log.d(TAG, "inDeleteRegion=" + inDeleteRegion + "
screenX=" + screenX
// + " mScrollZone=" + mScrollZone);
if (!inDeleteRegion && screenX < mScrollZone) {
    if (mScrollState == SCROLL_OUTSIDE_ZONE) {
        mScrollState = SCROLL_WAITING_IN_ZONE;
        mScrollRunnable.setDirection(SCROLL_LEFT);
        mHandler.postDelayed(mScrollRunnable,
SCROLL_DELAY);
    }
} else if (!inDeleteRegion && screenX >
scrollView.getWidth() - mScrollZone) {
    if (mScrollState == SCROLL_OUTSIDE_ZONE) {
        mScrollState = SCROLL_WAITING_IN_ZONE;
        mScrollRunnable.setDirection(SCROLL_RIGHT);
    }
}
```



```
mHandler.postDelayed(mScrollRunnable,
SCROLL_DELAY);
}
} else {
    if (mScrollState == SCROLL_WAITING_IN_ZONE) {
        mScrollState = SCROLL_OUTSIDE_ZONE;
        mScrollRunnable.setDirection(SCROLL_RIGHT);
        mHandler.removeCallbacks(mScrollRunnable);
    }
}
}
*/

break;
case MotionEvent.ACTION_UP:
    if (mDragging) {
        drop(screenX, screenY);
    }
    endDrag();
    break;
case MotionEvent.ACTION_CANCEL:
    cancelDrag();
}

return true;
}

private boolean drop(float x, float y) {

    final int[] coordinates = mCoordinatesTemp;
    DropTarget dropTarget = findDropTarget((int) x, (int) y,
coordinates);

    if (dropTarget != null) {
        dropTarget.onDragExit(mDragSource, coordinates[0],
coordinates[1],
(int) mTouchOffsetX, (int) mTouchOffsetY,
mDragView, mDragInfo);
        if (dropTarget.acceptDrop(mDragSource, coordinates[0],
coordinates[1],
(int) mTouchOffsetX, (int) mTouchOffsetY,
mDragView, mDragInfo)) {

            dropTarget.onDrop(mDragSource, coordinates[0],
coordinates[1],
(int) mTouchOffsetX, (int) mTouchOffsetY,
mDragView, mDragInfo);
            mDragSource.onDropCompleted((View) dropTarget,
true);

            return true;

        } else {
            mDragSource.onDropCompleted((View) dropTarget,
false);
            return true;
        }
    }
    return false;
}

private DropTarget findDropTarget(int x, int y, int[]
dropCoordinates) {
    final Rect r = mRectTemp;

    final ArrayList<DropTarget> dropTargets = mDropTargets;
    final int count = dropTargets.size();
    for (int i=count-1; i>=0; i--) {
        final DropTarget target = dropTargets.get(i);
```

```
target.getHitRect(r);
target.getLocationOnScreen(dropCoordinates);
r.offset(dropCoordinates[0] - target.getLeft(),
dropCoordinates[1] - target.getTop());
if (r.contains(x, y)) {
    dropCoordinates[0] = x - dropCoordinates[0];
    dropCoordinates[1] = y - dropCoordinates[1];
    return target;
}

}

return null;
}

/**
 * Get the screen size so we can clamp events to the screen size so
even if
 * you drag off the edge of the screen, we find something.
 */
private void recordScreenSize() {
    ((WindowManager)mContext.getSystemService(Context.WINDOW_
SERVICE))
        .getDefaultDisplay().getMetrics(mDisplayMetrics);
}

/**
 * Clamp val to be &gt;= min and &lt;= max.
 */
private static int clamp(int val, int min, int max) {
    if (val < min) {
        return min;
    } else if (val >= max) {
        return max - 1;
    } else {
        return val;
    }
}

public void setWindowToken(IBinder token) {
    mWindowToken = token;
}

/**
 * Sets the drag listener which will be notified when a drag starts or
ends.
 */
public void setDragListener(DragListener l) {
    mListener = l;
}

/**
 * Remove a previously installed drag listener.
 */
public void removeDragListener(DragListener l) {
    mListener = null;
}

/**
 * Add a DropTarget to the list of potential places to receive drop
events.
 */
public void addDropTarget(DropTarget target) {
    mDropTargets.add(target);
}

/**
 * Don't send drop events to <em>target</em> any more.
 */
public void removeDropTarget(DropTarget target) {
```



```
mDropTargets.remove(target);
}

}

DragLayer

/*
 * This is a modified version of a class from the Android Open Source
 * Project.
 * The original copyright and license information follows.
 *
 * Copyright (C) 2008 The Android Open Source Project
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
 * either express or implied.
 * See the License for the specific language governing permissions
 * and
 * limitations under the License.
 */

package com.example.markligot.embrasd.dragview;

import android.content.Context;
import android.graphics.Rect;
import android.util.AttributeSet;
import android.view.KeyEvent;
import android.view.MotionEvent;
import android.view.View;
import android.widget.Toast;

/**
 * A ViewGroup that coordinates dragging across its descendants.
 *
 * <p> This class used DragLayer in the Android Launcher activity
 * as a model.
 * It is a bit different in several respects:
 * (1) It extends MyAbsoluteLayout rather than FrameLayout; (2) it
 * implements DragSource and DropTarget methods
 * that were done in a separate Workspace class in the Launcher.
 */
public class DragLayer extends MyAbsoluteLayout
    implements DragSource, DropTarget
{
    DragController mDragController;

    /**
     * Used to create a new DragLayer from XML.
     *
     * @param context The application's context.
     * @param attrs The attributes set containing the Workspace's
     * customization values.
     */
    public DragLayer (Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    public void setDragController(DragController controller) {
        mDragController = controller;
    }

    @Override
    public boolean dispatchKeyEvent(KeyEvent event) {
```

```
        return mDragController.dispatchKeyEvent(event) ||
        super.dispatchKeyEvent(event);
    }

    @Override
    public boolean onInterceptTouchEvent(MotionEvent ev) {
        return mDragController.onInterceptTouchEvent(ev);
    }

    @Override
    public boolean onTouchEvent(MotionEvent ev) {
        return mDragController.onTouchEvent(ev);
    }

    @Override
    public boolean dispatchUnhandledMove(View focused, int
    direction) {
        return mDragController.dispatchUnhandledMove(focused,
    direction);
    }

    /**
     * // DragSource interfaces methods

    /**
     * This method is called to determine if the DragSource has something
     * to drag.
     *
     * @return True if there is something to drag
     */

    public boolean allowDrag () {
        // In this simple demo, any view that you touch can be dragged.
        return true;
    }

    /**
     * setDragController
     *
     */

    /** setDragController is already defined. See above. */

    /**
     * onDropCompleted
     *
     */

    public void onDropCompleted (View target, boolean success)
    {
        toast ("DragLayer2.onDropCompleted: " + target.getId () + "
    Check that the view moved.");
    }

    /**
     * // DropTarget interfaces implementation

    /**
     * Handle an object being dropped on the DropTarget.
     * This is the where a dragged view gets repositioned at the end of a
     * drag.
     *
     * @param source DragSource where the drag started
     * @param x X coordinate of the drop location
     * @param y Y coordinate of the drop location
     * @param xOffset Horizontal offset with the object being dragged
     * where the original
     * touch happened
     * @param yOffset Vertical offset with the object being dragged
```



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

85

```
where the original
* touch happened
* @param dragView The DragView that's being dragged around on
screen.
* @param dragInfo Data associated with the object being dragged
*
*/
public void onDrop(DragSource source, int x, int y, int xOffset, int
yOffset,
    DragView dragView, Object dragInfo)
{
    View v = (View) dragInfo;
    toast ("DragLayer2.onDrop accepts view: " + v.getId ()
    + "x, y, xO, yO : " + new Integer(x) + ", " + new Integer(y) +
    ", "
    + new Integer(xOffset) + ", " + new Integer(yOffset));

    int w = v.getWidth ();
    int h = v.getHeight ();
    int left = x - xOffset;
    int top = y - yOffset;
    DragLayer.LayoutParams lp = new DragLayer.LayoutParams (w,
h, left, top);
    this.updateViewLayout(v, lp);
}

public void onDragEnter(DragSource source, int x, int y, int xOffset,
int yOffset,
    DragView dragView, Object dragInfo)
{
}

public void onDragOver(DragSource source, int x, int y, int xOffset,
int yOffset,
    DragView dragView, Object dragInfo)
{
}

public void onDragExit(DragSource source, int x, int y, int xOffset,
int yOffset,
    DragView dragView, Object dragInfo)
{
}

/**
 * Check if a drop action can occur at, or near, the requested
location.
 * This may be called repeatedly during a drag, so any calls should
return
 * quickly.
 *
 * @param source DragSource where the drag started
 * @param x X coordinate of the drop location
 * @param y Y coordinate of the drop location
 * @param xOffset Horizontal offset with the object being dragged
where the
 * original touch happened
 * @param yOffset Vertical offset with the object being dragged
where the
 * original touch happened
 * @param dragView The DragView that's being dragged around on
screen.
 * @param dragInfo Data associated with the object being dragged
 * @return True if the drop will be accepted, false otherwise.
 */
public boolean acceptDrop(DragSource source, int x, int y, int
xOffset, int yOffset,
    DragView dragView, Object dragInfo)
{
    System.out.println("correct drop" + dragView.getTag());
}
```

```
return true;
}

/**
 * Estimate the surface area where this object would land if dropped
at the
 * given location.
 *
 * @param source DragSource where the drag started
 * @param x X coordinate of the drop location
 * @param y Y coordinate of the drop location
 * @param xOffset Horizontal offset with the object being dragged
where the
 * original touch happened
 * @param yOffset Vertical offset with the object being dragged
where the
 * original touch happened
 * @param dragView The DragView that's being dragged around on
screen.
 * @param dragInfo Data associated with the object being dragged
 * @param recycle {@link Rect} object to be possibly recycled.
 * @return Estimated area that would be occupied if object was
dropped at
 * the given location. Should return null if no estimate is found,
 * or if this target doesn't provide estimations.
 */
public Rect estimateDropLocation(DragSource source, int x, int y,
int xOffset, int yOffset,
    DragView dragView, Object dragInfo, Rect
recycle)
{
    return null;
}

/**
 */
// More methods

/**
 * Show a string on the screen via Toast.
 *
 * @param msg String
 * @return void
 */
public void toast (String msg)
{
    if (!DragActivityV4.Debugging) return;
    Toast.makeText (getContext (), msg,
    Toast.LENGTH_SHORT).show ();
} // end toast

} // end class

DragSource

/**
 * This is a modified version of a class from the Android Open Source
Project.
 * The original copyright and license information follows.
 *
 * Copyright (C) 2008 The Android Open Source Project
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
```



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

86

** distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either express or implied.
* See the License for the specific language governing permissions
and
* limitations under the License.
/

```
package com.example.markligot.embrasd.dragview;
```

```
import android.view.View;
```

```
/**
```

```
 * Interface defining an object where drag operations originate.
```

```
 */
```

```
public interface DragSource {
```

```
    /**
```

```
     * This method is called to determine if the DragSource has  
something to drag.
```

```
     */
```

```
     * @return True if there is something to drag
```

```
    */
```

```
    boolean allowDrag();
```

```
    /**
```

```
     * This method is used to tell the DragSource which drag controller  
it is working with.
```

```
     */
```

```
     * @param dragger DragController
```

```
    */
```

```
    void setDragController(DragController dragger);
```

```
    /**
```

```
     * This method is called on the completion of the drag operation so  
the DragSource knows  
     * whether it succeeded or failed.
```

```
     */
```

```
     * @param target View - the view that accepted the dragged object
```

```
     * @param success boolean - true means that the object was  
dropped successfully
```

```
    */
```

```
    void onDropCompleted (View target, boolean success);
```

```
 }
```

```
DragView
```

```
/*
```

```
 * This is a modified version of a class from the Android Open Source  
Project.
```

```
 * The original copyright and license information follows.
```

```
 */
```

```
 * Copyright (C) 2008 The Android Open Source Project
```

```
 */
```

```
 * Licensed under the Apache License, Version 2.0 (the "License");
```

```
 * you may not use this file except in compliance with the License.
```

```
 * You may obtain a copy of the License at
```

```
 */
```

```
 * http://www.apache.org/licenses/LICENSE-2.0
```

```
 */
```

```
 * Unless required by applicable law or agreed to in writing, software
```

```
 * distributed under the License is distributed on an "AS IS" BASIS,
```

```
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
```

```
 * either express or implied.
```

```
 * See the License for the specific language governing permissions
```

```
 and
```

```
 * limitations under the License.
```

```
*/
```

```
package com.example.markligot.embrasd.dragview;
```

```
import android.content.Context;
```

```
import android.graphics.Bitmap;
```

```
import android.graphics.Canvas;
```

```
import android.graphics.Matrix;
```

```
import android.graphics.Paint;
```

```
import android.graphics.PixelFormat;
```

```
import android.os.IBinder;
```

```
import android.view.Gravity;
```

```
import android.view.View;
```

```
import android.view.ViewGroup;
```

```
import android.view.WindowManager;
```

```
/**
```

```
 * A DragView is a special view used by a DragController. During a  
drag operation, what is actually moving
```

```
 * on the screen is a DragView. A DragView is constructed using a
```

```
 * bitmap of the view the user really
```

```
 * wants to move.
```

```
 */
```

```
*/
```

```
public class DragView extends View
```

```
{
```

```
    // Number of pixels to add to the dragged item for scaling. Should  
be even for pixel alignment.
```

```
    private static final int DRAG_SCALE = 0; // In Launcher, value  
is 40
```

```
    private Bitmap mBitmap;
```

```
    private Paint mPaint;
```

```
    private int mRegistrationX;
```

```
    private int mRegistrationY;
```

```
    private float mScale;
```

```
    private float mAnimationScale = 1.0f;
```

```
    private WindowManager.LayoutParams mLayoutParams;
```

```
    private WindowManager mWindowManager;
```

```
    /**
```

```
     * Construct the drag view.
```

```
     * <p>
```

```
     * The registration point is the point inside our view that the touch  
events should
```

```
     * be centered upon.
```

```
     */
```

```
     * @param context A context
```

```
     * @param bitmap The view that we're dragging around. We scale  
it up when we draw it.
```

```
     * @param registrationX The x coordinate of the registration point.
```

```
     * @param registrationY The y coordinate of the registration point.
```

```
     */
```

```
    public DragView(Context context, Bitmap bitmap, int
```

```
registrationX, int registrationY,
```

```
int left, int top, int width, int height) {
```

```
super(context);
```

```
    // mWindowManager = WindowManagerImpl.getDefault();
```

```
    mWindowManager = (WindowManager)
```

```
context.getSystemService(Context.WINDOW_SERVICE);
```

```
    Matrix scale = new Matrix();
```

```
    float scaleFactor = width;
```

```
    scaleFactor = mScale = (scaleFactor + DRAG_SCALE) /
```

```
scaleFactor;
```

```
    scale.setScale(scaleFactor, scaleFactor);
```

```
    mBitmap = Bitmap.createBitmap(bitmap, left, top, width,
```




Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

87

```
height, scale, true);

// The point in our scaled bitmap that the touch events are
located
mRegistrationX = registrationX + (DRAG_SCALE / 2);
mRegistrationY = registrationY + (DRAG_SCALE / 2);
}

@Override
protected void onMeasure(int widthMeasureSpec, int
heightMeasureSpec) {
    setMeasuredDimension(mBitmap.getWidth(),
mBitmap.getHeight());
}

@Override
protected void onDraw(Canvas canvas) {
    /*if (true) {
        // for debugging
        Paint p = new Paint();
        p.setStyle(Paint.Style.FILL);
        p.setColor(0x88dd0011);
        canvas.drawRect(0, 0, getWidth(), getHeight(), p);
    }*/
    float scale = mAnimationScale;
    if (scale < 0.999f) { // allow for some float error
        float width = mBitmap.getWidth();
        float offset = (width-(width*scale))/2;
        canvas.translate(offset, offset);
        canvas.scale(scale, scale);
    }
    canvas.drawBitmap(mBitmap, 0.0f, 0.0f, mPaint);
}

@Override
protected void onDetachedFromWindow() {
    super.onDetachedFromWindow();
    mBitmap.recycle();
}

public void setPaint(Paint paint) {
    mPaint = paint;
    invalidate();
}

public void setScale(float scale) {
    if (scale > 1.0f) mAnimationScale = 1.0f;
    else mAnimationScale = scale;
    invalidate();
}

/**
 * Create a window containing this view and show it.
 *
 * @param windowToken obtained from v.getWindowToken() from
one of your views
 * @param touchX the x coordinate the user touched in screen
coordinates
 * @param touchY the y coordinate the user touched in screen
coordinates
 */
public void show(IBinder windowToken, int touchX, int touchY)
{
    WindowManager.LayoutParams lp;
    int pixelFormat;

    pixelFormat = PixelFormat.TRANSLUCENT;

    lp = new WindowManager.LayoutParams(
        ViewGroup.LayoutParams.WRAP_CONTENT,
        ViewGroup.LayoutParams.WRAP_CONTENT,
```

```
touchX-mRegistrationX, touchY-mRegistrationY,
WindowManager.LayoutParams.TYPE_APPLICATION_SUB_PAN
EL,
WindowManager.LayoutParams.FLAG_LAYOUT_IN_SCREEN
|
WindowManager.LayoutParams.FLAG_LAYOUT_NO_LIMITS
/*
WindowManager.LayoutParams.FLAG_ALT_FOCUSABLE_IM*/,
pixelFormat);
// lp.token = mStatusBarView.getWindowToken();
lp.gravity = Gravity.LEFT | Gravity.TOP;
lp.token = windowToken;
lp.setTitle("DragView");
mLayoutParams = lp;

mWindowManager.addView(this, lp);
}

/**
 * Move the window containing this view.
 *
 * @param touchX the x coordinate the user touched in screen
coordinates
 * @param touchY the y coordinate the user touched in screen
coordinates
 */
void move(int touchX, int touchY) {
    // This is what was done in the Launcher code.
    WindowManager.LayoutParams lp = mLayoutParams;
    lp.x = touchX - mRegistrationX;
    lp.y = touchY - mRegistrationY;
    mWindowManager.updateViewLayout(this, lp);
}

void remove() {
    mWindowManager.removeView(this);
}

DropSpot

/**
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
 * either express or implied.
 * See the License for the specific language governing permissions
and
 * limitations under the License.
 */

package com.example.markligot.embrasd.dragview;

import android.content.Context;
import android.graphics.Rect;
import android.util.AttributeSet;
import android.view.View;
import android.widget.Toast;

import com.example.markligot.embrasd.R;
import
com.example.markligot.embrasd.interfaces.CorrectAnswerListener;
```




```
/**
 * This class describes an area within a DragLayer where a dragged
 * item can be dropped.
 * It is a subclass of MyAbsoluteLayout, which means that it is a
 * ViewGroup and views
 * can be added as child views.
 * <p>
 * <p> In the onDrop method, the view dropped is not added as a
 * child view. Instead, the
 * view is repositioned within the DragLayer that contains the
 * DropSpot.
 * If the DropSpot is not associated with a DragLayer, it will not
 * accept dropped objects.
 */

public class DropSpot extends MyAbsoluteLayout
    implements DropTarget, DragController.DragListener {

    public CorrectAnswerListener mListener; // Set This attribute from
    // DragActivityV2

    /**
     */
    // Constructors
    public DropSpot(Context context) {
        super(context);
    }

    public DropSpot(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    public DropSpot(Context context, AttributeSet attrs, int style) {
        super(context, attrs, style);
    }

    /**
     */
    // Instance Variables

    private DragController mDragController;
    private DragLayer mDragLayer;

    /**
     */
    // Properties

    /**
     * Get the value of the DragController property.
     *
     * @return DragController
     */

    public DragController getDragController() {
        return mDragController;
    } // end getDragController

    /**
     * Set the value of the DragController property.
     *
     * @param newValue DragController
     */

    public void setDragController(DragController newValue) {
        mDragController = newValue;
    } // end setDragController

    /**
     * Get the value of the DragLayer property.
     *
     * @return DragLayer
     */

    public DragLayer getDragLayer() {
        return mDragLayer;
    } // end getDragLayer

    /**
     * Set the value of the DragLayer property.
     *
     * @param newValue DragLayer
     */

    public void setDragLayer(DragLayer newValue) {
        mDragLayer = newValue;
    } // end setDragLayer

    /** Property SavedBackground */
    /**
     * This variable holds the value of the SavedBackground property.
     */

    private int mSavedBackground = R.color.drop_target_color1;

    /**
     * Get the value of the SavedBackground property.
     *
     * @return int
     */

    public int getSavedBackground() {
        //if (mSavedBackground == null) {}
        return mSavedBackground;
    } // end getSavedBackground

    /**
     * Set the value of the SavedBackground property.
     *
     * @param newValue int
     */

    public void setSavedBackground(int newValue) {
        mSavedBackground = newValue;
    } // end setSavedBackground
    /* end Property SavedBackground */

    /**
     */
    // DragListener interfaces implementation

    /**
     * A drag has begun
     *
     * @param source An object representing where the drag
     * originated
     * @param info The data associated with the object that is
     * being dragged
     * @param dragAction The drag action: either {@link
     * DragController#DRAG_ACTION_MOVE}
     * or {@link DragController#DRAG_ACTION_COPY}
     */

    public void onDragStart(DragSource source, Object info, int
    dragAction) {
        toast("onDragStart");
    }

    /**
     * The drag has ended
     */
}
```



```
*/
public void onDragEnd() {
    toast("onDragEnd");
}

/**
 *
 * // DropTarget interfaces implementation
 *
 * /**
 *  * Handle an object being dropped on the DropTarget.
 *  * This is the where a dragged view gets repositioned at the end of
 *  * a drag.
 *  *
 *  * @param source DragSource where the drag started
 *  * @param x X coordinate of the drop location
 *  * @param y Y coordinate of the drop location
 *  * @param xOffset Horizontal offset with the object being dragged
 *  * where the original
 *  * touch happened
 *  * @param yOffset Vertical offset with the object being dragged
 *  * where the original
 *  * touch happened
 *  * @param dragView The DragView that's being dragged around
 *  * on screen.
 *  * @param dragInfo Data associated with the object being dragged
 *  */
public void onDrop(DragSource source, int x, int y, int xOffset,
int yOffset,
    DragView dragView, Object dragInfo) {
    // We need the DragLayer to do the drop.
    if (mDragLayer == null) return;

    // The view being moved is stored in the dragInfo.
    View v = (View) dragInfo;

    // At some point, it would be nice to get away from use of the
    clone of deprecated AbsoluteLayout.
    // That's what DragLayer is defined in terms of.

    // Launcher code and my own DragView moves the DragView
    this way.
    // Don't think this is the way.
    // Go back and study the Launcher code some more. Look at
    CellLayout.
    /**
    WindowManager.LayoutParams lp = mLayoutParams;
    lp.x = touchX - mRegistrationX;
    lp.y = touchY - mRegistrationY;
    mWindowManager.updateViewLayout(this, lp);
    */

    // For now, take the view and update its position using the
    DragLayer layout object.
    // The coordinates given to onDrop are relative to the DropSpot.
    Add in its x and y location
    // so the drop occurs where the user expects it.
    // (At least the fact that the layout is absolute is pretty much
    hidden from the caller.)
    int viewX = this.getLeft();
    int viewY = this.getTop();
    int w = v.getWidth();
    int h = v.getHeight();
    int left = x - xOffset + viewX;
    int top = y - yOffset + viewY;
    DragLayer.LayoutParams lp = new DragLayer.LayoutParams(w,
h, left, top);
    mDragLayer.updateViewLayout(v, lp);

    if(mListener != null) {
```

```
mListener.onRightAnswerSelected();
    }
}

/**
 * React to a dragged object entering the area of this DropSpot.
 * Provide the user with some visual feedback.
 */
public void onDragEnter(DragSource source, int x, int y, int
xOffset, int yOffset,
    DragView dragView, Object dragInfo) {
    toast("onDragEnter");
    int bg = isEnabled() ? R.color.drop_target_enabled :
R.color.drop_target_disabled;
    setBackgroundResource(bg);
}

/**
 * React to something being dragged over the drop target.
 */
public void onDragOver(DragSource source, int x, int y, int
xOffset, int yOffset,
    DragView dragView, Object dragInfo) {
}

/**
 * React to a drag
 */
public void onDragExit(DragSource source, int x, int y, int
xOffset, int yOffset,
    DragView dragView, Object dragInfo) {
    toast("onDragExit");
    setBackgroundResource(getSavedBackground());
}

/**
 * Check if a drop action can occur at, or near, the requested
    location.
 * This may be called repeatedly during a drag, so any calls should
    return
 * quickly.
 *
 * @param source DragSource where the drag started
 * @param x X coordinate of the drop location
 * @param y Y coordinate of the drop location
 * @param xOffset Horizontal offset with the object being dragged
    where the
 * original touch happened
 * @param yOffset Vertical offset with the object being dragged
    where the
 * original touch happened
 * @param dragView The DragView that's being dragged around
    on screen.
 * @param dragInfo Data associated with the object being dragged
 * @return True if the drop will be accepted, false otherwise.
 */
public boolean acceptDrop(DragSource source, int x, int y, int
xOffset, int yOffset,
    DragView dragView, Object dragInfo) {
    return isEnabled();
}

/**
 * Estimate the surface area where this object would land if
    dropped at the
 * given location.
 *
 * @param source DragSource where the drag started
 * @param x X coordinate of the drop location
 * @param y Y coordinate of the drop location
 * @param xOffset Horizontal offset with the object being dragged
```



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740
Bachelor of Science in Information Technology

90

```
where the
* original touch happened
* @param yOffset Vertical offset with the object being dragged
where the
* original touch happened
* @param dragView The DragView that's being dragged around
on screen.
* @param dragInfo Data associated with the object being dragged
* @param recycle {@link Rect} object to be possibly recycled.
* @return Estimated area that would be occupied if object was
dropped at
* the given location. Should return null if no estimate is found,
* or if this target doesn't provide estimations.
*/
public Rect estimateDropLocation(DragSource source, int x, int y,
int xOffset, int yOffset,
    DragView dragView, Object dragInfo, Rect
recycle) {
    return null;
}

/**
*/
// Methods

/**
* Return true if this DropSpot is hooked up to a DragLayer.
* If it is, it means that it will accept dropped views.
*
* @return boolean
*/

public boolean isEnabled() {
    return (mDragLayer != null);
} // end getDragLayer

/**
* Set up the drop spot by connecting it to a drag layer and a drag
controller.
* When this method completes, the drop spot is listed as one of the
drop targets of the controller
* and the drop spot is also set as a DragListener.
* <p>
* If the dragLayer is null, the drop spot is set initially in a state
where nothing can be dropped on it.
*
* @param layer DragLayer
* @param controller DragController
* @param initialColor int - resource id of the initial background
color
*/

public void setup(DragLayer layer, DragController controller, int
initialColor) {
    mDragLayer = layer;
    mDragController = controller;
    setSavedBackground(initialColor);

    if (controller != null) {
        controller.setDragListener(this);
        controller.addDropTarget(this);

        //context.startActivitiy(context,DragActivityV2.class);

        //mListener.onRightAnswerSelected();
        if (mListener != null) {
            mListener.onRightAnswerSelected();
        }
    }
}
```

```

}

/**
*/
// More methods

/**
* Show a string on the screen via Toast.
*
* @param msg String
* @return void
*/

public void toast(String msg) {
    if (!DragActivityV4.Debugging) return;
    Toast.makeText(getContext(), msg,
Toast.LENGTH_SHORT).show();
} // end toast

} // end DropSpot

DropTarget

/**
* This is a modified version of a class from the Android Open Source
Project.
* The original copyright and license information follows.
*
* Copyright (C) 2008 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either express or implied.
* See the License for the specific language governing permissions
and
* limitations under the License.
*/

package com.example.markligot.embrasd.dragview;

import android.graphics.Rect;

/**
* Interface defining an object that can receive a view at the end of a
drag operation.
*/

public interface DropTarget {

/**
* Handle an object being dropped on the DropTarget
*
* @param source DragSource where the drag started
* @param x X coordinate of the drop location
* @param y Y coordinate of the drop location
* @param xOffset Horizontal offset with the object being dragged
where the original
* touch happened
* @param yOffset Vertical offset with the object being dragged
where the original
* touch happened
* @param dragView The DragView that's being dragged around
on screen.
*/
}
```



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

91

```

    * @param dragInfo Data associated with the object being
    dragged
    */
    */
    void onDrop(DragSource source, int x, int y, int xOffset, int
    yOffset,
        DragView dragView, Object dragInfo);

    /**
     * React to something started to be dragged.
     */
    void onDragEnter(DragSource source, int x, int y, int xOffset, int
    yOffset,
        DragView dragView, Object dragInfo);

    /**
     * React to something being dragged over the drop target.
     */
    void onDragOver(DragSource source, int x, int y, int xOffset, int
    yOffset,
        DragView dragView, Object dragInfo);

    /**
     * React to a drag
     */
    void onDragExit(DragSource source, int x, int y, int xOffset, int
    yOffset,
        DragView dragView, Object dragInfo);

    /**
     * Check if a drop action can occur at, or near, the requested
     location.
     * This may be called repeatedly during a drag, so any calls should
     return
     * quickly.
     *
     * @param source DragSource where the drag started
     * @param x X coordinate of the drop location
     * @param y Y coordinate of the drop location
     * @param xOffset Horizontal offset with the object being dragged
     where the
     * original touch happened
     * @param yOffset Vertical offset with the object being dragged
     where the
     * original touch happened
     * @param dragView The DragView that's being dragged around
     on screen.
     * @param dragInfo Data associated with the object being
     dragged
     * @return True if the drop will be accepted, false otherwise.
     */
    boolean acceptDrop(DragSource source, int x, int y, int xOffset,
    int yOffset,
        DragView dragView, Object dragInfo);

    /**
     * Estimate the surface area where this object would land if
     dropped at the
     * given location.
     *
     * @param source DragSource where the drag started
     * @param x X coordinate of the drop location
     * @param y Y coordinate of the drop location
     * @param xOffset Horizontal offset with the object being dragged
     where the
     * original touch happened
     * @param yOffset Vertical offset with the object being dragged
     where the
     * original touch happened
     * @param dragView The DragView that's being dragged around

```

```

on screen.
    * @param dragInfo Data associated with the object being
    dragged
    * @param recycle {@link Rect} object to be possibly recycled.
    * @return Estimated area that would be occupied if object was
    dropped at
    * the given location. Should return null if no estimate is
    found,
    * or if this target doesn't provide estimations.
    */
    Rect estimateDropLocation(DragSource source, int x, int y, int
    xOffset, int yOffset,
        DragView dragView, Object dragInfo, Rect
    recycle);

    // These methods are implemented in Views
    void getHitRect(Rect outRect);
    void getLocationOnScreen(int[] loc);
    int getLeft();
    int getTop();
}

```

Globals

```
package com.example.markligot.embrasd.dragview;
```

```

/**
 * Created by Mark Ligot on 1/11/2017.
 */

```

```
public class Globals {
```

```
    private String name;
```

```
    public String getName(){
        return name;
    }

```

```
    public void setName(String aName) {

        name = aName;
    }

```

```
    private static Globals instance = new Globals();
```

```

    // Getter-Setters
    public static Globals getInstance() {
        return instance;
    }

```

```
    public static void setInstance(Globals instance) {
        Globals.instance = instance;
    }

```

```
    private String notification_index;
```

```
    private Globals() {
```

```
    }
```

```
    public String getValue() {
        return notification_index;
    }

```

```
    public void setValue(String notification_index) {
        this.notification_index = notification_index;
    }

```



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

92

```
}  
  
}  
  
MyAbsoluteLayout  
  
/*  
 * This is a modified version of a class from the Android Open Source  
 * Project.  
 * The original copyright and license information follows.  
 *  
 * Copyright (C) 2006 The Android Open Source Project  
 *  
 * Licensed under the Apache License, Version 2.0 (the "License");  
 * you may not use this file except in compliance with the License.  
 * You may obtain a copy of the License at  
 *  
 * http://www.apache.org/licenses/LICENSE-2.0  
 *  
 * Unless required by applicable law or agreed to in writing, software  
 * distributed under the License is distributed on an "AS IS" BASIS,  
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,  
 * either express or implied.  
 * See the License for the specific language governing permissions  
 * and  
 * limitations under the License.  
 */
```

```
package com.example.markligot.embrasd.dragview;
```

```
import android.content.Context;  
import android.util.AttributeSet;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.RemoteViews.RemoteView;
```

```
/**  
 * A layout that lets you specify exact locations (x/y coordinates) of its  
 * children. Absolute layouts are less flexible and harder to maintain  
 * than  
 * other types of layouts without absolute positioning.  
 *  
 * <p><strong>XML attributes</strong><p> <p> See {@link  
 * android.R.styleable#ViewGroup ViewGroup Attributes}, {@link  
 * android.R.styleable#View View Attributes}<p>  
 *  
 * <p>Note: This class is a clone of AbsoluteLayout, which is now  
 * deprecated.  
 */
```

```
@RemoteView
```

```
public class MyAbsoluteLayout extends ViewGroup {  
    public MyAbsoluteLayout(Context context) {  
        super(context);  
    }  
  
    public MyAbsoluteLayout(Context context, AttributeSet attrs) {  
        super(context, attrs);  
    }  
  
    public MyAbsoluteLayout(Context context, AttributeSet attrs,  
        int defStyle) {  
        super(context, attrs, defStyle);  
    }  
  
    @Override  
    protected void onMeasure(int widthMeasureSpec, int  
        heightMeasureSpec) {
```

```
        int count = getChildCount();
```

```
        int maxHeight = 0;  
        int maxWidth = 0;
```

```
        // Find out how big everyone wants to be  
        measureChildren(widthMeasureSpec, heightMeasureSpec);
```

```
        // Find rightmost and bottom-most child
```

```
        for (int i = 0; i < count; i++) {  
            View child = getChildAt(i);  
            if (child.getVisibility() != GONE) {  
                int childRight;  
                int childBottom;
```

```
                MyAbsoluteLayout.LayoutParams lp  
                    = (MyAbsoluteLayout.LayoutParams)  
                    child.getLayoutParams();
```

```
                childRight = lp.x + child.getMeasuredWidth();  
                childBottom = lp.y + child.getMeasuredHeight();
```

```
                maxWidth = Math.max(maxWidth, childRight);  
                maxHeight = Math.max(maxHeight, childBottom);  
            }  
        }
```

```
        // Account for padding too  
        maxWidth += getPaddingLeft () + getPaddingRight ();  
        maxHeight += getPaddingTop () + getPaddingBottom ();  
        /* original  
        maxWidth += mPaddingLeft + mPaddingRight;  
        maxHeight += mPaddingTop + mPaddingBottom;  
        */
```

```
        // Check against minimum height and width  
        maxHeight = Math.max(maxHeight,  
            getSuggestedMinimumHeight());  
        maxWidth = Math.max(maxWidth,  
            getSuggestedMinimumWidth());
```

```
        setMeasuredDimension(resolveSize(maxWidth,  
            widthMeasureSpec),  
            resolveSize(maxHeight, heightMeasureSpec));  
    }
```

```
    /**  
     * Returns a set of layout parameters with a width of  
     * {@link  
     * android.view.ViewGroup.LayoutParams#WRAP_CONTENT},  
     * a height of {@link  
     * android.view.ViewGroup.LayoutParams#WRAP_CONTENT}  
     * and with the coordinates (0, 0).  
     */
```

```
    @Override  
    protected ViewGroup.LayoutParams  
        generateDefaultLayoutParams() {  
        return new LayoutParams(LayoutParams.WRAP_CONTENT,  
            LayoutParams.WRAP_CONTENT, 0, 0);  
    }
```

```
    @Override  
    protected void onLayout(boolean changed, int l, int t,  
        int r, int b) {  
        int count = getChildCount();
```

```
        int paddingL = getPaddingLeft ();  
        int paddingT = getPaddingTop ();  
        for (int i = 0; i < count; i++) {  
            View child = getChildAt(i);  
            if (child.getVisibility() != GONE) {
```



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

93

```
MyAbsoluteLayout.LayoutParams lp =
    (MyAbsoluteLayout.LayoutParams)
child.getLayoutParams();

int childLeft = paddingL + lp.x;
int childTop = paddingT + lp.y;
/*
int childLeft = mPaddingLeft + lp.x;
int childTop = mPaddingTop + lp.y;
*/
child.layout(childLeft, childTop,
    childLeft + child.getMeasuredWidth(),
    childTop + child.getMeasuredHeight());
}
}

@Override
public ViewGroup.LayoutParams
generateLayoutParams(AttributeSet attrs) {
    return new MyAbsoluteLayout.LayoutParams(getContext(),
        attrs);
}

// Override to allow type-checking of LayoutParams.
@Override
protected boolean checkLayoutParams(ViewGroup.LayoutParams
p) {
    return p instanceof MyAbsoluteLayout.LayoutParams;
}

@Override
protected ViewGroup.LayoutParams
generateLayoutParams(ViewGroup.LayoutParams p) {
    return new LayoutParams(p);
}

/**
 * Per-child layout information associated with MyAbsoluteLayout.
 * See
 * {@link android.R.styleable#MyAbsoluteLayout_Layout Absolute
 * Layout Attributes}
 * for a list of all child view attributes that this class supports.
 */
public static class LayoutParams extends
ViewGroup.LayoutParams {
    /**
     * The horizontal, or X, location of the child within the view
     group.
     */
    public int x;
    /**
     * The vertical, or Y, location of the child within the view group.
     */
    public int y;

    /**
     * Creates a new set of layout parameters with the specified
     width,
     * height and location.
     *
     * @param width the width, either {@link #MATCH_PARENT},
     *             {@link #WRAP_CONTENT} or a fixed size in pixels
     * @param height the height, either {@link
     *             #MATCH_PARENT},
     *             {@link #WRAP_CONTENT} or a fixed size in pixels
     * @param x the X location of the child
     * @param y the Y location of the child
     */
}
```

```
public LayoutParams(int width, int height, int x, int y) {
    super(width, height);
    this.x = x;
    this.y = y;
}

/**
 * Creates a new set of layout parameters. The values are
 * extracted from
 * the supplied attributes set and context. The XML attributes
 * mapped
 * to this set of layout parameters are:
 *
 * <ul>
 * <li><code>layout_x</code>: the X location of the
 * child</li>
 * <li><code>layout_y</code>: the Y location of the
 * child</li>
 * <li>All the XML attributes from
 * {@link android.view.ViewGroup.LayoutParams}</li>
 * </ul>
 *
 * @param c the application environment
 * @param attrs the set of attributes from which to extract the
 * layout
 * parameters values
 */
public LayoutParams(Context c, AttributeSet attrs) {
    super(c, attrs);
    /* FLX THIS eventually. Without this, I don't think you can put
    x and y in layout xml files.
    TypedArray a = c.obtainStyledAttributes(attrs,
    com.android.internal.R.styleable.AbsoluteLayout_Layout);
    x = a.getDimensionPixelOffset(
    com.android.internal.R.styleable.AbsoluteLayout_Layout_layout_x,
    0);
    y = a.getDimensionPixelOffset(
    com.android.internal.R.styleable.AbsoluteLayout_Layout_layout_y,
    0);
    a.recycle();
    */
}

/**
 * {@inheritDoc}
 */
public LayoutParams(ViewGroup.LayoutParams source) {
    super(source);
}

public String debug(String output) {
    return output + "Absolute.LayoutParams={width=" +
        sizeToString(width) + ", height=" +
        sizeToString(height)
        + " x=" + x + " y=" + y + "}";
}

/**
 * Converts the specified size to a readable String.
 *
 * @param size the size to convert
 * @return a String instance representing the supplied size
 *
 * @hide
 */
protected static String sizeToString(int size) {
    if (size == WRAP_CONTENT) {
        return "wrap-content";
    }
}
```



```
    }
    if (size == MATCH_PARENT) {
        return "match-parent";
    }
    return String.valueOf(size);
} // end class

} // end class

CorrectAnswerListener

package com.example.markligot.embrasd.interfaces;

/**
 * Created by chaitanya on 11/1/17.
 */

public interface CorrectAnswerListener {

    void onRightAnswerSelected();
}

AbstractFragment

package com.example.markligot.embrasd;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

/**
 * Created by markligot on 1/2/17.
 */

public class AbstractFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
        container,
        Bundle savedInstanceState) {

        View rootView = inflater.inflate(R.layout.fragment_abstract,
            container, false);

        return rootView;
    }
}

AnswerTable

package com.example.markligot.embrasd;

/**
 * Created by markligot on 1/4/17.
 */

public class AnswerTable {
    //private variables
    int _id;
    String QuesDesc;
    String CorrectAns;

    // Empty constructor
    public AnswerTable(){
    }
    // constructor
    public AnswerTable(int id, String quesDesc, String correctAns){
        this._id = id;
        this.QuesDesc = quesDesc;
    }
}
```

```
        this.CorrectAns = correctAns;
    }

    // constructor
    public AnswerTable(String quesDesc, String correctAns){
        this.QuesDesc = quesDesc;
        this.CorrectAns = correctAns;
    }
    // getting ID
    public int getID(){
        return this._id;
    }

    // setting id
    public void setID(int id){
        this._id = id;
    }

    // getting quesDesc
    public String getquesDesc(){
        return this.QuesDesc;
    }

    // setting quesDesc
    public void setquesDesc(String quesDesc){
        this.QuesDesc = quesDesc;
    }

    // getting correctAns
    public String getcorrectAns(){
        return this.CorrectAns;
    }

    // setting correctAns
    public void setcorrectAns(String correctAns){
        this.CorrectAns = correctAns;
    }
}
```

AudioRecorderService

```
package com.example.markligot.embrasd;

import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.media.MediaRecorder;
import android.os.Binder;
import android.os.Environment;
import android.os.IBinder;
import android.widget.Toast;

import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;

/**
 * Created by Mark Ligot on 1/17/2017.
 */

public class AudiorecorderService extends Service {

    private final IBinder mBinder = new
        AudiorecorderService.ServiceBinder();
    MediaRecorder myRecorder;
    String outputFile = null;

    MediaPlayer mPlayer;
    private int length = 0;
}
```




```
public AudiorecorderService() {
}

public class ServiceBinder extends Binder {
    public AudiorecorderService getService() {
        return AudiorecorderService.this;
    }
}

@Override
public IBinder onBind(Intent arg0) {
    return mBinder;
}

@Override
public void onCreate() {
    super.onCreate();
    SimpleDateFormat s = new
SimpleDateFormat("d0dMMyyyyhhmmss");
    String format = s.format(new Date());
    outputFile = Environment.getExternalStorageDirectory().
        getAbsolutePath() + "/EmbrASD" + format + ".3gpp";
    System.out.println("format " + format + " " + format.toString()
);
    myRecorder = new MediaRecorder();

    myRecorder.setAudioSource(MediaRecorder.AudioSource.CAMCO
RDER);

    myRecorder.setOutputFormat(MediaRecorder.OutputFormat.DEFA
ULT);

    myRecorder.setAudioEncoder(MediaRecorder.OutputFormat.MPEG
_4);
    myRecorder.setMaxDuration(10000);
    myRecorder.setOutputFile(outputFile);

    try {
        myRecorder.prepare();
        myRecorder.start();
    } catch (IllegalStateException e) {
        // start: it is called before prepare()
        // prepare: it is called after start() or before
        setOutputFormat()
        e.printStackTrace();
    } catch (IOException e) {
        // prepare() fails
        e.printStackTrace();
    }
}

myRecorder.setOnErrorListener(new
MediaRecorder.OnErrorListener() {

    public void onError(MediaRecorder mp, int what, int
extra) {

        onError(myRecorder, what, extra);
        return;
    }
});

public boolean onError(MediaRecorder mp, int what, int extra) {

    Toast.makeText(this, "music player failed",
Toast.LENGTH_SHORT).show();
    if (myRecorder != null) {
        try {
```

```
myRecorder.stop();
myRecorder.release();
} finally {
    myRecorder = null;
}
}

return false;
}
```

```
public void onStop(){
    myRecorder.stop();
    myRecorder.release();
    myRecorder = null;
}
```

```
public void stopRecording() {
    myRecorder.stop();
    myRecorder.release();
    myRecorder = null;
}
```

```
}
```

ColorsAndShapesFragment

```
package com.example.markligot.embrasd;
```

```
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
```

```
/**
 * Created by markligot on 1/2/17.
 */
```

```
public class ColorsandShapesFragment extends Fragment {

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
        Bundle savedInstanceState) {

        View rootView =
inflater.inflate(R.layout.fragment_colorsandshapes, container, false);

        return rootView;
    }
}
```

```
}
```

CvcandSightWordsFragment

```
package com.example.markligot.embrasd;
```

```
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
```

```
/**
 * Created by markligot on 1/2/17.
 */
```

```
public class CvcandSightWordsFragment extends Fragment {
```




Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

96

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup
container,
    Bundle savedInstanceState) {

    View rootView =
inflater.inflate(R.layout.fragment_cvcandsightwords, container,
false);

    return rootView;
}
}
```

LettersandNumbersFragment

```
package com.example.markligot.embrasd;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
```

```
/**
 * Created by markligot on 1/2/17.
 */
```

```
public class LettersandNumbersFragment extends Fragment {
```

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup
container,
    Bundle savedInstanceState) {

    View rootView =
inflater.inflate(R.layout.fragment_lettersandnumbers, container,
false);

    return rootView;
}
}
```

MainActivity

```
package com.example.markligot.embrasd;
```

```
import android.app.ActionBar;
import android.app.ActionBar.Tab;
import android.app.FragmentTransaction;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.ServiceConnection;
import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.os.IBinder;
import android.support.v4.app.FragmentActivity;
import android.support.v4.view.ViewPager;
import android.view.View;
import android.view.Window;
```

```
import com.example.markligot.embrasd.adapter.TabsPagerAdapter;
import com.example.markligot.embrasd.dragview.*;
```

```
public class MainActivity extends FragmentActivity implements
ActionBar.TabListener {
    public final static String EXTRA_MESSAGE =
"com.example.myfirstapp.MESSAGE";
    public final static String MUSIC_BACKGROUND =
```

```
"com.example.myfirstapp.MESSAGE";
```

```
private ViewPager viewPager;
private TabsPagerAdapter mAdapter;
```

```
private boolean mIsBound = false;
private boolean mIsBound2 = false;
private MusicService mServ;
private AudiorecorderService mServ2;
```

```
private ServiceConnection Scon =new ServiceConnection(){
```

```
    public void onServiceConnected(ComponentName name,
IBinder
        binder) {
```

```
        mServ = ((MusicService.ServiceBinder)binder).getService();
    }
```

```
    public void onServiceDisconnected(ComponentName name) {
        mServ = null;
    }
};
```

```
void doBindService(){
    bindService(new Intent(this,MusicService.class),
        Scon,Context.BIND_AUTO_CREATE);
    mIsBound = true;
}
```

```
void doUnbindService()
```

```
{
    if(mIsBound)
    {
        unbindService(Scon);
        mIsBound = false;
    }
}
///
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    setContentView(R.layout.activity_main);
```

```
super.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATI
ON_PORTRAIT);
```

```
new StartBGServicesHomeScreen().invoke();
new SetupViews().invoke(); //setup fragment views
```

```
}
```

```
@Override
public void onTabReselected(Tab tab, FragmentTransaction ft) {
}
```

```
@Override
public void onTabSelected(Tab tab, FragmentTransaction ft) {
    // on tab selected
    // show respected fragment view
    viewPager.setCurrentItem(tab.getPosition());
}
```

```
@Override
public void onTabUnselected(Tab tab, FragmentTransaction ft) {
}
```

```
public void onClickcolorsandshapes(View v1) {
    Intent intent = new Intent(this, DragActivityV4.class);
    String lessonModule = "1";
```



```
int musicbg = 1;
intent.putExtra(EXTRA_MESSAGE, lessonModule);
intent.putExtra("intVariableName", musicbg);

startActivity(intent);
}

public void onClickLettersandNumbers(View v2) {

    Intent intent = new Intent(this, DragActivityV4.class);
    String lessonModule = "2";
    int musicbg = 2;
    intent.putExtra(EXTRA_MESSAGE, lessonModule);
    intent.putExtra("intVariableName", musicbg);
    startActivity(intent);
}

public void onClickcvandsightwords(View v3) {

    Intent intent = new Intent(this, DragActivityV4.class);
    String lessonModule = "3";
    int musicbg = 3;
    intent.putExtra(EXTRA_MESSAGE, lessonModule);
    intent.putExtra("intVariableName", musicbg);
    startActivity(intent);
}

public void onClickabstractwords(View v4) {
    Intent intent = new Intent(this, DragActivityV4.class);
    String lessonModule = "4";
    int musicbg = 2;
    intent.putExtra(EXTRA_MESSAGE, lessonModule);
    intent.putExtra("intVariableName", musicbg);
    startActivity(intent);
}

public void onClickreadingcomp(View v5) {
    Intent intent = new Intent(this, DragActivityV4.class);
    String lessonModule = "5";
    int musicbg = 3;
    intent.putExtra(EXTRA_MESSAGE, lessonModule);
    intent.putExtra("intVariableName", musicbg);
    startActivity(intent);
}

public void onStart(){
    super.onStart();
}

public void onResume() {
    super.onResume();
    new StartBGServicesHomeScreen().invoke();
}

public void onPause(){
    super.onPause();
    Intent intentmusic = new Intent(this, MusicService.class);
    stopService(intentmusic);
    //MediaPlayer player =
    MediaPlayer.create(MainActivitytest.this, R.raw.bgmusic3);
    // player.pause(); //causes null pointer exception
}

public void onStop(){
    super.onStop();
    Intent intentmusic = new Intent(this, MusicService.class);
    stopService(intentmusic);
}

public void onDestroy(){
    super.onDestroy();
    Intent intentmusic = new Intent(this, MusicService.class);
    stopService(intentmusic);
}
```

```
private class SetupViews {
    public void invoke() {
        viewPager = (ViewPager) findViewById(R.id.pager);
        mAdapter = new
        TabsPagerAdapter(getSupportFragmentManager());
        viewPager.setAdapter(mAdapter);
    }
}

private class StartBGServicesHomeScreen {
    public void invoke() {
        Intent intent = new Intent(MainActivity.this,
        MusicService.class);
        startService(intent);
    }
}

modulefourGetterSetter

package com.example.markligot.embrasd;

/**
 * Created by markligot on 1/4/17.
 */

public class moduleFourGetterSetter {
    //private variables

    String quesNum1;
    String quesDesc1;
    String quesDesc2;
    String choiceA;
    String choiceB;
    String corrChoice;
    String descIMGFP;
    String quesDescAUDFP;
    String corrANSAUDFP;
    String choiceAAUDFP;
    String choiceBAUDFP;
    String lessonmodule;
    String QuesDescAUDFP2;
    String followupquesaud;
    String followupquestext;

    // Empty constructor
    public moduleFourGetterSetter(){

    }

    public moduleFourGetterSetter(String lessonmodule){
        this.lessonmodule = lessonmodule;
    }

    // constructor
    public moduleFourGetterSetter(String quesNum1, String
    quesDesc1, String quesDesc2,
        String choiceA, String choiceB, String
    corrChoice,
        String descIMGFP, String quesDescAUDFP,
    String corrANSAUDFP,
        String choiceAAUDFP, String choiceBAUDFP,
    String lessonmodule,
        String QuesDescAUDFP2, String
    followupquesaud, String followupquestext){
        this.quesNum1 = quesNum1;
        this.quesDesc1 = quesDesc1;
```



```
this.quesDesc2 = quesDesc2;
this.choiceA = choiceA;
this.choiceB = choiceB;
this.corrChoice = corrChoice;
this.descIMGFP = descIMGFP;
this.quesDescAUDFP = quesDescAUDFP;
this.corrANSAUDFP = corrANSAUDFP;
this.choiceAAUDFP = choiceAAUDFP;
this.choiceBAUDFP = choiceBAUDFP;
this.lessonmodule = lessonmodule;
this.QuesDescAUDFP2 = QuesDescAUDFP2;
this.followupquesaud = followupquesaud;
this.followupquestext = followupquestext;
}

// constructor

// getting quesNum
public String getQuesNum1(){
    return this.quesNum1;
}

// setting quesNum1
public void setQuesNum1(String quesNum1){
    this.quesNum1 = quesNum1;
}

// getting quesDesc1
public String getquesDesc1(){
    return this.quesDesc1;
}

// setting quesDesc1
public void setquesDesc1(String quesDesc1){
    this.quesDesc1 = quesDesc1;
}

// getting quesDesc2
public String getquesDesc2(){
    return this.quesDesc2;
}

// setting quesDesc2
public void setquesDesc2(String quesDesc2){
    this.quesDesc2 = quesDesc2;
}

// getting choiceA
public String getchoiceA(){
    return this.choiceA;
}

// setting choiceA
public void setchoiceA(String choiceA){
    this.choiceA = choiceA;
}

// getting choiceB
public String getchoiceB(){
    return this.choiceB;
}

// setting choiceB
public void setchoiceB(String choiceB){
    this.choiceB = choiceB;
}

// getting corrChoice
public String getcorrChoice(){
    return this.corrChoice;
}

// setting corrChoice
public void setcorrChoice(String corrChoice){
    this.corrChoice = corrChoice;
}

// getting descIMGFP
public String getdescIMGFP(){
    return this.descIMGFP;
}

// setting descIMGFP
public void setdescIMGFP(String descIMGFP){
    this.descIMGFP = descIMGFP;
}

// getting quesDescAUDFP
public String getqquesDescAUDFP(){
    return this.quesDescAUDFP;
}

// setting quesDescAUDFP
public void setquesDescAUDFP(String quesDescAUDFP){
    this.quesDescAUDFP = quesDescAUDFP;
}

// getting corrANSAUDFP
public String getcorrANSAUDFP(){
    return this.corrANSAUDFP;
}

// setting corrANSAUDFP
public void setcorrANSAUDFP(String corrANSAUDFP){
    this.corrANSAUDFP = corrANSAUDFP;
}

// getting choiceAAUDFP
public String getchoiceAAUDFP(){
    return this.choiceAAUDFP;
}

// setting choiceAAUDFP
public void setchoiceAAUDFP(String choiceAAUDFP){
    this.choiceAAUDFP = choiceAAUDFP;
}

// getting choiceBAUDFP
public String getchoiceBAUDFP(){
    return this.choiceBAUDFP;
}

// setting choiceBAUDFP
public void setchoiceBAUDFP(String choiceBAUDFP){
    this.choiceBAUDFP = choiceBAUDFP;
}

// getting lessonmodule
public String getlessonmodule(){
    return this.lessonmodule;
}

// setting lessonmodule
public void setlessonmodule(String lessonmodule){
    this.lessonmodule = lessonmodule;
}

// getting QuesDescAUDFP2
public String getQuesDescAUDFP2(){
    return this.QuesDescAUDFP2;
}
}
```



```
// setting QuesDescAUDFP2
public void setQuesDescAUDFP2(String QuesDescAUDFP2){
    this.QuesDescAUDFP2 = QuesDescAUDFP2;
}

// getting followupquesaud
public String getfollowupquesaud(){
    return this.followupquesaud;
}

// setting followupquesaud
public void setfollowupquesaud(String
followupquesaud){this.followupquesaud = followupquesaud;}

// getting followupquestext
public String getfollowupquestext(){
    return this.followupquestext;
}

// setting followupquestext
public void setfollowupquestext(String
followupquestext){this.followupquestext = followupquestext;}
}
```

MusicService

```
package com.example.markligot.embrasd;
```

```
import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.Binder;
import android.os.IBinder;
import android.widget.Toast;
```

```
public class MusicService extends Service implements
MediaPlayer.OnErrorListener {
```

```
    private final IBinder mBinder = new ServiceBinder();
    MediaPlayer mPlayer;
    private int length = 0;
```

```
    public MusicService() {
    }
```

```
    public class ServiceBinder extends Binder {
        public MusicService getService() {
            return MusicService.this;
        }
    }
```

```
@Override
public IBinder onBind(Intent arg0) {
    return mBinder;
}
```

```
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
```

```
    try {
        if (mPlayer != null) {
            mPlayer.stop();
            mPlayer.reset();
            mPlayer.release();
            mPlayer = null;
        }
    }
```

```
        mPlayer = MediaPlayer.create(this, R.raw.bgmusic2);
        mPlayer.setLooping(true);
        mPlayer.setVolume(100,100);
        mPlayer.start();
    } catch (Exception e) {
    }
    return START_STICKY;
}

public void pauseMusic() {
    if (mPlayer.isPlaying()) {
        mPlayer.pause();
        length = mPlayer.getCurrentPosition();
    }
}
```

```
public void resumeMusic() {
    if (mPlayer.isPlaying() == false) {
        mPlayer.seekTo(length);
        mPlayer.start();
    }
}
```

```
public void stopMusic() {
    mPlayer.stop();
    mPlayer.reset();
    mPlayer.release();
    mPlayer = null;
}
```

```
@Override
public void onDestroy() {
    super.onDestroy();
    if (mPlayer != null) {
        try {
            mPlayer.stop();
            mPlayer.reset();
            mPlayer.release();
        } finally {
            mPlayer = null;
        }
    }
}
```

```
    public boolean onError(MediaPlayer mp, int what, int extra) {

        Toast.makeText(this, "music player failed",
        Toast.LENGTH_SHORT).show();
        if (mPlayer != null) {
            try {
                mPlayer.stop();
                mPlayer.reset();
                mPlayer.release();
            } finally {
                mPlayer = null;
            }
        }
        return false;
    }
}
```

MusicService2

```
package com.example.markligot.embrasd;
```

```
import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.media.MediaPlayer.OnErrorListener;
```



```
import android.os.Binder;
import android.os.IBinder;
import android.widget.Toast;

public class MusicService2 extends Service implements
OnErrorListener {

    private final IBinder mBinder = new ServiceBinder();
    MediaPlayer mPlayer;
    private int length = 0;

    public MusicService2() {
    }

    public class ServiceBinder extends Binder {
        public MusicService2 getService() {
            return MusicService2.this;
        }
    }

    @Override
    public IBinder onBind(Intent arg0) {
        return mBinder;
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {

        try {
            if (mPlayer != null) {
                mPlayer.stop();
                mPlayer.reset();
                mPlayer.release();
                mPlayer = null;
            }

            mPlayer = MediaPlayer.create(this, R.raw.bgmusic2);
            mPlayer.setLooping(true);
            mPlayer.setVolume(.3f,.3f);
            mPlayer.start();
        } catch (Exception e) {
        }
        return START_STICKY;
    }

    public void pauseMusic() {
        if (mPlayer.isPlaying()) {
            mPlayer.pause();
            length = mPlayer.getCurrentPosition();
        }
    }

    public void resumeMusic() {
        if (mPlayer.isPlaying() == false) {
            mPlayer.seekTo(length);
            mPlayer.start();
        }
    }

    public void stopMusic() {
        mPlayer.stop();
        mPlayer.reset();
        mPlayer.release();
        mPlayer = null;
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
    }
}
```

```
if (mPlayer != null) {
    try {
        mPlayer.stop();
        mPlayer.reset();
        mPlayer.release();
    } finally {
        mPlayer = null;
    }
}

public boolean onError(MediaPlayer mp, int what, int extra) {

    Toast.makeText(this, "music player failed",
    Toast.LENGTH_SHORT).show();
    if (mPlayer != null) {
        try {
            mPlayer.stop();
            mPlayer.reset();
            mPlayer.release();
        } finally {
            mPlayer = null;
        }
    }
    return false;
}

}
```

MusicService3

```
package com.example.markligot.embrasd;

import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.media.MediaPlayer.OnErrorListener;
import android.os.Binder;
import android.os.IBinder;
import android.widget.Toast;

public class MusicService3 extends Service implements
OnErrorListener {

    private final IBinder mBinder = new ServiceBinder();
    MediaPlayer mPlayer;
    private int length = 0;

    public MusicService3() {
    }

    public class ServiceBinder extends Binder {
        public MusicService3 getService() {
            return MusicService3.this;
        }
    }

    @Override
    public IBinder onBind(Intent arg0) {
        return mBinder;
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        try {
            if (mPlayer != null) {
                mPlayer.stop();
                mPlayer.reset();
                mPlayer.release();
            }
        }
    }
}
```



```
        mPlayer= null;
    }

    mPlayer = MediaPlayer.create(this, R.raw.bgmusic2);
    mPlayer.setLooping(true);
    mPlayer.setVolume(.3f,.3f);
    mPlayer.start();
    } catch (Exception e) {
    }
    return START_STICKY;
}

public void pauseMusic() {
    if (mPlayer.isPlaying()) {
        mPlayer.pause();
        length = mPlayer.getCurrentPosition();
    }
}

public void resumeMusic() {
    if (mPlayer.isPlaying() == false) {
        mPlayer.seekTo(length);
        mPlayer.start();
    }
}

public void stopMusic() {
    mPlayer.stop();
    mPlayer.reset();
    mPlayer.release();
    mPlayer = null;
}

@Override
public void onDestroy() {
    super.onDestroy();
    if (mPlayer != null) {
        try {
            mPlayer.stop();
            mPlayer.reset();
            mPlayer.release();
        } finally {
            mPlayer = null;
        }
    }
}

public boolean onError(MediaPlayer mp, int what, int extra) {

    Toast.makeText(this, "music player failed",
    Toast.LENGTH_SHORT).show();
    if (mPlayer != null) {
        try {
            mPlayer.stop();
            mPlayer.reset();
            mPlayer.release();
        } finally {
            mPlayer = null;
        }
    }
    return false;
}

}

MusicService4

package com.example.markligot.embrasd;
```

```
import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.media.MediaPlayer.OnErrorListener;
import android.os.Binder;
import android.os.IBinder;
import android.widget.Toast;

public class MusicService4 extends Service implements
OnErrorListener {

    private final IBinder mBinder = new ServiceBinder();
    MediaPlayer mPlayer;
    private int length = 0;

    public MusicService4() {
    }

    public class ServiceBinder extends Binder {
        public MusicService4 getService() {
            return MusicService4.this;
        }
    }

    @Override
    public IBinder onBind(Intent arg0) {
        return mBinder;
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        try {
            if (mPlayer != null) {
                mPlayer.stop();
                mPlayer.reset();
                mPlayer.release();
                mPlayer= null;
            }

            mPlayer = MediaPlayer.create(this, R.raw.bgmusic2);
            mPlayer.setLooping(true);
            mPlayer.setVolume(.3f,.3f);
            mPlayer.start();
        } catch (Exception e) {
        }
        return START_STICKY;
    }

    public void pauseMusic() {
        if (mPlayer.isPlaying()) {
            mPlayer.pause();
            length = mPlayer.getCurrentPosition();
        }
    }

    public void resumeMusic() {
        if (mPlayer.isPlaying() == false) {
            mPlayer.seekTo(length);
            mPlayer.start();
        }
    }

    public void stopMusic() {
        mPlayer.stop();
        mPlayer.reset();
        mPlayer.release();
        mPlayer = null;
    }
}
```



```
@Override
public void onDestroy() {
    super.onDestroy();
    if (mPlayer != null) {
        try {
            mPlayer.stop();
            mPlayer.reset();
            mPlayer.release();
        } finally {
            mPlayer = null;
        }
    }
}

public boolean onError(MediaPlayer mp, int what, int extra) {

    Toast.makeText(this, "music player failed",
        Toast.LENGTH_SHORT).show();
    if (mPlayer != null) {
        try {
            mPlayer.stop();
            mPlayer.reset();
            mPlayer.release();
        } finally {
            mPlayer = null;
        }
    }
    return false;
}

}
```

MyDBEmbrASD

```
package com.example.markligot.embrasd;
```

```
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteQueryBuilder;
```

```
import java.util.ArrayList;
import java.util.List;
```

```
/**
 * Created by markligot on 1/7/17.
 */
```

```
public class MyDBEmbrASD extends SQLiteAssetHelper {
```

```
    private static final String DATABASE_NAME =
        "EmbrASDdb.db";
    private static final int DATABASE_VERSION = 1;
    public Cursor cursor;
```

```
    public MyDBEmbrASD(Context context) {
        super(context, DATABASE_NAME, null,
            DATABASE_VERSION);
```

```
        // you can use an alternate constructor to specify a database
        location
        // (such as a folder on the sd card)
        // you must ensure that this folder is available and you have
        permission
        // to write to it
        //super(context, DATABASE_NAME,
        context.getExternalFilesDir(null).getAbsolutePath(), null,
        DATABASE_VERSION);
    }
}
```

```
public Cursor getResources() {

    if (cursor != null) {
        cursor.close();
    }

    SQLiteDatabase db = getReadableDatabase();
    SQLiteQueryBuilder qb = new SQLiteQueryBuilder();

    String[] sqlSelect = {"0_id", "ImgFP", "AudFP"};
    String sqlTables = "ResTable";

    qb.setTables(sqlTables);
    cursor = qb.query(db, sqlSelect, null, null,
        null, null, null);

    cursor.moveToFirst();
    cursor.close();
    db.close();

    return cursor;
}

public Cursor getAnswerTable() {
    if (cursor != null) {
        cursor.close();
    }

    SQLiteDatabase db = getReadableDatabase();
    SQLiteQueryBuilder qb = new SQLiteQueryBuilder();

    String[] sqlSelect = {"0_id", "QuesDesc", "CorrectAns"};
    String sqlTables = "AnsTable";

    qb.setTables(sqlTables);
    cursor = qb.query(db, sqlSelect, null, null,
        null, null, null);

    cursor.moveToFirst();

    cursor.close();
    db.close();
    return cursor;
}

// Getting All Resource
public List<ResourceTable> getAllResource(String lsmodule) {

    if (cursor != null) {
        cursor.close();
    }

    List<ResourceTable> resourceTableList = new
    ArrayList<ResourceTable>();
    // Select All Query

    SQLiteDatabase db = getReadableDatabase();
    SQLiteQueryBuilder qb = new SQLiteQueryBuilder();

    String selectQuery = "SELECT * FROM ResTable WHERE
    LessonModule = " + lsmodule;

    cursor = db.rawQuery(selectQuery, null);

    // looping through all rows and adding to list
    if (cursor.moveToFirst()) {
        do {
            ResourceTable resource = new ResourceTable();
            resource.setID(Integer.parseInt(cursor.getString(0)));
            resource.setimgFP(cursor.getString(1));
            resource.setaudioFP(cursor.getString(2));
            // Adding resource to list
        } while (cursor.moveToNext());
    }
}
```



```
resourceTableList.add(resource);
} while (cursor.moveToNext());
}
// return resource list
cursor.close();
db.close();
return resourceTableList;
}
// Getting All Answer
public List<AnswerTable> getAllAnswer() {
    if (cursor != null) {
        cursor.close();
    }
    List<AnswerTable> answerTableList = new
    ArrayList<AnswerTable>();
    // Select All Query

    SQLiteDatabase db = getReadableDatabase();
    SQLiteQueryBuilder qb = new SQLiteQueryBuilder();
    String selectQuery = "SELECT * FROM AnsTable";
    cursor = db.rawQuery(selectQuery, null);

    // looping through all rows and adding to list
    if (cursor.moveToFirst()) {
        do {
            AnswerTable resource2 = new AnswerTable();
            resource2.setID(Integer.parseInt(cursor.getString(0)));
            resource2.setquesDesc(cursor.getString(1));
            resource2.setcorrectAns(cursor.getString(2));
            // Adding resource to list
            answerTableList.add(resource2);
        } while (cursor.moveToNext());
    }

    // return resource list
    cursor.close();
    db.close();
    return answerTableList;
}

//get Resource Count
public int getResourcesCount() {
    if (cursor != null) {
        cursor.close();
    }
    String countQuery = "SELECT * FROM ResTable";
    SQLiteDatabase db = this.getReadableDatabase();
    cursor = db.rawQuery(countQuery, null);

    // return count
    cursor.close();
    return cursor.getCount();
}

public int getAnswerCount() {
    if (cursor != null) {
        cursor.close();
    }
    String countQuery = "SELECT * FROM AnsTable";
    SQLiteDatabase db = this.getReadableDatabase();
    cursor = db.rawQuery(countQuery, null);

    // return count
    cursor.close();
    db.close();
    return cursor.getCount();
}

public List<QuestionGetterSetter> QuestionGetterSetter(String
lsnmodule) {
    if (cursor != null) {
        cursor.close();
    }
}
```

```
List<QuestionGetterSetter> answerTableList = new
ArrayList<QuestionGetterSetter>();

SQLiteDatabase db = getReadableDatabase();
SQLiteQueryBuilder qb = new SQLiteQueryBuilder();

String ansQuery = "SELECT a._id, a.QuesDesc,
a.CorrectAns, b.ImgFP, a.LessonModule, a.QImgFP, a.QAudFP,
b.AudFP FROM AnsTable a, ResTable b " +
"WHERE a.CorrectAns = b._ID AND a.LessonModule
= " + lsnmodule + " ORDER BY a._ID ASC";
cursor = db.rawQuery(ansQuery, null);

// looping through all rows and adding to list
if (cursor.moveToFirst()) {
    do {
        QuestionGetterSetter resource3 = new
        QuestionGetterSetter();
        resource3.setQuesNum(cursor.getString(0));
        resource3.setquesDesc(cursor.getString(1));
        resource3.setcorrAns(cursor.getString(2));
        resource3.setcorrAnsIMPID(cursor.getString(3));
        resource3.setlessnModule(cursor.getString(4));
        resource3.setqImgFP(cursor.getString(5));
        resource3.setqAudFP(cursor.getString(6));
        resource3.setcorrAnsAUDID(cursor.getString(7));

        // Adding resource to list
        answerTableList.add(resource3);
    } while (cursor.moveToNext());
}
// return resource list
cursor.close();
db.close();
return answerTableList;
}

public List<moduleFourGetterSetter>
moduleFourGetterSetter(String lsnmodule) {
    if (cursor != null) {
        cursor.close();
    }
    List<moduleFourGetterSetter> moduleFourTableList = new
    ArrayList<moduleFourGetterSetter>();

    SQLiteDatabase db = getReadableDatabase();
    SQLiteQueryBuilder qb = new SQLiteQueryBuilder();

    String ansQuery = "SELECT _id, QuesDesc1, QuesDesc2,
choiceA, choiceB, corrChoice, DescIMGFP, QuesDescAUDFP, " +
"CorrAnsAUDFP, choiceAAUDFP, choiceBAUDFP,
lessonmodule, QuesDescAUDFP2, followupquesaud, " +
"followupquestext FROM modulefour where
lessonmodule = " + lsnmodule + " ORDER BY _ID ASC";
    cursor = db.rawQuery(ansQuery, null);

    // looping through all rows and adding to list
    if (cursor.moveToFirst()) {
        do {
            moduleFourGetterSetter resource4 = new
            moduleFourGetterSetter();
            resource4.setQuesNum1(cursor.getString(0));
            resource4.setquesDesc1(cursor.getString(1));
            resource4.setquesDesc2(cursor.getString(2));
            resource4.setchoiceA(cursor.getString(3));
            resource4.setchoiceB(cursor.getString(4));
            resource4.setcorrChoice(cursor.getString(5));
            resource4.setdescIMGFP(cursor.getString(6));
            resource4.setquesDescAUDFP(cursor.getString(7));
}
```




```
resource4.setcorrANSAUDFP(cursor.getString(8));
resource4.setchoiceAAUDFP(cursor.getString(9));
resource4.setchoiceBAUDFP(cursor.getString(10));
resource4.setlessonmodule(cursor.getString(11));
resource4.setQuesDescAUDFP2(cursor.getString(12));
resource4.setfollowupquesaud(cursor.getString(13));
resource4.setfollowupquestext(cursor.getString(14));

// Adding resource to list
moduleFourTableList.add(resource4);
} while (cursor.moveToNext());
}
// return resource list
cursor.close();
db.close();
return moduleFourTableList;
}
```

QuestionGetterSetter

```
package com.example.markligot.embrasd;
```

```
/**
 * Created by markligot on 1/4/17.
 */
```

```
public class QuestionGetterSetter {
    //private variables
```

```
String quesNum;
String quesDesc;
String corrAns;
String corrAnsIMPID;
String lessnModule;
String qImgFP;
String qAudFP;
String corrAnsAUDID;
```

```
// Empty constructor
```

```
public QuestionGetterSetter(){
```

```
}
```

```
// constructor
```

```
public QuestionGetterSetter(String quesNum, String quesDesc,
String corrAns, String corrAnsIMPID, String lessnModule, String
qImgFP, String qAudFP, String corrAnsAUDID){
```

```
    this.quesNum = quesNum;
    this.quesDesc = quesDesc;
    this.corrAns = corrAns;
    this.corrAnsIMPID = corrAnsIMPID;
    this.lessnModule = lessnModule;
    this.qImgFP = qImgFP;
    this.qAudFP = qAudFP;
    this.corrAnsAUDID = corrAnsAUDID;
}
```

```
// constructor
```

```
public QuestionGetterSetter(String lessnModule){
```

```
    this.lessnModule = lessnModule;
```

```
}
```

```
// getting quesNum
```

```
public String getQuesNum(){
```

```
    return this.quesNum;
```

```
}
```

```
// setting quesNum
```

```
public void setQuesNum(String quesNum){
```

```
    this.quesNum = quesNum;
```

```
}
```

```
// getting quesDesc
```

```
public String getquesDesc(){
```

```
    return this.quesDesc;
```

```
}
```

```
// setting quesDesc
```

```
public void setquesDesc(String quesDesc){
```

```
    this.quesDesc = quesDesc;
```

```
}
```

```
// getting corrAns
```

```
public String getcorrAns(){
```

```
    return this.corrAns;
```

```
}
```

```
// setting corrAns
```

```
public void setcorrAns(String corrAns){
```

```
    this.corrAns = corrAns;
```

```
}
```

```
// getting corrAnsIMPID
```

```
public String getcorrAnsIMPID(){
```

```
    return this.corrAnsIMPID;
```

```
}
```

```
// setting corrAnsIMPID
```

```
public void setcorrAnsIMPID(String corrAnsIMPID){
```

```
    this.corrAnsIMPID = corrAnsIMPID;
```

```
}
```

```
// getting lessnModule
```

```
public String getlessnModule(){
```

```
    return this.lessnModule;
```

```
}
```

```
// setting lessnModule
```

```
public void setlessnModule(String lessnModule){
```

```
    this.lessnModule = lessnModule;
```

```
}
```

```
// getting qImgFP
```

```
public String getqImgFP(){
```

```
    return this.qImgFP;
```

```
}
```

```
// setting qImgFP
```

```
public void setqImgFP(String qImgFP){
```

```
    this.qImgFP = qImgFP;
```

```
}
```

```
// getting qAudFP
```

```
public String getqAudFP(){
```

```
    return this.qAudFP;
```

```
}
```

```
// setting qAudFP
```

```
public void setqAudFP(String qAudFP){
```

```
    this.qAudFP = qAudFP;
```

```
}
```

```
// getting corrAnsAUDID
```

```
public String getcorrAnsAUDID(){
```

```
    return this.corrAnsAUDID;
```

```
}
```

```
// setting corrAnsAUDID
```

```
public void setcorrAnsAUDID(String corrAnsAUDID){
```

```
    this.corrAnsAUDID = corrAnsAUDID;
```

```
}
```

```
}
```



ReadingComprehensionFragment

package com.example.markligot.embrasd;

/**
 * Created by markligot on 1/2/17.
 */

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class ReadingComprehensionFragment **extends** Fragment {

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState) {

View rootView =
inflater.inflate(R.layout.fragment_readingcomprehension, container,
false);

return rootView;
}

ResourceTable

package com.example.markligot.embrasd;

/**
 * Created by markligot on 1/4/17.
 */

public class ResourceTable {

//private variables
int _id;
String **ImgFP**;
String **AudioFP**;

// Empty constructor
public ResourceTable(){

}

// constructor

public ResourceTable(**int** id, String imgFP, String audFP){
this._id = id;
this.**ImgFP** = imgFP;
this.**AudioFP** = audFP;
}

// constructor

public ResourceTable(String imgFP, String audioFP){
this.**ImgFP** = imgFP;
this.**AudioFP** = audioFP;
}

// getting ID
public int getID(){
return this._id;
}

// setting id
public void setId(**int** id){
this._id = id;
}

// getting imgFP

public String getimgFP(){
return this.**ImgFP**;
}

// setting imgFP
public void setimgFP(String imgFP){
this.**ImgFP** = imgFP;
}

// getting audioFP
public String getaudioFP(){
return this.**AudioFP**;
}

// setting audioFP
public void setaudioFP(String audioFP){
this.**AudioFP** = audioFP;
}

RetrieveImgResGetter

package com.example.markligot.embrasd;

/**
 * Created by markligot on 1/4/17.
 */

public class RetrieveImgResGetter {

//private variables
String[] **myShuffledArray**;

// Empty constructor

public RetrieveImgResGetter() {

}

// constructor

public RetrieveImgResGetter(String[] myShuffledArray) {
this.**myShuffledArray** = myShuffledArray;
}

}

// getting imgFP
public String[] getmyShuffledArray(){
return this.**myShuffledArray**;
}

// setting imgFP
public void setmyShuffledArray(String[] shufldArray){
this.**myShuffledArray** = **myShuffledArray**;
}

SplashScreen

package com.example.markligot.embrasd;

/**
 * Created by Mark Ligot on 2/8/2017.
 */

import android.app.Activity;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.os.Handler;
import android.view.Window;



```
public class SplashScreen extends Activity {

    // Splash screen timer
    private static int SPLASH_TIME_OUT = 1000;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.testing_splash);

        super.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        new Handler().postDelayed(new Runnable() {

            /*
             * Showing splash screen with a timer. This will be useful when
             * you want to show case your app logo / company
             */

            @Override
            public void run() {
                // This method will be executed once the timer is over
                // Start your app main activity
                Intent i = new Intent(SplashScreen.this,
                MainActivity.class);
                startActivity(i);

                // close this activity
                finish();
            }
        }, SPLASH_TIME_OUT);
    }
}
```

SQLiteAssetHelper

```
/*
 * Copyright (C) 2011 readyState Software Ltd, 2007 The Android
 * Open Source Project
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
 * either express or implied.
 * See the License for the specific language governing permissions
 * and
 * limitations under the License.
 */

package com.example.markligot.embrasd;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteDatabase.CursorFactory;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
```

```
import java.io.InputStream;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.zip.ZipInputStream;

/**
 * A helper class to manage database creation and version
 * management using
 * an application's raw asset files.
 *
 * This class provides developers with a simple way to ship their
 * Android app
 * with an existing SQLite database (which may be pre-populated with
 * data) and
 * to manage its initial creation and any upgrades required with
 * subsequent
 * version releases.
 *
 * <p>This class makes it easy for {@link
 * android.content.ContentProvider}
 * implementations to defer opening and upgrading the database until
 * first use,
 * to avoid blocking application startup with long-running database
 * upgrades.
 *
 * <p>For examples see <a href="https://github.com/jgilfelt/android-
 * sqlite-asset-helper">
 * https://github.com/jgilfelt/android-sqlite-asset-helper</a>
 *
 * <p class="note"><strong>Note:</strong> this class assumes
 * monotonically increasing version numbers for upgrades. Also,
 * there
 * is no concept of a database downgrade; installing a new version of
 * your app which uses a lower version number than a
 * previously-installed version will result in undefined behavior. </p>
 */
```

```
public class SQLiteAssetHelper extends SQLiteOpenHelper {

    private static final String TAG =
    SQLiteAssetHelper.class.getSimpleName();
    private static final String ASSET_DB_PATH = "";

    private final Context mContext;
    private final String mName;
    private final CursorFactory mFactory;
    private final int mNewVersion;

    private SQLiteDatabase mDatabase = null;
    private boolean mIsInitializing = false;

    private String mDatabasePath;

    private String mAssetPath;

    private String mUpgradePathFormat;

    private int mForcedUpgradeVersion = 0;

    /**
     * Create a helper object to create, open, and/or manage a
     * database in
     * a specified location.
     * This method always returns very quickly. The database is not
     * actually
     * created or opened until one of {@link #getWritableDatabase} or
     * {@link #getReadableDatabase} is called.
     *
     * @param context to use to open or create the database
     * @param name of the database file
     * @param storageDirectory to store the database file upon
     */
```



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740
Bachelor of Science in Information Technology

107

```
creation; caller must
* ensure that the specified absolute path is available and can be
written to
* @param factory to use for creating cursor objects, or null for the
default
* @param version number of the database (starting at 1); if the
database is older,
* SQL file(s) contained within the application assets folder will
be used to
* upgrade the database
*/
public SQLiteAssetHelper(Context context, String name, String
storageDirectory, CursorFactory factory, int version) {
    super(context, name, factory, version);

    if (version < 1) throw new IllegalArgumentException("Version
must be >= 1, was " + version);
    if (name == null) throw new
IllegalArgumentException("Database name cannot be null");

    mContext = context;
    mName = name;
    mFactory = factory;
    mNewVersion = version;

    mAssetPath = ASSET_DB_PATH + " " + name;
    if (storageDirectory != null) {
        mDatabasePath = storageDirectory;
    } else {
        mDatabasePath = context.getApplicationInfo().dataDir + " ";
    }
    mUpgradePathFormat = ASSET_DB_PATH + " " + name +
" _upgrade_%s-%s.sql";
}

/**
 * Create a helper object to create, open, and/or manage a
database in
 * the application's default private data directory.
 * This method always returns very quickly. The database is not
actually
 * created or opened until one of {@link #getWritableDatabase} or
 * {@link #getReadableDatabase} is called.
 *
 * @param context to use to open or create the database
 * @param name of the database file
 * @param factory to use for creating cursor objects, or null for the
default
 * @param version number of the database (starting at 1); if the
database is older,
 * SQL file(s) contained within the application assets folder will
be used to
 * upgrade the database
 */
public SQLiteAssetHelper(Context context, String name,
CursorFactory factory, int version) {
    this(context, name, null, factory, version);
}

/**
 * Create and/or open a database that will be used for reading and
writing.
 * The first time this is called, the database will be extracted and
copied
 * from the application's assets folder.
 *
 * <p>Once opened successfully, the database is cached, so you
can
 * call this method every time you need to write to the database.
 * (Make sure to call {@link #close} when you no longer need the
database.)
```

```
* Errors such as bad permissions or a full disk may cause this
method
* to fail, but future attempts may succeed if the problem is
fixed.</p>
*
* <p class="caution">Database upgrade may take a long time,
you
* should not call this method from the application main thread,
including
* from {@link android.content.ContentProvider#onCreate
ContentProvider.onCreate()}.
*
* @throws SQLiteException if the database cannot be opened for
writing
* @return a read/write database object valid until {@link #close}
is called
*/
@Override
public synchronized SQLiteDatabase getWritableDatabase() {
    if (mDatabase != null && mDatabase.isOpen() &&
!mDatabase.isReadOnly()) {
        return mDatabase; // The database is already open for
business
    }

    if (mIsInitializing) {
        throw new IllegalStateException("getWritableDatabase
called recursively");
    }

    // If we have a read-only database open, someone could be using
it
    // (though they shouldn't), which would cause a lock to be held
on
    // the file, and our attempts to open the database read-write
would
    // fail waiting for the file lock. To prevent that, we acquire the
// lock on the read-only database, which shuts out other users.

    boolean success = false;
    SQLiteDatabase db = null;
    if (mDatabase != null) mDatabase.lock();
    try {
        mIsInitializing = true;
        //if (mName == null) {
        //    db = SQLiteDatabase.create(null);
        //} else {
        //    db = mContext.openOrCreateDatabase(mName, 0,
mFactory);
        //}
        db = createOrOpenDatabase(false);

        int version = db.getVersion();

        // do force upgrade
        if (version != 0 && version < mForcedUpgradeVersion) {
            db = createOrOpenDatabase(true);
            db.setVersion(mNewVersion);
            version = db.getVersion();
        }

        if (version != mNewVersion) {
            db.beginTransaction();
            try {
                if (version == 0) {
                    onCreate(db);
                } else {
                    if (version > mNewVersion) {
                        Log.w(TAG, "Can't downgrade read-only
database from version " +
                            version + " to " + mNewVersion + ": " +
```



```
db.getPath());
    }
    onUpgrade(db, version, mNewVersion);
    }
    db.setVersion(mNewVersion);
    db.setTransactionSuccessful();
    } finally {
        db.endTransaction();
    }
}

onOpen(db);
success = true;
return db;
} finally {
    mIsInitializing = false;
    if (success) {
        if (mDatabase != null) {
            try { mDatabase.close(); } catch (Exception e) { }
            //mDatabase.unlock();
        }
        mDatabase = db;
    } else {
        //if (mDatabase != null) mDatabase.unlock();
        if (db != null) db.close();
    }
}
}

/**
 * Create and/or open a database. This will be the same object
 * returned by
 * {@link #getWritableDatabase} unless some problem, such as a
 * full disk,
 * * requires the database to be opened read-only. In that case, a
 * read-only
 * * database object will be returned. If the problem is fixed, a future
 * call
 * * to {@link #getWritableDatabase} may succeed, in which case
 * the read-only
 * * database object will be closed and the read/write object will be
 * returned
 * * in the future.
 * *
 * * <p class="caution">Like {@link #getWritableDatabase}, this
 * method may
 * * take a long time to return, so you should not call it from the
 * * application main thread, including from
 * * {@link android.content.ContentProvider#onCreate
 * ContentProvider.onCreate()}.
 * *
 * * @throws SQLiteException if the database cannot be opened
 * * @return a database object valid until {@link
 * #getWritableDatabase}
 * * or {@link #close} is called.
 */
@Override
public synchronized SQLiteDatabase getReadableDatabase() {
    if (mDatabase != null && mDatabase.isOpen()) {
        return mDatabase; // The database is already open for
        business
    }

    if (mIsInitializing) {
        throw new IllegalStateException("getReadableDatabase
        called recursively");
    }

    try {
        return getWritableDatabase();
    } catch (SQLiteException e) {
        if (mName == null) throw e; // Can't open a temp database
        read-only!
        Log.e(TAG, "Couldn't open " + mName + " for writing
        (will try read-only);", e);
    }

    SQLiteDatabase db = null;
    try {
        mIsInitializing = true;
        String path = mContext.getDatabasePath(mName).getPath();
        db = SQLiteDatabase.openDatabase(path, mFactory,
        SQLiteDatabase.OPEN_READONLY);
        if (db.getVersion() != mNewVersion) {
            throw new SQLiteException("Can't upgrade read-only
            database from version " +
            db.getVersion() + " to " + mNewVersion + ": " +
            path);
        }

        onOpen(db);
        Log.w(TAG, "Opened " + mName + " in read-only mode");
        mDatabase = db;
        return mDatabase;
    } finally {
        mIsInitializing = false;
        if (db != null && db != mDatabase) db.close();
    }
}

/**
 * Close any open database object.
 */
@Override
public synchronized void close() {
    if (mIsInitializing) throw new IllegalStateException("Closed
    during initialization");

    if (mDatabase != null && mDatabase.isOpen()) {
        mDatabase.close();
        mDatabase = null;
    }
}

@Override
public final void onConfigure(SQLiteDatabase db) {
    // not supported!
}

@Override
public final void onCreate(SQLiteDatabase db) {
    // do nothing - createOrOpenDatabase() is called in
    // getWritableDatabase() to handle database creation.
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
    newVersion) {

    Log.w(TAG, "Upgrading database " + mName + " from
    version " + oldVersion + " to " + newVersion + "...");

    ArrayList<String> paths = new ArrayList<String>();
    getUpgradeFilePaths(oldVersion, newVersion-1, newVersion,
    paths);

    if (paths.isEmpty()) {
        Log.e(TAG, "no upgrade script path from " + oldVersion +
        " to " + newVersion);
        throw new SQLiteAssetException("no upgrade script path
        from " + oldVersion + " to " + newVersion);
    }
}
```

```
    } catch (SQLiteException e) {
        if (mName == null) throw e; // Can't open a temp database
        read-only!
        Log.e(TAG, "Couldn't open " + mName + " for writing
        (will try read-only);", e);
    }

    SQLiteDatabase db = null;
    try {
        mIsInitializing = true;
        String path = mContext.getDatabasePath(mName).getPath();
        db = SQLiteDatabase.openDatabase(path, mFactory,
        SQLiteDatabase.OPEN_READONLY);
        if (db.getVersion() != mNewVersion) {
            throw new SQLiteException("Can't upgrade read-only
            database from version " +
            db.getVersion() + " to " + mNewVersion + ": " +
            path);
        }

        onOpen(db);
        Log.w(TAG, "Opened " + mName + " in read-only mode");
        mDatabase = db;
        return mDatabase;
    } finally {
        mIsInitializing = false;
        if (db != null && db != mDatabase) db.close();
    }
}

/**
 * Close any open database object.
 */
@Override
public synchronized void close() {
    if (mIsInitializing) throw new IllegalStateException("Closed
    during initialization");

    if (mDatabase != null && mDatabase.isOpen()) {
        mDatabase.close();
        mDatabase = null;
    }
}

@Override
public final void onConfigure(SQLiteDatabase db) {
    // not supported!
}

@Override
public final void onCreate(SQLiteDatabase db) {
    // do nothing - createOrOpenDatabase() is called in
    // getWritableDatabase() to handle database creation.
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
    newVersion) {

    Log.w(TAG, "Upgrading database " + mName + " from
    version " + oldVersion + " to " + newVersion + "...");

    ArrayList<String> paths = new ArrayList<String>();
    getUpgradeFilePaths(oldVersion, newVersion-1, newVersion,
    paths);

    if (paths.isEmpty()) {
        Log.e(TAG, "no upgrade script path from " + oldVersion +
        " to " + newVersion);
        throw new SQLiteAssetException("no upgrade script path
        from " + oldVersion + " to " + newVersion);
    }
}
```



```
}

Collections.sort(paths, new VersionComparator());
for (String path : paths) {
    try {
        Log.w(TAG, "processing upgrade: " + path);
        InputStream is = mContext.getAssets().open(path);
        String sql = Utils.convertStreamToString(is);
        if (sql != null) {
            List<String> cmds = Utils.splitSqlScript(sql, ';');
            for (String cmd : cmds) {
                //Log.d(TAG, "cmd=" + cmd);
                if (cmd.trim().length() > 0) {
                    db.execSQL(cmd);
                }
            }
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}

Log.w(TAG, "Successfully upgraded database " + mName + "
from version " + oldVersion + " to " + newVersion);

}

@Override
public final void onDowngrade(SQLiteDatabase db, int
oldVersion, int newVersion) {
    // not supported!
}

/**
 * Bypass the upgrade process (for each increment up to a given
version) and simply
 * overwrite the existing database with the supplied asset file.
 *
 * @param version bypass upgrade up to this version number -
should never be greater than the
 * latest database version.
 *
 * @deprecated use {@link #setForcedUpgrade} instead.
 */
@Deprecated
public void setForcedUpgradeVersion(int version) {
    setForcedUpgrade(version);
}

/**
 * Bypass the upgrade process (for each increment up to a given
version) and simply
 * overwrite the existing database with the supplied asset file.
 *
 * @param version bypass upgrade up to this version number -
should never be greater than the
 * latest database version.
 */
public void setForcedUpgrade(int version) {
    mForcedUpgradeVersion = version;
}

/**
 * Bypass the upgrade process for every version increment and
simply overwrite the existing
 * database with the supplied asset file.
 */
public void setForcedUpgrade() {
    setForcedUpgrade(mNewVersion);
}
```

```
private SQLiteDatabase createOrOpenDatabase(boolean force)
throws SQLiteAssetException {

    // test for the existence of the db file first and don't attempt open
    // to prevent the error trace in log on API 14+
    SQLiteDatabase db = null;
    File file = new File(mDatabasePath + "/" + mName);
    if (file.exists()) {
        db = returnDatabase();
    }
    //SQLiteDatabase db = returnDatabase();

    if (db != null) {
        // database already exists
        if (force) {
            Log.w(TAG, "forcing database upgrade!");
            copyDatabaseFromAssets();
            db = returnDatabase();
        }
        return db;
    } else {
        // database does not exist, copy it from assets and return it
        copyDatabaseFromAssets();
        db = returnDatabase();
        return db;
    }
}

private SQLiteDatabase returnDatabase(){
    try {
        SQLiteDatabase db =
SQLiteDatabase.openDatabase(mDatabasePath + "/" + mName,
mFactory, SQLiteDatabase.OPEN_READWRITE);
        Log.i(TAG, "successfully opened database " + mName);
        return db;
    } catch (SQLiteException e) {
        Log.w(TAG, "could not open database " + mName + " - " +
e.getMessage());
        return null;
    }
}

private void copyDatabaseFromAssets() throws
SQLiteAssetException {
    Log.w(TAG, "copying database from assets...");

    String path = mAssetPath;
    String dest = mDatabasePath + "/" + mName;
    InputStream is;
    boolean isZip = false;

    try {
        // try uncompressed
        is = mContext.getAssets().open(path);
    } catch (IOException e) {
        // try zip
        try {
            is = mContext.getAssets().open(path + ".zip");
            isZip = true;
        } catch (IOException e2) {
            // try gzip
            try {
                is = mContext.getAssets().open(path + ".gz");
            } catch (IOException e3) {
                SQLiteAssetException se = new
SQLiteAssetException("Missing " + mAssetPath + " file (or .zip, .gz
archive) in assets, or target folder not writable");
                se.setStackTrace(e3.getStackTrace());
                throw se;
            }
        }
    }
}
```



```
    }

    try {
        File f = new File(mDatabasePath + "/" + " ");
        if (!f.exists()) { f.mkdir(); }
        if (isZip) {
            ZipInputStream zis = Utils.getFileFromZip(is);
            if (zis == null) {
                throw new SQLiteAssetException("Archive is missing
a SQLite database file");
            }
            Utils.writeExtractedFileToDisk(zis, new
FileOutputStream(dest));
        } else {
            Utils.writeExtractedFileToDisk(is, new
FileOutputStream(dest));
        }

        Log.w(TAG, "database copy complete");

    } catch (IOException e) {
        SQLiteAssetException se = new
SQLiteAssetException("Unable to write " + dest + " to data
directory");
        se.setStackTrace(e.getStackTrace());
        throw se;
    }

    private InputStream getUpgradeSQLStream(int oldVersion, int
newVersion) {
        String path = String.format(mUpgradePathFormat, oldVersion,
newVersion);
        try {
            return mContext.getAssets().open(path);
        } catch (IOException e) {
            Log.w(TAG, "missing database upgrade script: " + path);
            return null;
        }
    }

    private void getUpgradeFilePaths(int baseVersion, int start, int
end, ArrayList<String> paths) {

        int a;
        int b;

        InputStream is = getUpgradeSQLStream(start, end);
        if (is != null) {
            String path = String.format(mUpgradePathFormat, start, end);
            paths.add(path);
            //Log.d(TAG, "found script: " + path);
            a = start - 1;
            b = start;
            is = null;
        } else {
            a = start - 1;
            b = end;
        }

        if (a < baseVersion) {
            return;
        } else {
            getUpgradeFilePaths(baseVersion, a, b, paths); // recursive
call
        }

    }

    /**
     * An exception that indicates there was an error with SQLite asset
```

```
retrieval or parsing.
 */
@SuppressWarnings("serial")
public static class SQLiteAssetException extends
SQLiteException {

    public SQLiteAssetException() {}

    public SQLiteAssetException(String error) {
        super(error);
    }
}

Utils

package com.example.markligot.embrasad;

import android.util.Log;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
import java.util.zip.ZipEntry;
import java.util.zip.ZipInputStream;

class Utils {

    private static final String TAG =
SQLiteAssetHelper.class.getSimpleName();

    public static List<String> splitSqlScript(String script, char delim)
    {
        List<String> statements = new ArrayList<String>();
        StringBuilder sb = new StringBuilder();
        boolean inLiteral = false;
        char[] content = script.toCharArray();
        for (int i = 0; i < script.length(); i++) {
            if (content[i] == '"') {
                inLiteral = !inLiteral;
            }
            if (content[i] == delim && !inLiteral) {
                if (sb.length() > 0) {
                    statements.add(sb.toString().trim());
                    sb = new StringBuilder();
                }
            } else {
                sb.append(content[i]);
            }
        }
        if (sb.length() > 0) {
            statements.add(sb.toString().trim());
        }
        return statements;
    }

    public static void writeExtractedFileToDisk(InputStream in,
OutputStream outs) throws IOException {
        byte[] buffer = new byte[1024];
        int length;
        while ((length = in.read(buffer)) > 0) {
            outs.write(buffer, 0, length);
        }
        outs.flush();
        outs.close();
        in.close();
    }
}
```




```
public static ZipInputStream getFileFromZip(InputStream
zipFileStream) throws IOException {
    ZipInputStream zis = new ZipInputStream(zipFileStream);
    ZipEntry ze;
    while ((ze = zis.getNextEntry()) != null) {
        Log.w(TAG, "extracting file: '" + ze.getName() + "'...");
        return zis;
    }
    return null;
}
```

```
public static String convertStreamToString(InputStream is) {
    return new Scanner(is).useDelimiter("\\A").next();
}
```

VersionComparator

```
package com.example.markligot.embrasd;
```

```
import android.util.Log;
```

```
import java.util.Comparator;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
```

```
/**
 * Compare paths by their upgrade version numbers, instead of using
 * alphanumeric comparison on plain file names. This prevents the
 * upgrade
 * scripts from being applied out of order when they first move to
 * double-,
 * triple-, etc. digits.
 * <p>
 * For example, this fixes an upgrade that would apply 2 different
 * upgrade
 * files from version 9 to 11 (<code>...updated_9_10</code> and
 * <code>...updated_10_11</code>) from using the
 * <em>incorrect</em>
 * alphanumeric order of <code>10_11</code> before
 * <code>9_10</code>.
 * </p>
 */
```

```
class VersionComparator implements Comparator<String> {
```

```
    private static final String TAG =
    SQLiteAssetHelper.class.getSimpleName();
```

```
    private Pattern pattern = Pattern
    .compile(".*_upgrade_([0-9]+)-([0-9]+).*");
```

```
/**
 * Compares the two specified upgrade script strings to determine
 * their
 * relative ordering considering their two version numbers.
 * Assumes all
 * database names used are the same, as this function only
 * compares the
 * two version numbers.
 *
 * @param file0
 * an upgrade script file name
 * @param file1
 * a second upgrade script file name to compare with file0
 * @return an integer < 0 if file0 should be applied before file1, 0 if
 * they are equal (though that shouldn't happen), and > 0 if
 * file0 should be applied after file1.
 *
 * @exception SQLiteAssetHelper.SQLiteAssetException
```

```
* thrown if the strings are not in the correct upgrade
* script format of:
```

```
<code>dbname_fromVersionInteger_toVersionInteger</code>
*/
```

```
@Override
```

```
public int compare(String file0, String file1) {
    Matcher m0 = pattern.matcher(file0);
    Matcher m1 = pattern.matcher(file1);
```

```
    if (!m0.matches()) {
        Log.w(TAG, "could not parse upgrade script file: " + file0);
        throw new
```

```
SQLiteAssetHelper.SQLiteAssetException("Invalid upgrade script
file");
    }
```

```
    if (!m1.matches()) {
        Log.w(TAG, "could not parse upgrade script file: " + file1);
        throw new
```

```
SQLiteAssetHelper.SQLiteAssetException("Invalid upgrade script
file");
    }
```

```
    int v0_from = Integer.valueOf(m0.group(1));
    int v1_from = Integer.valueOf(m1.group(1));
    int v0_to = Integer.valueOf(m0.group(2));
    int v1_to = Integer.valueOf(m1.group(2));
```

```
    if (v0_from == v1_from) {
        // 'from' versions match for both; check 'to' version next
```

```
        if (v0_to == v1_to) {
            return 0;
        }
```

```
        return v0_to < v1_to ? -1 : 1;
```

```
    }
    return v0_from < v1_from ? -1 : 1;
```

```
}
```

XML Layout

activity_main

```
<android.support.v4.view.ViewPager
xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/pager"
android:layout_width="match_parent"
android:layout_height="match_parent">
```

```
</android.support.v4.view.ViewPager>
```

fragment_abstract

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<RelativeLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:background="#ff8400" >
```

```
<ImageView
```

```
android:src="@drawable/abstractwords"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:id="@+id/imageView4" />
```

```
<ImageButton
```




```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:src="@drawable/abstractwordsbtn"
        android:id="@+id/abstractwordsclick"
        android:clickable="false"
        android:onClick="onClickAbstractWords"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />

</RelativeLayout>

fragment_colorsandshapes

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#fa6a6a" >

    <ImageView
        android:src="@drawable/colorsandshapes"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentBottom="true"
        android:id="@+id/imageView"
        android:layout_alignParentTop="true" />

    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:src="@drawable/colorsandshapesbtn"
        android:id="@+id/colorsandshapesclick"
        android:clickable="true"
        android:onClick="onClickColorsandShapes"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />

</RelativeLayout>

fragment_cvcsightwords

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#17df0d">

    <ImageView
        android:src="@drawable/cvcsightwords"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/imageView2"
        android:layout_alignParentBottom="true"
        android:layout_alignParentTop="true" />

    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:src="@drawable/cvcsightwordsbtn"
        android:id="@+id/cvcsightwordsclick"
        android:clickable="true"
        android:onClick="onClickCvcsightWords"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />

</RelativeLayout>

fragment_lettersandnumbers

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#ff8400" >

    <ImageView
        android:src="@drawable/lettersnumbers"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/imageView3" />

    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:src="@drawable/lettersnumbersbtn"
        android:id="@+id/lettersnumbersclick"
        android:clickable="true"
        android:onClick="onClickLettersandNumbers"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />

</RelativeLayout>

fragment_readingcomprehension

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#fa6a6a" >

    <ImageView
        android:src="@drawable/readingcompbg"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/imageView4"/>

    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:src="@drawable/readingcompbtn"
        android:id="@+id/readingcompclick"
        android:clickable="false"
        android:onClick="onClickReadingComp"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />

</RelativeLayout>

main2

<?xml version="1.0" encoding="utf-8"?>
<com.example.markligot.embrasd.dragview.DragLayer
```



```
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:launcher="http://schemas.android.com/apk/res-auto"
android:id="@+id/drag_layer"
android:layout_width="match_parent"
android:layout_height="match_parent">
```

<RelativeLayout

```
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:background="@drawable/background">
```

<VideoView

```
android:layout_width="wrap_content"
android:layout_height="match_parent"
android:id="@+id/videoView"
android:background="@drawable/box_four"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_alignParentBottom="true"
android:layout_alignParentTop="true" />
```

<ImageView

```
android:src="@android:drawable/ic_media_previous"
android:layout_width="100dp"
android:layout_height="100dp"
android:onClick="onClickmainmenu"
android:clickable="true"
android:id="@+id/returnMenu"
android:layout_alignParentTop="true"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true" />
```

<ImageView

```
android:src="@drawable/box_one"
android:layout_width="100dp"
android:layout_height="100dp"
android:id="@+id/qimg"
android:layout_centerVertical="true"
android:layout_centerHorizontal="true"
android:background="@drawable/box_two" />
```

<ImageView

```
android:src="@android:drawable/ic_media_play"
android:layout_width="100dp"
android:layout_height="100dp"
android:id="@+id/playAgain"
android:onClick="onClickplayagain"
android:clickable="true"
android:layout_alignParentTop="true"
android:layout_alignParentRight="true"
android:layout_alignParentEnd="true" />
```

<TextView

```
android:text="TextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/tv1"
android:layout_above="@+id/drop_spot2"
android:layout_centerHorizontal="true" />
```

<com.example.markligot.embrasd.dragview.DropSpot

```
android:id="@+id/drop_spot2"
android:background="@color/drop_target_color2"
android:adjustViewBounds="true"
android:padding="8dp"
android:layout_width="match_parent"
android:layout_height="150dp"
android:layout_alignParentBottom="true"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true" />
```

</RelativeLayout>

</com.example.markligot.embrasd.dragview.DragLayer>

main2jb

<?xml version="1.0" encoding="utf-8"?>

```
<com.example.markligot.embrasd.dragview.DragLayer
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:launcher="http://schemas.android.com/apk/res-auto"
android:id="@+id/drag_layer"
android:layout_width="match_parent"
android:layout_height="match_parent">
```

<RelativeLayout

```
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:background="@drawable/background">
```

<ImageView

```
android:src="@android:drawable/ic_media_previous"
android:layout_width="100dp"
android:layout_height="100dp"
android:onClick="onClickmainmenu"
android:clickable="true"
android:id="@+id/returnMenu"
android:layout_alignParentTop="true"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true" />
```

<ImageView

```
android:src="@drawable/blankbox"
android:layout_width="100dp"
android:layout_height="100dp"
android:id="@+id/qimg"
android:layout_centerVertical="true"
android:layout_centerHorizontal="true"/>
```

<ImageView

```
android:src="@android:drawable/ic_media_play"
android:layout_width="100dp"
android:layout_height="100dp"
android:id="@+id/playAgain"
android:onClick="onClickplayagain"
android:clickable="true"
android:layout_alignParentTop="true"
android:layout_alignParentRight="true"
android:layout_alignParentEnd="true" />
```

<TextView

```
android:text="TextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/tv1"
android:layout_above="@+id/drop_spot2"
android:layout_centerHorizontal="true" />
```

<com.example.markligot.embrasd.dragview.DropSpot

```
android:id="@+id/drop_spot2"
android:background="@color/drop_target_color2"
android:adjustViewBounds="true"
android:padding="8dp"
android:layout_width="match_parent"
android:layout_height="150dp"
android:layout_alignParentBottom="true"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true" />
```

</RelativeLayout>



```
</com.example.markligot.embrasd.dragview.DragLayer>

main3

<?xml version="1.0" encoding="utf-8"?>
<com.example.markligot.embrasd.dragview.DragLayer
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:launcher="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drag_layer"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/background">

<VideoView
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:id="@+id/videoView"
    android:background="@drawable/box_four"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentBottom="true"
    android:layout_alignParentTop="true" />

<ImageView
    android:src="@android:drawable/ic_media_previous"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:onClick="onClickmainmenu"
    android:clickable="true"
    android:id="@+id/returnMenu"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<TextView
    android:text="TextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/drop_spot2"
    android:layout_centerHorizontal="true"
    android:id="@+id/tv1" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="24dp"
    android:id="@+id/tv2"
    android:layout_alignParentBottom="true"
    android:layout_alignLeft="@+id/tv1"
    android:layout_alignStart="@+id/tv1" />

<ImageView
    android:src="@android:drawable/ic_media_play"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:id="@+id/playAgain"
    android:onClick="onClickplayagain"
    android:clickable="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />

<ImageView
```

```
    android:src="@drawable/box_one"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:id="@+id/qimg"
    android:layout_alignTop="@+id/drop_spot2"
    android:layout_toRightOf="@+id/drop_spot2"
    android:layout_toEndOf="@+id/drop_spot2" />
```

```
<com.example.markligot.embrasd.dragview.DropSpot
    android:id="@+id/drop_spot2"
    android:background="@color/drop_target_color2"
    android:adjustViewBounds="true"
    android:padding="8dp"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_above="@+id/tv2"
    android:layout_centerHorizontal="true" />
```

```
</RelativeLayout>
```

```
</com.example.markligot.embrasd.dragview.DragLayer>
```

```
main3jb
```

```
<?xml version="1.0" encoding="utf-8"?>
<com.example.markligot.embrasd.dragview.DragLayer
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:launcher="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drag_layer"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/background">
```

```
<ImageView
    android:src="@android:drawable/ic_media_previous"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:onClick="onClickmainmenu"
    android:clickable="true"
    android:id="@+id/returnMenu"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />
```

```
<TextView
    android:text="TextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/drop_spot2"
    android:layout_centerHorizontal="true"
    android:id="@+id/tv1" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="24dp"
    android:id="@+id/tv2"
    android:layout_alignParentBottom="true"
    android:layout_alignLeft="@+id/tv1"
    android:layout_alignStart="@+id/tv1" />
```

```
<ImageView
    android:src="@android:drawable/ic_media_play"
```



```
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:id="@+id/playAgain"
        android:onClick="onClickplayagain"
        android:clickable="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />

<ImageView
    android:src="@drawable/blankbox"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:id="@+id/qimg"
    android:layout_alignTop="@+id/drop_spot2"
    android:layout_toRightOf="@+id/drop_spot2"
    android:layout_toEndOf="@+id/drop_spot2"/>

<com.example.markligot.embrasd.dragview.DropSpot
    android:id="@+id/drop_spot2"
    android:background="@color/drop_target_color2"
    android:adjustViewBounds="true"
    android:padding="8dp"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_above="@+id/tv2"
    android:layout_centerHorizontal="true" />

</RelativeLayout>

</com.example.markligot.embrasd.dragview.DragLayer>

main4

<?xml version="1.0" encoding="utf-8"?>
<com.example.markligot.embrasd.dragview.DragLayer
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:launcher="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drag_layer"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/background">

<VideoView
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:id="@+id/videoView"
    android:background="@drawable/box_four"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentBottom="true"
    android:layout_alignParentTop="true" />

<ImageView
    android:src="@android:drawable/ic_media_previous"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:onClick="onClickmainmenu"
    android:clickable="true"
    android:id="@+id/returnMenu"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<TextView
    android:text="TextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_above="@+id/drop_spot2"
    android:layout_centerHorizontal="true"
    android:gravity="center"
    android:id="@+id/tv1" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="24dp"
    android:id="@+id/tv2"
    android:layout_alignParentBottom="true"
    android:gravity="center"
    android:layout_alignLeft="@+id/tv1"
    android:layout_alignStart="@+id/tv1" />

<ImageView
    android:src="@android:drawable/ic_media_play"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:id="@+id/playAgain"
    android:onClick="onClickplayagain"
    android:clickable="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />

<com.example.markligot.embrasd.dragview.DropSpot
    android:id="@+id/drop_spot2"
    android:background="@color/drop_target_color2"
    android:adjustViewBounds="true"
    android:padding="8dp"
    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:layout_above="@+id/tv2"
    android:layout_centerHorizontal="true" />

<ImageView
    android:src="@drawable/box_one"
    android:layout_width="300dp"
    android:layout_height="350dp"
    android:id="@+id/qimg"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true" />

</RelativeLayout>

</com.example.markligot.embrasd.dragview.DragLayer>

main4jb

<?xml version="1.0" encoding="utf-8"?>
<com.example.markligot.embrasd.dragview.DragLayer
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:launcher="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drag_layer"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/background">
```



```
<ImageView
    android:src="@android:drawable/ic_media_previous"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:onClick="onClickmainmenu"
    android:clickable="true"
    android:id="@+id/returnMenu"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />
```

```
<TextView
    android:text="TextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_above="@+id/drop_spot2"
    android:layout_centerHorizontal="true"
    android:gravity="center"
    android:id="@+id/tv1" />
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="24dp"
    android:id="@+id/tv2"
    android:gravity="center"
    android:layout_alignParentBottom="true"
    android:layout_alignLeft="@+id/tv1"
    android:layout_alignStart="@+id/tv1" />
```

```
<ImageView
    android:src="@android:drawable/ic_media_play"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:id="@+id/playAgain"
    android:onClick="onClickplayagain"
    android:clickable="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />
```

```
<com.example.markligot.embrasd.dragview.DropSpot
    android:id="@+id/drop_spot2"
    android:background="@color/drop_target_color2"
    android:adjustViewBounds="true"
    android:padding="8dp"
    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:layout_above="@+id/tv2"
    android:layout_centerHorizontal="true" />
```

```
<ImageView
    android:src="@drawable/box_one"
    android:layout_width="300dp"
    android:layout_height="350dp"
    android:id="@+id/qimg"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true" />
```

</RelativeLayout>

<com.example.markligot.embrasd.dragview.DragLayer>

main5

<?xml version="1.0" encoding="utf-8"?>

<com.example.markligot.embrasd.dragview.DragLayer

```
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:launcher="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drag_layer"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/background">
```

```
<VideoView
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:id="@+id/videoView"
    android:background="@drawable/box_four"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentBottom="true"
    android:layout_alignParentTop="true" />
```

```
<ImageView
    android:src="@android:drawable/ic_media_previous"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:onClick="onClickmainmenu"
    android:clickable="true"
    android:id="@+id/returnMenu"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />
```

```
<ImageView
    android:src="@android:drawable/ic_media_play"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:id="@+id/playAgain"
    android:onClick="onClickplayagain"
    android:clickable="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/tv2"
    android:gravity="center"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />
```

```
<com.example.markligot.embrasd.dragview.DropSpot
    android:id="@+id/drop_spot2"
    android:background="@color/drop_target_color2"
    android:adjustViewBounds="true"
    android:padding="8dp"
    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:layout_above="@+id/tv2"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />
```

```
<ImageView
    android:src="@drawable/box_one"
    android:layout_width="300dp"
    android:layout_height="350dp"
    android:id="@+id/qimg"
```



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

117

```
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true" />

<TextView
    android:text="TextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:id="@+id/tv1"
    android:layout_below="@+id/qimg"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

</RelativeLayout>

</com.example.markligot.embrasd.dragview.DragLayer>

main5jb

<?xml version="1.0" encoding="utf-8"?>
<com.example.markligot.embrasd.dragview.DragLayer
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:launcher="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drag_layer"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/background">

<ImageView
    android:src="@android:drawable/ic_media_previous"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:onClick="onClickmainmenu"
    android:clickable="true"
    android:id="@+id/returnMenu"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<ImageView
    android:src="@android:drawable/ic_media_play"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:id="@+id/playAgain"
    android:onClick="onClickplayagain"
    android:clickable="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/tv2"
    android:gravity="center"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<ImageView
    android:src="@drawable/box_one"
    android:layout_width="300dp"
    android:layout_height="350dp"

        android:id="@+id/qimg"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true" />

<com.example.markligot.embrasd.dragview.DropSpot
    android:id="@+id/drop_spot2"
    android:background="@color/drop_target_color2"
    android:adjustViewBounds="true"
    android:padding="8dp"
    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:layout_above="@+id/tv2"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<TextView
    android:text="TextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:id="@+id/tv1"
    android:layout_below="@+id/qimg"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

</RelativeLayout>

</com.example.markligot.embrasd.dragview.DragLayer>

maintest

<RelativeLayout
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/surface_camera"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_gravity="center"
    android:layout_height="fill_parent">

<VideoView
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:id="@+id/videoView"
    android:background="@drawable/box_four"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentBottom="true"
    android:layout_alignParentTop="true" />

<ImageView
    android:src="@android:drawable/ic_media_play"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:id="@+id/playAgain"
    android:onClick="onClickplayagain2"
    android:clickable="true"
    android:layout_above="@+id/drop_spot2"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_marginBottom="187dp" />

<ImageView
    android:src="@android:drawable/ic_media_previous"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:onClick="onClickmainmenu2"
    android:clickable="true"
    android:id="@+id/returnMenu"
```




```
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

<ImageView
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true"
    android:id="@+id/corrAnsIMG" />

<ImageView
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:id="@+id/qimgFP"
    android:layout_alignTop="@+id/corrAnsIMG"
    android:layout_toRightOf="@+id/corrAnsIMG"
    android:layout_toEndOf="@+id/corrAnsIMG" />

</RelativeLayout>

maintest2

<RelativeLayout
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/surface_camera"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_gravity="center"
    android:layout_height="match_parent">

    <VideoView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:id="@+id/videoView"
        android:background="@drawable/box_four"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentBottom="true"
        android:layout_alignParentTop="true" />

    <ImageView
        android:src="@android:drawable/ic_media_play"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:id="@+id/playAgain"
        android:onClick="onClickplayagain2"
        android:clickable="true"
        android:layout_above="@+id/drop_spot2"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_marginBottom="187dp" />

    <ImageView
        android:src="@android:drawable/ic_media_previous"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:onClick="onClickmainmenu2"
        android:clickable="true"
        android:id="@+id/returnMenu"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <ImageView
        android:layout_width="200dp"
        android:layout_height="250dp"
        android:id="@+id/sellImgDesc"
        android:layout_alignParentTop="true"

        android:layout_centerHorizontal="true" />

<TextView
    android:text="TextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/corrAnsIMG"
    android:layout_alignRight="@+id/tvString"
    android:layout_alignEnd="@+id/tvString"
    android:layout_centerHorizontal="true"
    android:gravity="center"

    android:id="@+id/tv2String" />

<ImageView
    android:layout_width="200dp"
    android:layout_height="200dp"
    android:id="@+id/corrAnsIMG"
    android:layout_below="@+id/playAgain"
    android:gravity="center"
    android:layout_alignLeft="@+id/sellImgDesc"
    android:layout_alignStart="@+id/sellImgDesc" />

<TextView
    android:text="TextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/tvString"
    android:layout_below="@+id/sellImgDesc"
    android:layout_centerHorizontal="true"
    android:gravity="center"
    android:layout_marginTop="20dp" />

</RelativeLayout>

maintest2jb

<RelativeLayout
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/surface_camera"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_gravity="center"
    android:layout_height="fill_parent">

    <ImageView
        android:src="@android:drawable/ic_media_play"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:id="@+id/playAgain"
        android:onClick="onClickplayagain2"
        android:clickable="true"
        android:layout_above="@+id/drop_spot2"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_marginBottom="187dp" />

    <ImageView
        android:src="@android:drawable/ic_media_previous"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:onClick="onClickmainmenu2"
        android:clickable="true"
        android:id="@+id/returnMenu"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />
```



Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

119

```
<ImageView
    android:layout_width="200dp"
    android:layout_height="250dp"
    android:id="@+id/sellmgDesc"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true" />

<TextView
    android:text="TextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/corrAnsIMG"
    android:layout_alignRight="@+id/tvString"
    android:layout_alignEnd="@+id/tvString"
    android:layout_centerHorizontal="true"
    android:gravity="center"
    android:id="@+id/tv2String" />

<ImageView
    android:layout_width="200dp"
    android:layout_height="200dp"
    android:id="@+id/corrAnsIMG"
    android:layout_below="@+id/playAgain"
    android:layout_alignLeft="@+id/sellmgDesc"
    android:layout_alignStart="@+id/sellmgDesc" />

<TextView
    android:text="TextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/tvString"
    android:layout_below="@+id/sellmgDesc"
    android:layout_centerHorizontal="true"
    android:gravity="center"
    android:layout_marginTop="20dp" />
</RelativeLayout>

maintestjb

<RelativeLayout
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/surface_camera"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_gravity="center"
    android:layout_height="fill_parent">

    <ImageView
        android:src="@android:drawable/ic_media_play"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:id="@+id/playAgain"
        android:onClick="onClickplayagain2"
        android:clickable="true"
        android:layout_above="@+id/drop_spot2"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_marginBottom="187dp" />

    <ImageView
        android:src="@android:drawable/ic_media_previous"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:onClick="onClickmainmenu2"
        android:clickable="true"
        android:id="@+id/returnMenu"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <ImageView
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:id="@+id/qimgFP"
        android:layout_alignTop="@+id/corrAnsIMG"
        android:layout_toRightOf="@+id/corrAnsIMG"
        android:layout_toEndOf="@+id/corrAnsIMG"
        android:src="@drawable/blankbox" />

</RelativeLayout>

testing_splash

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:src="@drawable/ic_launcher"
        android:id="@+id/imageView5" />

</LinearLayout>

styles
<resources>

    <!--
        Base application theme, dependent on API level. This theme is
        replaced
        by AppBaseTheme from res/values-vXX/styles.xml on newer
        devices.
    -->
    <style name="AppBaseTheme"
        parent="android:Theme.Light.NoTitleBar">
        <!--
            Theme customizations available in newer API levels can go in
            res/values-vXX/styles.xml, while customizations related to
            backward-compatibility can go here.
        -->
    </style>

    <!-- Application theme. -->
    <style name="AppTheme" parent="AppBaseTheme">
        <!-- All customizations that are NOT specific to a particular
        API-level can go here. -->
    </style>

    <style name="AppTheme.Fullscreen">
        <item name="windowActionBar">false</item>
        <item name="windowNoTitle">true</item>
        <item name="android:windowFullscreen">true</item>
        <item name="android:windowNoTitle">true</item>
        <item name="android:windowActionBar">false</item>
    </style>
</resources>
```




Southville International School and Colleges

Luxembourg St., BF Homes International, Las Piñas City 1740

Bachelor of Science in Information Technology

120

Gradle Script

```
apply plugin: 'com.android.application'
```

```
android {  
    compileSdkVersion 25  
    buildToolsVersion "25.0.2"  
    defaultConfig {  
        applicationId "com.example.markligot.embrasd"  
        minSdkVersion 16  
        targetSdkVersion 25  
        versionCode 1  
        versionName "1.0"  
        testInstrumentationRunner  
        "android.support.test.runner.AndroidJUnitRunner"  
    }  
    buildTypes {  
        release {  
            minifyEnabled false
```

```
        proguardFiles getDefaultProguardFile('proguard-  
android.txt'), 'proguard-rules.pro'  
    }  
}  
  
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
  
    androidTestCompile('com.android.support.test.espresso:  
core:2.2.2', {  
        exclude group: 'com.android.support', module: 'support-  
annotations'  
    })  
    compile 'com.android.support:appcompat-v7:25.1.1'  
    compile 'com.android.support:design:25.1.1'  
    testCompile 'junit:junit:4.12'  
}
```