

Mobile Programming

GreenyDroid

MI141

Heramia, Johanna Marisse C.  
Tadeo, Jose Lorenzo G.

Professor: Mr. Noel Anonas

**Table of Contents**

Description………………………………………………………………...3

Target Audience……………………………………………………….….3

Project Objective………………………………………………………….3

Scope and Limitation………………………………………....................3

Review of Related Literature…………………………………………….4

Game Mechanics……………………………………………………...….6

Game Instruction………………………………………………………….6

Technologies used…………………………………………………….….6

Game Map / Level………………………………………………………...7

Use Case Diagram……………………………………………………….22

Activity Diagram………………………………………………………….23

Java Code Snippets……………………………………………………..25

Graphics Assets and Art Work Example………………………………30

Lessons Learned………………………………………………...………21

Resume of each member……………………………………………….32

Revision Table…………………………………………………………...34

Requirement Checklist……………………………………………....….35

**GREENYDROID**

**Theme:** *Health /* Game

*GreenyDroid* is a game application that gives the user fictional stories and questions, some simple details, some facts or trivia, and a mini game base on sweet version of Android starting from Cupcake up to KitKat. GreenyDroid has different levels from easiest level (Cupcake to Eclair) to hardest level (Ice Cream Sandwich to KitKat).

In order to go to the next level, the user should complete the challenges, but if the user cannot do the challenges, they will repeat the challenge from the beginning of their current level. Challenges in every level are composed of questions (multiple choice) and mini games.

The target audience are people from 8-15 years old for they will be entertained and they will be able to understand the game easily and playfully. Seven years old below and 18 and above may also use it for their entertainment.

The main objective of this project is to be able to know more about the simple details base on every sweet version of android, to be able to give information to the user, to be more knowledgeable in certain food, and to give user the *entertainment*.

Scope

* Welcome page with animated image
* Different game level (Easy, Medium and Hard)
* In the game setting, background music can be turn on and off
* Animated game play in mini games
* Cut scene before and after the game play
* Game timer
* Display high score and current score in mini game
* Displaying the total score with name of the user
* It has a “Game Over” feature
* Sub menus such as Play, Settings and About

Limitation

* Health tips, stories and questions are just simple and few.
* Versions from Lollipop and Nougat are not included

**Review of Related Literature**

*Category Quiz (Trivia)*

Category Quiz is a quiz where the questions must be answered by category such as sports, entertainment, food, geography, music, news etc. If all questions in a category are answered correctly, user will get and extra points, but if question is answered incorrectly, the category gets locked. If the question timer is used (can be enabled/disabled using the game options) the user has 30 seconds to answer the question. Like on our project, it also has question and answer portion that needs to be answered correctly in every level of the game but the user cannot disable nor enable the timer of the game.



*Nutrition Tips*

This application, like on our application, gives useful nutrition tips and nutritional health facts that can help to and improve a person’s health.

*Pou*

Pou is a pet that you can take care of, like user should give food, give shower, give good outfit, and give some fun. It has mini games that user should play to earn money to buy items and to earn level to unlock items.



**Game Mechanics**

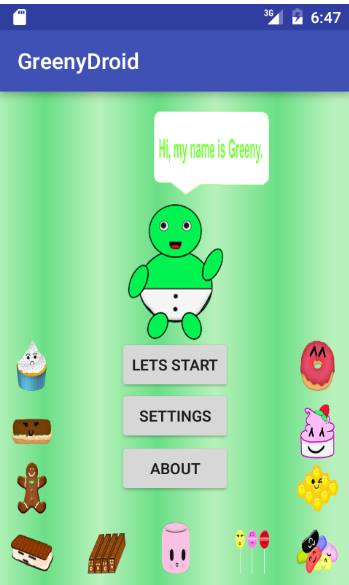
Before the game will start, an instruction will be given to be able to know the objective of the challenge on each level. After the story, certain questions will be asked and questions should be answered quickly. When all the questions are answered correctly, user can now be able to go the next activity, which is the mini game, otherwise, the user need to answer the previous questions again. Before the mini game, an instruction will be given and the user should complete the requirements. Everything must be completed to be able to go to the next level.

**Game Instruction**

In question and answer, click the corresponding button for the correct answer. User should get all the correct answers to be able to go to the next activity, otherwise, user should answer the previous questions again. Since the user cannot go back to the story, he/she should understand it very carefully. In mini games, an instruction will be given that needs to be completed to be able to go to the next level.

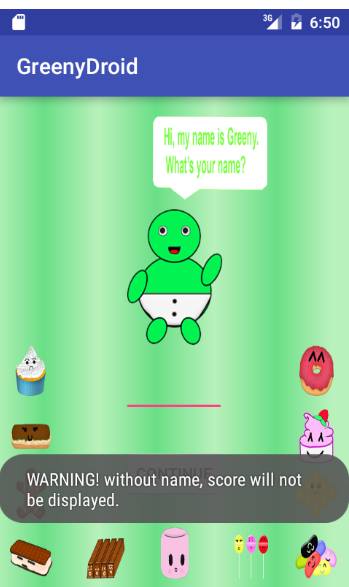
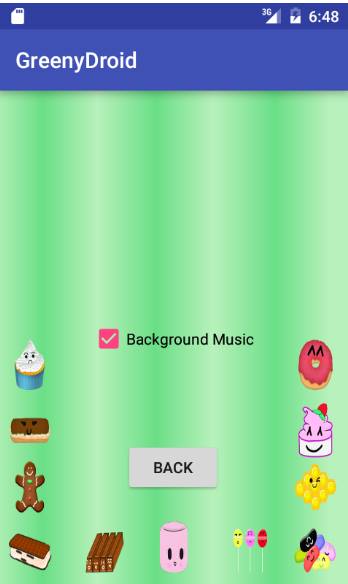
**Technical Requirements**

The technologies or software that we used to develop this application are: (1) Android Studio. Android Studio provides the fastest tools for building apps on every type of Android device. We use Android Studio for code editing, debugging, performance tooling, a flexible build system, and an instant build/deploy system. (2) Also, we use Photoshop for creating image and Word Document for creating background.

**Game Map**

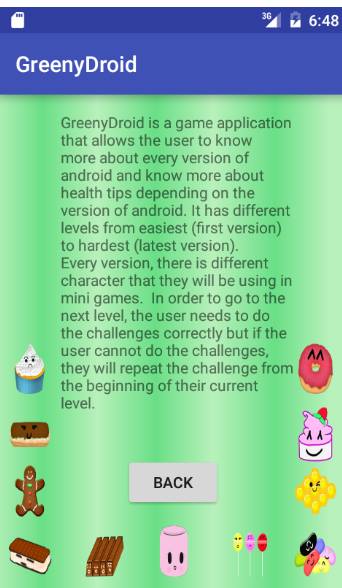
MAIN PAGE  
 This is the main page of the application. There are three buttons: LET’S START, SETTING, and ABOUT.

WELCOME SCREEN  
 This is the welcome screen of the application. While loading, the GreenyDroid logo will turn around.



SETTING SCREEN  
 When the user clicks the ‘Setting’ button, it will display a screen where the user may turn on or off the background music

GET NAME SCREEN  
 When the user clicks the ‘Let’s start’ button, it will display a screen where the user should put his/her name.

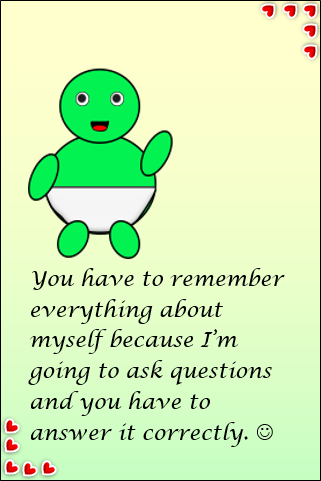


BEGIN

INTRODUCTION SCREEN  
 After the user enters his/her name, there are introduction / instruction before the story.

ABOUT SCREEN  
 When the user clicks the ‘Let’s start’ button, it will display a screen where the user should put his/her name.





STORY  
 When the user clicks the ‘Begin’ button, it will display the story.

BEGIN

INTRO  
 Introduction before the questions



QUESTIONS  
 This is the questions that are related to the story that the user should answer correctly



RESULT PAGE  
 When the user gets a score below three, this result page will display

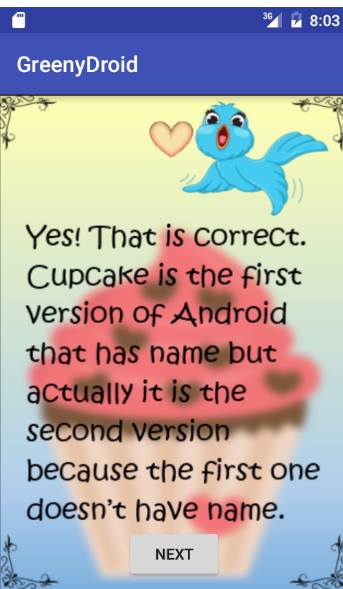


CONTINUE

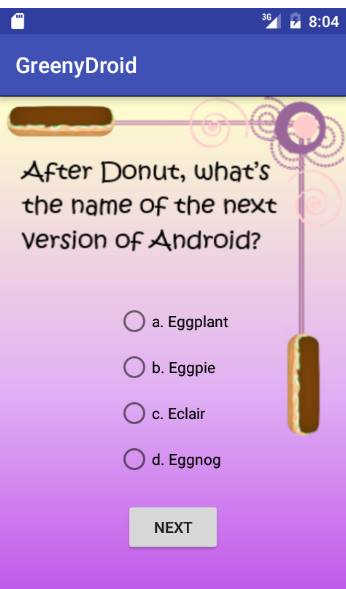
START OF 1st LEVEL  
 Start of the 1st level

INTRODUCTION ON 1st LEVEL  
 This is the introduction for the 1st level of the game.

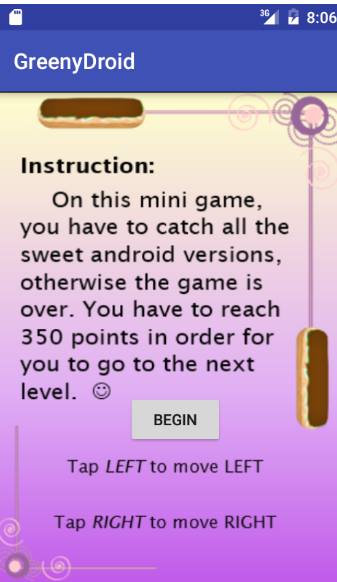
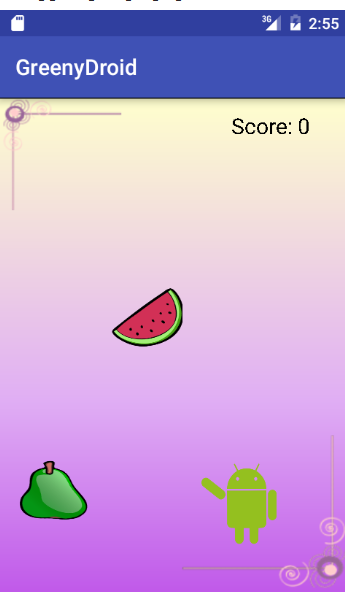
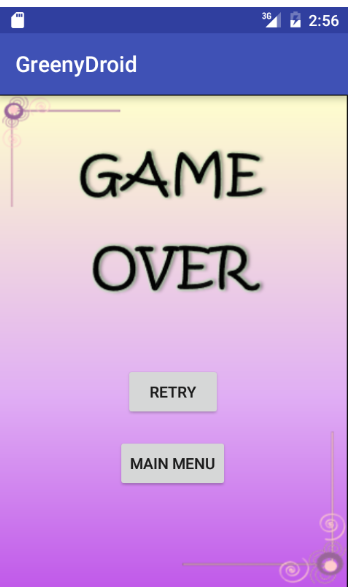
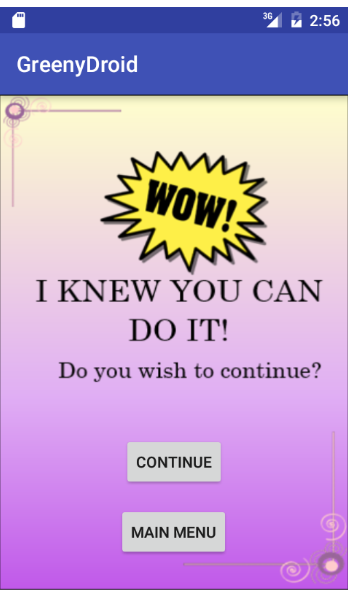
 







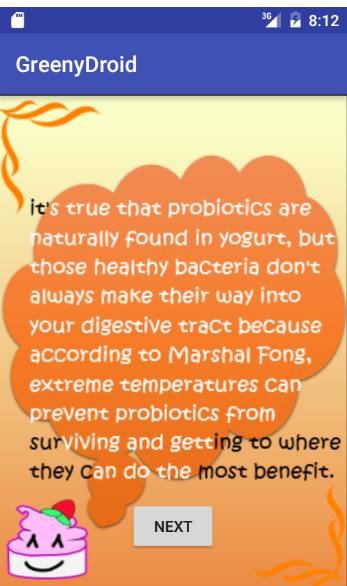


****

MINI GAME  
 Mini game for the 1st level.

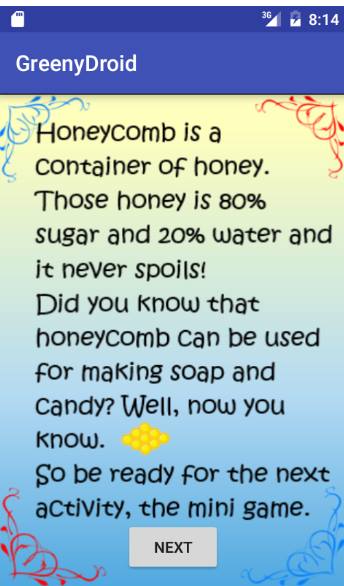
CONTINUE ON 2nd LEVEL  
 When the user was able to the requirement in mini game, he/she can now continue to the next level.

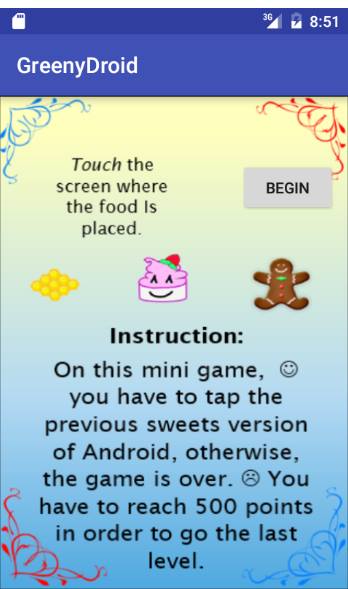
GAMOVER ON 1st LEVEL  
 When the user was not able to the requirement in mini game, the game is over.

****

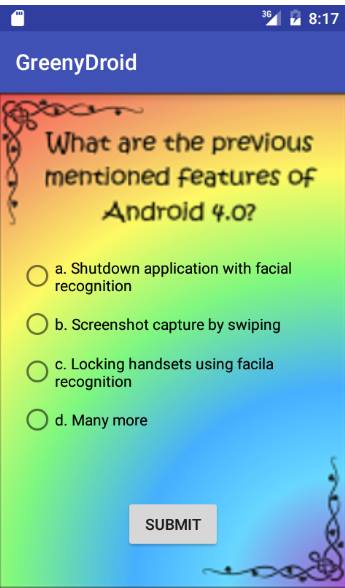
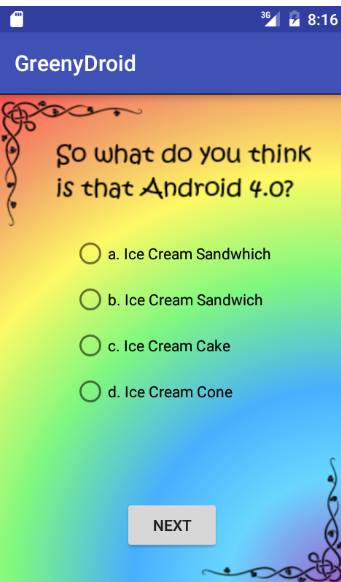
START OF 2nd LEVEL  
 Start of the 2nd level.

INTRODUCTION ON 2nd LEVEL  
 This is the introduction for the 2nd level of the game.

****

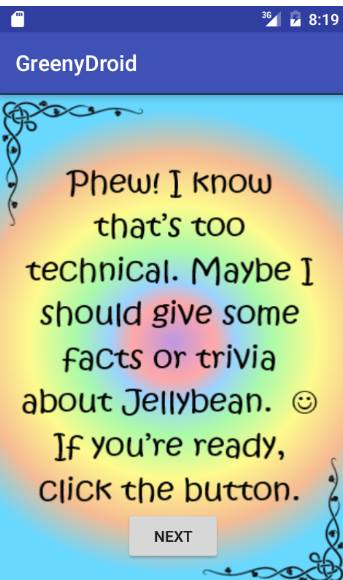
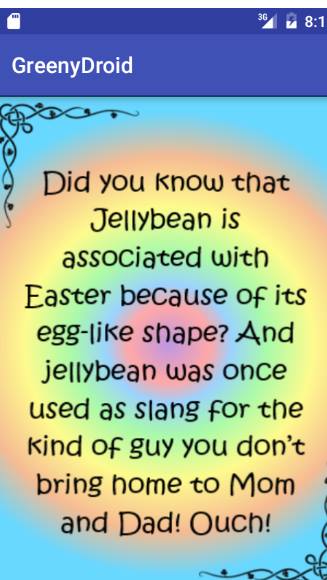
****

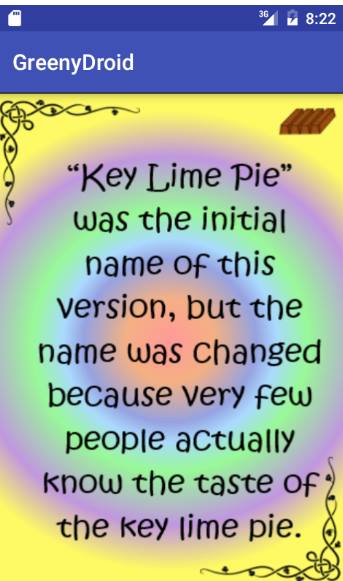
MINI GAME  
 Mini game for the 2nd level.

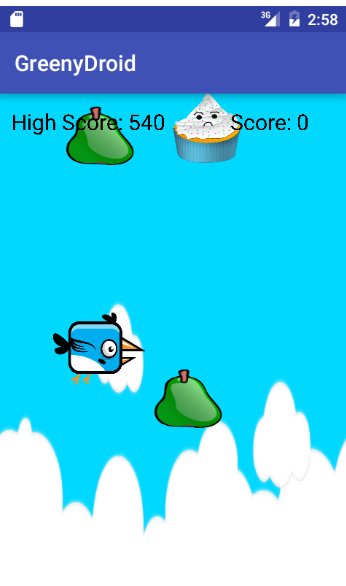
****

START OF 3rd LEVEL  
 Start of the 3rd level

INTRODUCTION ON 3rd LEVEL  
 This is the introduction for the 3srdlevel of the game.

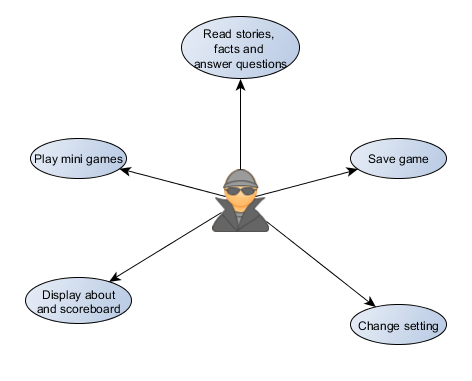
****

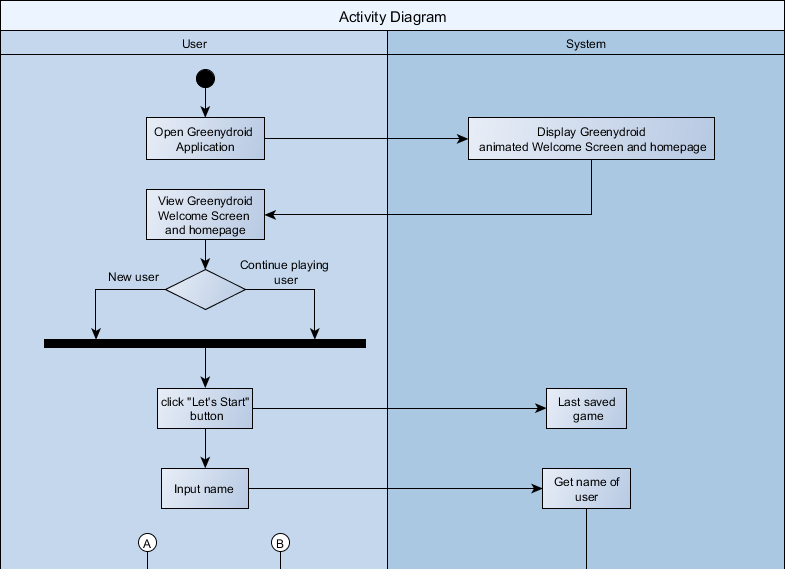
****

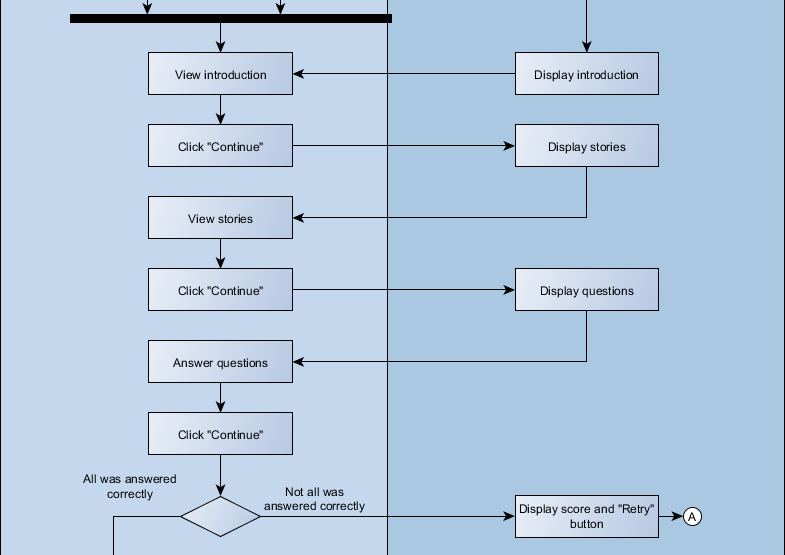
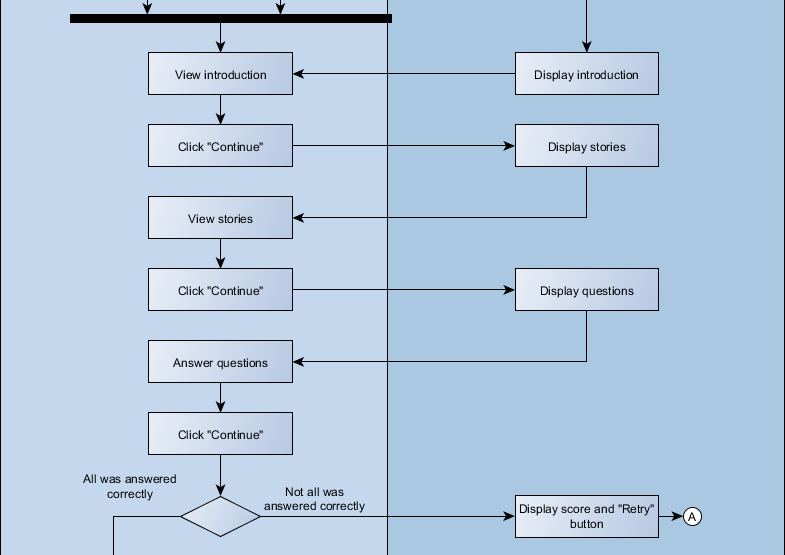
****

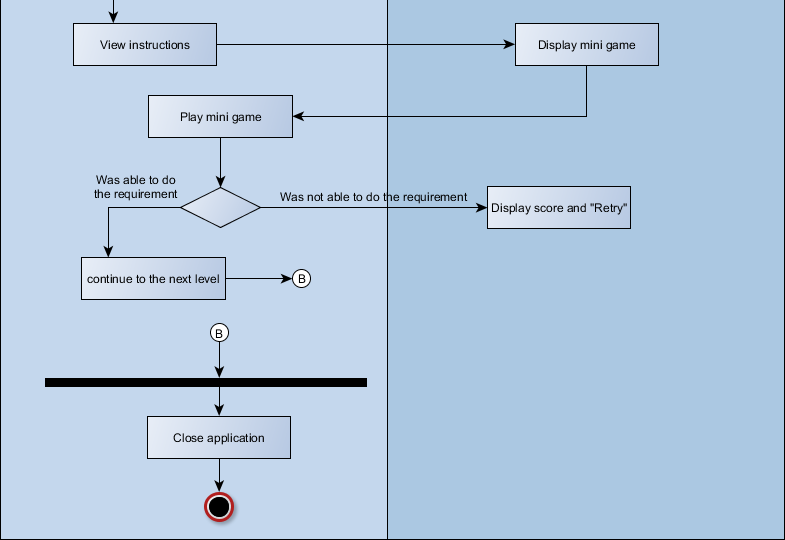
MINI GAME  
 Mini game for the 3rd level.

USE CASE DIAGRAM



**ACTIVITY DIAGRAM**





**JAVA CODES SNIPPET**

1. When button is clicked, new activity will appear.

private void settingButton() {  
 Button setting = (Button) findViewById(R.id.btnSetting);  
 setting.setOnClickListener(new View.OnClickListener() {  
 public void onClick(View v) {  
 Intent i = new Intent(MainActivity.this, SettingActivity.class);  
 startActivity(i);  
 finish();  
 }  
 });  
}

1. When user wants to go back (Back button)

public void backButton() {  
 Button back = (Button) findViewById(R.id.btnBack);  
 back.setOnClickListener(new View.OnClickListener() {  
 public void onClick(View v) {  
 bgdCheck = 1;  
 Intent i = new Intent(SettingActivity.this, MainActivity.class);  
 startActivity(i);  
 finish();  
 =  
 }  
 });  
}

1. Turning on and off of music background.

public static int bgdCheck = 0;  
static CheckBox music;  
  
@Override  
protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_setting);  
 backButton();  
  
 music = (CheckBox) findViewById(R.id.checkBox);  
 if (MainActivity.bgdMusic.isPlaying()) {  
 SettingActivity.music.setChecked(true);  
 } else {  
 SettingActivity.music.setChecked(false);  
 }  
  
 music.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
 public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
 if (isChecked) {  
 SharedPreferences.Editor editor = getSharedPreferences("GreenyDroid.txt", MODE\_PRIVATE).edit();  
 editor.putBoolean("Resume", true);  
 editor.commit();  
 music.setChecked(true);  
 MainActivity.bgdMusic.start();  
 } else {  
 SharedPreferences.Editor editor = getSharedPreferences("GreenyDroid.txt", MODE\_PRIVATE).edit();  
 editor.putBoolean("Pause", false);  
 editor.commit();  
 music.setChecked(false);  
 MainActivity.bgdMusic.pause();  
 }  
 }  
 });

1. Getting answer from the user using the radio button.

static RadioButton A, B, C, D;  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_acupcakequestion);  
  
 A = (RadioButton)findViewById(R.id.radioButtonA);  
 A.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
 @Override  
 public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
 if(isChecked){  
 nextButton();  
 }else{  
 nextButton();  
 }  
 }  
 });  
 B = (RadioButton)findViewById(R.id.radioButtonB);  
 B.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
 @Override  
 public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
 if(isChecked){  
 nextButton();  
 }else{  
 nextButton();  
 }  
 }  
 });  
 C = (RadioButton)findViewById(R.id.radioButtonC);  
 C.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
 @Override  
 public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
 if(isChecked){  
 SharedPreferences sp = getSharedPreferences("QuestionCupcake",MODE\_PRIVATE);  
 SharedPreferences.Editor spsave = sp.edit();  
 spsave.commit();  
 nextButton();  
 }else{  
 nextButton();  
 }  
 }  
 });  
}

1. Putting delay before the fruits would drop in mini game

class Timer implements Runnable{

@Override

public void run() {

while(count != 5){

SystemClock.sleep(1000);

count++;

}

}

}

1. Displaying high score

SharedPreferences sp = getSharedPreferences("highscore.txt", MODE\_PRIVATE);  
lastHighScore = sp.getInt("highscorebeat", highScore);  
highScore = lastHighScore;

7. Adding score

int c = 0;

String name = StartActivity.name;

if(CCupcakeQuestionActivity.D.isChecked()){

c+=1;

}

if(BCupCakeQuestionActivity.C.isChecked()){

c+=1;

}

if(ACupcakeQuestionActivity.C.isChecked()){

c+=1;

}

8. If the user’s score is equals to three, he/she may proceed to the next activity, else if less than three, user should try again.

Static int c = 0;

If(AIceCreamQuestion.B.isChecked()){

C+=1;

}

If(BIceCreamQuestion.D.isChecked()){

C+=1;

}

Switch(c){

Case 2:

Intent I = new Intent(IceCreamCalculating.this, AJellyStory.class);

startActivity(i);

finish();

break;

default:

Intent l = new Intent(IceCreamCalculating.this, IceCreamResultR.class);

startActivity(l);

finish();

break;

}

9. Making the bird move in different directions.

int my, mx;

public boolean onTouchEvent(MotionEvent event){

my = (int) event.getY();

mx = (int) event.getX();

if(my <= 400) {

if (birdTop > 0) {

birdTop -= 50;

birdBottom -= 50;

}else{

birdTop -= 0;

birdBottom -= 0;

}

}else{

if(birdBottom < 650){

birdTop += 50;

birdBottom += 50;

}else{

birdTop += 0;

birdBottom += 0;

}

}

if(mx <= 240){

if(birdLeft > 0){

birdLeft -= 50;

birdRight -= 50;

}else{

birdLeft -= 0;

birdRight -= 0;

}

}else{

if(birdRight < 470){

birdLeft += 50;

birdRight += 50;

}else{

birdLeft += 0;

birdRight += 0;

}

}

birdDst.top = birdTop;

birdDst.bottom = birdBottom;

birdDst.left = birdLeft;

birdDst.right = birdRight;

return super.onTouchEvent(event);

}

10. Catching the sweets and fruits by the character.

if(birdLeft >= pearsLeft2 && birdLeft <= pearsRight2 || birdRight >= pearsLeft2 && birdRight <= pearsRight2){

if(birdTop >= pearsTop2 && birdTop <= pearsBottom2 || birdBottom >= pearsTop2 && birdBottom <= pearsBottom2){

Intent i = new Intent(LastMiniGame.this, LastMiniGameCongrats.class);

startActivity(i);

finish();

break;

}

}

if(birdLeft >= cupcakeLeft && birdLeft <= cupcakeRight || birdRight >= cupcakeLeft && birdRight <= cupcakeRight){

if(birdTop >= cupcakeTop && birdTop <= cupcakeBottom || birdBottom >= cupcakeTop && birdBottom <= cupcakeBottom){

c = new Random();

cupcakeLeft = c.nextInt(350);

cupcakeRight = cupcakeLeft + 100;

cupcakeTop = -105;

cupcakeBottom = -5;

score += 20;

if(score >= highScore){

highScore = score;

SharedPreferences sp = getSharedPreferences("highscore.txt", MODE\_PRIVATE);

SharedPreferences.Editor spsave = sp.edit();

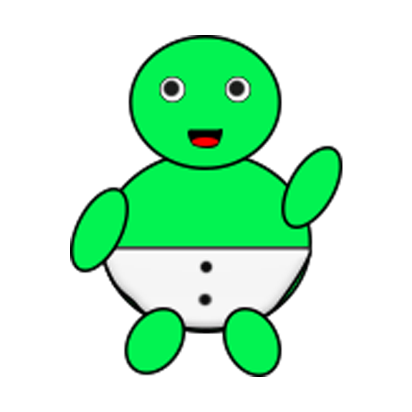
spsave.putInt("highscorebeat",highScore);

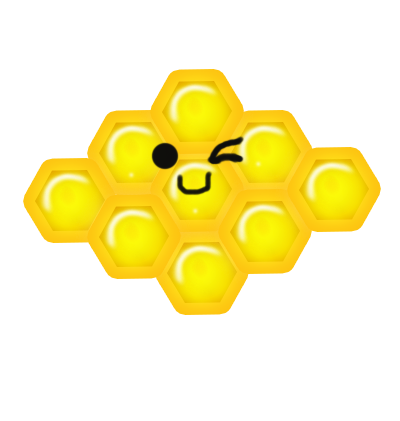
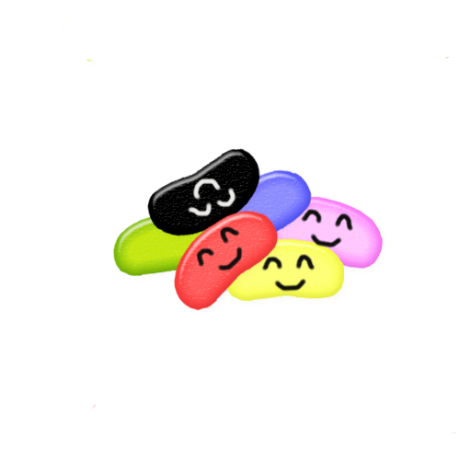
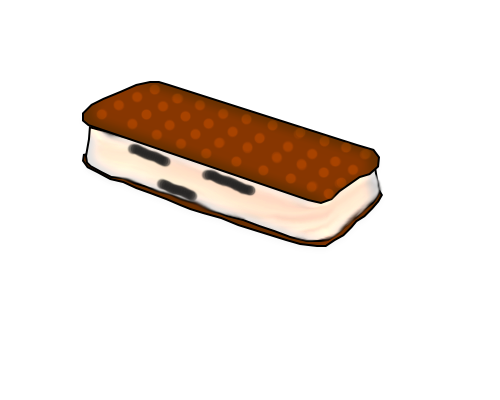
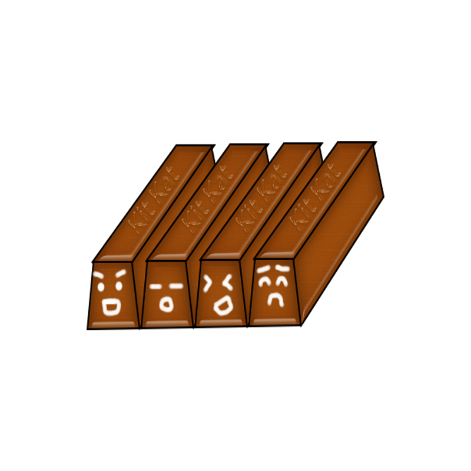
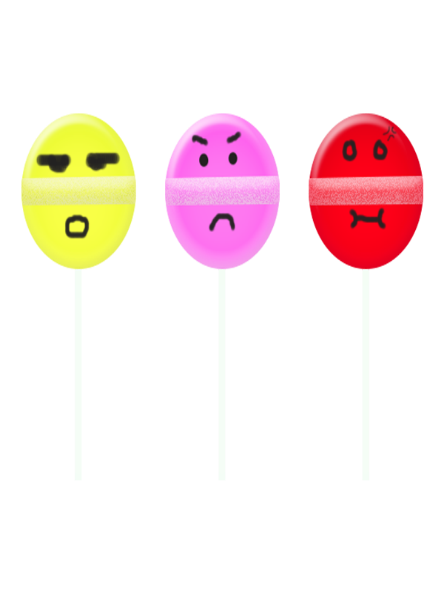
spsave.commit();

}

}

}

**Graphics Asset and Artwork Sample**



Ice Cream, Jellybean, KitKat, Lollipop and Marshmallow can be seen on the last level of the game.

Froyo, Gingerbread and Honeycomb can be seen on the second level of the game and on the last mini game.

Éclair, Donut and Cupcake can be seen on the first part of the game and to the last mini game.

This is GreenyDroid. You can see him on the main page of the app.

**Lesson Learned**

Having a subject like Mobile Programing requires time to learn and willingness to learn for us to be able to pass. Analyzation, memorization, and time management is one of the most important thing that every individual should learn and should have. From the start, it was not easy, but as the time goes by, you will “ENJOY”. You will enjoy learning xml codes and java codes, and even if creating image is not part of this subject, but we do not have choice but to use Photoshop (like no basic knowledge) to build characters, images and backgrounds that are needed in our application. Even if it is hard and always we get errors, when we get it, it is the best feeling in the world. No matter how hard it is, in the end, it is worth the effort.

**JOHANNA MARISSE CREDITO HERAMIA**

P 53 – 20 12th 15th St., Villamor Air Base, Pasay City

jcheramia@student.apc.edu.ph

+63926 152 3128

**EDUCATION**

June 2014 – Present Asia Pacific College

#3 Humabon Place, Magallanes, Makati City

Bachelor of Science in Information Technology

Major in Mobile and Internet Technology

June 2010 – March 2014 Pasay City South High School

Piccio Garden, Villamor Air Base Pasay City

June 2004 – March 2010 Villamor Air Base Elementary School

Piccio Garden, Villamor Air Base Pasay City

**ORGANIZATIONS**

2015-2016 Junior Philippine Computer Society

Member

2016 – Present Junior Informations System Security Association

Documentations Head

**RESEARCH PROJECT**

THE NEGATIVE EFFECTS OF COURSES CHOSEN BY PARENTS TO THE ACADEMIC PERFORMANCE OF STUDENTS IN ASIA PACIFIC COLLEGE, ACADEMIC YEAR 2015-2016

BALCENA, D.P., GARDON, J.G., HERAMIA, J.C.

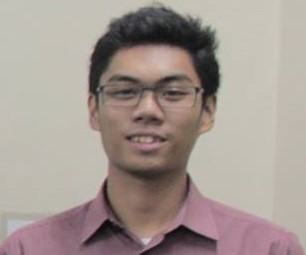
**PERSONAL DATA**

December 2, 1997

Pasay City

Tagalog

Single

**JOSE LORENZO GONZALES TADEO**

8291 B. Dapitan St. Guadalupe Nuevo, Makati City

jgtadeo@student.apc.edu.ph

09354811320

**EDUCATION**

June 2014 – Present Asia Pacific College

#3 Humabon Place, Magallanes, Makati City

Bachelor of Science in Information Technology

Major in Mobile and Internet Technology

June 2010 – March 2014 Guadalupe Catholic School

1923 Orense St, Guadalupe Nuevo, Makati City

June 2006 – March 2010 Guadalupe Catholic School

1923 Orense St, Guadalupe Nuevo, Makati City

June 2004 – March 2006 Paco Catholic School

1521 Paz St, Paco, Manila

**ORGANIZATIONS**

2014 – 2015 Junior Philippine Computer Society

Member

2016 – Present Microsoft Community

Assistant Documentations Head

**RESEARCH PROJECT**

SOPHOMORE THE BENEFITS OF PHYSICAL EDUCATION TO

STUDENTS WITH HEALTH PROBLEMS IN ASIA PACIFIC COLLEGE, ACADEMIC YEAR 2015 – 2016

TADEO, J, G., SAMSON, J, O., VICTORINO, M, I.

**PERSONAL DATA**

March 26, 1998

Makati City

Tagalog

Single

**Revision Table**

|  |  |  |
| --- | --- | --- |
| **DATE** | **TASK** |  |
| **07/26/16** | Started creating images and background | Johanna Marisse Jose Lorenso |
| **08/02/16** | Started creating the application using Android Studio | Johanna Marisse Jose Lorenzo |
| **08/02/16** | Put all the images in drawable and mipmap-hdpi | Johanna Marisse Jose Lorenzo |
| **08/02/16** | Started creating java activity and xml layouts | Johanna Marisse Jose Lorenzo |
| **08/04/16** | Started coding and connecting of activities | Jose Lorenzo |
| **08/05/16** | Coding for the setting activity for turning on and off of background music | Jose Lorenzo |
| **08/12/16** | Started animating the welcome page | Jose Lorenzo |
| **08/13/16** | Started using shared preference for the score and name | Jose Lorenzo |
| **08/13/16** | Started putting all the stories and questions with time limit | Johanna Marisse Jose Lorenzo |
| **08/13/16** | Started creating the mini game | Jose Lorenzo |
| **08/13/16** | Moving of the character | Jose Lorenzo |
| **08/18/16** | Animating falling food | Johanna Marisse Jose Lorenzo |
| **08/18/16** | Coding for the score of the mini game | Johanna Marisse Jose Lorenzo |
| **08/18/16** | Getting and Displaying High Score  Limit of the mini game | Jose Lorenzo |
| **08/18/16** | Started creating the 2nd level | Johanna Marisse Jose Lorenzo |
| **08/20/16** | Doing the mini game for the next level | Jose Lorenzo |
| **08/21/16** | Creating more background | Johanna Marisse |
| **08/23/16** | Updating Requirement | Johanna Marisse |
| **08/23/16** | Continuing the 2nd level | Jose Lorenzo |
| **08/24/16** | Started creating the 3rd level | Jose Lorenzo |
| **08/29/16** | Finalizing documents | Johanna Marisse Jose Lorenzo |
| **08/29/16** | Finalizing project | Johanna Marisse Jose Lorenzo |
| **09/02/16** | Project Revision | Johanna Marisse Jose Lorenzo |

**Requirement Checklist**

*Project Document*

|  |  |  |  |
| --- | --- | --- | --- |
| Name | TO DO | DOING | DONE |
| 1. Overview |  |  | ✓ |
| 1. If game is selected |  |  | ✓ |
| 1. Technical Requirements |  |  | ✓ |
| 1. Graphics Asset and Samples |  |  | ✓ |
| 1. Glossary of Terms |  |  |  |
| 1. Lesson Learned |  |  | ✓ |
| 1. Resume of each member |  |  | ✓ |
| 1. Revision Table |  |  | ✓ |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | TO DO | DOING | DONE |
| 1. Video Presentation |  |  |  |
| 1. PowerPoint Presentation |  |  | ✓ |
| 1. 2 Printed Documents |  |  | ✓ |

*Project Application (GreenyDroid)*

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | TO DO | DOING | DONE |
| 1. Welcome Page with animated image |  |  | ✓ |
| 1. Game level |  |  | ✓ |
| Easy |  |  | ✓ |
| Medium |  |  | ✓ |
| Hard |  |  | ✓ |
| 1. Displaying scoreboard with name |  |  | ✓ |
| 1. Game settings (on/off) |  |  | ✓ |
| 1. Animated mini game |  |  | ✓ |
| 1. Cut scene before and after game play |  |  | ✓ |
| 1. Game timer |  |  | ✓ |
| 1. Game over feature |  |  | ✓ |
| 1. Sub menus (Play, Setting, About) |  |  | ✓ |

*Reviewer Comments/Issues*

The issues that we encountered in creating this mini game were:

We proposed to our professor that we were going to do stories and mini games in each of every level, so all in all we had 12 levels since Android has now 12 versions (not including the first one), but we realize that we cannot do it for a short period of time because we had to think of 12 mini games and 12 stories. As a result, we think of a solution. Since there are 12 levels, we divide it into 4, but we only did 3 levels in equivalent for Easy, Medium and Hard levels. Also, the reason why we need to change the mechanics of the game, because we encountered memory error because there are lots of java classes and xml layouts since we had 12 levels. We had almost 80 java classes and xml layouts in just 2 levels so we made that solution.