Project_Part 2

Team: Yun Zhou, Sorayya Niazi,Tiantian Xie

Title: Best greedy snake game on Android

Project Summary:

An advanced snake game is to be developed in android platform. People can use their cell phone to play this small game at their spare time and get relaxed.

The game is played by single player, whose goal is to obtain the highest score by eating good food. Before the game begins, the player need to enter the username, the player can choose easy/medium/hard modes of games. The player can also modify the game interface. Once the game begins, the player controls the movement of the snake. At any time, the snake can move in one of the 4 directions which the player can control through touching the screen. When the snake eats food, its length and speed will change. At the same time, the player wins scores. When the snake hit itself or hit the borders , it will die and game is over.

When the game is over, the player can save the information and restart the game or exit.

Requirements:

User Requirements				
ID	**Description**	**Topic Area**	**User**	**Priority**
UR-01	**As a client,I want the system asks for player to enter an unique username before playing game so that the database can retrieve the player’s username and related score.**	**Entering Username**	**Player**	**Critical**
UR-02	**As a client,I want the system provides play game button so that the player be able to begin game in a easy process (no login required).**	**Playing Game**	**Player**	**High**
UR-03	**As a player, I want the system provides setting button so that I am able to adjust the color, difficulty mode and volume.**	**Modifying Setting**	**Player**	**Medium**
UR-04	**As a player, I can view top ten scores by clicking view score button on the main menu.**	**View Score**	**Player**	**Medium**
UR-05	**As a player, I can use pause/resume button during playing the game so that I can take a break and play again.**	**Pause Game**	**Player**	**Low**
UR-06	**As a player, I want to be provided an exit button so that the I can exit the game. I want to be able to do this either after the game is over or even while playing the game.**	**Exit Game**	**Player**	**High**

Functional Requirements

ID	Description	Topic Area	User	Priority
FR-01	The direction of the snake will be controlled by touching screen(up, down, left and right).	Control Game	Player	High
FR-02	The game will be displayed on visual display of specific size.	Display	System	High
FR-03	The game will have a main menu to show help, setting, view score, play game and exit buttons.	Title Screen	Player	High
FR-04	A message “game over ” pops up when the game over.	Game Over	System	High
FR-05	Scoring in game will be based on the amount of good food eaten by the snake.	Score	System	High
FR-06	The current score will be displayed on the screen when player is playing the game.	Score	System	High
FR-07	The game will have background music when the player is playing the game.	Sound	System	Medium

Non-Functional Requirements				
ID	**Description**	**Topic Area**	**User**	**Priority**
NFR-01	Performance : 1) The game loading time should be under 10s.	System’s Functionality	Player System	High
NFR-02	Performance : 2) All pages should be displayed under 3s.	System’s Functionality	Player System	High
NFR-03	Performance : 3) It only supports one player in the game.	System’s Functionality	Player System	High
NFR-04	Performance : 4) The requirement of resource for data storage is medium.	System’s Functionality	Player System	High
NFR-05	Platform Constraints: The game will be installed just on android system.	Platform	Player	Medium
NFR-06	Usability: The game interface should be designed to be familiar for the player. It should have similar design with classical snake game.	Interface	All	High
NFR-07	Supportability: In future it can be extended to other different platforms.	Support	Player	Low
NFR-08	Interface: This system will interact with a database system which stores the players’ usernames and scores.	Intercation	System	Medium

Business Requirements				
ID	Description	Topic area	User	Priority
BR-01	Players can share their names and scores through Facebook or Twitter.	Social Network	Player	Low

User and Tasks:

In this android app, only one single player can play the game. In the welcome page, the player is asked to enter a user name. The system then uses the username to record the user id and their score in the database. Then, the user can modify setting, including both volume and color. The user can also choose difficult mode. The system changes initial speed of snake based on this choice. During playing, the player controls the direction of snake via touching the screen or clicking keyboard buttons. When the snake eats good food, the score increases and its length and speed will decrease. The real time score can be displayed on screen. When the snake eats bad food, the length of snake will increase and the speed will also increase. The player can pause, resume and exit game when playing. When the snake hits its body or hits the border, obstacles of the game field, the game is over. The system will record the ID and score. The player can choose to play again, return to the main menu or exit game in the end.

The Use Case Diagram is shown in Figure 1:

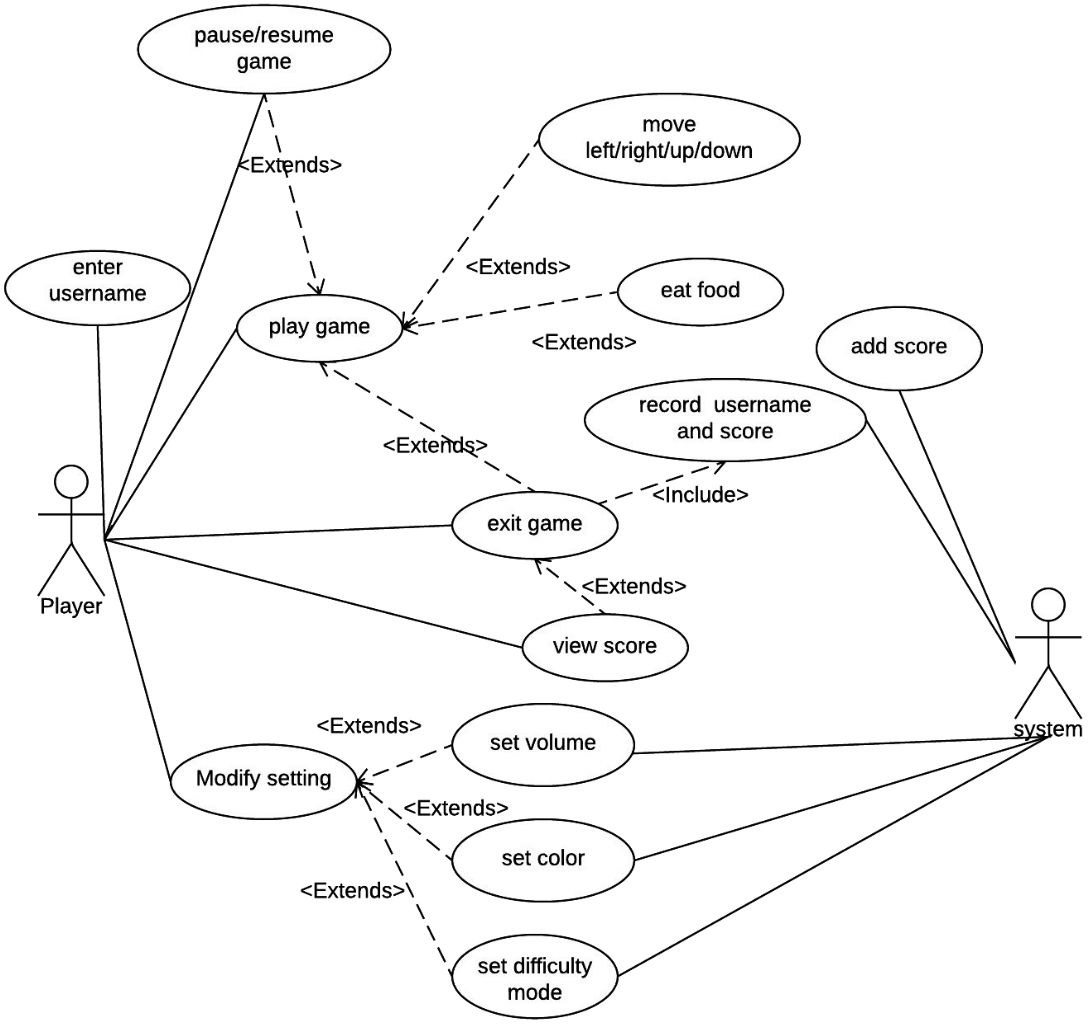
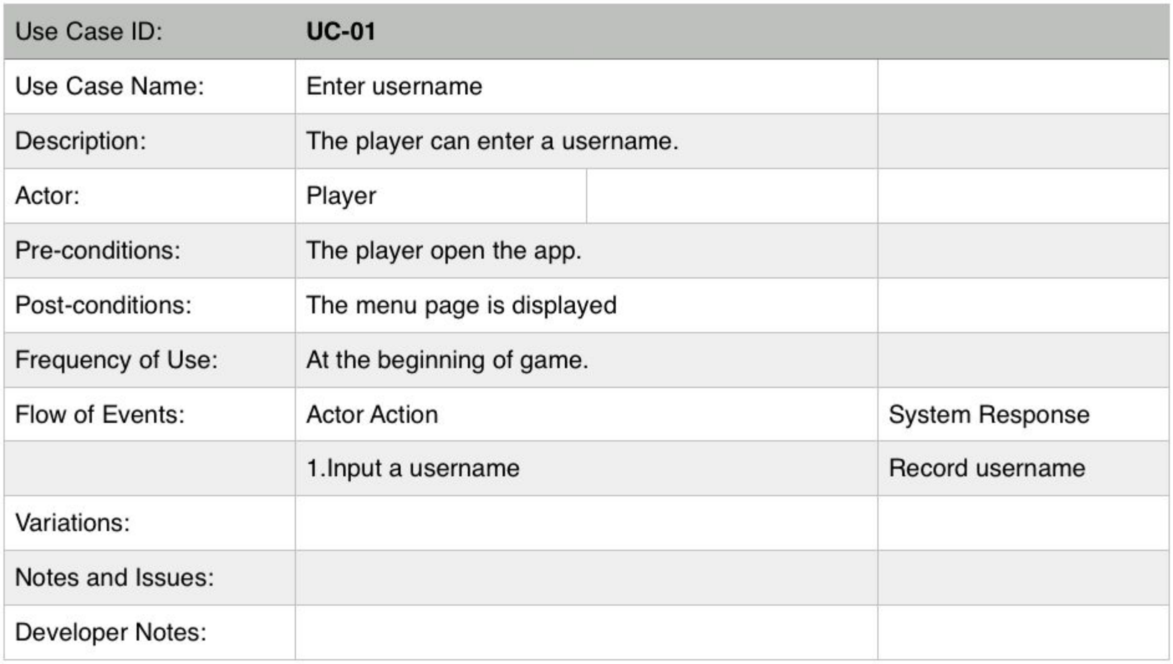
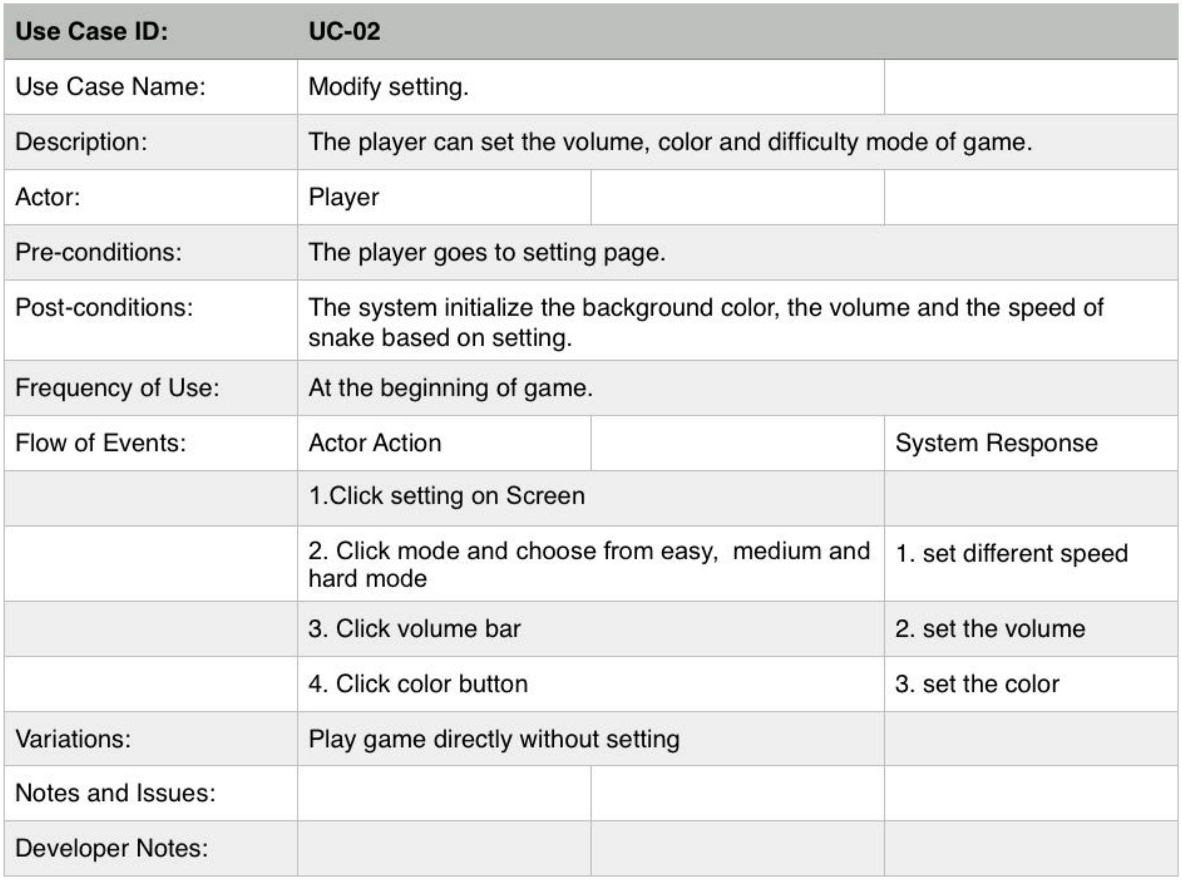


Figure 1: Use Case Diagram

The tables below are the Use Case documents:



Use Case ID:	**UC-01**	
Use Case Name:	Enter username	
Description:	The player can enter a username.	
Actor:	Player	
Pre-conditions:	The player open the app.	
Post-conditions:	The menu page is displayed	
Frequency of Use:	At the beginning of game.	
Flow of Events:	Actor Action	System Response
	1.Input a username	Record username
Variations:		
Notes and Issues:		
Developer Notes:		



Use Case ID:	**UC-02**	
Use Case Name:	Modify setting.	
Description:	The player can set the volume, color and difficulty mode of game.	
Actor:	Player	
Pre-conditions:	The player goes to setting page.	
Post-conditions:	The system initialize the background color, the volume and the speed of snake based on setting.	
Frequency of Use:	At the beginning of game.	
Flow of Events:	Actor Action	System Response
	1.Click setting on Screen	
	2. Click mode and choose from easy, medium and hard mode	1. set different speed
	3. Click volume bar	2. set the volume
	4. Click color button	3. set the color
Variations:	Play game directly without setting	
Notes and Issues:		
Developer Notes:		

Use Case:	**UC-03**				
Use Case Name:	Play game.				
Description:	The player is in control of a snake moving around the a square field of cells to four directions. The player is in control of a snake to eat different kinds of food.				
Actors:	Player				
Pre-conditions:	The player start to play game. In the game canvas, the system draw a square field of cells and a snake with a number of cells.				
Post-conditions:	The snake move four direction: left, right, up and down. The length of snake increase or decrease. The velocity of snake changes.				
Frequency of Use:	During playing				
Flow of Events:	Actor Action			System Response	
	1. Touch the screen to move left, right, up and down.			Change the direction of snake and move the snake. If the new position is food, then the snake will eat food. The length of snake increase or decrease. The velocity of snake changes.	
Variations:	The player pause the game, turn off sound, and exit game.				
Notes and Issues:					
Developer Notes:					

Use Case ID:	**UC-04**		
Use Case Name:	View Score.		
Description:	The player can view the score before the game start and after the game finish.		
Actor:	Player		
Pre-condition:	The system record the score and User ID in database file.		
Post-condition:	The player know the highest 10 score history at the beginning and end of the game.		
Frequency of Use:	At the beginning and at the end of game.		
Flow of Events:	Actor Action		System Response
	1. At the beginning, click view score on screen		List the highest 10 scores by this id
	3. At the end, once the game finish, the score and ranking of this playing will display automatically		List the score and ranking of this playing
Variations:	Set the color, volume or play game.		
Notes and Issues:			
Developer Notes:			

User Case ID:	**UC-05**			
Use Case Name:	Pause/Resume game			
Description:	The player can pause and resume game during playing.			
Actor:	Player			
Pre-conditions:	The player are playing game.			
Post-conditions:	The game are paused or resumed after pressing the pause/resume button.			
Frequency of Use:	During playing game.			
Flow of Events:	Actor Action		System Response	
	1. Click the pause button		The system is paused and the pause button changes to resume button.	
	2. Click the resume button		The system is resumed and the resume button changes to pause button.	
Variations:	The player play game until game is over.			
Notes and Issues:				
Developer Notes:				

Use Case ID:	**UC-06**		
Use Case Name:	Exit the game.		
Description:	The player can exit the game at any time.		
Actor:	Player		
Pre-condition:	The player open the app.		
Post-condition:	The game exit and the app shut down.		
Frequency of Use:	All the time		
Flow of Events:	Actor Action		System Response
	The player click the Exit button		The app shut down
Variation:	The player play game until game is over and Restart game when game is over		
Notes and Issues:			
Developer Notes:			

Use Case ID:	**UC-07**		
Use Case Name:	Record username and score		
Description:	The system record the username and score after game is over.		
Actor:	System		
Pre-condition:	The game is finished, and the player is willing to save the information about this play.		
Post-condition	The score and user name about this play will be saved in the database.		
Frequency of Use:	At the game is over		
Flow of Events:	Actor Action		System Response
	1. Click the SAVE button		Save username and score in Database.
Variations:	Exit game or restart game directly.		
Notes and Issues:			
Developer Notes:			

Use Case ID:	**UC-08**		
Use Case Name:	Add score		
Description:	The system will count score and display on screen during playing game.		
Actor:	System		
Pre-condition:	The player start to play game and eat food.		
Post-condition:	The real time score is displayed above the game canvas.		
Frequency of Use:	During playing game.		
Flow of Events:	Actor Action		System Response
	update the real time score		Update UI
Variations:	None		
Notes and Issues:			
Developer Notes:			

Activity Diagram:
This activity diagram describes the logic process involving activities from a player loading game to exiting game. When the player starts playing game, he or she controls the movement of the snake to avoid hitting border, obstacle or self, and also to eat as many good food as it can to get scores. Once the snake hits border, obstacle or self, it dies and the game is over. Then the system shows the top ten players and their related scores. The player can choose to return to the main menu, exit or play again.

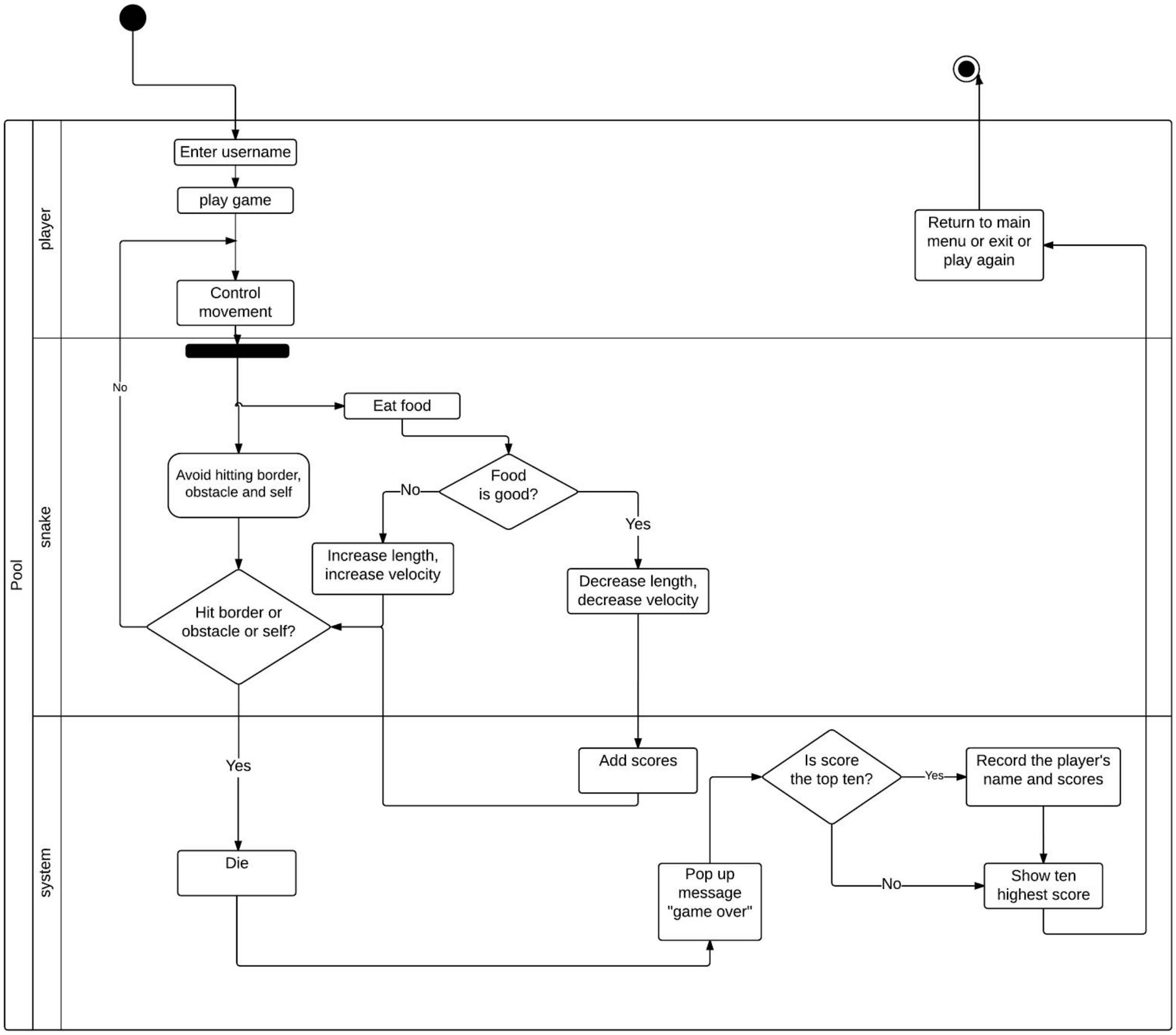

Pool
player
snake
system
Enter username
play game
Control movement
No
Eat food
Avoid hitting border, obstacle and self
Food is good?
No
Yes
Increase length, increase velocity
Decrease length, decrease velocity
Hit border or obstacle or self?
Yes
Add scores
Is score the top ten?
Yes
Record the player's name and scores
Die
Pop up message "game over"
No
Show ten highest score
Return to main menu or exit or play again

Figure 2: Activity Diagram of Greedy Snake Game.

Data Storage:
The SQLite database is used to stores data. Android has built in SQL database implementation. The SQLite is connected to the score class. Once the game is over, the user name and score will be insert to SQLite database.

UI Mockups:

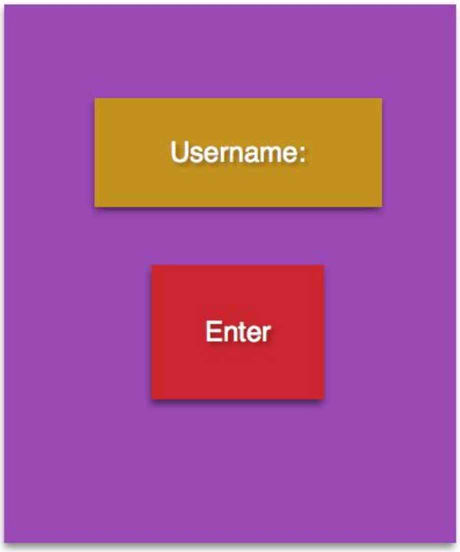
Here are the screen mockups for user interface of different parts of the application.

The user will see the following use cases as he/she goes work through. First screen requires user to enter username. Then user push enter button to go to next screen. Second screen is the menu page. In this screen user can modify setting, view top ten scores and see help document. Setting screen includes changing color, difficulty mode of game and modifying sound.

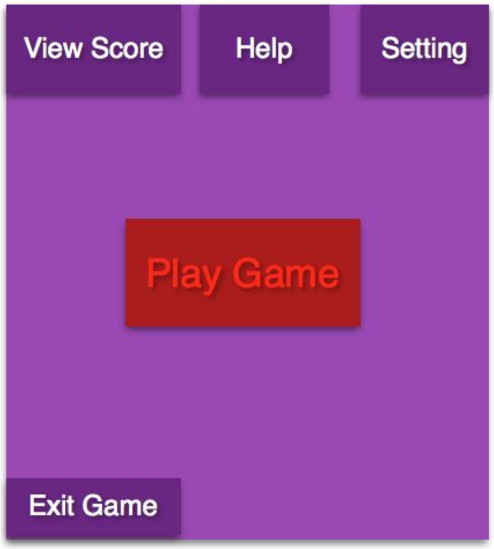
Then user will go to the playing page, and be ready to move snake. At this stage when user is playing the game, he or she can modify sound, pause/resume game ,view current score or exit the game. Once the game is over, a “ Game Over ” message will pop up and user can save information, exit game, return to main menu or play again.

UI Mockups of Greedy Snake Game are shown in Figure 3:

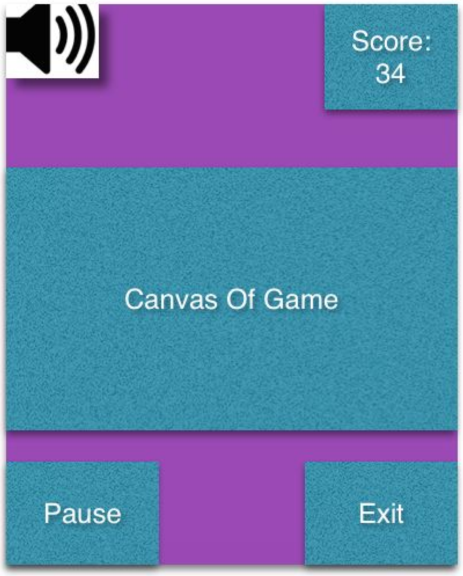
Part A :



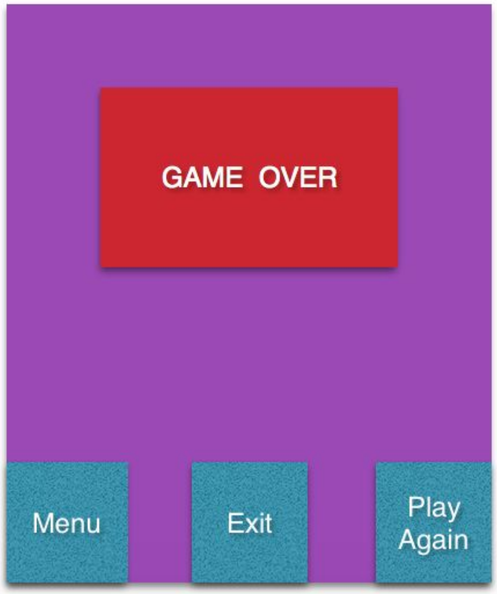
Part B:



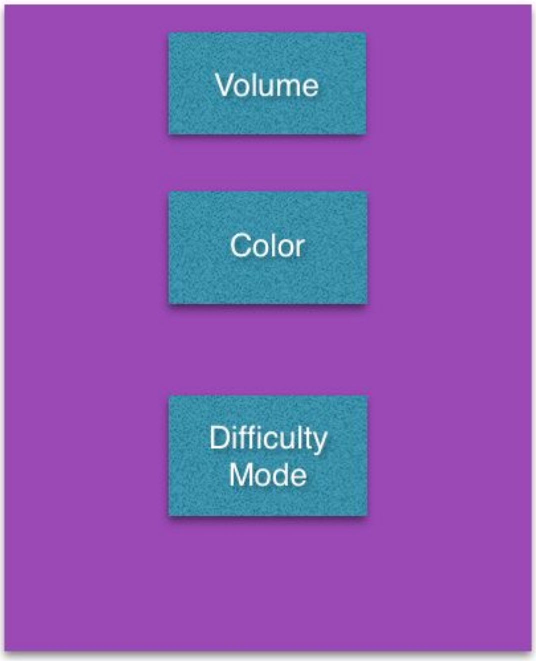
Part C:



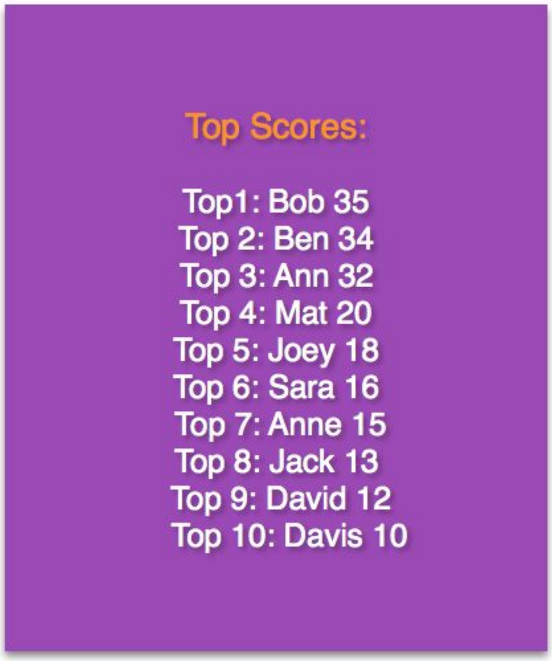
Part D:



Part B.1

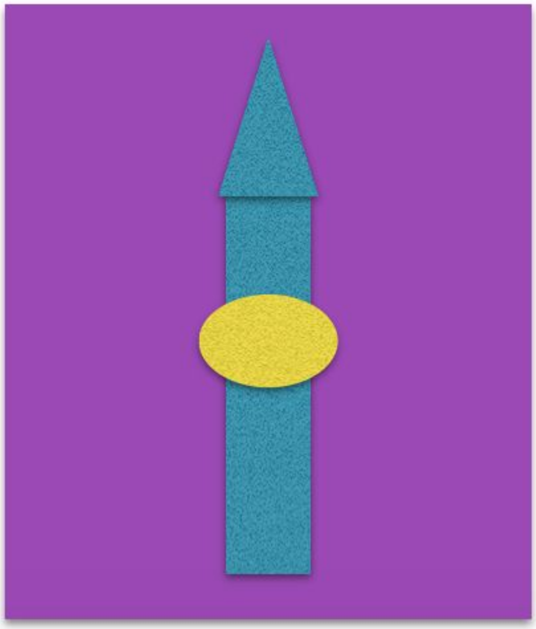


Part B.2

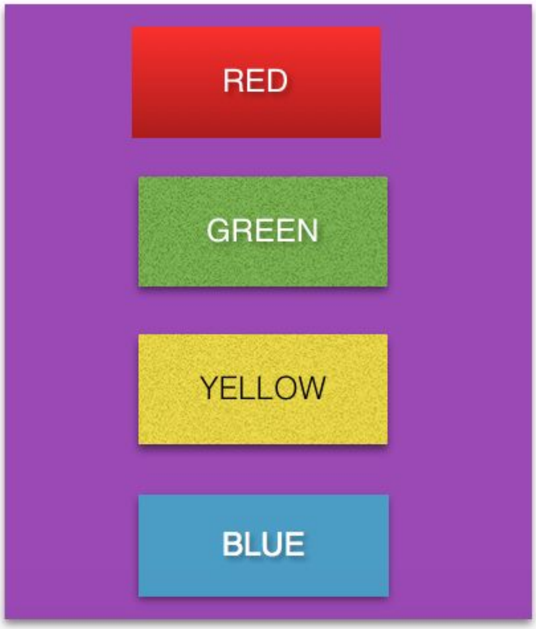

Top Scores:

Top1: Bob 35
Top 2: Ben 34
Top 3: Ann 32
Top 4: Mat 20
Top 5: Joey 18
Top 6: Sara 16
Top 7: Anne 15
Top 8: Jack 13
Top 9: David 12
Top 10: Davis 10

Part B.1.1



Part B.1.2



Part B.1.3

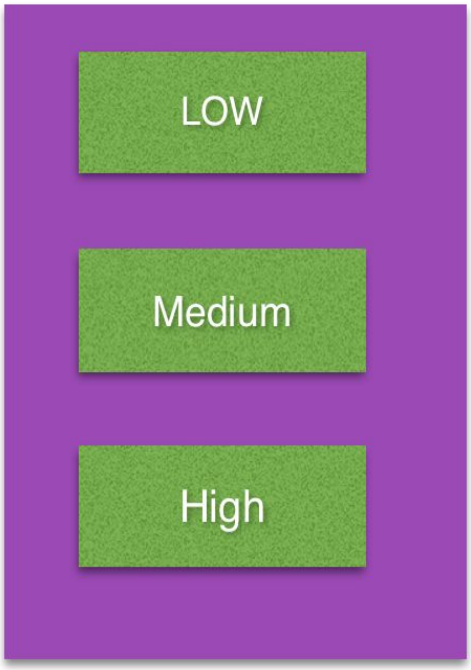


Figure 3: Mockups of Greedy Snake Game

User Interactions:

When the app is open, the WelcomeActivity class gets the username from player. When the player sets username and clicks enter button, the Menu Page is displayed in MenuActivity class. The menu page includes ViewScore, Setting, Help and Play buttons. When the player clicks setting button, the system goes to setting page which is in SettingActivity class. The player can set difficulty mode - the initial speed and set volume which are controlled in Utility class. The player can also set color which is in TextAdapter class. The TextAdaper then display the new view for player. When the player finish setting, he/she can go back and begin to play game clicking Play button. The sequence process is show in Figure 4.

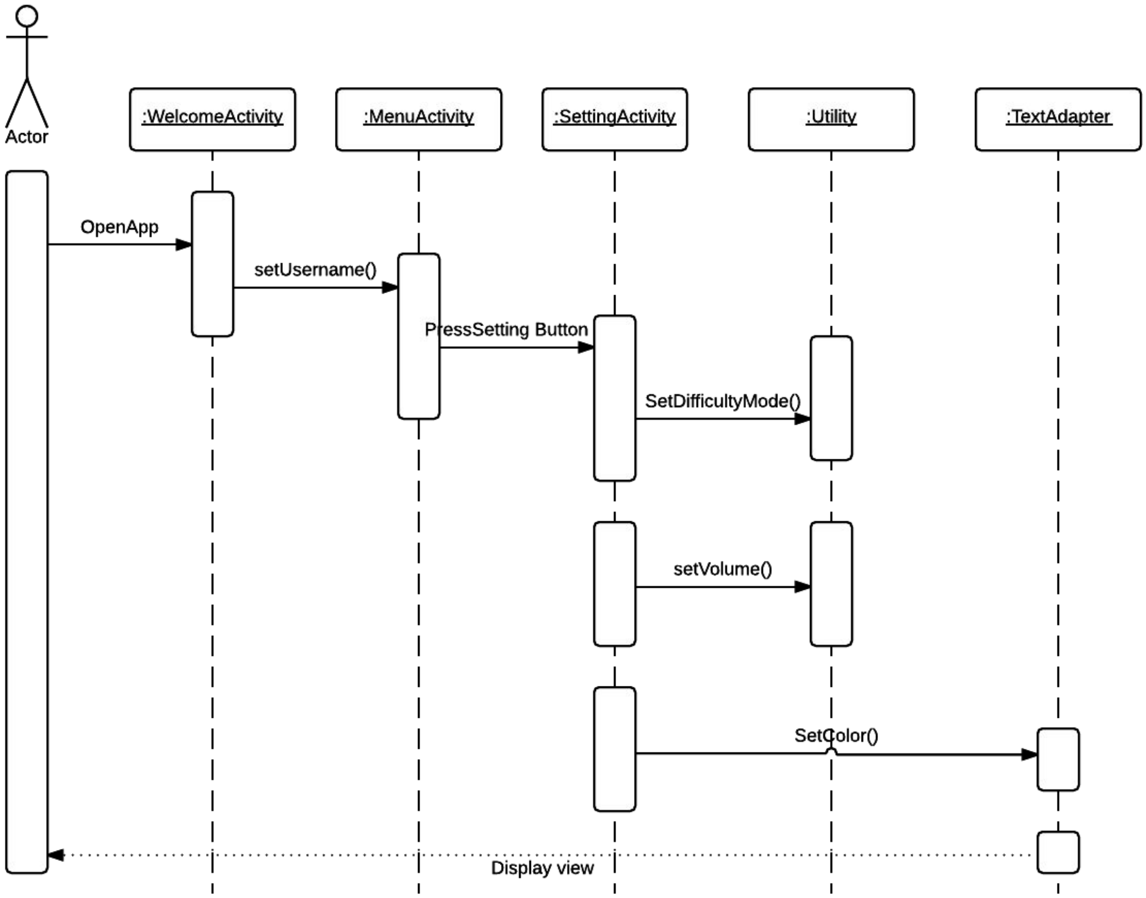


Figure 4: The sequence diagram of setting

In the play section, once the player touch the screen, the snakeActivity class gets the direction. The snakeActivity class then sends the direction input to GameController class. The GameContoller moves the snake to the direction according to the input. If the new position of snake head has food cell. The Snake class will eat the food cell. After moving and eating, the GameController will judge whether game is over via call setHitSelf method in Snake class and isWalkable GameField class. The GameController then updates score in Score class and update GUI in TextAdapter class. The sequence diagram of playing is shown in Figure 5.

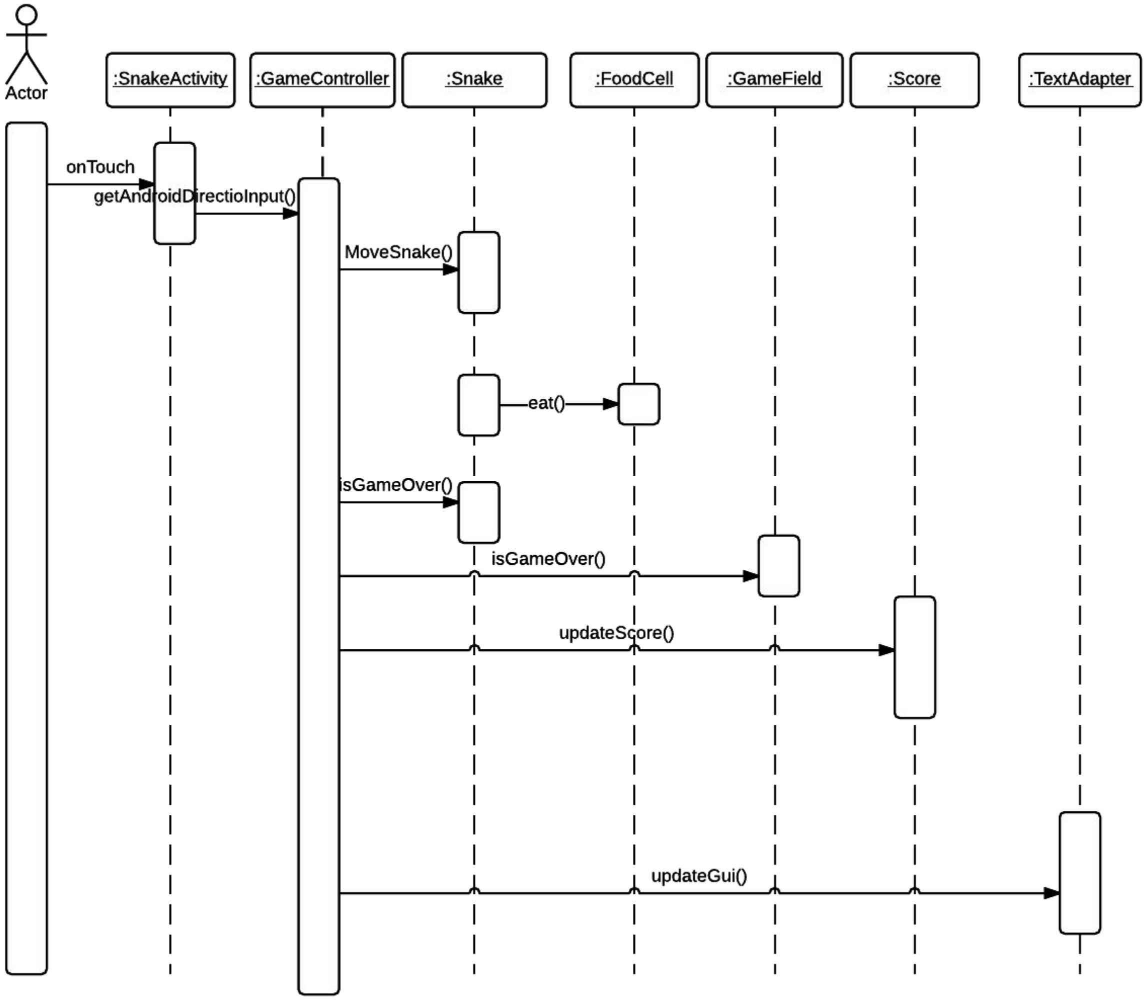


Figure 5: The sequence diagram of playing

Once the snake is dead, the GameController class will send a gameover message to GameoverActivity class to activate the game over page. In the game over page, the player can click the save button to save ID and score. The player can also click menu to restart game. The sequence diagram after the game over is shown in Figure 6.

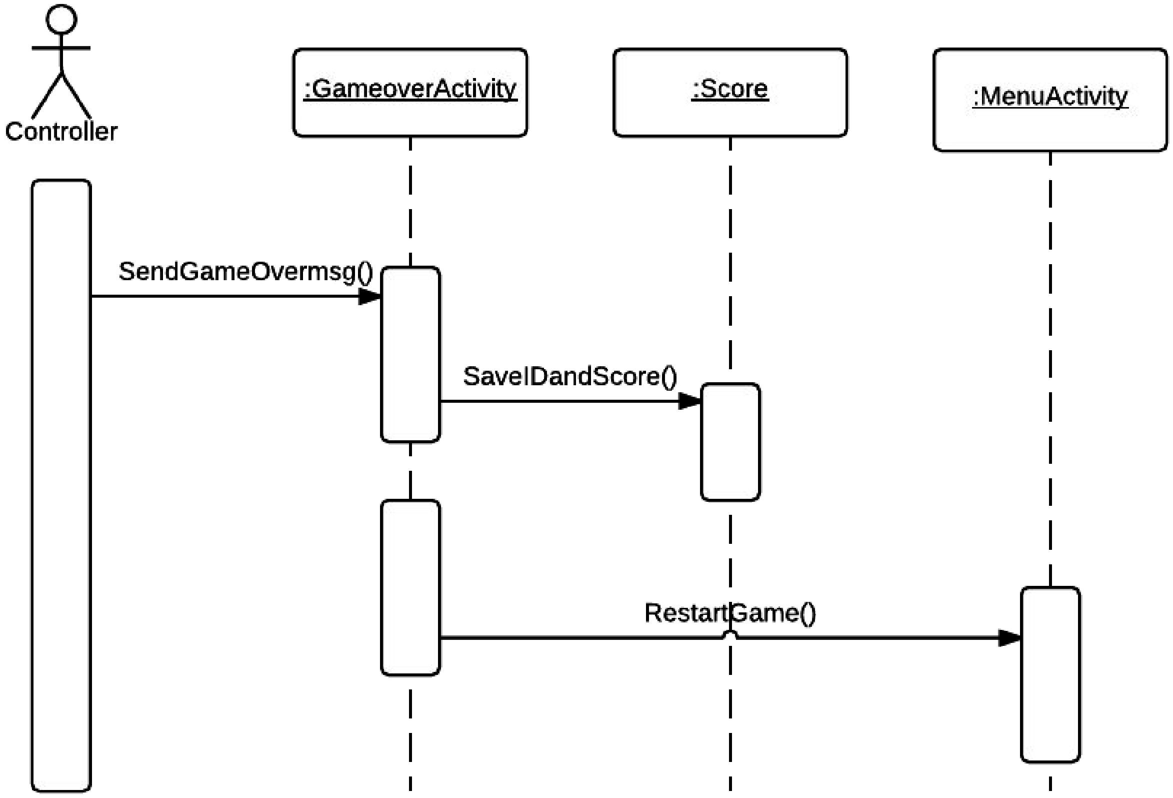


Figure 6: The sequence diagram after game over.

Class Diagram:

In this class diagram design process, we take advantage of MVC framework here. In this way, our class diagram consists of four packages: Android, Model, Control and View packages.

The Model is the part of the game application that handles the logic for the application data. Model objects also retrieve players' data (and store data) from the database.

The View is the parts of the game application that handles the display. The views are created from the model data.

The Controller is the part of the game application that handles user interaction. Controllers read data from a view, control user input, and send input data to the model.

The Android is the user terminal part of the game application that receive user actions from players and deliver these actions to Controller and View.

The class diagram is shown in Figure 7.

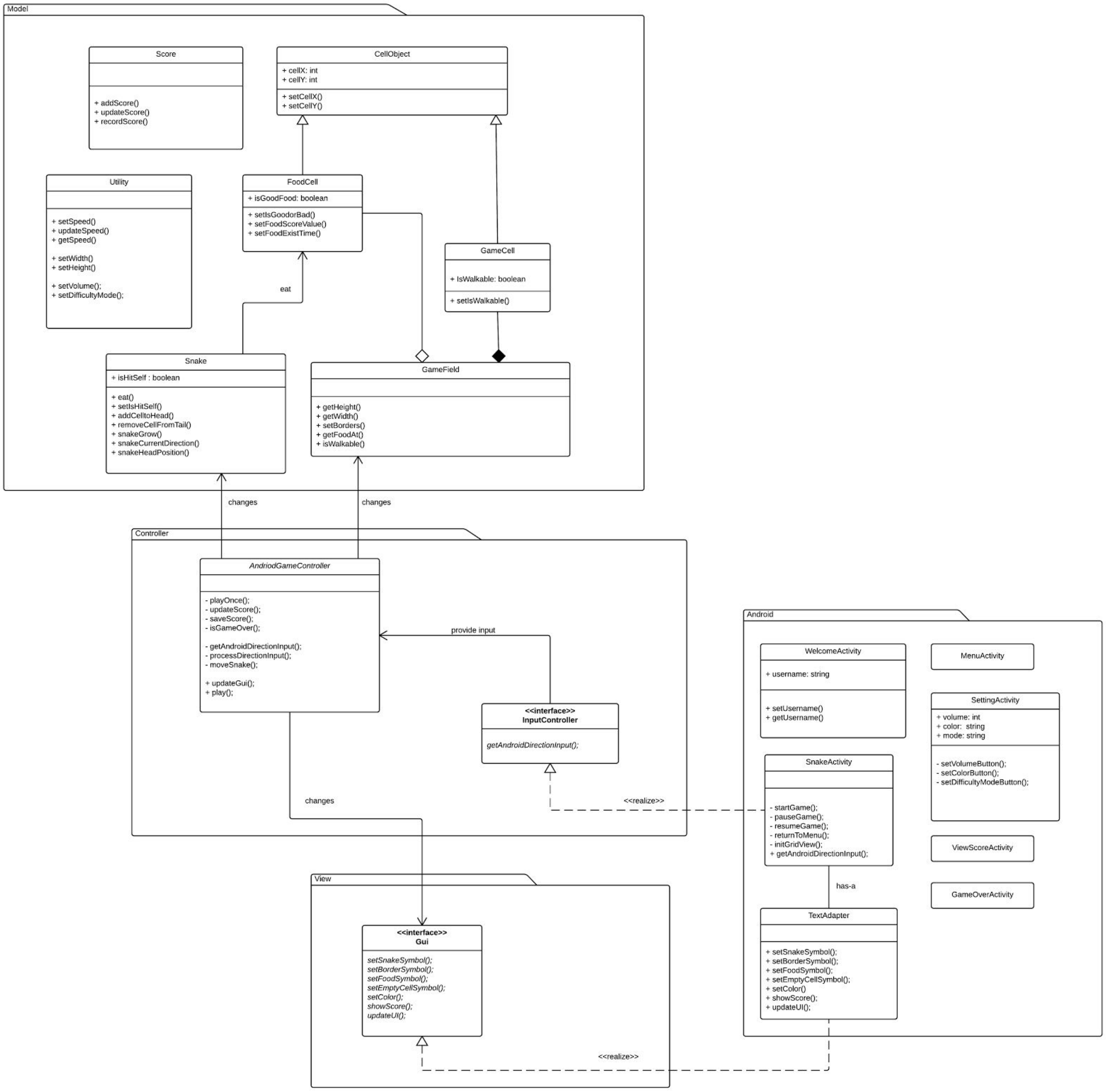


Figure 7. Class Diagram of Greedy Snake Game.

State Diagram:

1. Player State Diagram:

This diagram shows the overall states of the player. It starts by entering username. After loading game the player has options of play game, modify setting and exit. Player can also see the top ten score as well. For setting there are three buttons which are setting color, setting difficulty mode and changing volume. Inside of play game, again there are three buttons: pause/resume, turnoff/on sound and exit. Figure 8 and 9 show the game state and snake state diagram respectively.

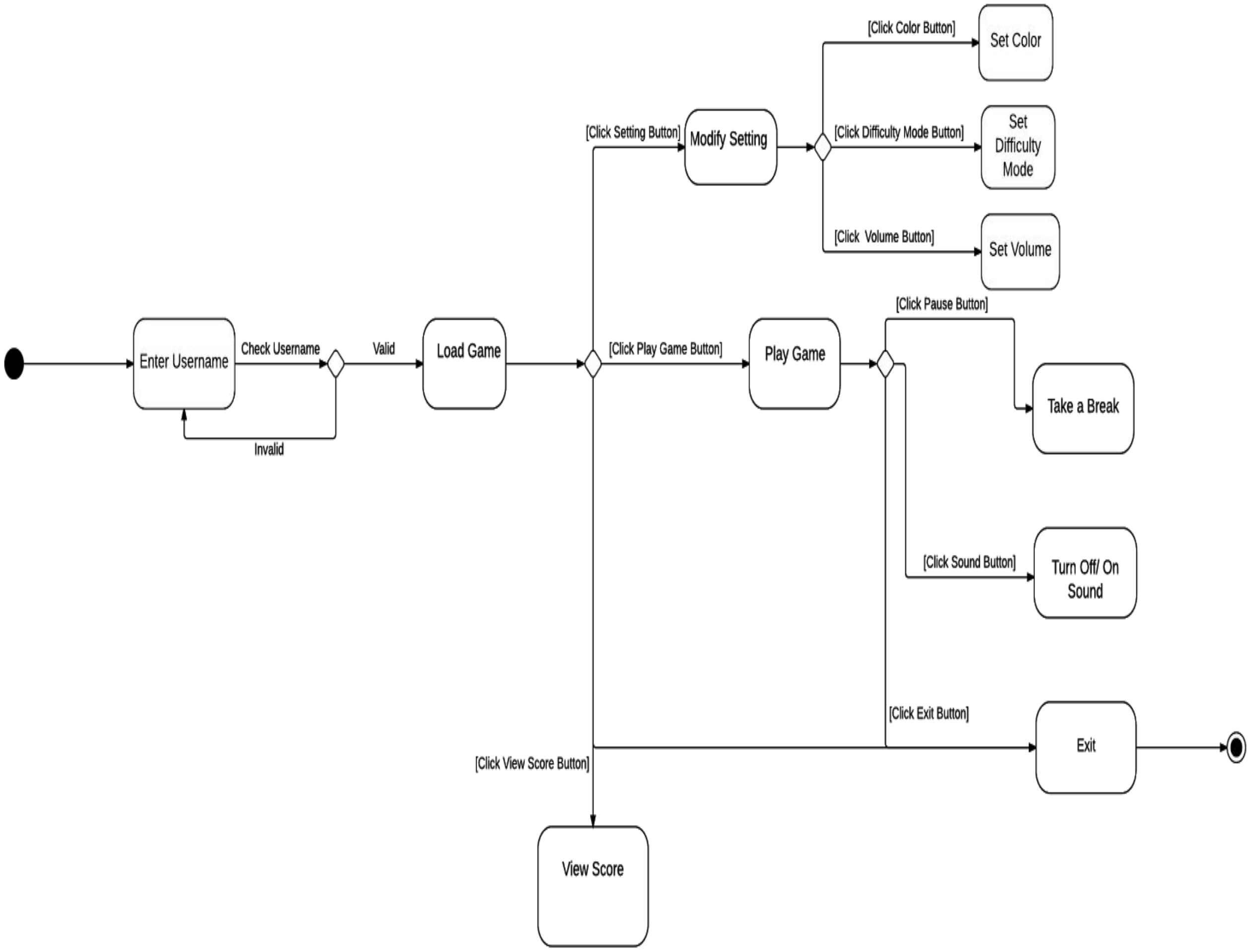


Figure 8. Player State Diagram of Greedy Snake Game.

2. Snake State Diagram:

This diagram shows the snake states during playing. Player starts pressing and gives touch gestures on screen. The snake changes direction correspondingly. If the snake eats good food, its speed and its length will decrease. If the snake eat bad food, its speed will increase. If snake hits border, obstacles or itself, it will die.

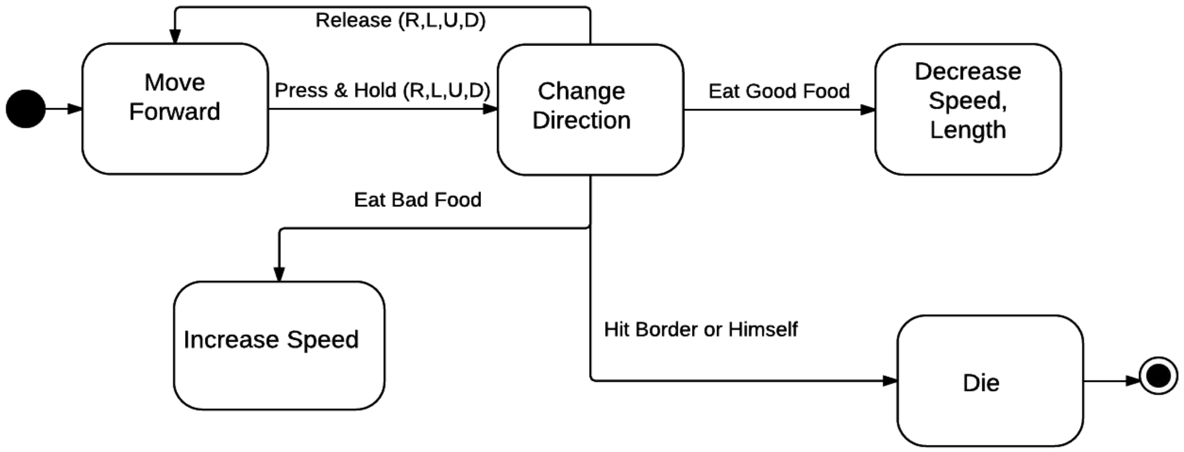


Figure 9. Snake State Diagram of Greedy Snake Game.