Extra Credit For Project Part 2

Name: Sorayya Niazi

Title: Greedy Snake game

Project Summary:

A snake game will be developed that can be used on android phones.
The game is played by one player, whose goal is to obtain the highest score by eating good food. For playing game, first the player needs to enter the username. Then the player can choose easy/medium/hard modes of games. Once the game begins, the player take control of the movement of the snake. At any time, the snake can move in one of the four directions which the player will 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 play again or exit.

Requirements Document:

User requirements				
ID	**Description**	**Topic Area**	**User**	**Priority**
UR-01	As a client, I want the system asks the player to enter a username before the player starts to play so that I be able to have access to player's username and saved score in database.	Entering Username	Player	Critical
UR-02	As a player, I want the system provides me setting button so that I can adjust Difficulty Mode.	Modifying Setting	Player	Medium
	As a player, I want the system provides me setting button so that I can adjust Volume button.	Modifying Setting	Player	High
UR-03	As a player, I want the system provides me setting button so that I can adjust Color button.	Modifying Setting	Player	Low

Functional requirements				
ID	**Description**	**Topic Area**	**User**	**Priority**
FR-01	The game will have a main menu to show help, view score, play game and exit buttons.	Title Screen	Player	High
FR-02	The game will have background music while player is playing the game.	Sound	Player	Medium
FR-03	The game will have a main menu to show help, view score, play game and exit buttons.	Title Screen	Player	High
FR-4	The game will have a main menu to show setting button.	Title Screen	Player	High

Non-Functional requirements				
ID	**Description**	**Topic Area**	**User**	**Priority**
NFR-01	Performance: 1) The game loading time should be under 20s.	System's Functionality	Player, System	High
NFR-02	Performance: 2) It only supports one player in the game.	System's Functionality	Player, 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	High

User and tasks:

In this android app, only one 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.The Use Case Diagram is shown in Figure 1.

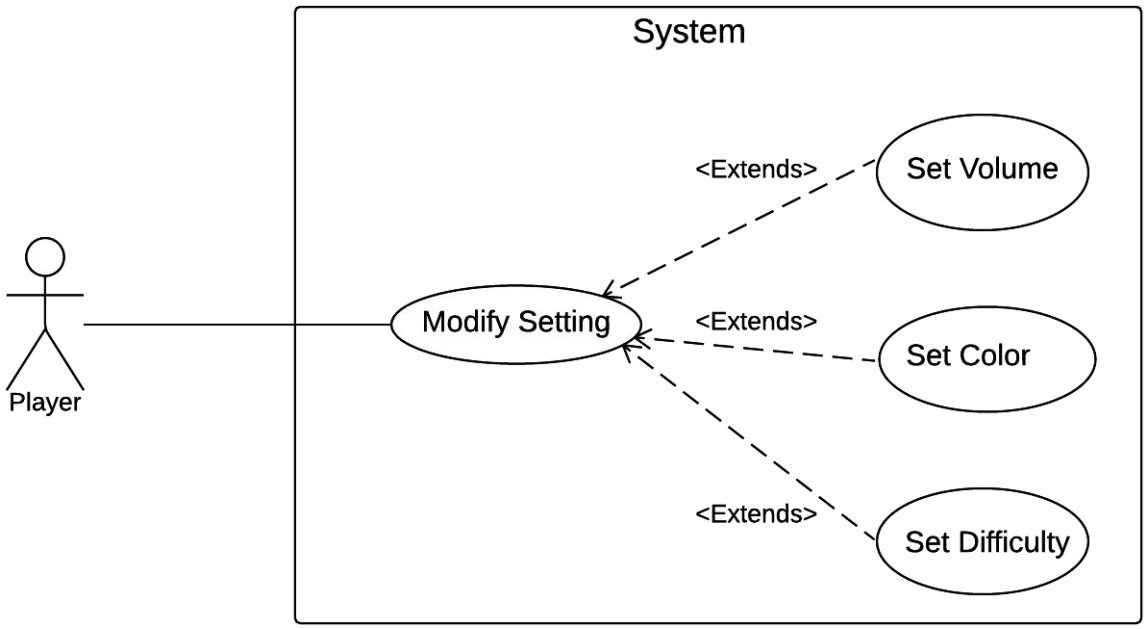


Figure 1: Use case Diagram

Use Case Document:

The following table shows the Use Case Document:

Use Case ID:	UC-01
Use Case Name:	Modify Setting
Description:	The player can change the volume, color and difficulty mode of the game.

Actor:	Player		
Pre-conditions:	The player opens setting page.		
Post-conditions:	The system initializes background color, the volume and speed of snake according to difficulty mode.		
Frequency of Use:	At the beginning of the game.		
Flow of Events:	**Actor Action**		**System Response**
	1	Click setting on screen.	
	2	Click Mode and choose options of difficulty from easy to medium to difficult one.	Set different speed for snake.
	3	Click Volume bar.	Set the volume.
	4	Click Color page.	Set the color.
Variations:	Play game directly without changing setting.		
Notes and Issues:			
Developer Notes:			

Activity Diagram:

This activity diagram describes the logic process involving activities from while player is 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. Activity Diagram is shown in Figure 2.

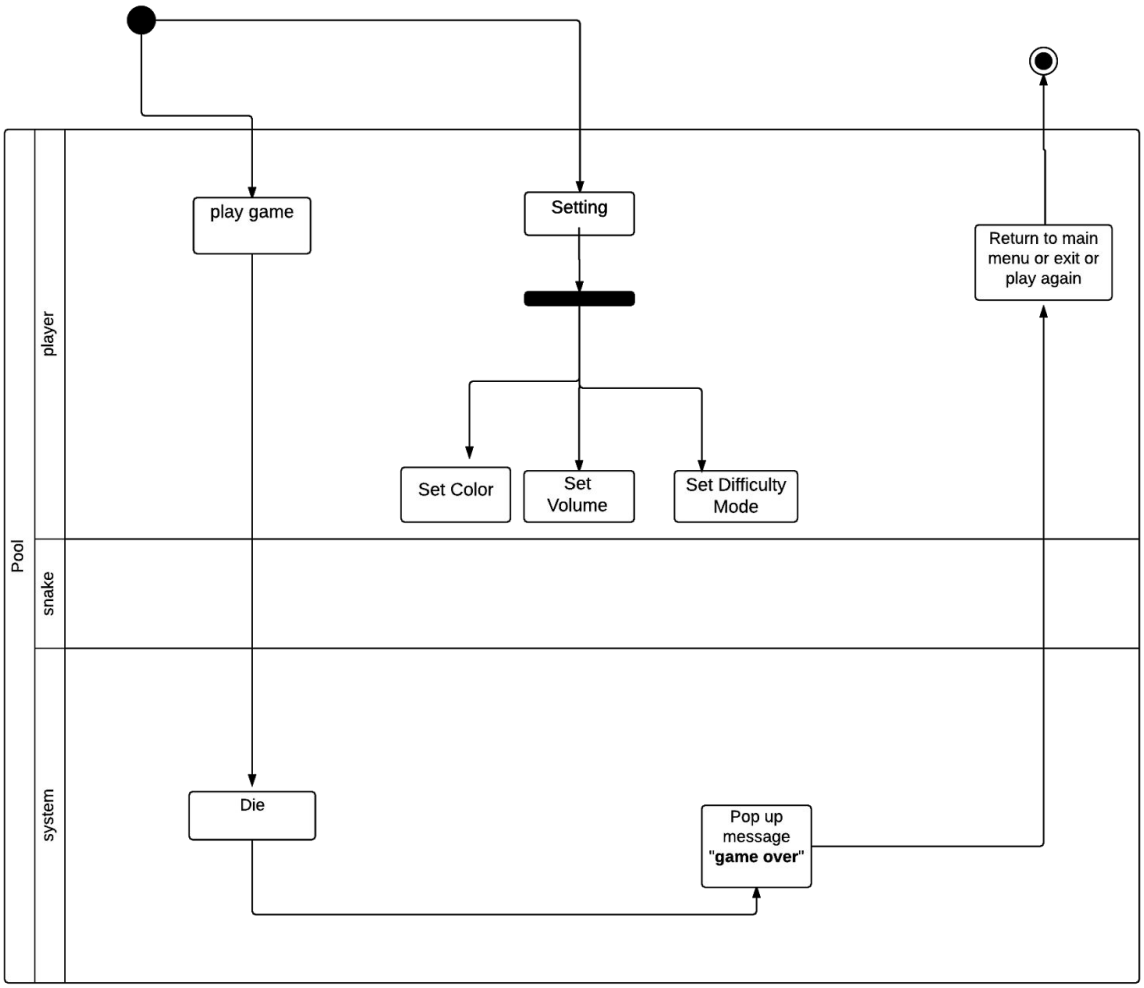


Figure 2: Activity Diagram of Greedy Snake Game.

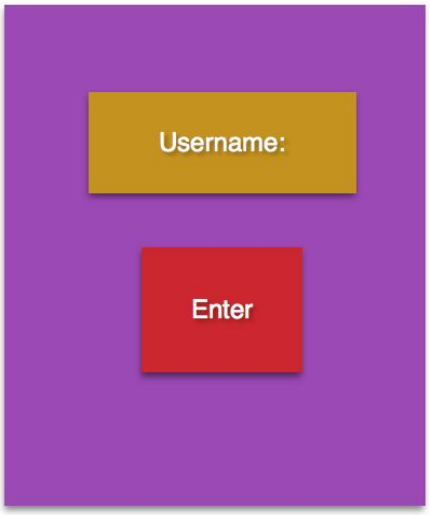
UI Mockups:

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

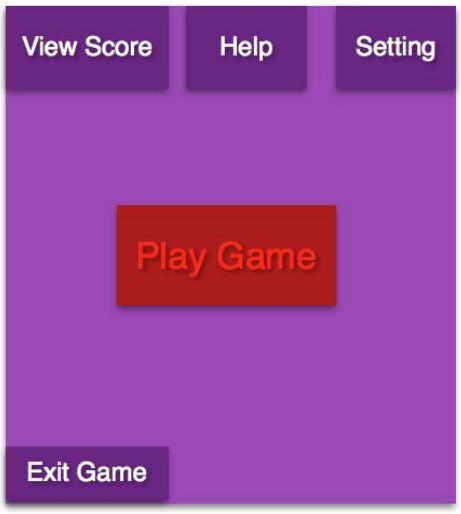
The user will see the following use cases as he/she goes work through. First screen requires user to enter username. Then user push enter button to go to next screen. Second screen is the menu page. In this screen user can modify setting, view top ten scores and see help document. Setting screen includes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

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

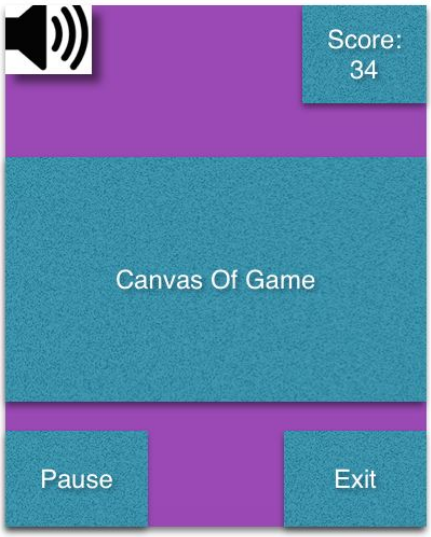
Part A :



Part B:

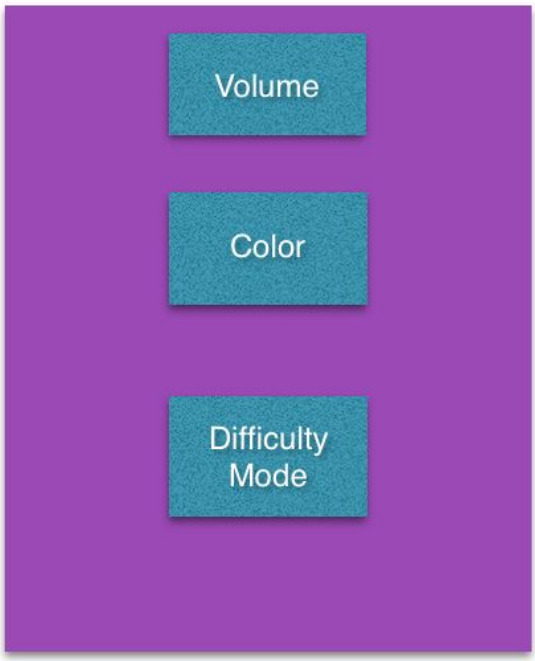


Part C:



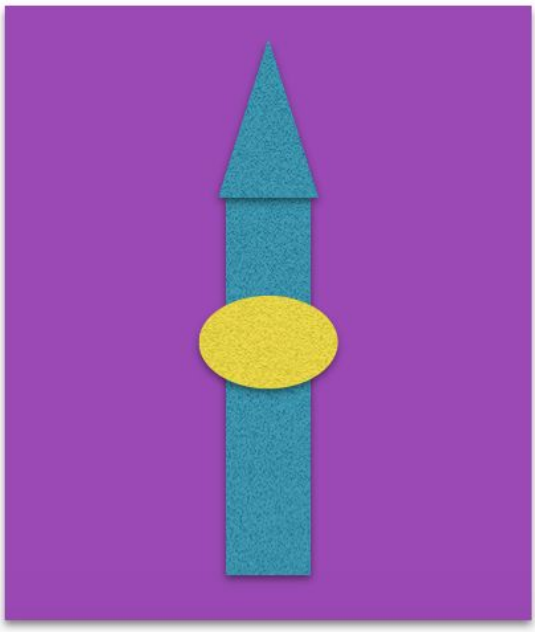
Part D:

Part B.1

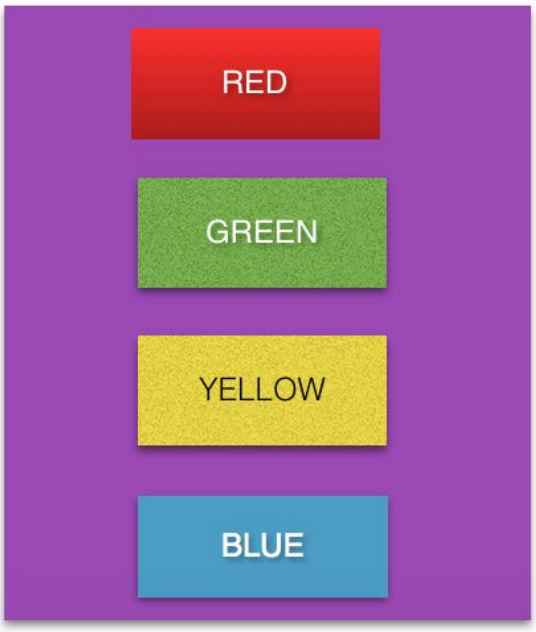


Part B.2

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 opened, 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.

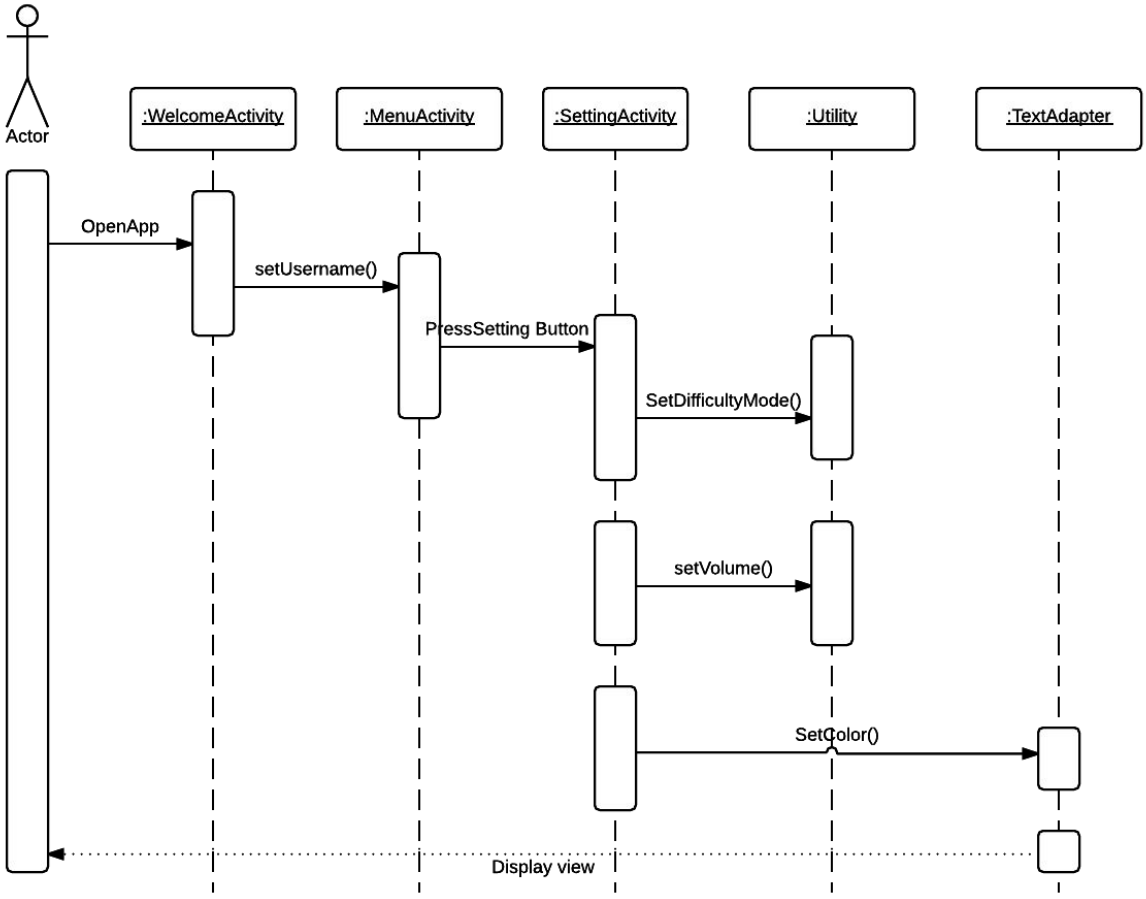


Figure 4: The Sequence diagram of setting of Greedy Snake Game.

Class Diagram:

In this class diagram design, we applied MVC framework here.The 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 5.

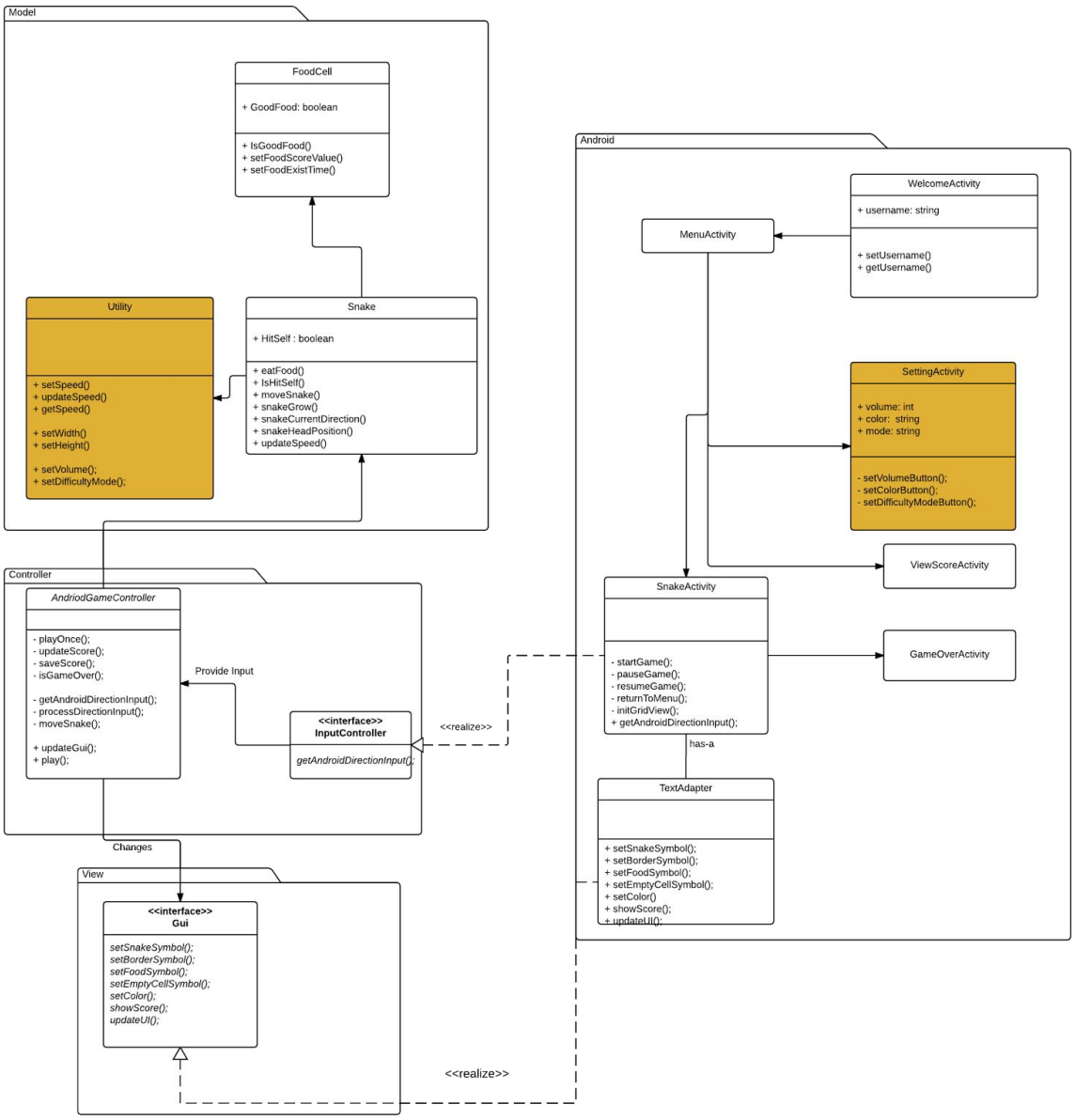


Figure 5: Class diagram of setting of Greedy Snake Game.

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. Figure 6 shows the player state diagram.

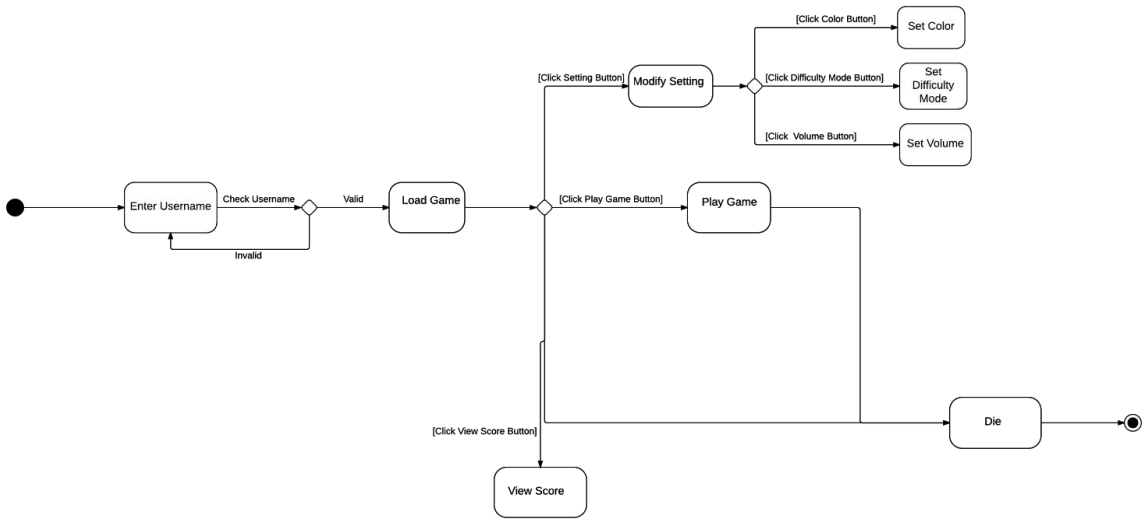


Figure 6: Player state diagram of Greedy Snake Game.

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.