COSC2659 - iOS Development
Assignment 2
Application Documentation

App Name: My Favorite Football Players

Name: Nguyen Manh Khang

ID: s3864131

1. INTRODUCTION

"MONSTER ESCAPE" (ME) is a board game based on the concept of one of the most OG game "Minesweepers". I remembered when I was a kid, I used to spend a lot of time playing Minesweepers and I really enjoyed it. Therefore, that is the reason why I choose this game concept to develop.

Moreover, since I am a big fan of Fantasy world, especially the fantasy in the pixel world. So, I decide the use this theme to develop "MONSTER ESCAPE". Players of ME will be the King and try to expand the Kingdom by exploring more wonderland. However, there are a lot of dangerous monsters wandering outside. Players can magic seal the monster; however, try to avoid them as much as possible.

2. GAME RULE

As mentioned above, players will try to explore the map and avoid all the monsters. In some case, they can magic seal the monster to disable them To help the players can avoid all the monsters, there are some hints on the map. The hints will be displayed as a number which indicates that eight cells that are adjacent to the current cell could have that number of monsters.
Ex: User explore the cell which contains number 2 => There are exactly two monsters among the eight cells that adjacent to the current location
By using those hints, players can easily avoid the monsters and explore more lands. In some case, players can user the Magic Seal option to mark that cell contains the Monster (like the Flag in Minesweepers)

- Winning condition: Explore all the normal cells in the map (not contains the monster cell)
- Losing condition:
  - Explore the cell that contains a monster
  - Time is over

There are three difficulties in the game: Easy, Medium, and Hard. The difference among them is the time limit of each game play. The more difficulty the gameplay has, the more score you get and you will be put in the leaderboard

- Easy: 3 minutes
- Medium: 2 minutes
- Hard: 1 minutes

3. GAME FEATURES

Main features:

- Menu options: Start game, Leaderboard, and How to Play
  Explain: use one container view named MotherView and the environment object ViewRouter to display the corresponding View based on which button is pressed
- Start game: Gameplay like mentioned above
  Explain: build a GameLogic class load the settings of the game from Setting class. The GameLogic will control the logic of the game, especially the Cell view and Cell View Model.
- Leaderboard: list of sorted players based on the score they got
  Explain: Fetch the list of player from Core Data, using sortdescriptor to sort the score from highest to lowest
- How to play: show the instructions how to play the game

  Explain: simple view with ZStack, Vstack and Hstack containing the static data

Other features:

- Difficulty menu: choose the difficulty for the game play
  Explain: each difficulty will set the time limit value to an environment object CurrentPlayer for later usage in the MapView
- Insert new player: Insert the name of new player to start the game
  - Check existence of player based on name
    - Fetch a single player from core data by name
  - Show error when the player's name is already existed
- Check current player then show Greetings message

  Explain: use Environment Object CurrentPlayer to keep track of current player's name, score and time limit
- Background sound: play suitable sounds like background sound, winning sound, game-over sound or some effects

  Explain: use AVFoundation

4. SCREENSHOT

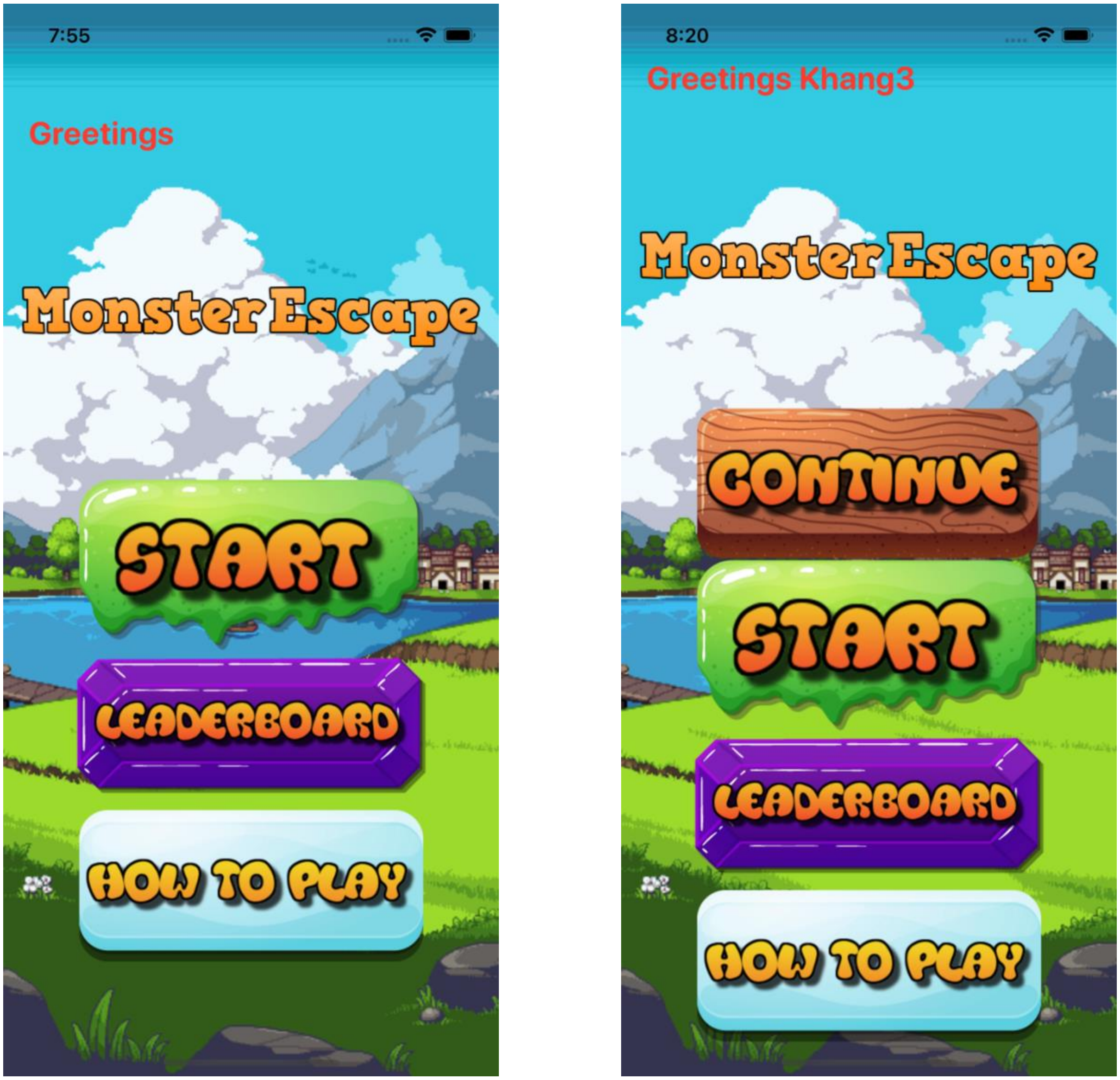
Flow of the app:

Menu <-> Start -> Insert new name -> Choose difficulty -> Play
Menu <-> Continue -> Choose difficulty -> Play
Menu <-> Leaderboard

Menu <-> How-to-Play

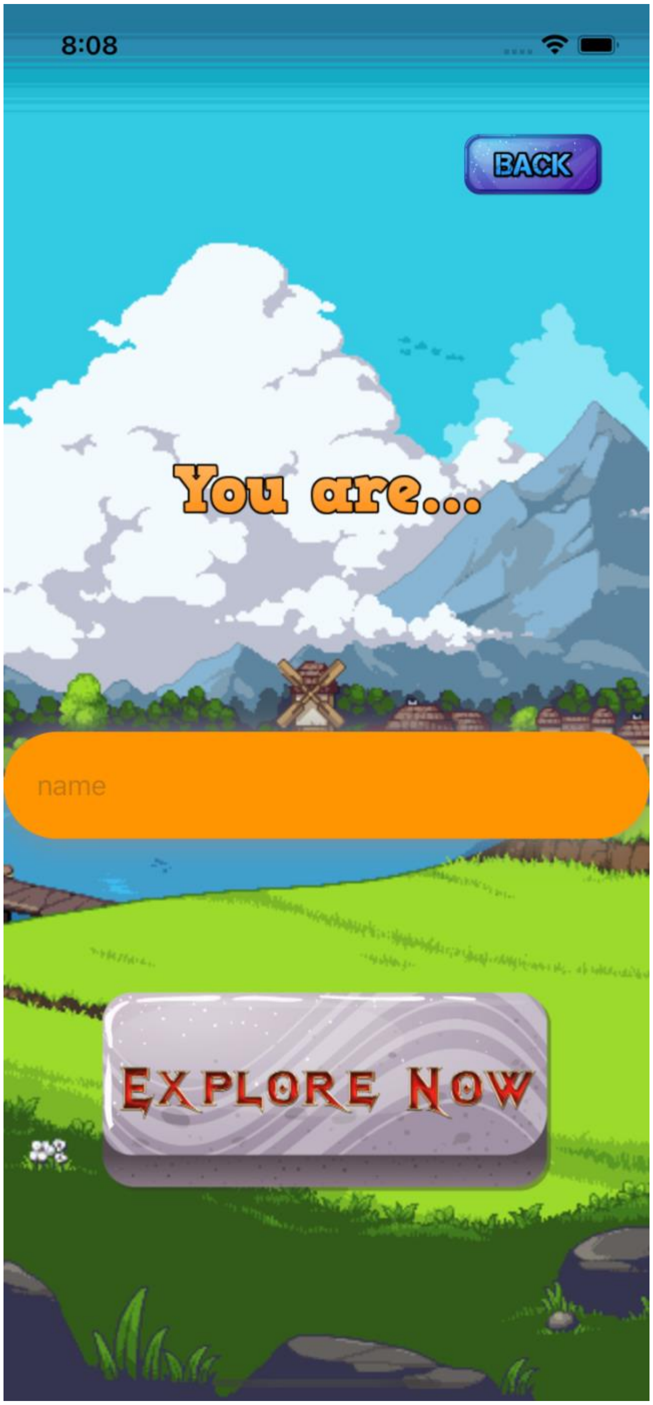
App icon

MonsterEscape

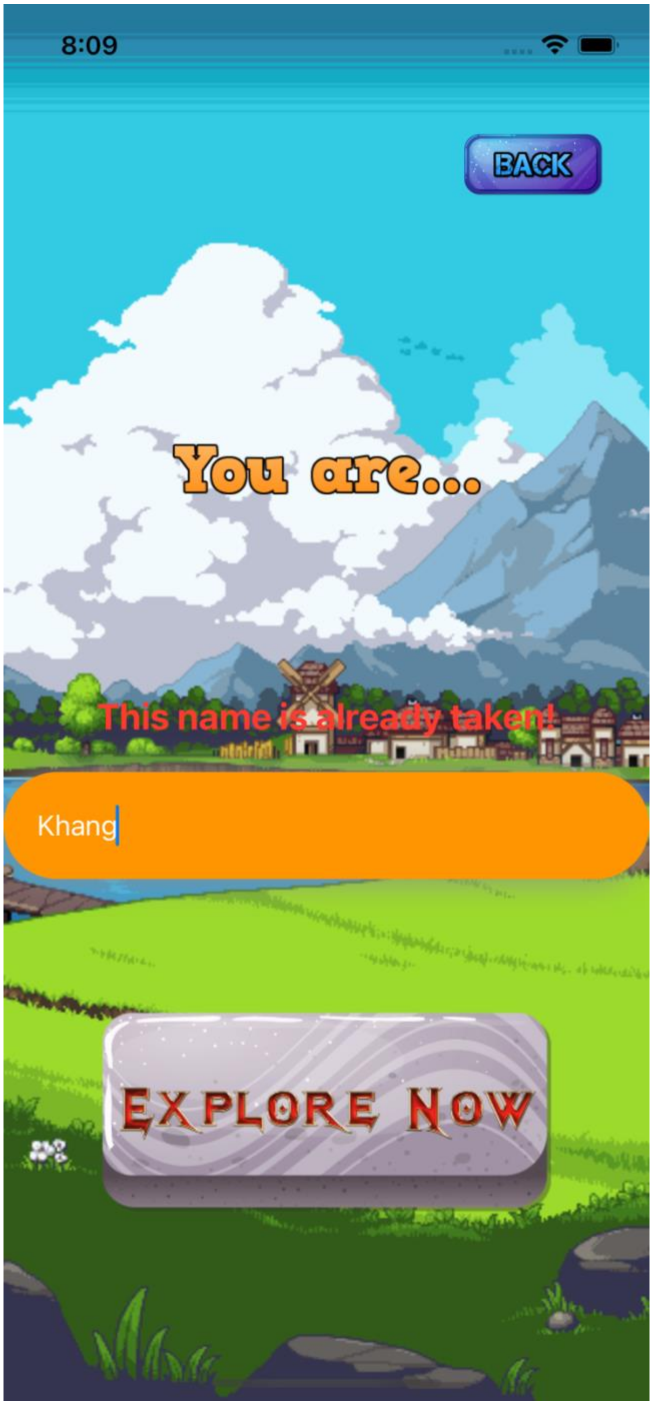


(Without current player) **(with current player)**

Game Menu View



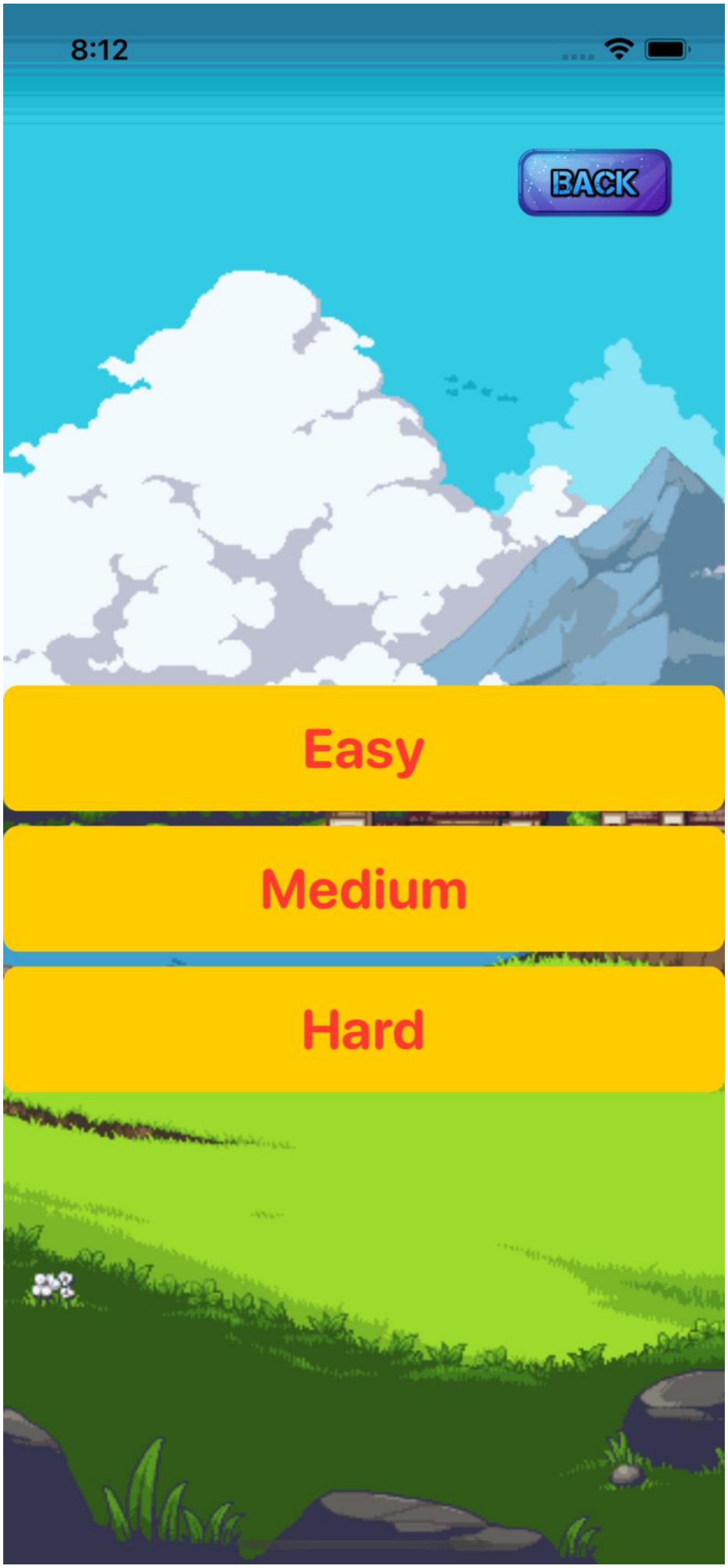
(Insert new player View)



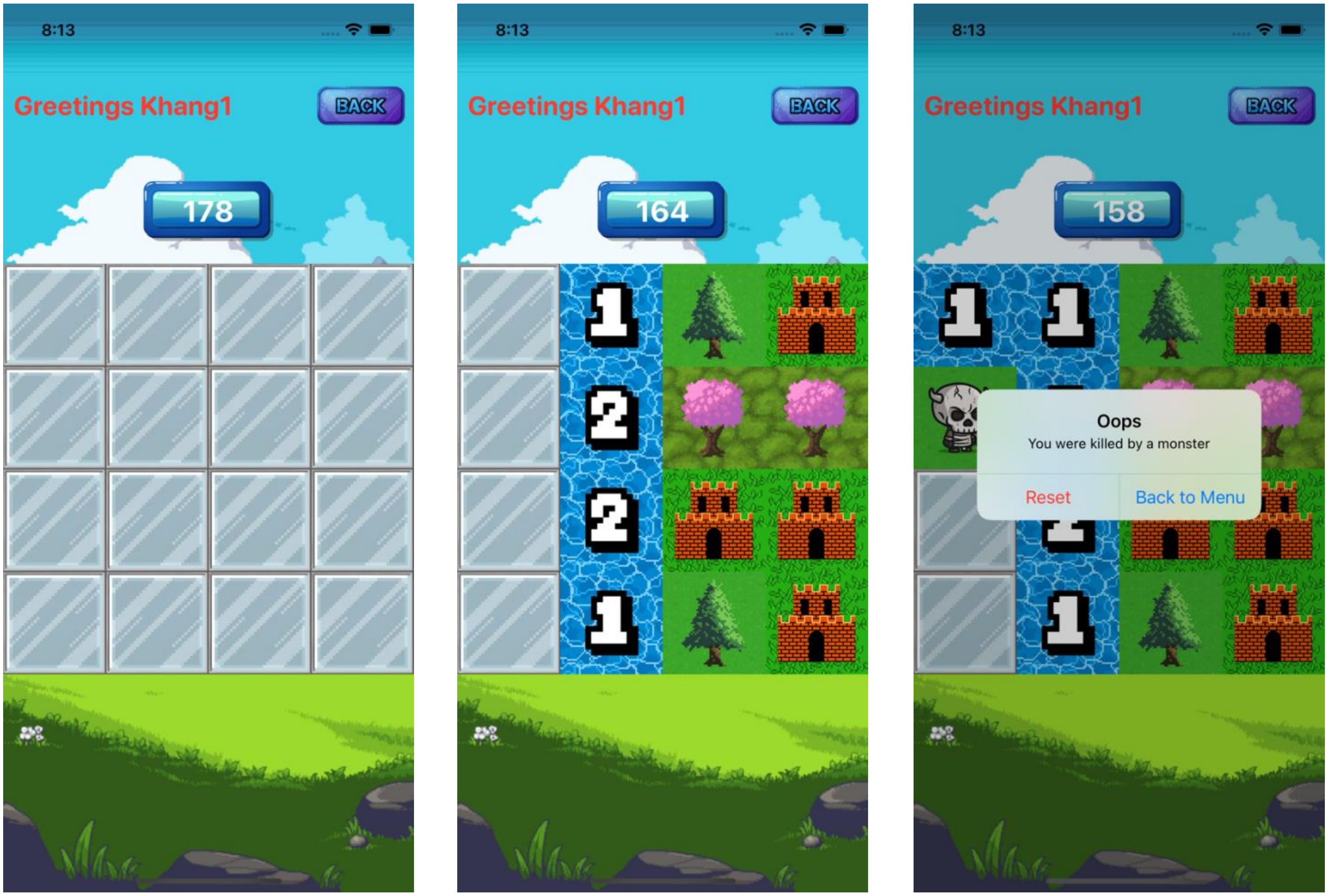
This name is already taken!

Khang

(Error duplicate username)



(Game difficulty view)



Oops

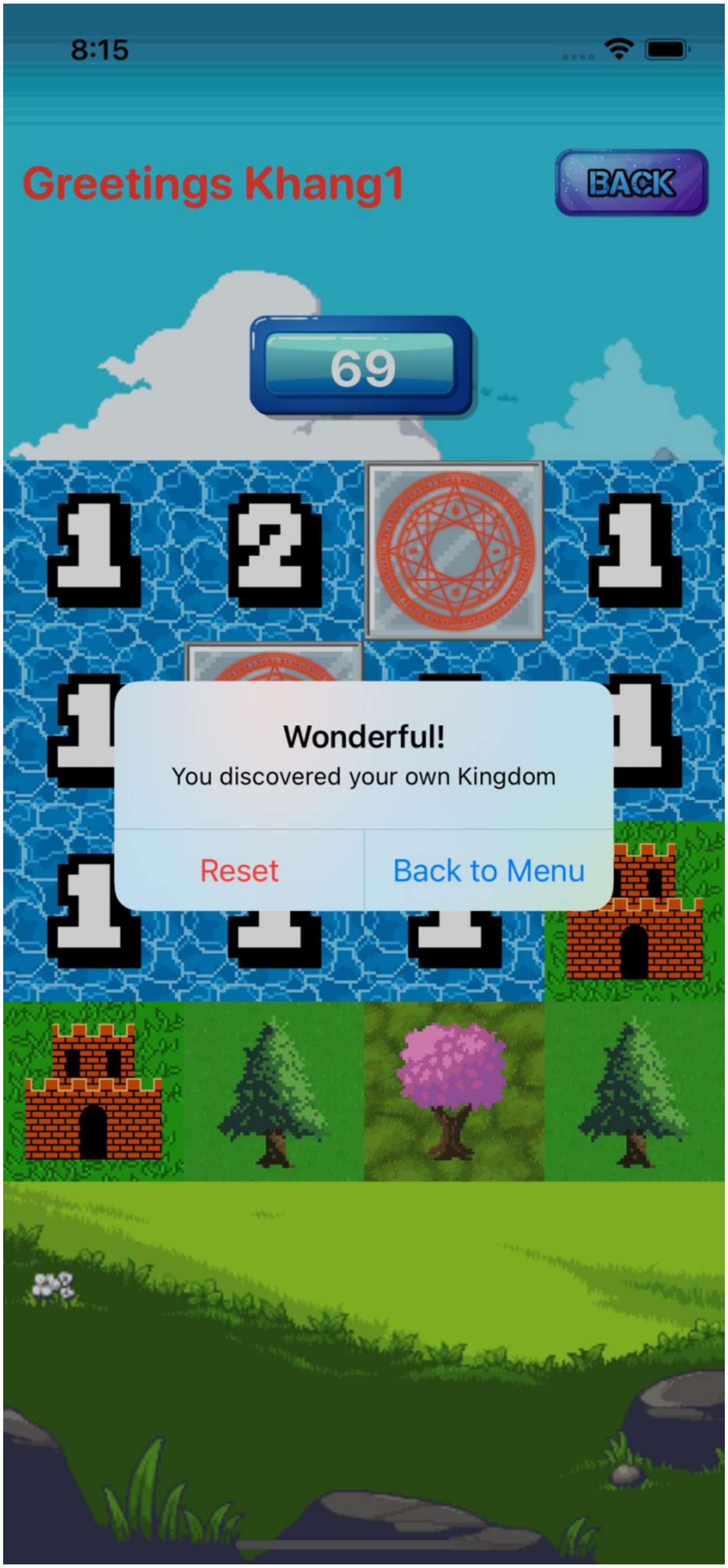
You were killed by a monster

Some gameplay View:

Pic1(from the left): Initial map

Pic2: In the middle of the game

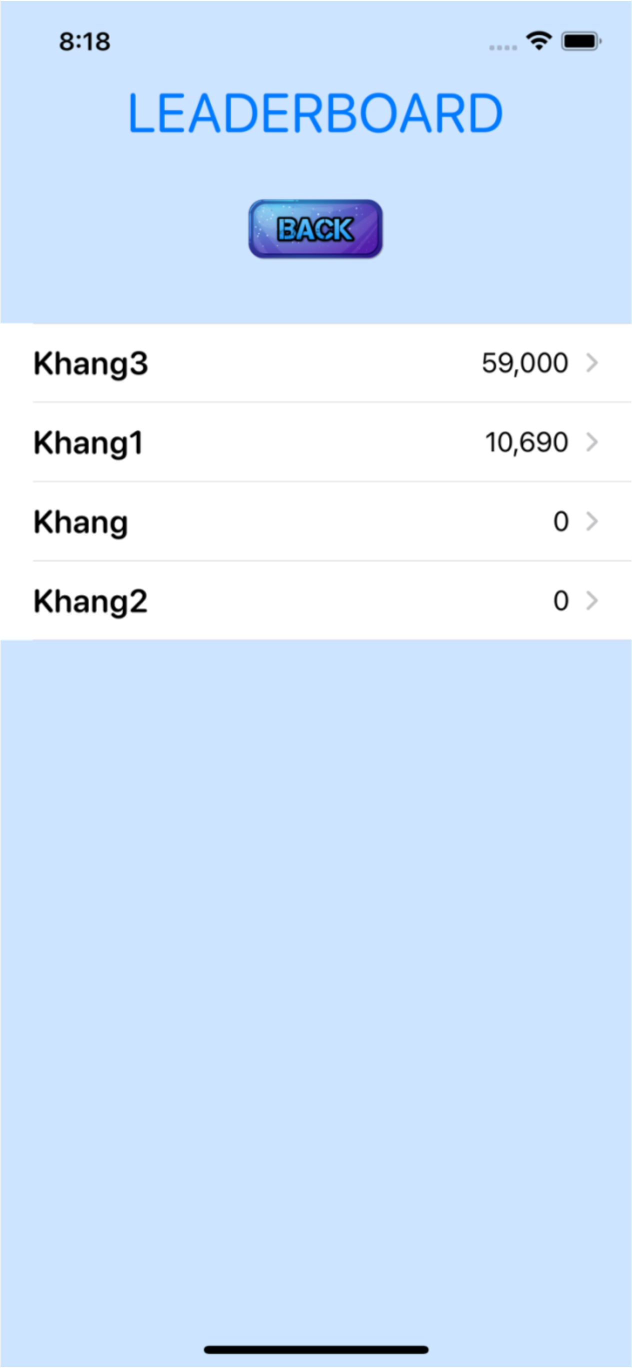
Pic3: Discover the monster => alert losing stage



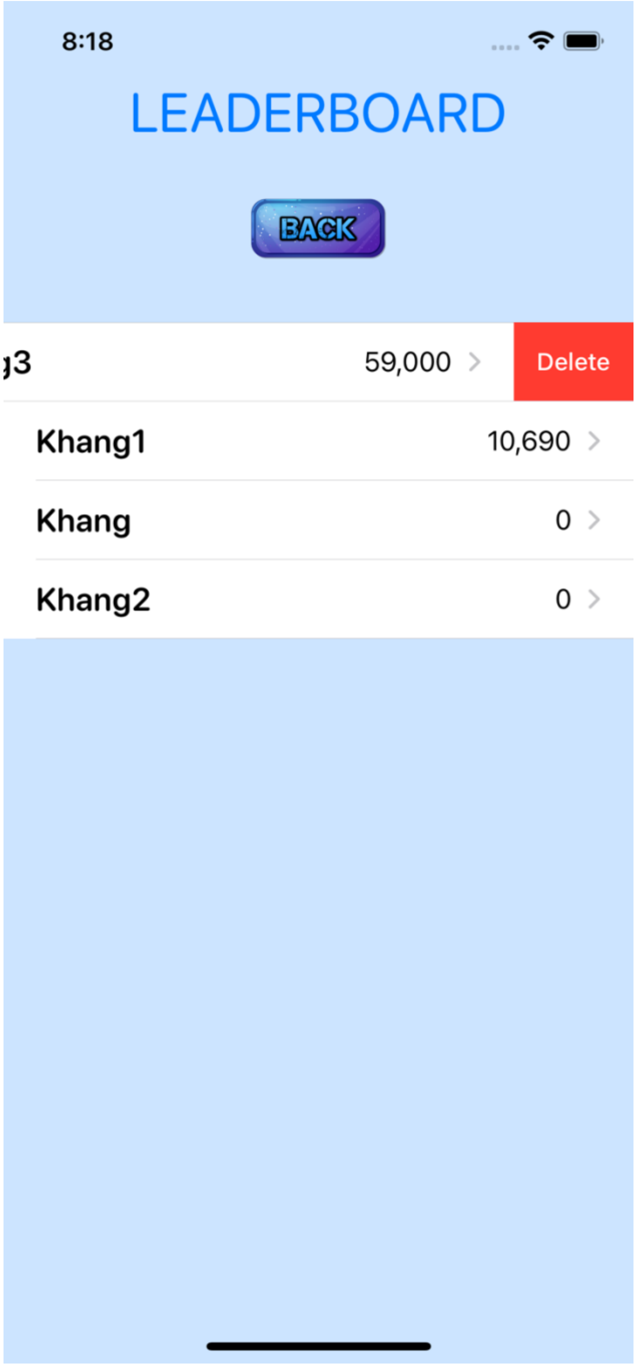
Wonderful!

You discovered your own Kingdom

(Winning stage: Finish discovering all cells without facing the monster)

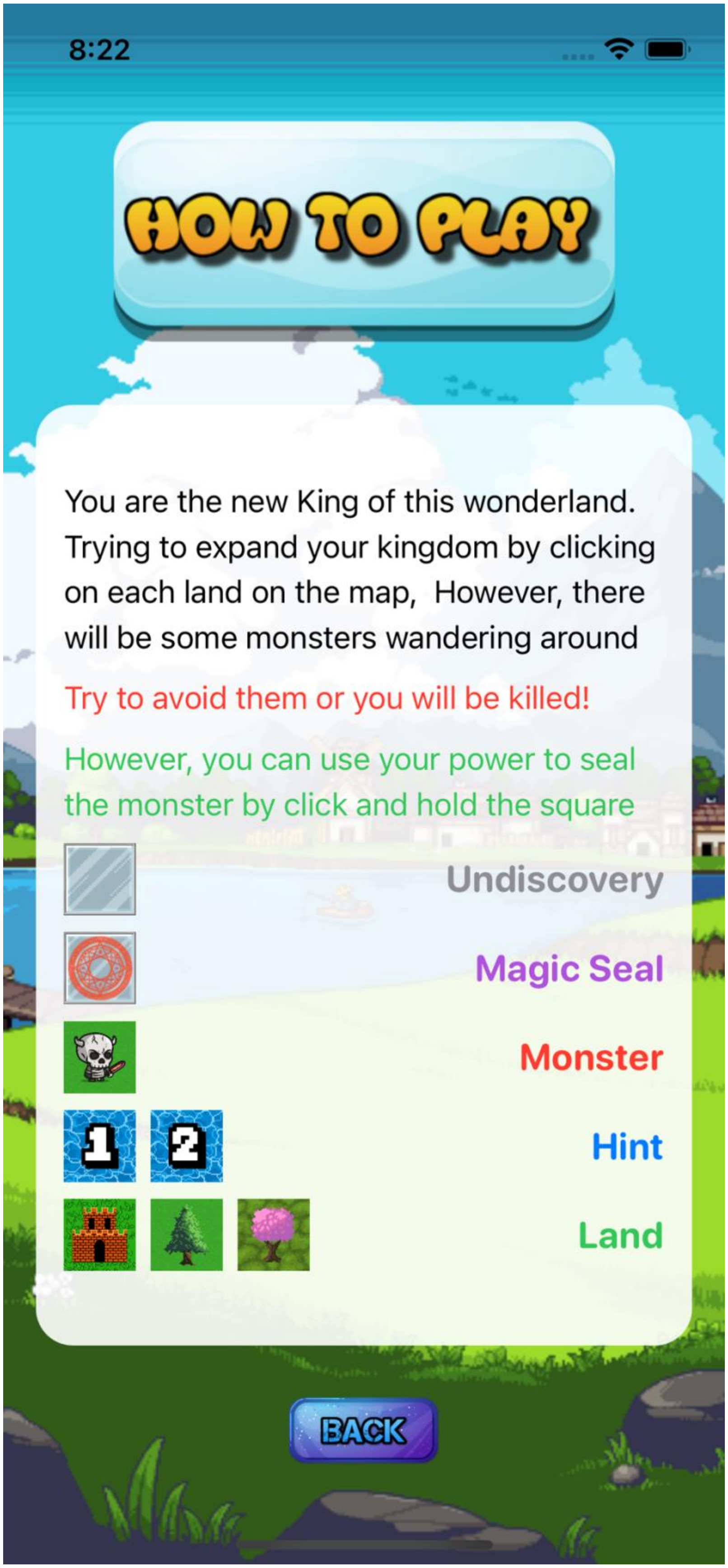


Khang3	59,000 >
Khang1	10,690 >
Khang	0 >
Khang2	0 >



;3	59,000 >
Khang1	10,690 >
Khang	0 >
Khang2	0 >

(Leaderboard View)



You are the new King of this wonderland. Trying to expand your kingdom by clicking on each land on the map, However, there will be some monsters wandering around

Try to avoid them or you will be killed!

However, you can use your power to seal the monster by click and hold the square

▨	Undiscovery
○	Magic Seal
☠	Monster
1 2	Hint
🟩 🟩 🟩	Land

(How-to-Play View)

5. VIDEO DEMONSTRATION

https://rmiteduau-my.sharepoint.com/personal/s3864131_rmit_edu_vn/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fs3864131%5Frmit%5Fedu%5Fvn%2FDocuments%2FIOS%2FAssignment2%5FDemoGif%2Egif&parent=%2Fpersonal%2Fs3864131%5Frmit%5Fedu%5Fvn%2FDocuments%2FIOS&ga=1

Video version: Assignment2_Demo_Video.mp4

6. FUTURE DEVELOPMENT

- Save the stage of the map to the player, next time when player start the app, they can continue the last stage in the game: I try to save the stage of the map which is the 2D array of class Cell to Core Data.
  - 1st Approach: Implement 2D array of class as Transformable attribute in Core Data
  - 2nd Approach: transform 2D array of class to JSON String then store in Core Data. After retrieving it, parsing JSON back to 2D array
  - 3rd Approach: implement Cell as one Entity in Core Data and it will have the One-to-Many relationship with Entity Player, but I am not sure if a player can have 2D array of Cell
- Make the map has more options in size of row and column. Currently, the size of the map and the number of monsters is static. The only different for three difficulty (Easy, Medium and Hard) is the time limit to finish exploring the map. In the future, I will implement the different way for three difficulties (the more difficult the game is, the bigger the map is)
- Add more animation effect for magic sealing or the animation that the monster will attack the player