**SPL-1 Project Report, 2020**

**Mammal Royale: A 2D strategy board game**

SE 305: Software Project Lab I

Submitted by

**Fahim Morshed**

BSSE Roll No.: 1102

BSSE Session: 2018-19

Supervised by

Dr. Md. Shariful Islam

Designation: Professor

Institute of Information Technology



A project submitted to

Institute of Information Technology

University of Dhaka

24 August, 2021

**Signature Page**

|  |  |
| --- | --- |
| **PROJECT:** | Mammal Royale: A 2D strategy board game |
|  |  |
| **AUTHOR:** | Fahim Morshed |
|  |  |
| **DATE SUBMITTED:** | 24 Aug, 2021 (2nd year 3rd Semester) |
|  |  |
| **SUPERVISED BY:** | Dr. Md. Shariful Islam  Designation: Professor  Institute of Information Technology |
| **SUPERVISOR’S APPROVAL:** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**Index**

1. **Introduction……………………………………………………01**
   1. **Objective……………………………………………………01**
   2. **Scope………………………………………………………..02**
   3. **Background Studies…………………………………….….02**
   4. **Challenges……………………………………………….….04**
2. **Project Description……………………………………………05**
   1. **Implementing the game loop………………………………05**
   2. **Implementing the game logics…..…………………………06**
3. **User Manual……………………………………………………08**
   1. **General Rules for all players………………………………14**
   2. **Rules for the DEER player………….……………….…….15**
   3. **Rules for the HUNTER player……….……………………15**
   4. **Rules for the TIGER player……………………………….16**
4. **Conclusion……………………………………………………...17**
5. **Appendix…………………….…………………………………17**
6. **Introduction**

In this project, I tried to build a 2D strategy board game inspired by the popular format of Ludo, “Snake and Ladders”. In Snake and Ladders, all the players have the same objectives, advantages and disadvantages. And winning the game is fully random and only depends on luck to get the ladders and avoid the mouths of the snakes. So, it can get pretty boring very quickly. In a summer vacation me and my younger brother tried to modify the game to include some strategical thinking that is required in the Classic Ludo. So, I implemented some new rules where different players have different objectives in the game. These players were a Deer, a Hunter and a Tiger. These players have different strengths and different objectives compared to the other players. I also implemented rivers with upstream and downstream current instead of the snakes and ladders. Instead of a fixed progression by a specific ladder or a fixed demotion by a specific snake, now a player can choose how much they want to progress or retract based on what is their objective and strategy. So, in this Project, I chose to make a digital version of that board game. Here I built an offline multi-player version of the game.

**1.1. Objective**

* Making an easy, fun and entertaining game.
* Using all original arts to do so.
* Making the code clean and easily understandable.
* Making the game smooth and functional at the same time.
* Making the code modification friendly for the future.

**1.2. Scope**

I only Intend to make an offline multiplayer version of this game which you can play with your friends in one device. Although, I wanted to make a single player version where the player could play against computer but I was restricted by my lack of knowledge and experience in this field. So, I removed the whole section from my project.

**1.3. Background Studies**

To implement this project some background studies was needed –

**JavaFX GUI**

As I was trying to build a game in Java, I had to choose from some of the most popular Graphical User Interface (GUI) for the language such as JavaFX, Abstract Widget Toolkit (AWT) or Java Swing. I chose JavaFX because of the abundance of resources that I found on this topic. As I was not allowed to use more than necessary library methods, I had to learn the basics of this GUI development kit.

**Game Loop**

As I was still completely inexperienced on the Game development field, I had to start learning from the basic concepts of game development. Game Loop is one of the main concepts of this field. Every game effectively runs on a basic loop where the game has to take input from the player, update the variables of the games, update the game world (in my case the game board) and take input again and continue. This loop is called a game loop. I had to learn from almost scratch of the concept. Although my game being a lot simpler in nature made my job a little easier.

**Multithreading**

Multithreading is a concept where we split the flow of the program into separate threads where they can run concurrently or sequentially. So basically, we split the program into different timelines where different actions can take place independently. In a game, a lot of stuffs have to happen concurrently to make it work. Sometime one method has to wait while the other one is running, sometime one action has to occur on regular intervals while the rest of the game has to go on. So here the concept of multithreading comes into really handy. As I had no prior experience of working on multiple threads, I had to learn the basic concepts of it applied on Java. Then I studied some more topics like sleeper threads and how to use that concept in JavaFX.

**Animation**

As no game is actually fun unless it has some animation in it, because everything seems abrupt and extremely hard to follow. So, I had to learn some basic animation to make the game a lot smoother. But being restricted from using the built-in animation methods of JavaFX, I had to study on ways to make my own animation methods. Here I drew the same image over and over again in a regular interval to make the transitions a lot smoother. In order to do so, I had studied the concepts of Key Frame and Timeline. I used the timeline to make a keyframe of the images I was trying to animate and showed them on succession or different positions in a regular interval to make it work.

**Digital Art**

As every game is dependent on a lot of graphics, I had to make some arts and other assets for the game. In order to do that, I learnt the basics of digital art using Adobe Illustrator and Photoshop.

**1.4. Challenges**

There is a lot of challenges that comes with game development, no matter how smaller the game is in terms of scope. My project is also no exception as I have faced many challenges while implementing the project. Here are some of them –

* Working on game development for the first time created some confusion as from where to jump in.
* Working with Graphical User Interface for the first time
* Working with Multithreading for the first time.
* Learning CSS commands to Customize the style of my components.
* Handling numerous variables, Lists and flags to keep track of the current game state and debugging the numerous bugs that came from them.
* Working out every single scenario of the game decision tree and covering them so that no unwanted bugs never appear.
* Animating the transition of each player from one point to another. Animating the rolling of the dice.
* Saving the game state while closing the game and later parsing the saved file to extract and replicating the same game state.
* Debugging any kind of bug, as I was not used to debugging any game bug and all sorts of new bugs that it creates.
* Being too ambitious at times as it led to spectacular failures and the mental fallout that comes with it.
* General unproductiveness while quarantining.

**2. Project Description**

My project can be broadly divided into two parts-

* **Implementing the game loop**
* **Implementing the logics, decision tree and game rules**

So, I am describing them into details bellow-

**2.1. Implementing the game loop**

As my game is a turn-based game, so the game loop I needed to implement had to depend on that too. The popular way of implementing a game loop is like this-

While (true){

tick(); //a method to take input and update the game variables

render(); //a method to render the updated game board

}

But as my game didn’t depend on high fps and there was going to be a lot of moment where nothing would happen, so instead of using an infinity loop which would be a lot more inefficient, I opted to use a version which is more dependent on the user inputs. The basic structure of my loop was like this-

deerPressed(){

if(isDeerTurn){

//updating variables for deer’s turn

//updating the game board for those updated variables

//unlocking the next players turn and locking deer’s turn

}

}

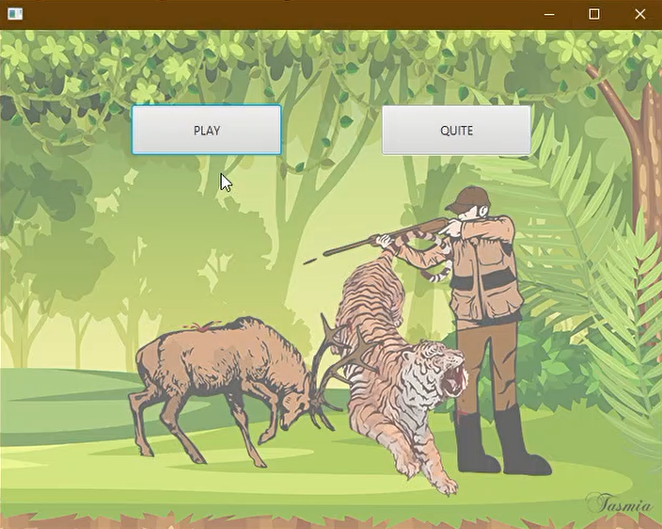
Here I used tick() and render() only when the user gave some input. I made similar methods for all the players depending on their own game rules and decision tree. Also, which player’s turn is going to be locked or unlocked was also dependent on which player was alive or dead or what the current player had done in this turn. This slight variation is still a loop because it works like **A – B – C – A – B – C**. This variation had saved me a lot of unnecessary looping and complexity.

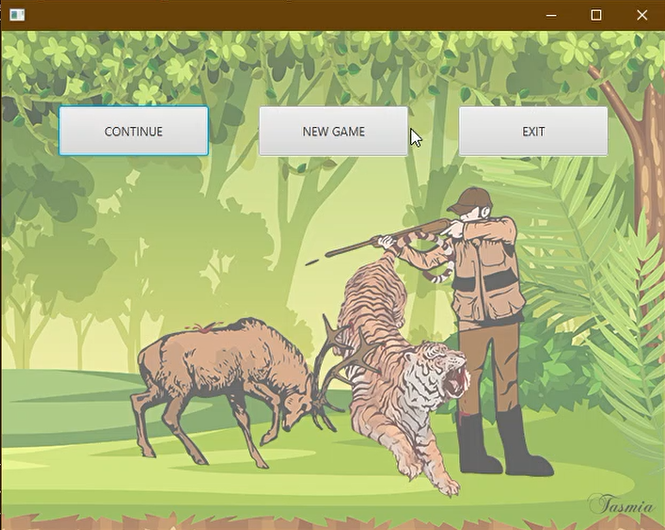
**2.2. Implementing the game logics**

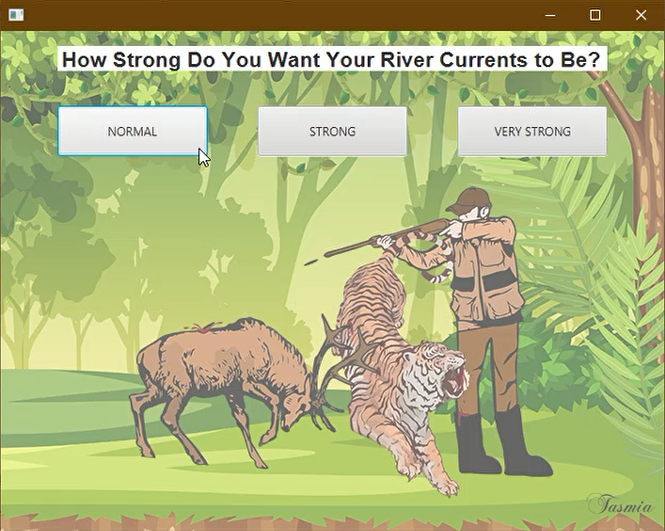
At first, I had to Implement the killing mechanism work out how the turn chain would react without that killed player. I had to check if any players kill, injured any other player or eat from any fruit trees. After that, I worked out all the possible outcomes of the games and check for them each turn to see if one of them had come true. If yes then we would stop end the game. The ending decisions of the game are-

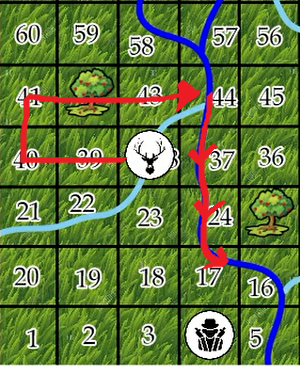
1. The DEER crosses 100th square having been fed, The DEER wins.
2. The DEER crosses 100th square having not been fed while the TIGER had eaten the HUNTER will result in the TIGER’s win.
3. The DEER crosses 100th square having not been fed while the TIGER is killed will result in a win for the HUNTER.
4. Both the HUNTER and the TIGER’s death will result in a win for the DEER.
5. The HUNTER crosses 100th square having been fed will result in a win for HUNTER.
6. The HUNTER crosses 100th square having not been fed while the TIGER had killed the DEER will result in a win for the TIGER.
7. The HUNTER crosses 100th square having not been fed while the TIGER is dead will result in a win for the DEER.
8. Both the TIGER and DEER being killed will result in a win for the HUNTER.
9. The TIGER crossing the 100th square while the DEER being killed will result in a win for the HUNTER.
10. The TIGER crossing the 100th square while the HUNTER being killed will result in a win for the DEER.
11. Both the DEER and HUNTER being killed will result in a win for the TIGER.

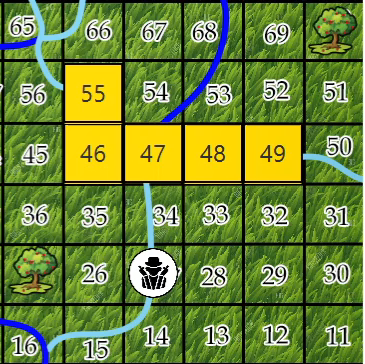
**3. User Manual**

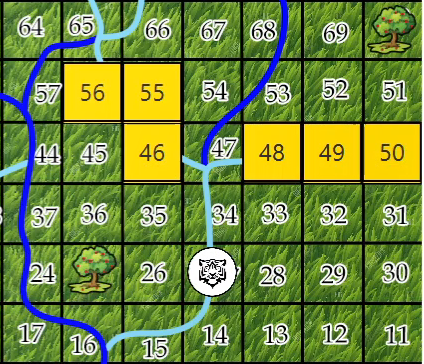
When launching this application, the user would get this window. Here the “PLAY” button will start the game while “QUITE” doing the opposite.

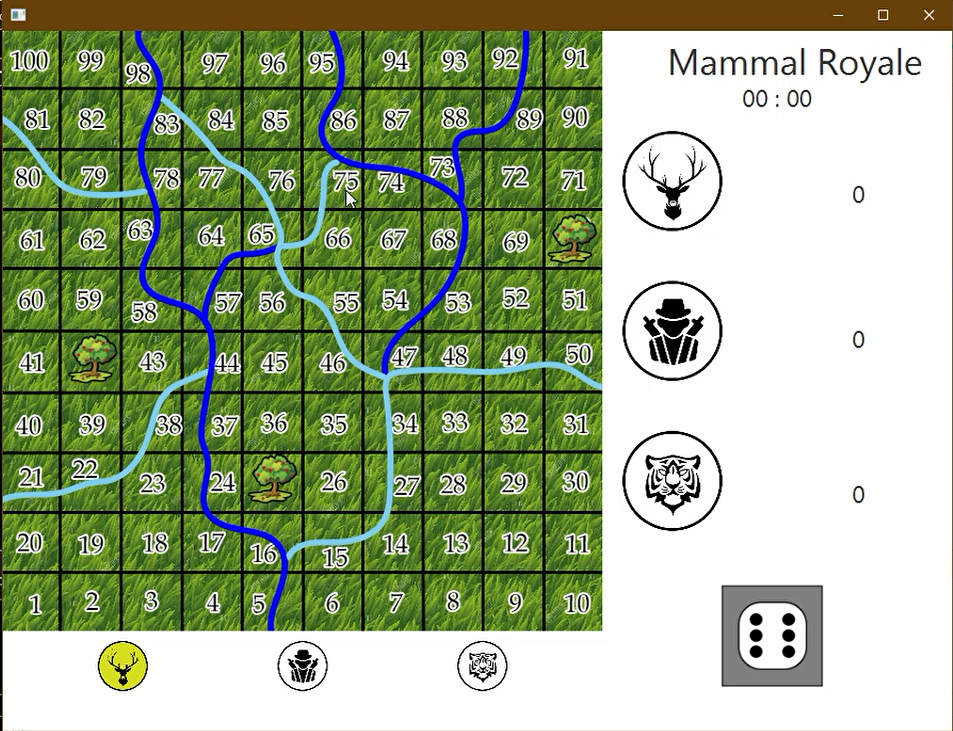
After deciding to play, the player can either choose to “CONTINUE” a previous saved game or decide to play a “NEW GAME”. He can also “EXIT” to the previous window.

After choosing to play a new game, the player has to select one of the three Levels or river current which are “NORMAL”, “STRONG” to “VERY STRONG”. These three types of strength of river current determines the game strategy of the players.

In the game with NORMAL current, the players when entering a square with rivers on can travel 1-3 squares following the river stream. This tends to make the players range smaller with a greater accuracy.

In the game with STRONG current, the players when entering a square with rivers on can travel 2-4 squares following the river stream.

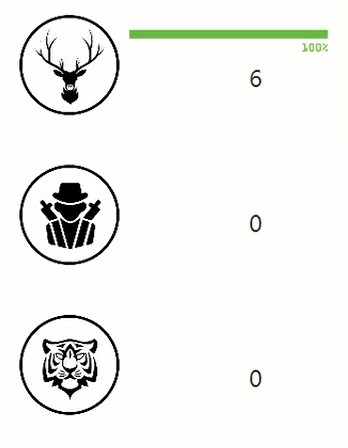
Similarly in a game with VERY STRONG current, the players can travel 3-5 square following the river stream making the game far more chaotic.

After selecting one of these three game mode, the user would be served with this game window-

Here in the bottom the three buttons are for each player.

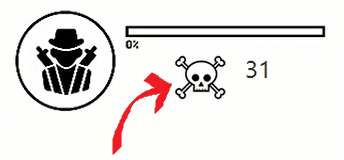
When it’s the turn of a player, the button associated with him will be lit yellow.

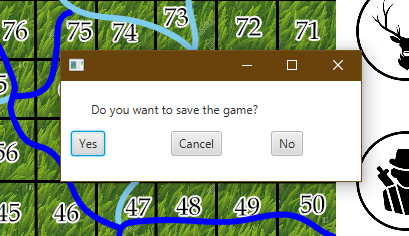
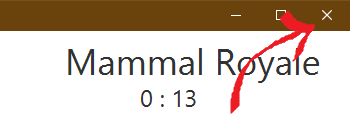
When one of the players would die, the button associated with him would be permanently turn grey and functionless.

The right side of the window will show each player’s state.

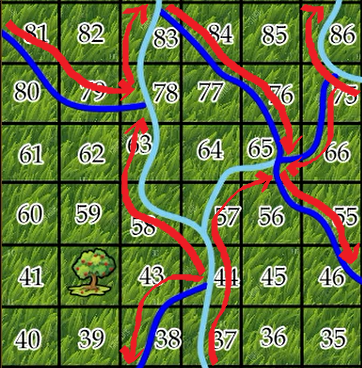
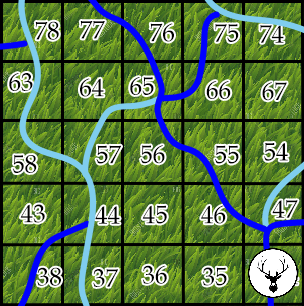
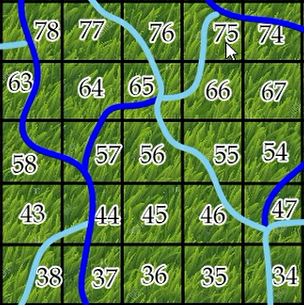
This is the health bar of the player and just right bellow is his position.

This sign will appear when the player has eaten.

This sign will appear after the player’s death.

The user can also choose to close the game with the cross button at top right. Upon doing so a prompt will appear asking whether he would like to save the game for later. Selecting “YES” will save the game and close the game. Selecting “NO” will not save the game but close the game while selecting to “CANCEL” will let him continue with the game.

**3.1. General Rules for all players:**

* Each player can enter the board only after they score a 6.
* When entering a square with rivers in it, each player has to choose one of the Golden Squares that is reachable from that river following the river stream depending on their objective and strategy.
* The rivers have two different colors signifying their different stream. The Light Blue colored rivers are upstream and the Deep Bluecolored rivers are downstream river.
* Each 30 seconds, the river currents will reverse, forcing the players to adapt quickly to the new current pattern.

* After killing or injuring another player, the player will have another turn on their hand as a reward to get ahead.

**3.2. Rules for the DEER player:**

* **Goals:**
  + The DEER has to eat one of the three fruits on the board.
  + After eating, it has to escape the jungle or in other words has to cross the 100th square.
* **Disadvantages:**
  + He has to reach the fruit trees before the HUNTER player.
  + Both HUNTER and TIGER can kill him in one hit, so he has to stay clear off their path.
* **Powers:**
  + He can injure both the HUNTER and the TIGER.
  + He can kill an injured HUNTER and TIGER.

**3.3. Rules for the HUNTER player:**

* **Goals:**
  + The HUNTER has to either eat one of the three fruits or the DEER.
  + After eating, he has to escape the jungle or in other words cross the 100th square.
* **Disadvantages:**
  + He has to reach the fruit trees before the DEER player.
  + The TIGER can kill him in one hit. The DEER can injure him and kill him if he is already injured.
* **Powers:**
  + He can injure the TIGER and kill him if he is already injured.
  + He can Kill the DEER in one hit.

**3.4. Rules for the TIGER player:**

* **Goals:**
  + The TIGER has to prevent the other players from achieving their goals by killing both of them.
* **Disadvantages:**
  + He cannot cross the 100th square. If he does, he would be considered dead.
  + He can be injured by both the DEER and the HUNTER.
  + He can be killed by both the DEER and the HUNTER if is already injured.
* **Powers:**
  + He can kill both the DEER and the HUNTER in one hit.

**4. Conclusion**

This project helped me the world of game development for the first time which was a dream of mine for a long time. Learning the basics of different game developing principles and trying to implement them in my game was a challenging as well as adventurous endeavor. Multithreading made me pull off my hair some times. But as I got used to, it made me more confident and prepared me for new challenges. The more I worked on the project, the farthest the finish line appeared to me. But as I kept pushing, I became more and more confident about my skills and knowledge. When I tried to add animation on my project the amount of exception it was throwing was unprecedented for me. It had improved my patience and persistence greatly. Every time I tried to draw something it seemed all to be just beyond my artistic capability. But significant practicing had paid off and I was able to produce some decent artworks. I never worked on a project of this magnitude which taught me the importance of breaking a bigger project into smaller manageable steps. I have learnt a great deal of basic about Object Oriented Concepts and JavaFX GUI. I learnt to parse a file and pick needed values by manipulating or just reading a string. It was a great experience and I cannot be more thankful to my supervisor who constantly helped me wherever I struggled and led my path. I really look forward to using this experience in future projects and programming works.

**5. Appendix**

I got introduced with the Unity game engine and building 3D open world games which I look forward to work on in the future.