CMPT 276: Phase 4 Report

Efe Erhan

Brianna Espena

Ruochun (Jasmine) Liu

Moon Wang

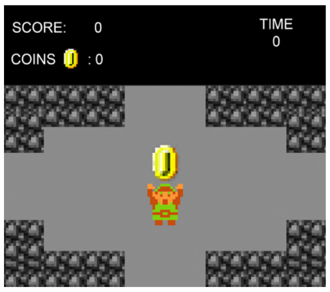
The Game: Mr. Green's Adventure

Our game is called Mr.Green's adventure. It is a multi-room maze game that features four different rooms/levels each with varying difficulty and layout. In each of the four rooms there will be barriers, moving enemies, non-animated enemies, and rewards. The player is able to access the different rooms through the doorways. The goal of the game is to move the main character, Mr. Green, around each room collecting all of the rewards and avoiding the enemies. The game will automatically end if the character touches a moving enemy or if their score count drops below 0. If the player is able to collect all the regular rewards in all four rooms and retain a score of 0 or greater then they have successfully completed the game and won.

How the final product varied from our original plan and design:

About Design: What we ended up with is really close to our design in terms of how it looks,.

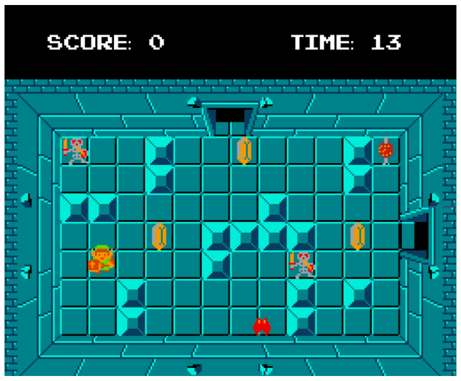
Mock-up User Interface:



SCORE: 0
TIME 347
COINS : 0

SCORE: 0
TIME 0
COINS : 0
GAME OVER

Our Final Product:



<u>Functionality:</u>

- We wanted the player to be able to pause the game, but we did not implement this.
- The original plan is to use “wasd” keys to control the player but finally “up down left right” keys are used.
- If the time is paused, the player will also lose the game. However, we can’t decide which time escape is better for the player to play the game, so we did not implement it.
- We misunderstood the assignment originally, and believed that each room in the maze would not be fully navigable, or have sub-mazes inside them. Upon re-reading, we fleshed it out to reach the scope of the actual requirements.

<u>UML design:</u>

- We planned to have a class called “position” to contain the position info of each base element of the game, such as barriers and the player. Later, we figured out that it would make coding complicated, so we deleted it. Instead, we let the base elements inherit JComponent directly because the JComponent object can provide its position via getX() and getY() methods.
- The class “TimeTracker” directly used the Timer object. So We can directly use the “start()” and “stop()” methods in the object and don’t need to write these methods by ourselves.
- We do not track the bonus score penalties in the “ScoreTracker” class as shown in UML. We track two variables “marks” and “realMarks” in our final project. “Marks” is the marks the player gets after the player encounters a penalty, bonus, and regular rewards. “realMarks” is the marks count for regular rewards. When “realMarks” is equal to a specified number (17), the player collects all regular rewards and wins the game.

The most important lessons we have learned throughout the creation of our game are that the UML is only considered the model part of the game and that you do need to follow the original plan perfectly. For the implementation, we were more focused on the view part, so there were lots of differences between UML and our final project. Although we created a detailed plan for designing and implementing, we made several changes throughout the process that made the overall quality of our code and game better.

Video Link:
https://drive.google.com/file/d/1h_825hc3hRr1vR8n3COf_YIuw3bJK6hs/view?usp=sharing

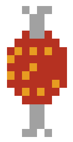
<u>Tutorial</u>

- The objective of the game is to keep the score at 0 or higher, avoid the moving enemies, and to collect every regular reward in each room
- The player is controlling the game’s main character, **Mr. Green** (pictured on left).

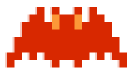
- When the game starts Mr. Green will appear in the bottom left corner of the first maze room
- The player uses their left, right, up, and down arrow keys to move Mr. Green around the board

- The game has 4 different rooms with differing layout and difficulty
- There are 6 things that the player may encounter in a room of the game

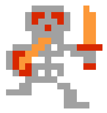
1. Regular Rewards (coin)
   a. All regular rewards in every room must be collected in order to win
   b. Adds 1 point to your score



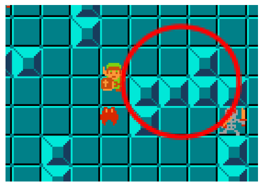
2. Bonus Rewards (meat)
   a. Adds 3 points to your score
   b. Does not need to be collected to win the game



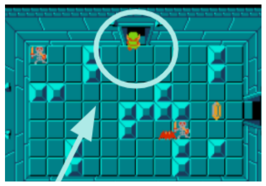
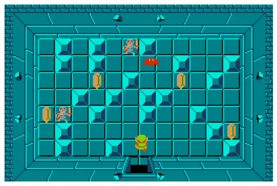
3. Moving Enemy (red bat)
   a. The red bats move around the board and if the main character touches a red bat, then the game automatically ends



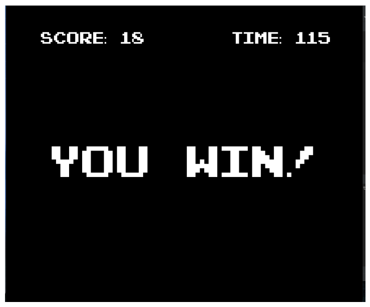
4. Non-Animated Enemy (skeleton)
   a. If the main character touches a skeleton, then their score decreases by 1 point



5. Barriers/Walls
   a. The character cannot move through the barriers but must instead find a path around them to get to their location
6. Doorways
   a. To enter another room, the player must go through a doorway of the room they are currently in

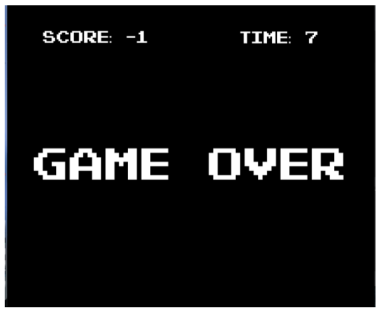



- There are three ways for the game to end
  1. Collect all rewards, have a score of 0 or greater and win the game
  2. Encounter more non-animated enemies than rewards and have a score below 0.
  3. Encounter a moving enemy and lose.



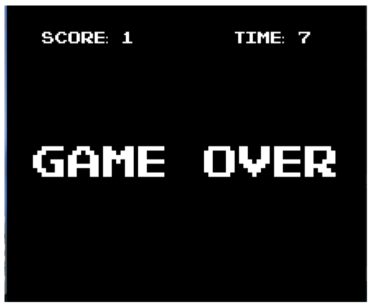
SCORE: 18 TIME: 115

YOU WIN!



SCORE: -1 TIME: 7

GAME OVER



SCORE: 1 TIME: 7

GAME OVER