

CS4303 P1 - Report

190022658

March 17, 2023

1 Overview

In this practical we had to implement a Rogue-like game in Java using Processing. In this game, the player needs to finish a total of three levels comprising of a dungeon each. A dungeon here has multiple rooms containing items and monsters. The player can gain weapons, gain skills and use other items. They also need to fight or avoid enemies while going to the last room and going to the next level.

2 Design

2.1 Dungeon

The dungeon room creation has been specified in the implementation. Then we store all the the rooms in the Dungeon class. We then create corridors for them. The dungeon also creates Monster and Items depending on the level of dungeon.

2.2 Characters and Movement

The game has a player and monsters. The player moves using the arrow keys or 'wasd'. The player can only stay in the playable area, which are the rooms and corridors. There is a collision detection mechanism to ensure this. The same algorithm also checks for collision with items and monsters which have their respective behaviour specified. The monsters' default movement is a wandering movement and when the player enters the monsters' room, the movement changes to kinematic seek. The monsters also have collision detection which prevent them from moving past walls. On collision with monsters the player engage in text combat.

2.3 Character Attributes

The player and monsters are subclass of the Combatant superclass. It has the following attributes:

- HP. It is their health.
- Attack. Determines how much the attack damages the other combatant.
- Defence. This is a barrier which defends your health.
- Speed. It is their movement speed.

2.4 Treasure, Items and equipment

The following are available for use by the player:

- Potion. Gain a certain amount of hp.
- Poison. Lose a certain amount of hp.
- Sword. Gain skills and attack by collecting a certain number of swords.
- Shield. Gain defense.
- Gate. You have completed the level and qualify for the next one.

2.5 Combat

The monsters and player have a number of battle skills available that they can learn. For example, the level 0 monsters only learn Growl and Scratch and so on. The monsters also have a ratio of the skill used. The players can learn skills by picking up Swords like Cut, Slash and Three sword style. There are no misses in attacks here. Level increase also improves monsters' speed.

- Damage. An attack gives the following damage - $\text{skill.damage} * (1 + \text{attack}/100)$

2.6 UI

The UI uses a titlemap from [<https://www.kenney.nl/>]. The top left part of the screen contains the hp bar of the user and when in combat, the hp bar of the monster too. The defence and attack are hidden stats and are therefore not shown anywhere.

3 Implementation

3.1 Procedural Generation

The dungeon has been procedurally generated using the `DungeonArea` which is a BSP tree node and splits into two nodes of itself and splits until a satisfactory size has been achieved. It then generates rooms in a somewhat random manner.

When splitting, first we randomly choose if we want to split the area horizontally or vertically. Also take into consideration the ratio of width and height of the current split. We then choose a random value with a certain amount of padding to create a split there. When the size of split is too small to house any more splits, we create a room there.

3.2 AI

Here we have chances of the monster using some type of attacks depending on their levels. For example it uses less effective skills when it is level one, and so on. They also have improved speed and attack. There is also a simple movement implemented.

4 Screenshots

4.1 Start screen

4.2 Dungeon

4.3 Fight screens

4.4 Game over

4.5 Win screen

5 Conclusion

Given more time, I would have created a more sophisticated AI and added more movements for monsters. I would also add a better combat system. I would also find a better way to create dungeon corridors. The code would also be neater.

NotRogue

Move using 'wasd'
Press space to start

Figure 1: Start screen

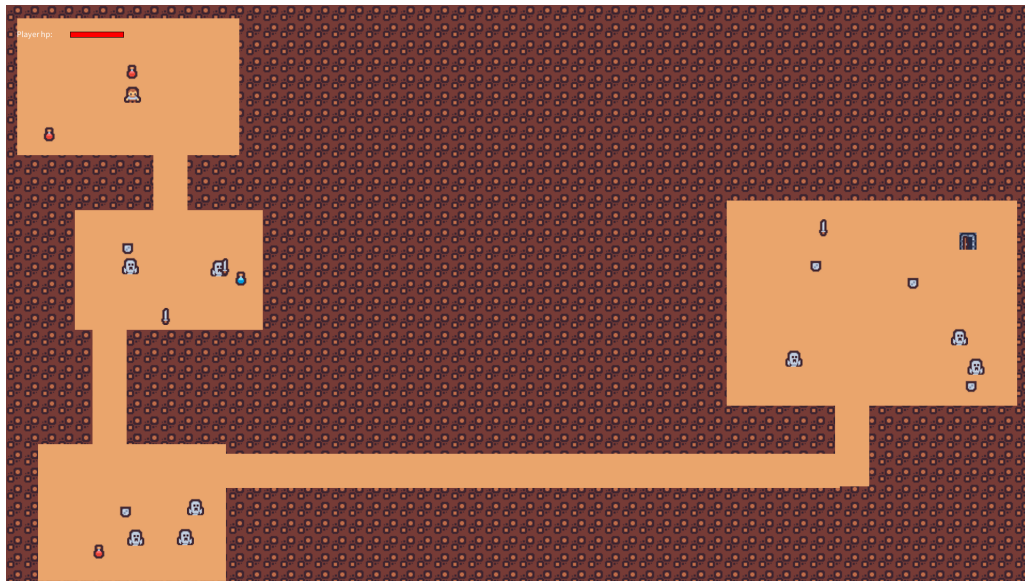


Figure 2: Dungeon

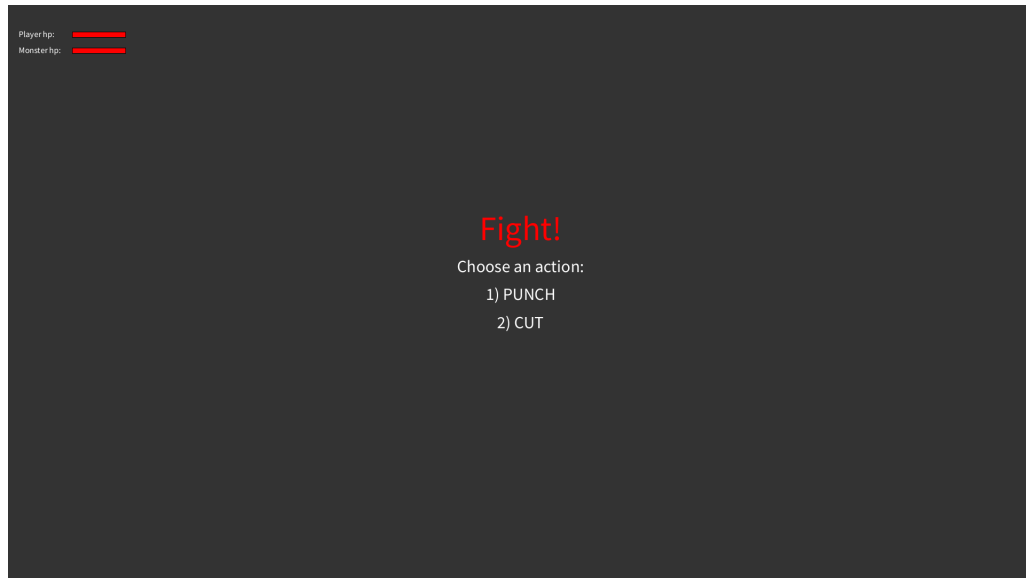


Figure 3: Fight intro

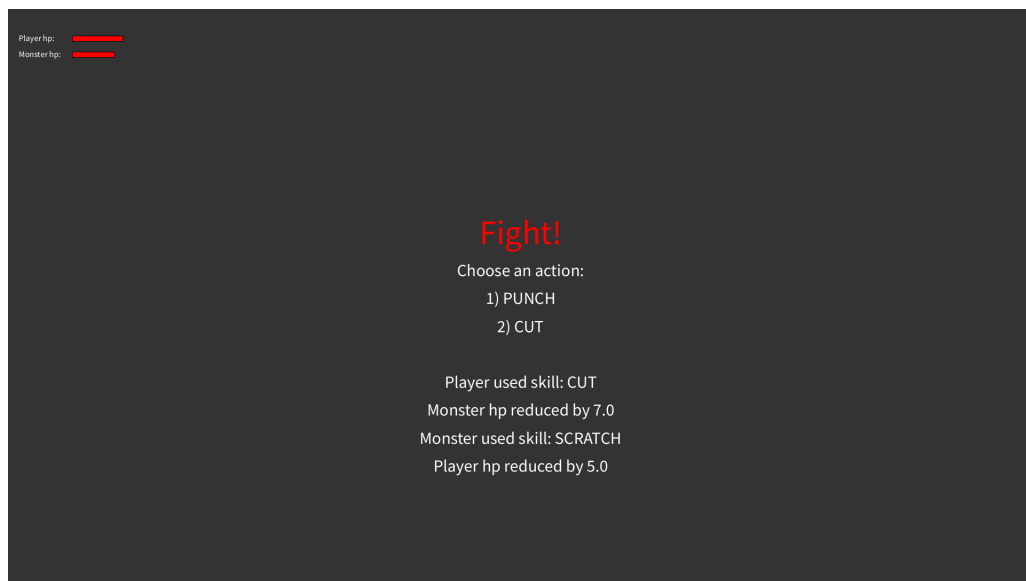


Figure 4: Fight screen

Game Over!

Figure 5: Game over

You win!

Figure 6: Win