CS 246 Design Document

Assignment 5 (CC3K)

Susu Dong & Xiner Ning

Table of Contents

Introduction

This document outlines the thoughts on design and architecture of the CC3K program put forth by both team members of cs246_118; Susu Dong(s2dong) and Xiner Ning (x4ning) for CS 246 (Object-Oriented Software Development) Assignment 5 Group Project.

Project – Plan vs. Actual

There were several changes between the initial designs of the project versus the project after completion. The team's statement of intent was indeed fulfilled as all the basic features outlined in the assignment specification were successfully implemented. Also, during game time, player has the chance to cheat 3 times to help win the game by setting his HP back to maximum. Player will then either die or manually quit the game or win the game by reaching the 8th floor. However, when it came to other extra features for bonus marks, both members felt overambitious on some aspects. Also, some changes were made upon the initial design to allow for the easiness of the implementation.

The specific differences in the UML are in:

1. gameObjects

- playerCharacter
- Enemy
    - split setCoordsAndGrid to setCoords and setGrid (trivial)
    - make getType non-virtual (not necessary)
    - remove the move() method (implement this function in the Grid class)
    - add the attack(playerCharacter*) method
    - for the subclass Dragon, add setTreasureLocation(Tile *) method for the convenience of notifying dragon that PC is near the dragonHorde
- Item
    - add a class Item from which Potion and Treasure inherits (so that duplicate code can be reduced in the generation of Potion and Treasure)
    - make it inherits from gameObject
- Potion
    - remove decorator to playerCharacter and all of its subclass (fail to adopt decorator pattern for the implementation of potions which have temporary effect)
    - make usePotion virtual (so different effect of potions is implemented)
    - add getType and getRace method
- Treasure
    - add getType and getRace method

- for the subclass dragonHorde, make it inherits from Tile rather than Dragon(trivial)

- floorElem
  - add a class Item from which hWall, vWall, Passage, Door, Ground, Stairway inherits, and it inherits from gameObject
  - add getType and getRace method
- Factory
  - add a class Factory which encapsulates the generation of different kinds of gameObjects
  - Grid owns-a factory

2. Floor

- Grid
  - remove setPotions and setTreasure method (unnecessary)
  - make initFloor pass in a char for the convenience of using command-line option

3. combat

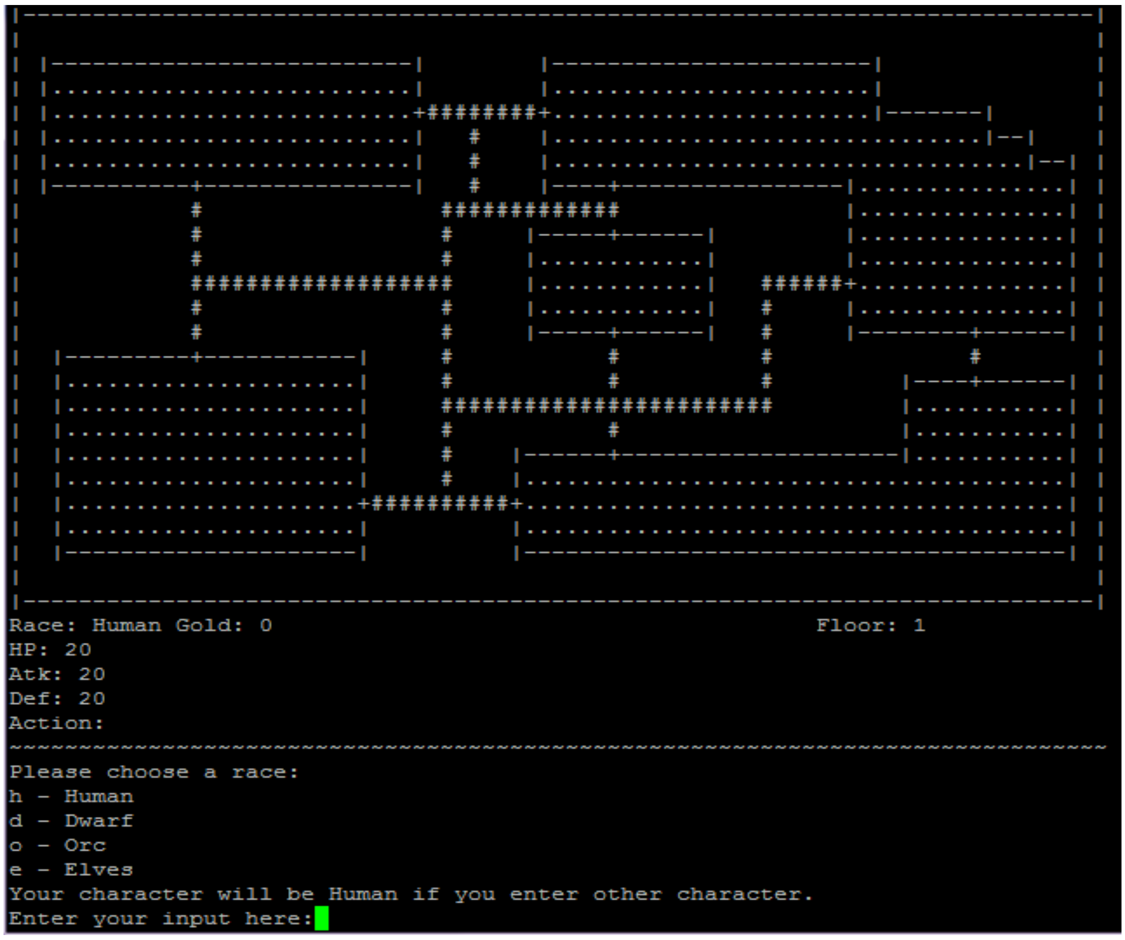
- Mediator
  - removes combatMediator class which inherits from playerCharacter and Enemy
  - add a Mediator class which inherits from Grid
  - Mediator has only combat method (now it just needs to take in attack and defend values of playerCharacter and Enemy)

Assignment Specific Approach

- Floor setup.

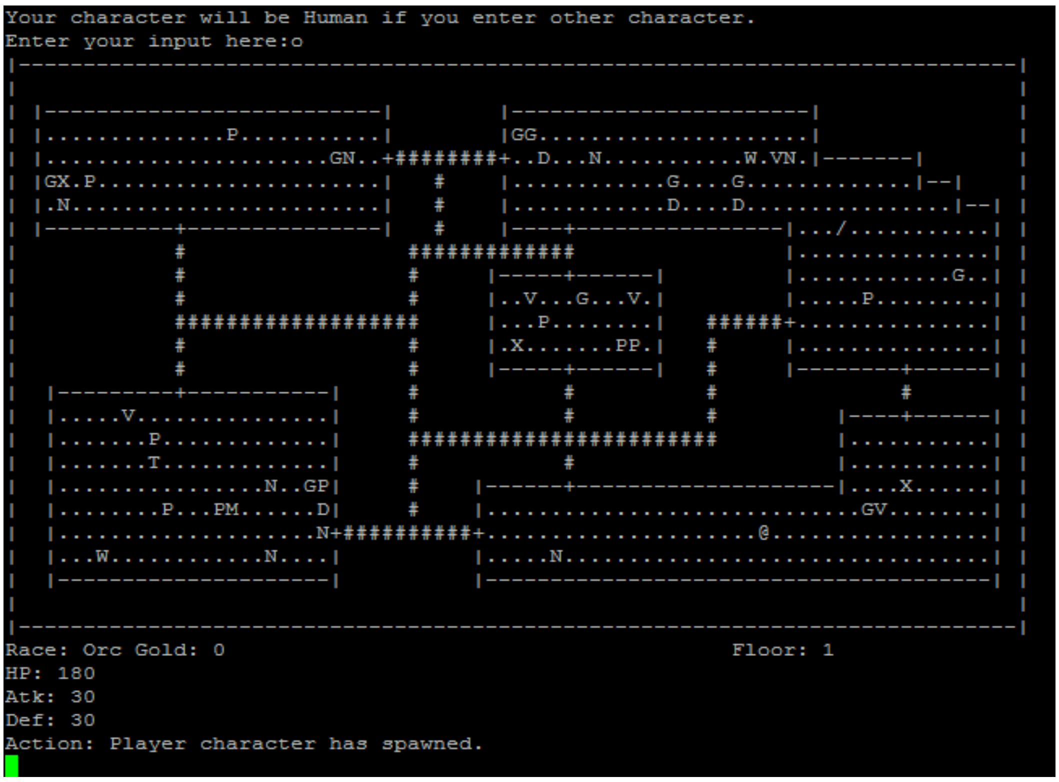
1) We have our game setup based on a class called "Grid". It is basically a m x n grid with each cell an object of class "Tile". Every "Tile" has a private file called "thing" which is a pointer points to an object of "gameObject" where "gameObject" is the base class of every game element class or base class of more specific game elements in this game. For example, "gameObject" is the base class of player character and class "Item" where "Item" is the base class of class "potion" and class "treasure".
2) We have managed to print out the basic pattern floor in front of player at the beginning of the game and messages informing the player to enter the desired race.
   See figure 1.
3) When player enters a command, the corresponding player character will be generated and every other elements will be generated and shown in the floor on the screen. The basic and key point to randomly generates elements is by using function srand() and rand() where srand() reset the rand() and rand() randomly generates a number in a specific range which represents a specific enemy or item("potion", "treasure"). This number will be parsed and a corresponding string (eg. "Troll") will be passed to an a method of an object of class "Factory" and "Factory" takes the responsibility to generate different types of enemies or items and return a pointer of that. After the generation, the result will be notified to an object "TextDisplay" , composed with the "Grid",   and "TextDisplay will update its record. For here, the design patter "Observer pattern" is used where "Grid" is the subject which contains m x n Tiles and "TextDisplay" is the "Observer".

   (Note: There is a special case when generate dragon. It is generated if and only if a dragonHorde is generated. Thus, when a dragonHorde is randomly created on the floor, the 8 positions around it will be checked to see it is alreay occupied(if PC or stairway is generated in these 8 positions). We will randomly choose a position by the use of rand() function by taking the randomly generated number mod of 8 to have a number between 0 and 7(inclusive). Each of them represent a specific position. After picking up a position and detecting its occupation, an enemy dragon will be spawned. The dragonHorde will have a "Tile" pointer which points to this dragon.)
   See figure 2.



```
|-----------------------------------------------------------------------------|
|                                                                             |
| |--------------------------|        |-----------------------|               |
| |..........................|        |.......................|               |
| |..........................+########+.......................|-------|       |
| |..........................|   #    |...............................|--|    |
| |..........................|   #    |..................................|--| |
| |----------+---------------|   #    |----+----------------|...............| |
|            #                 #############                |...............| |
|            #                 #     |-----+------|          |...............| |
|            #                 #     |............|          |...............| |
|            ###################     |............|   ######+...............| |
|            #                 #     |............|   #     |...............| |
|            #                 #     |-----+------|   #     |--------+------| |
|  |---------+-----------|     #           #          #              #        |
|  |.....................|     #           #          #         |----+------| |
|  |.....................|     ########################         |...........| |
|  |.....................|     #           #                    |...........| |
|  |.....................|     #    |------+--------------------|...........| |
|  |.....................|     #    |.......................................| |
|  |.....................+##########+.......................................| |
|  |.....................|          |.......................................| |
|  |---------------------|          |---------------------------------------| |
|                                                                             |
|-----------------------------------------------------------------------------|
Race: Human Gold: 0                                          Floor: 1
HP: 20
Atk: 20
Def: 20
Action:
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Please choose a race:
h - Human
d - Dwarf
o - Orc
e - Elves
Your character will be Human if you enter other character.
Enter your input here:
```

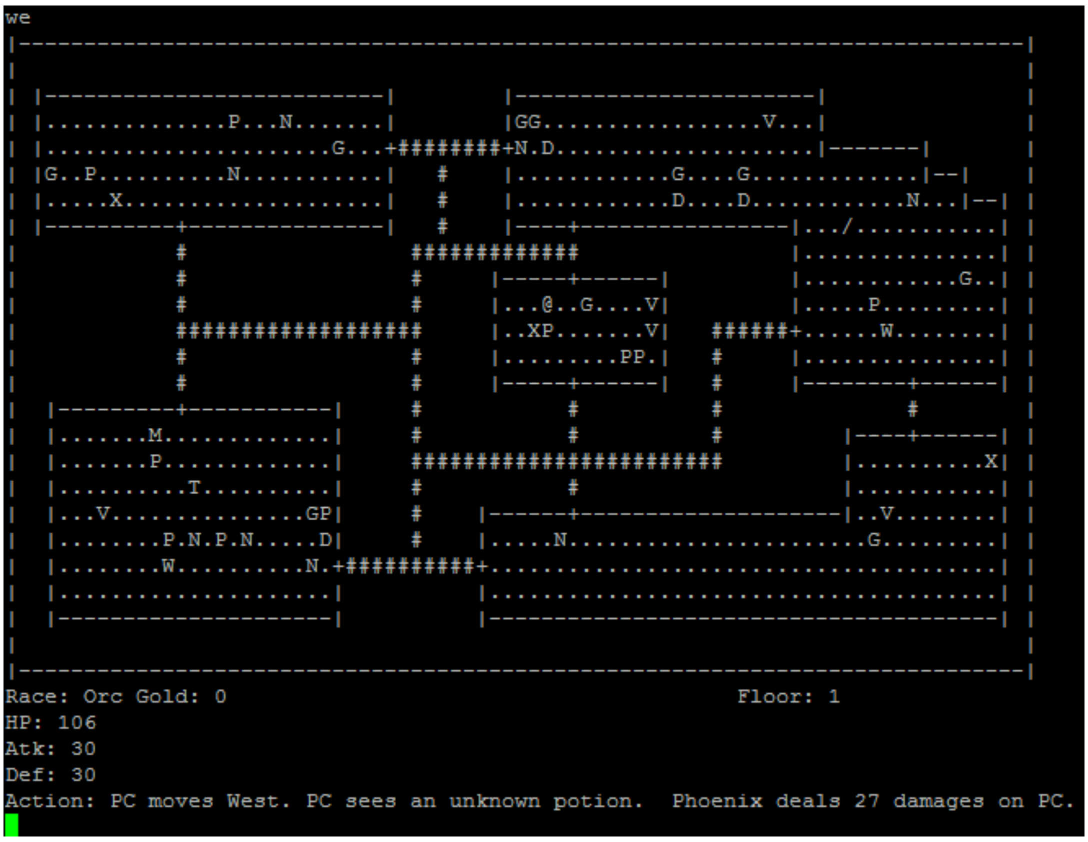
figure 1



```
Your character will be Human if you enter other character.
Enter your input here:o
|-----------------------------------------------------------------------------|
|                                                                             |
| |--------------------------|        |-----------------------|               |
| |..............P...........|        |GG.....................|               |
| |......................GN..+########+..D...N...........W.VN.|-------|       |
| |GX.P......................|   #    |............G....G.............|--|    |
| |.N........................|   #    |............D....D................|--| |
| |----------+---------------|   #    |----+----------------|.../...........| |
|            #                 #############                |...............| |
|            #                 #     |-----+------|          |............G..| |
|            #                 #     |..V...G...V.|          |.....P.........| |
|            ###################     |...P........|   ######+...............| |
|            #                 #     |.X.......PP.|   #     |...............| |
|            #                 #     |-----+------|   #     |--------+------| |
|  |---------+-----------|     #           #          #              #        |
|  |.....V...............|     #           #          #         |----+------| |
|  |.......P.............|     ########################         |...........| |
|  |.......T.............|     #           #                    |...........| |
|  |................N..GP|     #    |------+--------------------|....X......| |
|  |........P...PM......D|     #    |.............................GV........| |
|  |....................N+##########+.....................@.................| |
|  |...W............N....|          |.....N.................................| |
|  |---------------------|          |---------------------------------------| |
|                                                                             |
|-----------------------------------------------------------------------------|
Race: Orc Gold: 0                                            Floor: 1
HP: 180
Atk: 30
Def: 30
Action: Player character has spawned.
```

- figure 2

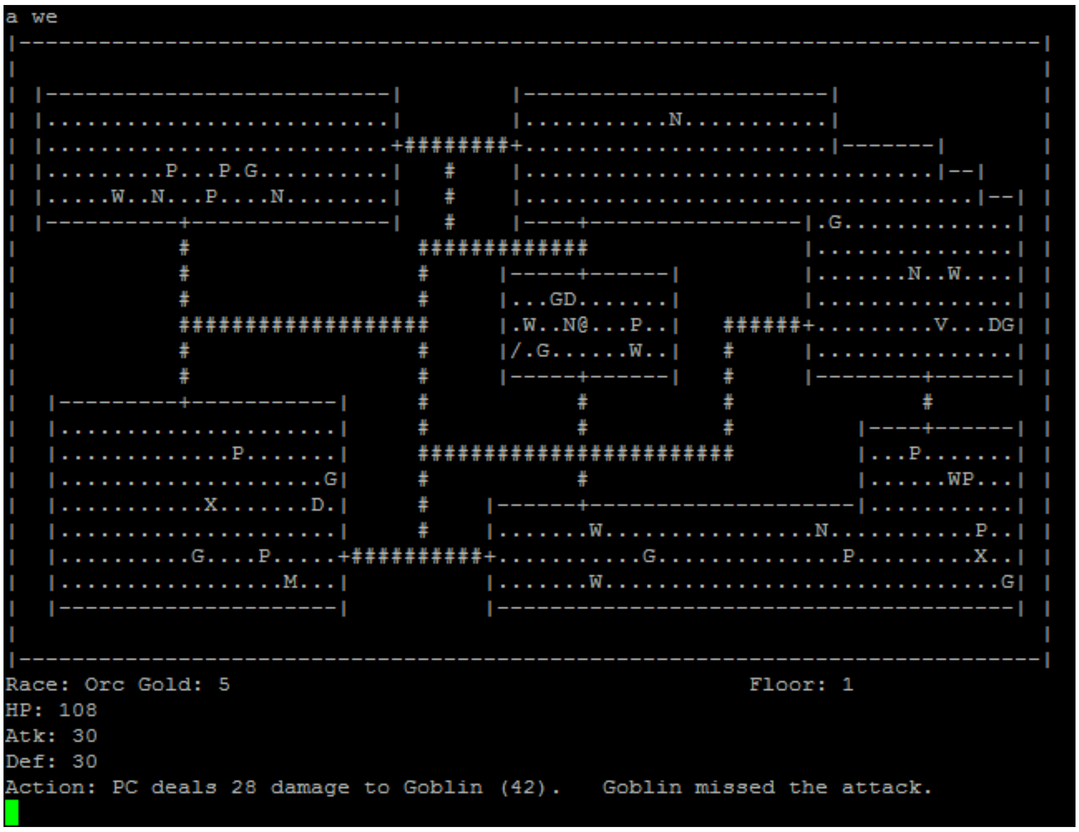
- Command Interpretation
  1) If user enters direction “no, we, so, ea ,nw, ne,sw, se”, the player character will move in the direction. An enemy will attack PC if PC enters within a block of an enemy. PC might see a potion if PC enters within a block of an enemy and PC might be able to pick up a treasure. Of course, these may happen at the same time. The way to implement this so we can print out the complete message and update the grid is by calling a few methods and checking special cases after we call PC ‘s move() method and these methods return a string and we print out those strings together. We first call player’s “move” method, where it returns a bool. If it succeed, we will print out “PC moves to” and direction plus the strings come from the result we check if there is a dragonHorde(special case) or a potion or an hostile enemy is 1 block within. if it failed, we will print out “Invalid movement” with other strings come from the checking above.
     See figure 3.



```
we
|-----------------------------------------------------------------------------|
|                                                                             |
| |--------------------------|        |-----------------------|               |
| |..............P...N.......|        |GG.................V...|               |
| |......................G...+########+N.D....................|-------|       |
| |G..P..........N...........|   #    |............G....G.............|--|    |
| |.....X....................|   #    |............D....D............N...|--| |
| |----------+---------------|   #    |----+----------------|.../...........| |
|            #                 #############                |...............| |
|            #                 #     |-----+------|         |............G..| |
|            #                 #     |...@..G....V|         |.....P.........| |
|            ###################     |..XP.......V|   ######+......W........| |
|            #                 #     |.........PP.|   #     |...............| |
|            #                 #     |-----+------|   #     |--------+------| |
|  |---------+-----------|     #           #          #              #        |
|  |.......M.............|     #           #          #         |----+------| |
|  |.......P.............|     ########################         |..........X| |
|  |..........T..........|     #           #                    |...........| |
|  |...V...............GP|     #    |------+--------------------|..V........| |
|  |........P.N.P.N.....D|     #    |.....N.......................G.........| |
|  |........W..........N.+##########+.......................................| |
|  |.....................|          |.......................................| |
|  |---------------------|          |---------------------------------------| |
|                                                                             |
|-----------------------------------------------------------------------------|
Race: Orc Gold: 0                                       Floor: 1
HP: 106
Atk: 30
Def: 30
Action: PC moves West. PC sees an unknown potion.  Phoenix deals 27 damages on PC.
```

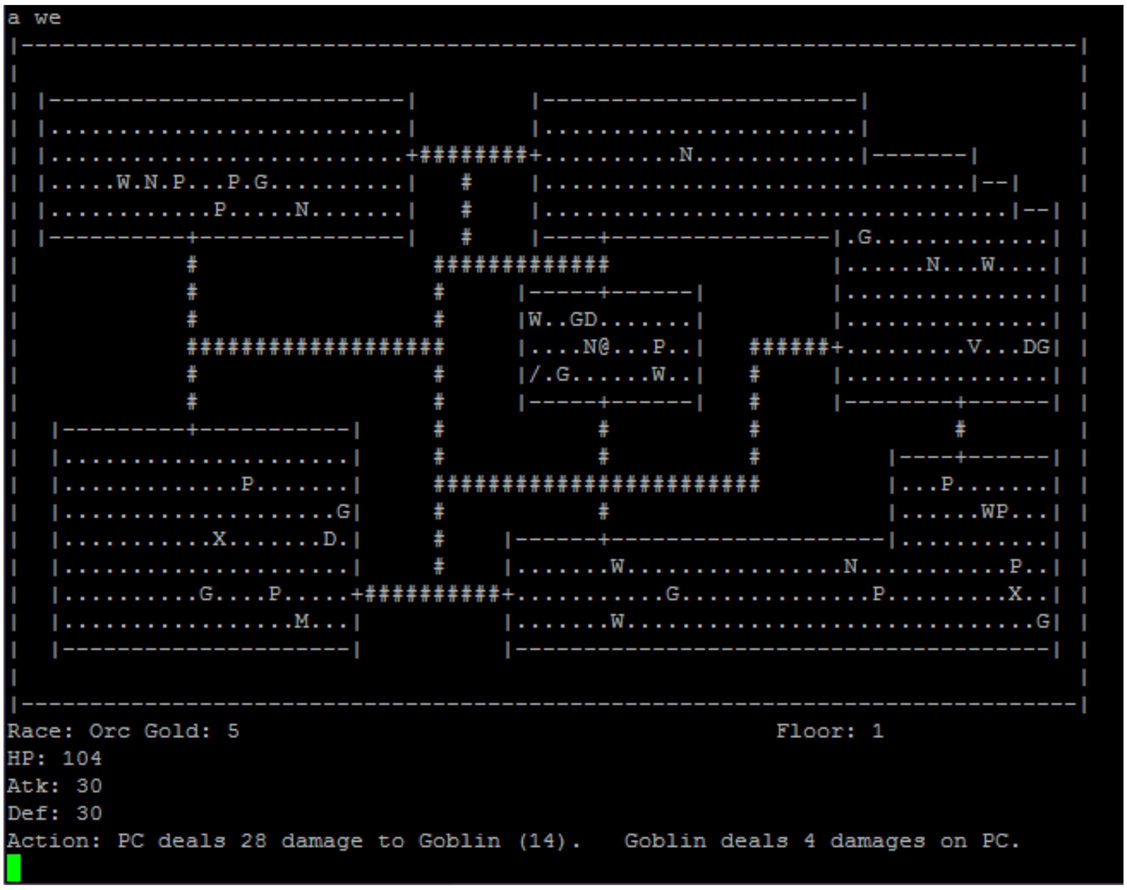
figure 3

2) If user enters attack "a" and its direction. We will call an object "mediator" 's combat method which takes in the x, y position of the attacker and the defender.For here, we use the design pattern "Mediator pattern", the PC will no longer call enemy directly when it wants to attack it but instead both of them(the coordinates) will be passed to the mediator. This lowers the dependencies between objects, thus it lowers the coupling. Then the mediator will check if a real enemy is in the direction, if so, it deals the damage from PC to enemy and return a string of the result with two possibilites that " PC kills the enemies" or "PC deals X damage to the enemy", if not, the mediator will return a string "Invalid attack". After this, grid will call its enemyRandomMove() method and an enemy will not move if it is dealing a combat with the PC and it will attack PC with a 50% success. This enemyRandom() will return the string of the result whether enemies successfully strike on PC or miss it or just no enemies are attacking PC. A more specific explanation about combat is explained later in the section of combat including special cases with Merchant and Dragon.
See figure 4 and figure 5.



```
a we
|-----------------------------------------------------------------------------|
|                                                                             |
| |--------------------------|        |-----------------------|               |
| |..........................|        |...........N...........|               |
| |..........................+########+.......................|-------|       |
| |.........P...P.G..........|   #    |...............................|--|    |
| |.....W..N...P....N........|   #    |..................................|--| |
| |----------+---------------|   #    |----+----------------|.G.............| |
|            #                 #############                |...............| |
|            #                 #     |-----+------|          |.......N..W....| |
|            #                 #     |...GD.......|          |...............| |
|            ###################     |.W..N@...P..|   ######+.........V...DG| |
|            #                 #     |/.G......W..|   #      |...............| |
|            #                 #     |-----+------|   #      |--------+------| |
|  |---------+-----------|     #           #          #               #        |
|  |.....................|     #           #          #          |----+------| |
|  |..............P......|     ########################          |...P.......| |
|  |....................G|     #           #                     |......WP...| |
|  |...........X.......D.|     #    |------+--------------------|...........| |
|  |.....................|     #    |.......W................N...........P..| |
|  |..........G....P.....+##########+...........G..............P.........X..| |
|  |.................M...|          |.......W..............................G| |
|  |---------------------|          |---------------------------------------| |
|                                                                             |
|-----------------------------------------------------------------------------|
Race: Orc Gold: 5                                        Floor: 1
HP: 108
Atk: 30
Def: 30
Action: PC deals 28 damage to Goblin (42).   Goblin missed the attack.
```

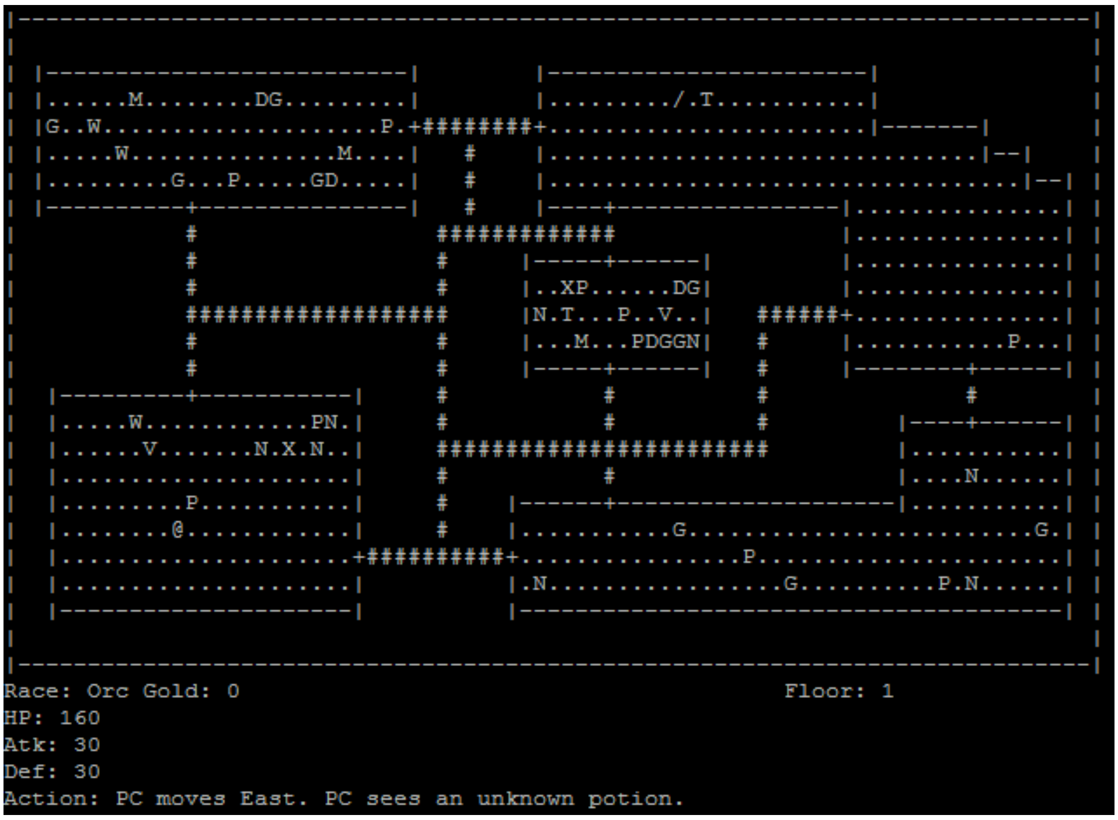
figure 4



```
a we
|-----------------------------------------------------------------------------|
|                                                                             |
| |--------------------------|        |-----------------------|               |
| |..........................|        |.......................|               |
| |..........................+########+..........N............|-------|       |
| |.....W.N.P...P.G..........|   #    |...............................|--|    |
| |............P.....N.......|   #    |..................................|--| |
| |----------+---------------|   #    |----+----------------|.G.............| |
|            #                 #############                |......N...W....| |
|            #                 #     |-----+------|          |...............| |
|            #                 #     |W..GD.......|          |...............| |
|            ###################     |....N@...P..|   ######+.........V...DG| |
|            #                 #     |/.G......W..|   #      |...............| |
|            #                 #     |-----+------|   #      |--------+------| |
|  |---------+-----------|     #           #          #               #        |
|  |.....................|     #           #          #          |----+------| |
|  |.............P.......|     ########################          |...P.......| |
|  |....................G|     #           #                     |......WP...| |
|  |...........X.......D.|     #    |------+--------------------|...........| |
|  |.....................|     #    |.......W................N...........P..| |
|  |..........G....P.....+##########+...........G..............P.........X..| |
|  |.................M...|          |.......W..............................G| |
|  |---------------------|          |---------------------------------------| |
|                                                                             |
|-----------------------------------------------------------------------------|
Race: Orc Gold: 5                                       Floor: 1
HP: 104
Atk: 30
Def: 30
Action: PC deals 28 damage to Goblin (14).   Goblin deals 4 damages on PC.
```

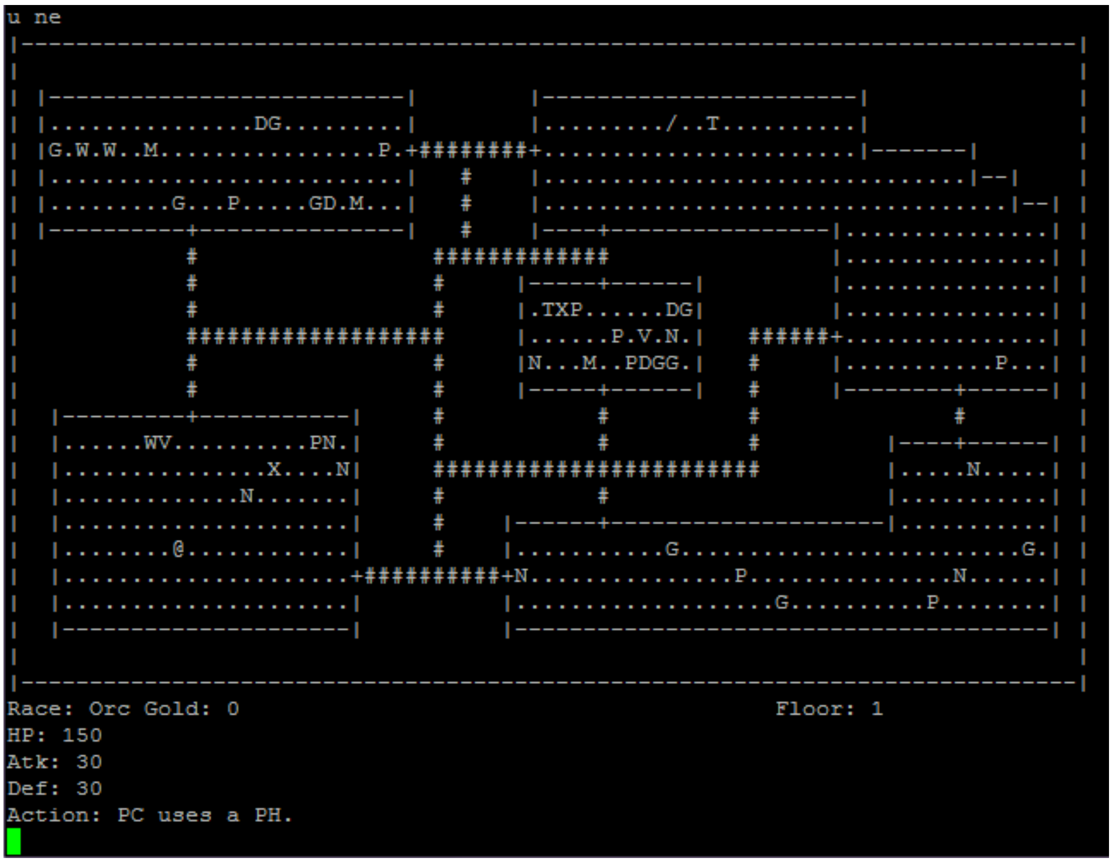
figure 5

3) If user enters “u” and its direction. We call player character’s usePotion() method which takes the pointer of the potion. The base class “playerCharacter” implement the usePotion() method as a virtual method, so that it will dynamically call the usePotion() method of a race whoever the poiner points to. Elves convert the negative effect into positve effects. Other races follow the positve effects and the negative effects of the potion.
A specific implement of seeing a potion and using a potion with different printed out message is shown later.
See figure 6 and figure 7.



```
|-----------------------------------------------------------------------------|
|                                                                             |
| |--------------------------|        |-----------------------|               |
| |......M........DG.........|        |........./.T...........|               |
| |G..W....................P.+########+.......................|-------|       |
| |.....W...............M....|   #    |...............................|--|    |
| |.........G...P.....GD.....|   #    |..................................|--| |
| |----------+---------------|   #    |----+----------------|...............| |
|            #                 #############                |...............| |
|            #                 #     |-----+------|         |...............| |
|            #                 #     |..XP......DG|         |...............| |
|            ###################     |N.T...P..V..|   ######+...............| |
|            #                 #     |...M...PDGGN|   #     |...........P...| |
|            #                 #     |-----+------|   #     |--------+------| |
|  |---------+-----------|     #           #          #              #        |
|  |.....W............PN.|     #           #          #         |----+------| |
|  |......V.......N.X.N..|     ########################         |...........| |
|  |.....................|     #           #                    |....N......| |
|  |.........P...........|     #    |------+--------------------|...........| |
|  |........@............|     #    |...........G.........................G.| |
|  |.....................+##########+................P......................| |
|  |.....................|          |.N.................G..........P.N......| |
|  |---------------------|          |---------------------------------------| |
|                                                                             |
|-----------------------------------------------------------------------------|
Race: Orc Gold: 0                                       Floor: 1
HP: 160
Atk: 30
Def: 30
Action: PC moves East. PC sees an unknown potion.
```

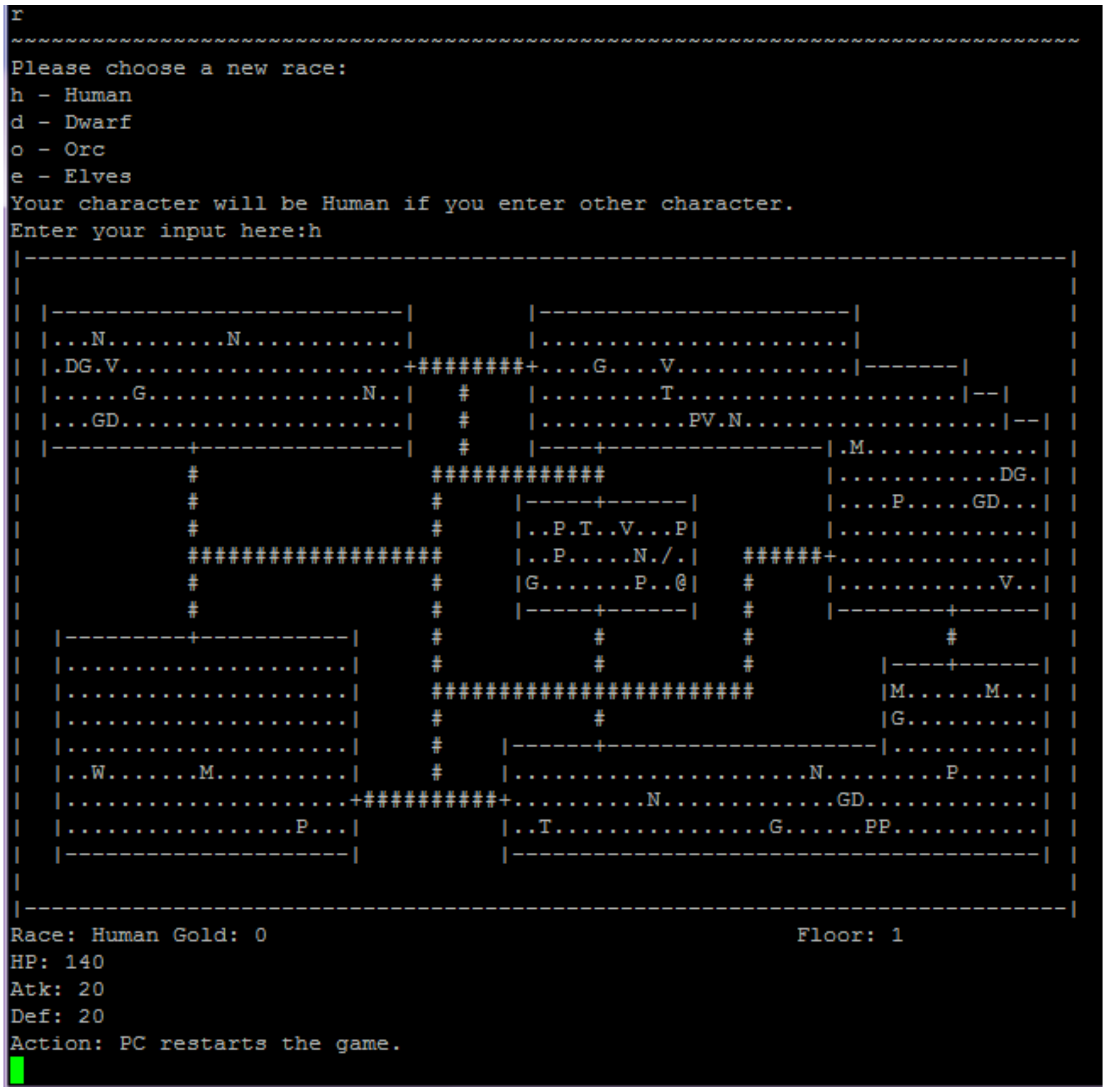
figure 6



```
u ne
|-----------------------------------------------------------------------------|
|                                                                             |
| |--------------------------|        |-----------------------|               |
| |...............DG.........|        |........./..T..........|               |
| |G.W.W..M................P.+########+.......................|-------|       |
| |..........................|   #    |...............................|--|    |
| |.........G...P.....GD.M...|   #    |..................................|--| |
| |----------+---------------|   #    |----+----------------|...............| |
|            #                 #############                |...............| |
|            #                 #     |-----+------|         |...............| |
|            #                 #     |.TXP......DG|         |...............| |
|            ###################     |......P.V.N.|   ######+...............| |
|            #                 #     |N...M..PDGG.|   #     |...........P...| |
|            #                 #     |-----+------|   #     |--------+------| |
|  |---------+-----------|     #           #          #              #        |
|  |......WV..........PN.|     #           #          #         |----+------| |
|  |...............X....N|     ########################         |.....N.....| |
|  |.............N.......|     #           #                    |...........| |
|  |.....................|     #    |------+--------------------|...........| |
|  |........@............|     #    |...........G.........................G.| |
|  |.....................+##########+N...............P...............N......| |
|  |.....................|          |...................G..........P........| |
|  |---------------------|          |---------------------------------------| |
|                                                                             |
|-----------------------------------------------------------------------------|
Race: Orc Gold: 0                                       Floor: 1
HP: 150
Atk: 30
Def: 30
Action: PC uses a PH.
```

figure 7

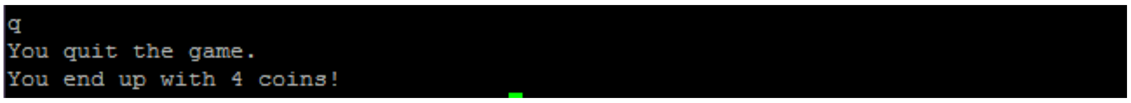
4) if user enters “r” , the game will be reset and a new race will be chosen. This is implemented by simply deleting the previous grid, thus deleting everything , and create a whole new game with a new race.
   See figure 8.



```
r
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Please choose a new race:
h - Human
d - Dwarf
o - Orc
e - Elves
Your character will be Human if you enter other character.
Enter your input here:h
|-----------------------------------------------------------------------------|
|                                                                             |
| |--------------------------|        |-----------------------|               |
| |...N.........N............|        |.......................|               |
| |.DG.V.....................+########+....G....V.............|-------|       |
| |......G................N..|   #    |.........T.....................|--|    |
| |...GD.....................|   #    |...........PV.N...................|--| |
| |----------+---------------|   #    |----+----------------|.M.............| |
|            #                 #############                |............DG.| |
|            #                 #     |-----+------|         |....P.....GD...| |
|            #                 #     |..P.T..V...P|         |...............| |
|            ###################     |..P.....N./.|   ######+...............| |
|            #                 #     |G.......P..@|   #     |............V..| |
|            #                 #     |-----+------|   #     |--------+------| |
|  |---------+-----------|     #           #          #              #        |
|  |.....................|     #           #          #         |----+------| |
|  |.....................|     ########################         |M......M...| |
|  |.....................|     #           #                    |G..........| |
|  |.....................|     #    |------+--------------------|...........| |
|  |..W.......M..........|     #    |......................N..........P.....| |
|  |.....................+##########+..........N.............GD.............| |
|  |.................P...|          |..T................G......PP...........| |
|  |---------------------|          |---------------------------------------| |
|                                                                             |
|-----------------------------------------------------------------------------|
Race: Human Gold: 0                                         Floor: 1
HP: 140
Atk: 20
Def: 20
Action: PC restarts the game.
```

figure 8

5) If user enters “q”, he will quit the game and score will be calculated. The whole grid will then be deleted.
   See figure 9.



```
q
You quit the game.
You end up with 4 coins!
```

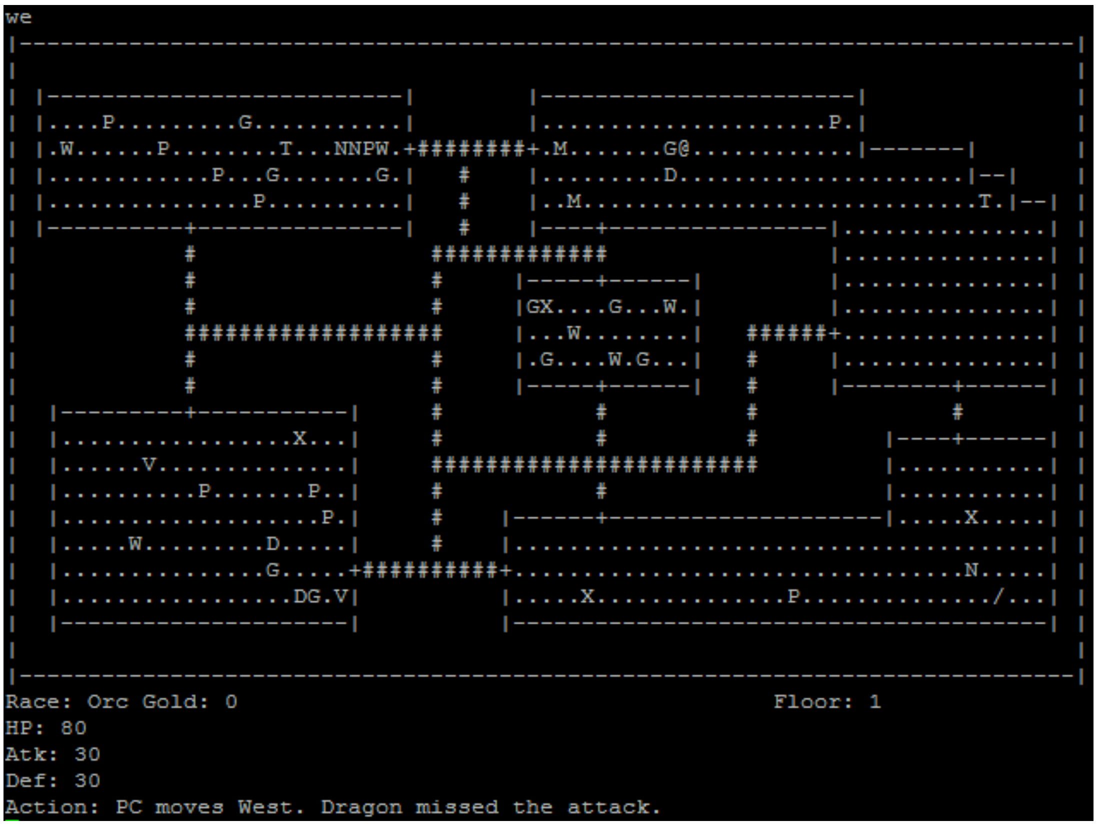
figure 9

- Combat between PC and Enemies

- When user decide to attack an enemy, we will pass the coordinates of the attacker and defenser to mediator and deal this combat. However, enemies' strike method is called within the gird.
- Now we have 7 different kind of enemies. Troll, Phenix, Goblin, Dragon, Merchant, Vampire and Werewolf. They will be randomly spawned on the floor with the same possibility distribution between chambers and tiles. Whenever player input a command, either valid or invalid, enemies(except dragon which shall protect its dragonHorde and will not move) will randomly move to a block chosen from the eight blocks around it. This enemyRandomMove() is inside grid and called by grid. Before set an enemy to a new postion, it will check whether the PC happens to be in the 8 positions around the enemy or not. If yes, enemy's attack method will be called and passed the pointer of the PC and it will return a string representing the result of this strike, either success or failure with a 50% probability by using rand() % 2 where 0 represents "failure" and 1 represents "success". i.e "Enemy deals X damage to PC" or " Enemy missed the attack". If no, the enemy will be set to a new position.

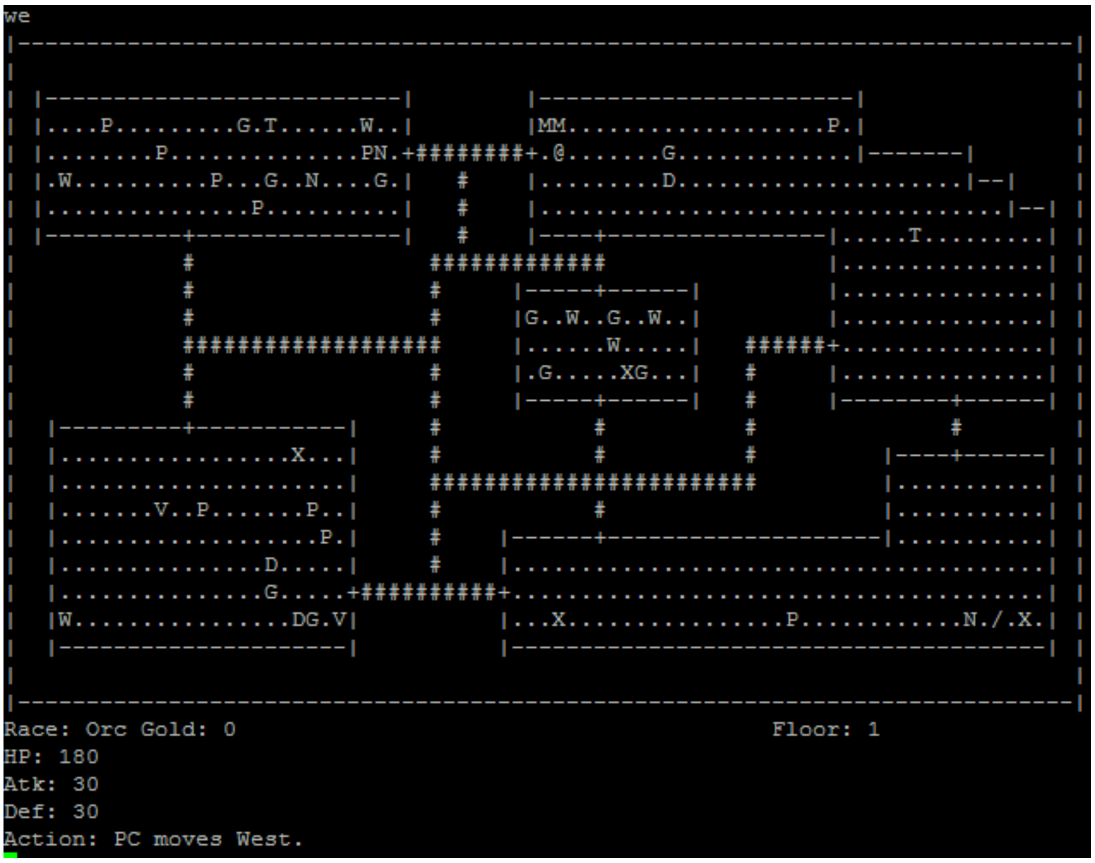
- There are two special enemies, dragon and merchant.
- 1) Dragon will attack PC if and only if the PC is within 1 block of its dragonHorde. This is implemented by checking the 8 positions around the PC every time when player enters a command. If a dragonHorde is found, we will check if the pointer inside dragonHorde which points to the location of its protector dragon is a dragon or not a dragon by checking the "type" of that specific tile. we call its getType() method which will give us the specific type if there is anything on that tile. i.e. It can return "Dragon" which is the protector the dragonHorde(means dragon is not dead yet). or "Ground"(means the dragon is dead), or "PC"(means the dragon is dead and now PC stands on its position) or any other enemies type(except "Dragon) . See Figure 10.

- 2) Merchant will attack PC if and only if PC striked a merchant before or killed one merchant before. Before the merchant is going to strike PC, we check if it is hostile to PC by checking a static field "isHostileToPc" inside the "Merchant" class. Every merchant enemy is related to the class, thus every merchant can check this static field. It then will decide to call its attack method on the PC or ignore PC. Thus, whenever PC strike a merchant or killed one, we will set the static field "isHostileToPc" of the merchant class to "true". If a merchant is dead, it will drop a gold on the ground. See Figure 11, Figure 12.



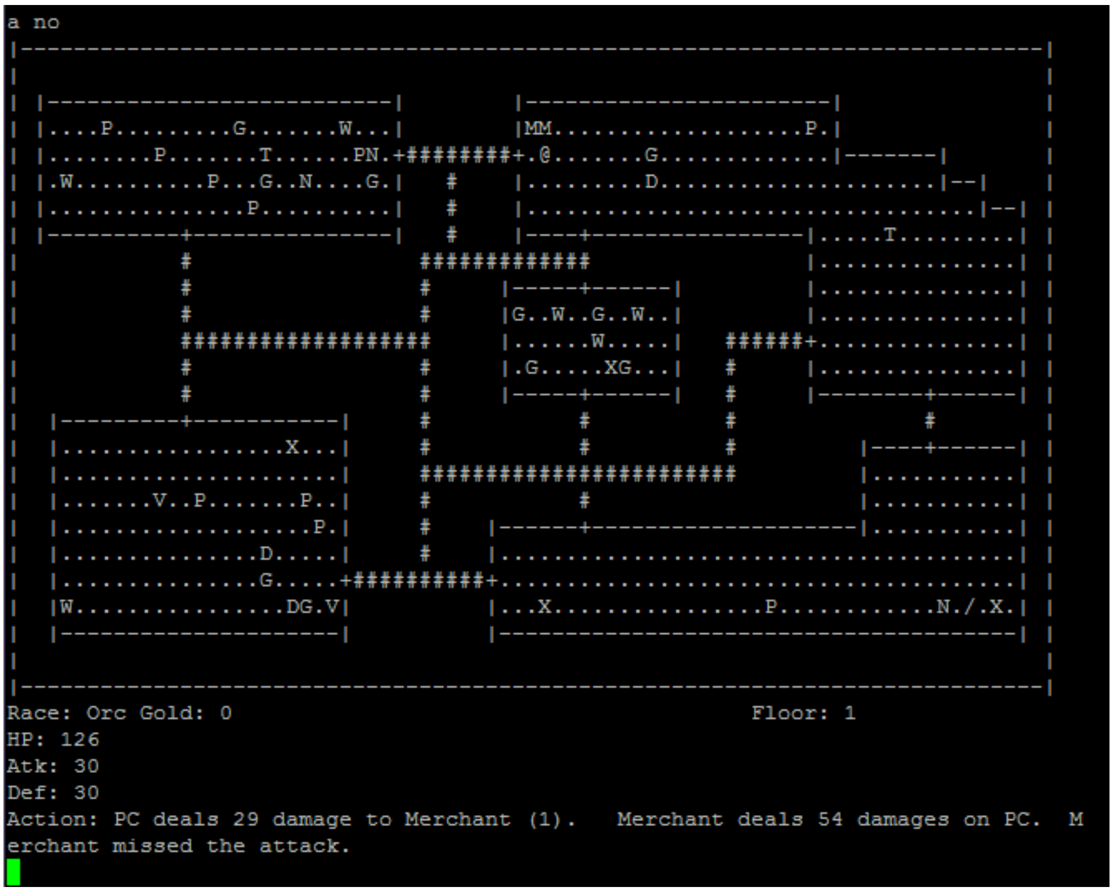
```
we
|-----------------------------------------------------------------------------|
|                                                                             |
| |--------------------------|        |-----------------------|               |
| |....P.........G...........|        |.....................P.|               |
| |.W......P........T...NNPW.+########+.M.......G@............|-------|       |
| |............P...G.......G.|   #    |.........D.....................|--|    |
| |...............P..........|   #    |..M.............................T.|--| |
| |----------+---------------|   #    |----+----------------|...............| |
|            #                 #############                |...............| |
|            #                 #     |-----+------|          |...............| |
|            #                 #     |GX....G...W.|          |...............| |
|            ###################     |...W........|   ######+...............| |
|            #                 #     |.G....W.G...|   #      |...............| |
|            #                 #     |-----+------|   #      |--------+------| |
|  |---------+-----------|     #           #          #               #        |
|  |.................X...|     #           #          #          |----+------| |
|  |......V..............|     ########################          |...........| |
|  |..........P.......P..|     #           #                     |...........| |
|  |...................P.|     #    |------+--------------------|.....X.....| |
|  |.....W.........D.....|     #    |.......................................| |
|  |...............G.....+##########+.................................N.....| |
|  |.................DG.V|          |.....X..............P.............../...| |
|  |---------------------|          |---------------------------------------| |
|                                                                             |
|-----------------------------------------------------------------------------|
Race: Orc Gold: 0                                           Floor: 1
HP: 80
Atk: 30
Def: 30
Action: PC moves West. Dragon missed the attack.
```

figure 10



```
we
|-----------------------------------------------------------------------------|
|                                                                             |
| |--------------------------|        |-----------------------|               |
| |....P.........G.T......W..|        |MM...................P.|               |
| |........P..............PN.+########+.@.......G.............|-------|       |
| |.W..........P...G..N....G.|   #    |.........D.....................|--|    |
| |...............P..........|   #    |..................................|--| |
| |----------+---------------|   #    |----+----------------|.....T.........| |
|            #                 #############                |...............| |
|            #                 #     |-----+------|          |...............| |
|            #                 #     |G..W..G..W..|          |...............| |
|            ###################     |......W.....|   ######+...............| |
|            #                 #     |.G.....XG...|   #      |...............| |
|            #                 #     |-----+------|   #      |--------+------| |
|  |---------+-----------|     #           #          #               #        |
|  |.................X...|     #           #          #          |----+------| |
|  |.....................|     ########################          |...........| |
|  |.......V..P.......P..|     #           #                     |...........| |
|  |...................P.|     #    |------+--------------------|...........| |
|  |...............D.....|     #    |.......................................| |
|  |...............G.....+##########+.......................................| |
|  |W................DG.V|          |...X................P............N./.X.| |
|  |---------------------|          |---------------------------------------| |
|                                                                             |
|-----------------------------------------------------------------------------|
Race: Orc Gold: 0                                           Floor: 1
HP: 180
Atk: 30
Def: 30
Action: PC moves West.
```

figure 11

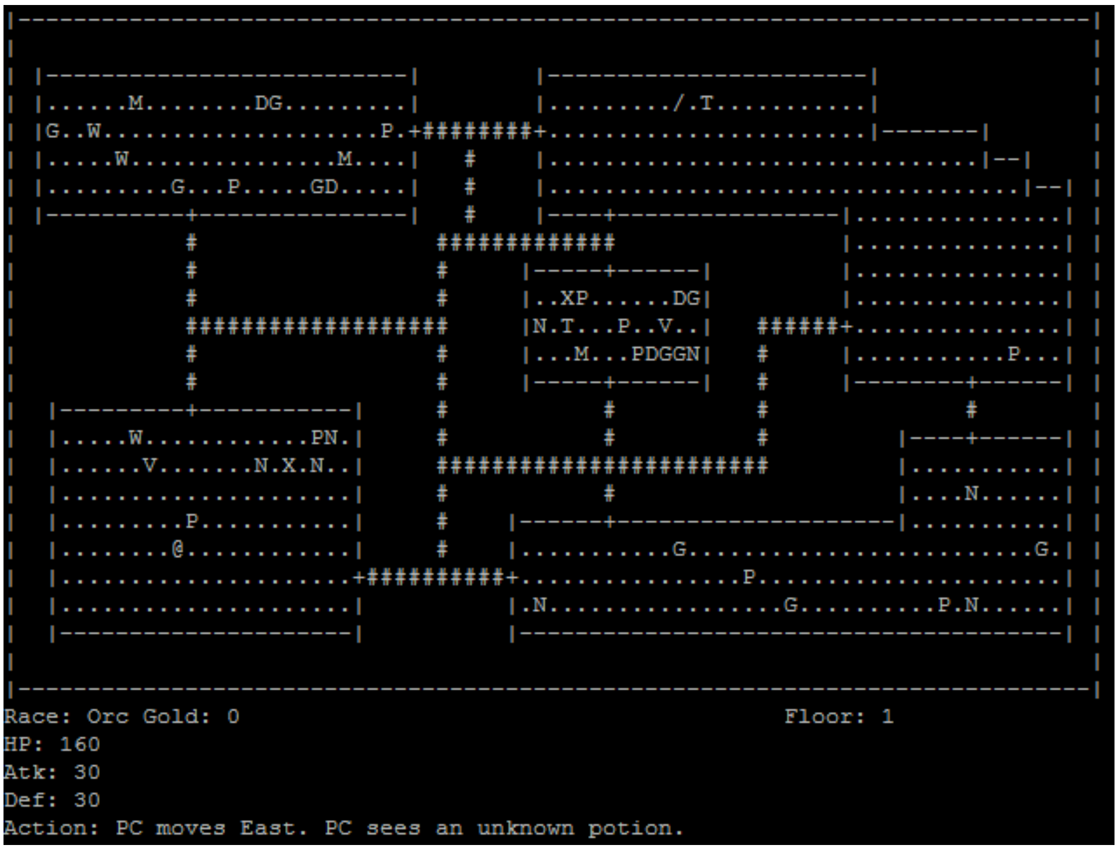


```
a no
|-----------------------------------------------------------------------------|
|                                                                             |
| |--------------------------|        |-----------------------|               |
| |....P.........G.......W...|        |MM...................P.|               |
| |........P.......T......PN.+########+.@.......G.............|-------|       |
| |.W..........P...G..N....G.|   #    |.........D.....................|--|    |
| |...............P..........|   #    |..................................|--| |
| |----------+---------------|   #    |----+----------------|.....T.........| |
|            #                 #############                |...............| |
|            #                 #     |-----+------|          |...............| |
|            #                 #     |G..W..G..W..|          |...............| |
|            ###################     |......W.....|   ######+...............| |
|            #                 #     |.G.....XG...|   #      |...............| |
|            #                 #     |-----+------|   #      |--------+------| |
|  |---------+-----------|     #           #          #               #        |
|  |.................X...|     #           #          #          |----+------| |
|  |.....................|     ########################          |...........| |
|  |.......V..P.......P..|     #           #                     |...........| |
|  |...................P.|     #    |------+--------------------|...........| |
|  |...............D.....|     #    |.......................................| |
|  |...............G.....+##########+.......................................| |
|  |W................DG.V|          |...X................P............N./.X.| |
|  |---------------------|          |---------------------------------------| |
|                                                                             |
|-----------------------------------------------------------------------------|
Race: Orc Gold: 0                                         Floor: 1
HP: 126
Atk: 30
Def: 30
Action: PC deals 29 damage to Merchant (1).   Merchant deals 54 damages on PC.  M
erchant missed the attack.
```

figure 12

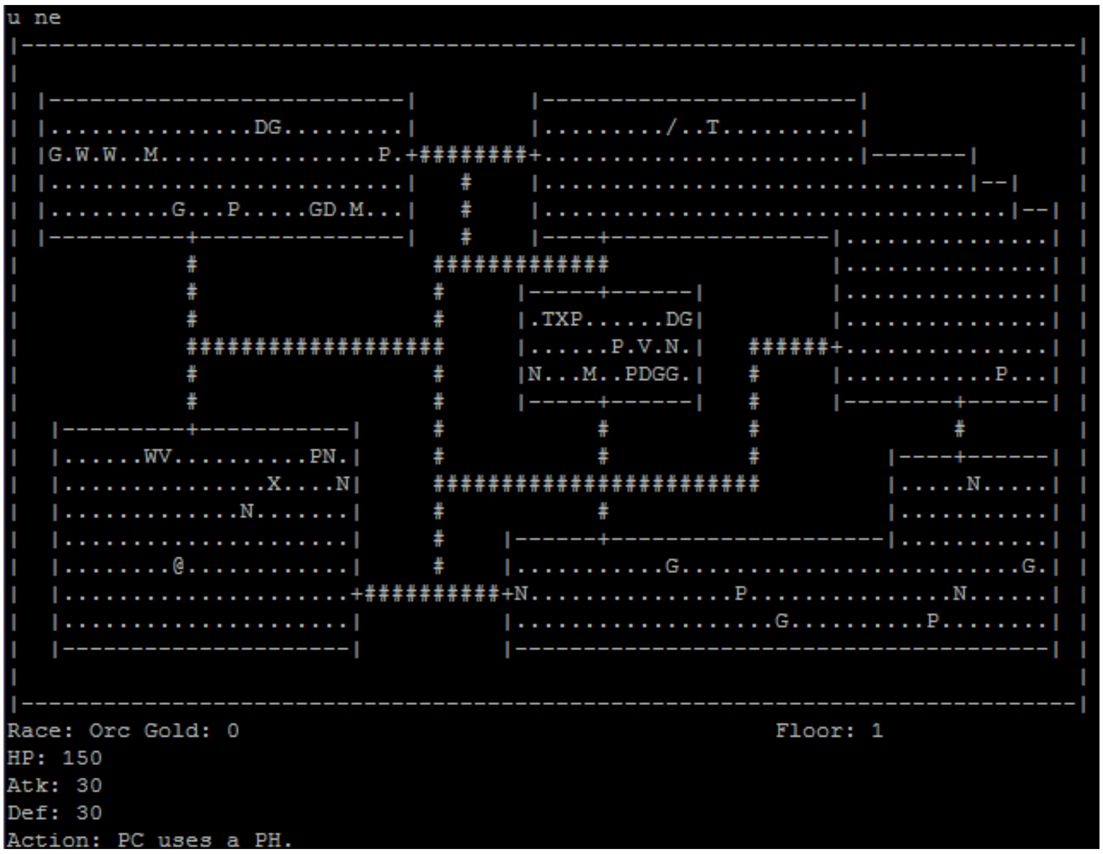
- Seeing a Potion and Using a Potion

  When the PC approaches a potion, it will be checked to see whether that potion has been seen before or not by checking a “mapping” implemented in the main control flow, m[potion name] = bool. If the bool is false, we print “PC sees an unknown potion”. If the bool is true, we print “PC sees <potion name>”. If the PC decide to use the potion, we turn that bool to true and print “PC uses <potion name>.
  See figure 13 and figure 14.



```
|-----------------------------------------------------------------------------|
|                                                                             |
| |--------------------------|        |-----------------------|               |
| |......M........DG.........|        |........./.T...........|               |
| |G..W....................P.+########+.......................|-------|       |
| |.....W...............M....|   #    |...............................|--|    |
| |.........G...P.....GD.....|   #    |..................................|--| |
| |----------+---------------|   #    |----+----------------|...............| |
|            #                 #############                |...............| |
|            #                 #     |-----+------|         |...............| |
|            #                 #     |..XP......DG|         |...............| |
|            ###################     |N.T...P..V..|   ######+...............| |
|            #                 #     |...M...PDGGN|   #     |...........P...| |
|            #                 #     |-----+------|   #     |--------+------| |
|  |---------+-----------|     #           #          #              #        |
|  |.....W............PN.|     #           #          #         |----+------| |
|  |......V.......N.X.N..|     ########################         |...........| |
|  |.....................|     #           #                    |....N......| |
|  |.........P...........|     #    |------+--------------------|...........| |
|  |........@............|     #    |...........G.........................G.| |
|  |.....................+##########+................P......................| |
|  |.....................|          |.N.................G..........P.N......| |
|  |---------------------|          |---------------------------------------| |
|                                                                             |
|-----------------------------------------------------------------------------|
Race: Orc Gold: 0                                      Floor: 1
HP: 160
Atk: 30
Def: 30
Action: PC moves East. PC sees an unknown potion.
```

figure 13



```
u ne
|-----------------------------------------------------------------------------|
|                                                                             |
| |--------------------------|        |-----------------------|               |
| |...............DG.........|        |........./..T..........|               |
| |G.W.W..M................P.+########+.......................|-------|       |
| |..........................|   #    |...............................|--|    |
| |.........G...P.....GD.M...|   #    |..................................|--| |
| |----------+---------------|   #    |----+----------------|...............| |
|            #                 #############                |...............| |
|            #                 #     |-----+------|         |...............| |
|            #                 #     |.TXP......DG|         |...............| |
|            ###################     |......P.V.N.|   ######+...............| |
|            #                 #     |N...M..PDGG.|   #     |...........P...| |
|            #                 #     |-----+------|   #     |--------+------| |
|  |---------+-----------|     #           #          #              #        |
|  |......WV..........PN.|     #           #          #         |----+------| |
|  |...............X....N|     ########################         |.....N.....| |
|  |.............N.......|     #           #                    |...........| |
|  |.....................|     #    |------+--------------------|...........| |
|  |........@............|     #    |...........G.........................G.| |
|  |.....................+##########+N...............P...............N......| |
|  |.....................|          |...................G..........P........| |
|  |---------------------|          |---------------------------------------| |
|                                                                             |
|-----------------------------------------------------------------------------|
Race: Orc Gold: 0                                      Floor: 1
HP: 150
Atk: 30
Def: 30
Action: PC uses a PH.
```

figure 14

Assignment Design Specification Question

☆Question. How could your design your system so that each race could be easily generated? Additionally, how difficult does such a solution make adding additional classes?

Answer:

We design our system by defining a pure virtual base class "playerCharacter" and defining each race subclass inheriting from it.

When the player chooses a certain kind of race, by simply passing in the name of the race to createPlayerCharacter(string) method in Factory class, a pointer to the base class "playerCharacter" will be returned and the desired race will be initialized it. (in the previous plan: we want to create a pointer points to the base class and pass in each race's different HP, Atk and Def, which is not as good as this encapsulation method.) After that, we set the pc field in the Grid to the playerCharacter pointer that we generate and track it through the game.

When there are additional classes, we add subclass to "playerCharacter". In the subclass, we just need to implement constructor which pass in the HP, Atk and Def and usePotion method which overrides the pure virtual usePotion in the playerCharacter. In this way, it will be relatively easy to add new classes and create new races' objects.

☆Question. How does your system handle generating different enemies? Is it different from how you generate the player character? Why or why not?

Answer:

We create a base class "Enemy" and define each different subclass of different enemy inheriting from it. However, we use the design pattern "Factory Pattern" to generate different enemies. This is different from generating our player character since there might be only one single player character but the enemies will be several of them. When generating different enemies, the rand() function will be used to generate random numbers between 0 and 18. According to different probability distribution of different enemy, a certain range of numbers will be attached to certain kind of enemy as a "signal". Thus, when a number in that range is generated, a corresponding enemy will be generated.

☆Question. How could you implement special abilities for different enemies. For example, gold stealing for goblins, health regeneration for trolls, health stealing for vampires, etc.

Answer: We could declare a pure virtual method about special abilities in the base class "Enemy" and implement it differently according different abilities of different enemy in each enemy's subclass. Thus, when it's in the middle of combat, in every enemy's turn, this method will be invoked dynamically depending on what kind of enemy to carry out their special abilities.

**Question. What design pattern could you use to model the effects of temporary potions (Wound/Boost Atk/Def) so that you do not need to explicitly track which potions the player character has consumed on any particular floor?*

Answer: We give up the idea of using decorator pattern in the previous plan. Instead, we add two fields to "playerCharacter" class which represent the temporary change of Atk and Def. When the PC uses potion that changes Atk or Def, we display the current Atk or Def together with the change that we record. When PC enters new level, we reset the change field.

**Question. How could you generate items so that the generation of Treasure and Potions reuses as much code as possible? That is, how would you structure your system so that the generation of a potion and then generation of treasure does not duplicate code?*

Answer: The generation of treasure and potions by reusing as much code as possible is achieved through mapping and creating a class "Item" from which both Treasure and Potion inherit (in the previous plan: we did not consider this point.). For mapping, the key is an integer generated by rand() function, and the value is a string which represents the corresponding item type. In addition, since we have an "Item" class, we just need to pass the name of type to createItem in the factory class and get a pointer to Item returned. (in the previous plan: we did not consider this point.) Besides, we use only one method for generating both treasure and potions and pass in a char to indicate whether treasure or potions are desired. Hence, the generation doesn't use duplicated code.

Lessons Learnt

⁎Question. What lessons did this project teach you about developing software in teams?

Answer:
1) The most important thing this project taught us is about developing software step by step. Make sure we have our current stage working before we add new features to it rather than finish everything and try to run. The latter will be hard to debug, time-consuming, and easily break codes if we change anything.
2) Second, develop a good interface is a good way to work in teams where a good interface means other members don't need to look at your code in order to understand and use your classes or methods.
3) Third, a well-developed plan of attack is always important. Rather than working randomly, sticking to a plan can be efficient and systematic.

⁎Question. What would you have done differently if you had the chance to start over?

Answer:
1) Developing this CC3K game step by step. Make sure our current stage is working perfectly before we add additional features to it.
2) Try to avoid dynamic_cast used in the development of the game by taking the advantage of virtual method rather than transferring pointer types between base class and derived classes.
3) Add bonus feature. If we stick the step by step plan, we could have extra time to allow us to develop some additional features to this game to make it more interesting than before.

Conclusion

Overall, both team members felt that the project was a fantastic experience and gave a small taste of what team software development in a professional working environment is like. This project has helped further the understanding of good software engineering practices and the importance of time management for both team members. The team hopes to be able to indulge in a similar endeavour in the near future.