Assignment #2

Exercise 2

Name: Sahand Sabour

Student ID Number: 1614650

Module code: EEE102

Problem statement

The second problem (Exercise 2) asks us to complete the given header and source files, which were provided on ICE. Moreover, we are asked to also add two other classes to the given program in order to increase the variety of the characters that the user can choose from. Essentially, in this exercise, we are asked to complete an RPG style game where the users choose their class of choice to fight different stages of randomly generated opponents. For solving this problem, we should determine the CRC cards of the classes and draw their hierarchy chart in order to understand how they cooperate with each other. Consequently, we can complete the code for each part respectively.

Analysis

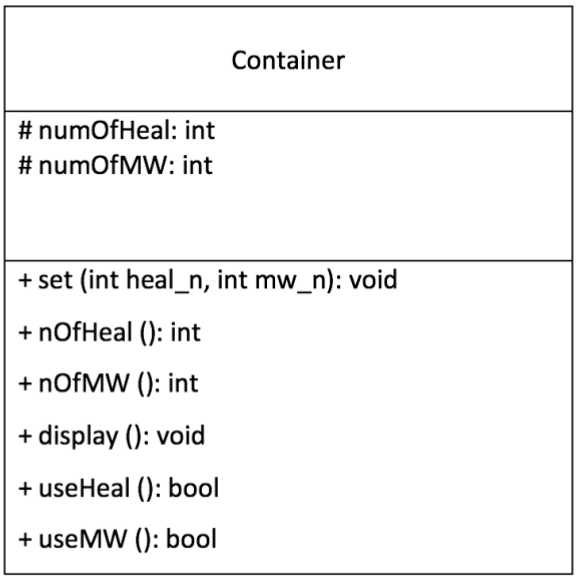
The input for this program is a string variable for the name of the user's character. We also ask the user to choose a number from one to three to choose their character's class, one being swordsman, two being archer, and three being mage.

The output of this program is the player's and enemy's attributes, such as their health, magic water, etc. Moreover, different stages of the game alongside the class of each opponent is also displayed as the output in the program. In addition, a message that shows whether the user has won the game and can move on to the next stage or has lost the game would be displayed as the final form of the output.

There are numerous variables used to program this mini-game. These variables can be summarized to be variables regarding the players' and the enemy's attributes (HP, HPmax, MP, etc.) variables that are set randomly in order to choose a class, as well as variables that determine the character's name and class.

Design

There are different stages to designing this program. First, as required in the task sheet, the CRC cards of the classes were generated and are provided respectively below (Figures 1-5). Moreover, the hierarchy chart of the mentioned classes is drawn. The hierarchy chart clearly demonstrates the relationship and the way of cooperation between these classes (Figure 6).

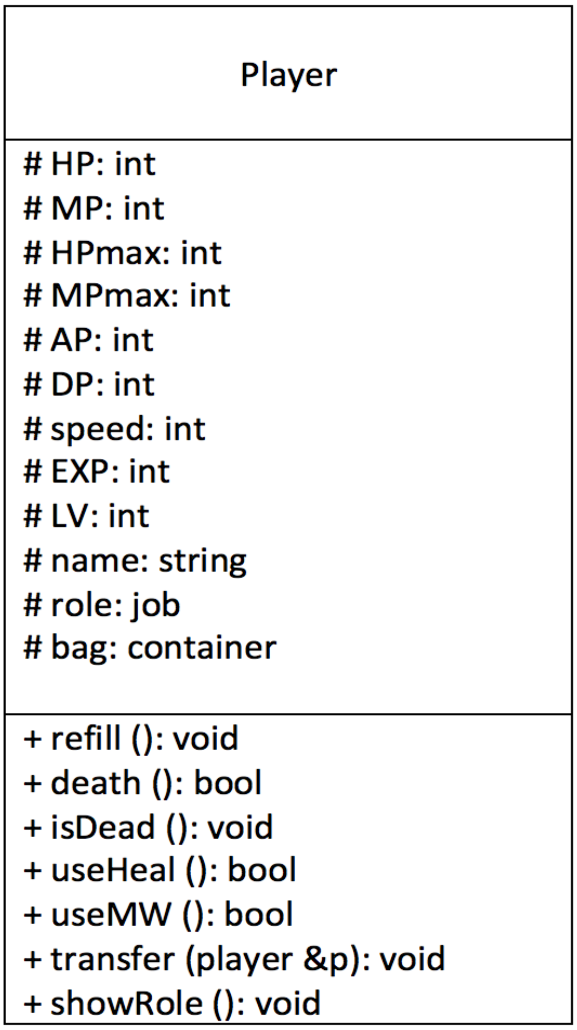


Container

numOfHeal: int
numOfMW: int

+ set (int heal_n, int mw_n): void
+ nOfHeal (): int
+ nOfMW (): int
+ display (): void
+ useHeal (): bool
+ useMW (): bool

Figure 1: Container class CRC card

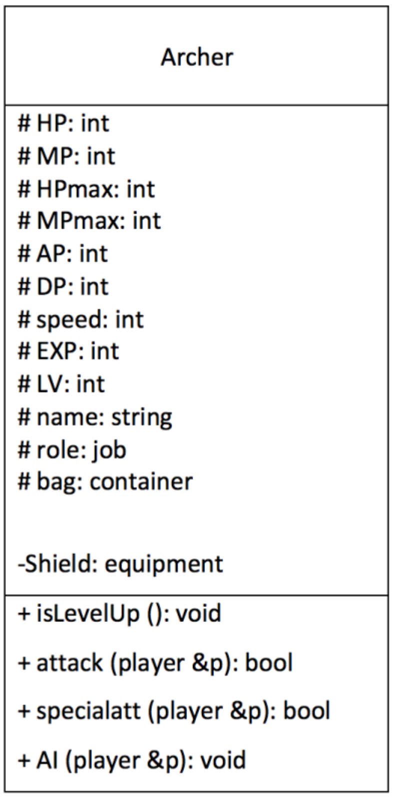


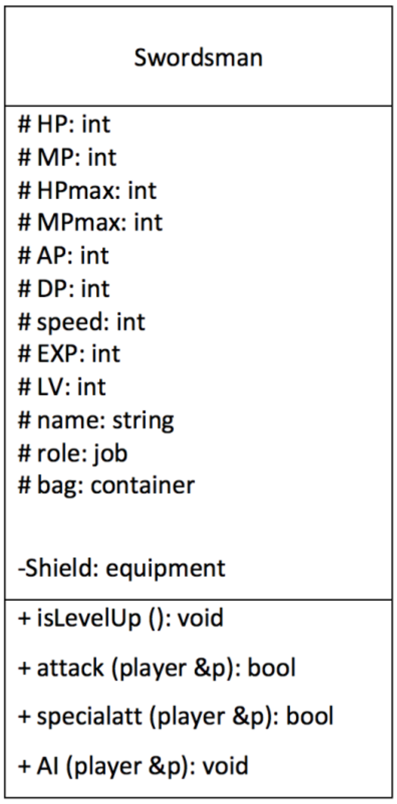
Player

HP: int
MP: int
HPmax: int
MPmax: int
AP: int
DP: int
speed: int
EXP: int
LV: int
name: string
role: job
bag: container

+ refill (): void
+ death (): bool
+ isDead (): void
+ useHeal (): bool
+ useMW (): bool
+ transfer (player &p): void
+ showRole (): void

Figure 2: Player class CRC card





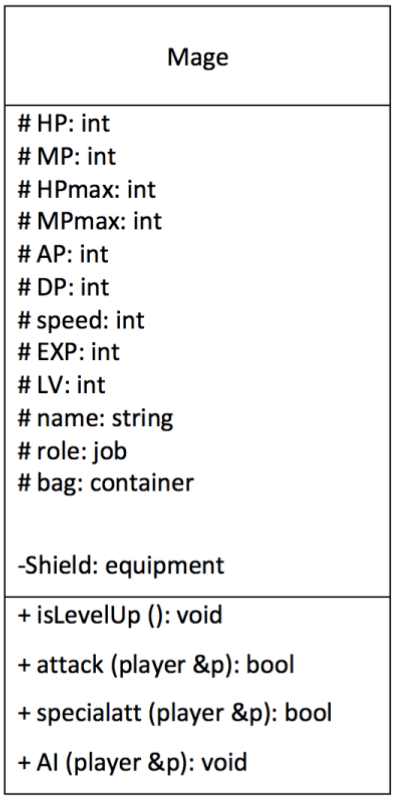


Figure 3: Archer CRC Figure 4: Swordsman CRC Figure 5: Mage CRC

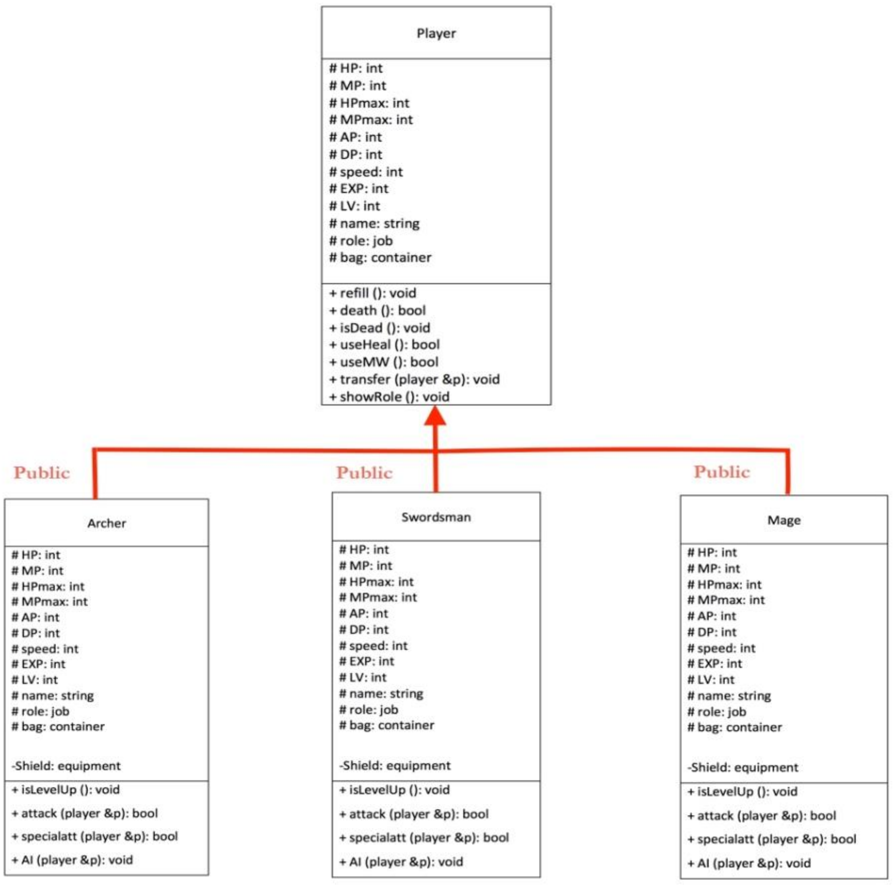


Figure 6: The hierarchy chart of the classes

The algorithm used for designing this program/game is as follows:
First, we ask the user for a name and a number from one to three, which corresponds to the class they choose, to create their character. Then we generate a random number from one to three that decides the name and the class of the opponent of the user (enemy). With each random number a new object of a character class will be created.

Second, is the part for the gameplay. As the first part of the gameplay, we show the human and enemy's info: Their HP level, MW level, etc. Next, we ask the user to choose one of the five given choices. The algorithm for each option is provided respectively below:

Attack
In this case, we use the attack function. Then, we check whether the character has enough EXP to level up. As the final step, we need to check whether the enemy has died or not to determine whether to continue this stage or move on to the next stage.

Special Attack
In this case, we use the specialatt function. Then, we check whether the character has enough EXP to level up. As the final step, we need to check whether the enemy has died or not to determine whether to continue this stage or move on to the next stage.

Use Heal
In this case, we use the useHeal function in order to store the player's HP back to HPmax.

Use Magic Water
In this case, we use the useHeal function in order to store the player's MP back to MPmax.

Exit Game
If the user chooses this option, we ask the user whether they are sure about their decision or not. If they respond with "Y", then the program shuts down. Else, the program ask the user to choose another option.

Lastly, we need to check whether the user's character has died or the enemy has died in order to show a victory/loss message. After showing the message, we need to show the information regarding the two mentioned characters. In the case that the user beats all of the opponents, we would display a message, which congratulates them for their victory.
Changes made to the code:

Note: since it is believed that the majority of the code was provided for this project, further explanation of each function and much detailed algorithm seems not to be needed. As there were seven different parts marks with question marks that we needed to complete, the explanation for the answer for each of these parts is provided respectively below:

1_????????????? ==> #ifndef
We are asked to insert a conditional compilation and therefore, #ifndef is added before #define to complete this part of the task.

2_????????????? ==> numOfHeal --
In this part, we are asked to complete the useHeal () function. In order to do so, we can take a look at useMW () function and do the same thing but with a different corresponding variable.

3_????????????? ==> bag.set(p.bag.nOfHeal (), p.bag.nOfMW ());
In this part, we were required to store the opponent's items inside the player's container (bag). Therefore, we use the set function in order to fulfill this task.

4_????????????? ==> void showinfo (player &p1, player &p2)
Showinfo () is a function that was declared in the player class but not defined. We can use this function in order to meet the requirement for this part, that is displaying character's job.

5_????????????? ==> public player
We are asked to make the swordsman class a sub-class of the player class. Therefore, this part was written based on the definition of sub-class.

6_????????????? ==> human -> isDead();
The requirement is to declare that the human is dead at this point. This can be achieved by calling the isDead () function.

7_????????????? ==> showinfo(*human, *enemy);
As the last part, we are required to display the information about the user's character and their opponent before the game ends. Therefore, we have used the previously created showinfo() function.

Moreover, two instances of human and enemy object were created as an initialization of the two objects. It is believed that without these instances, the program would not run properly and would face and error when compiling.

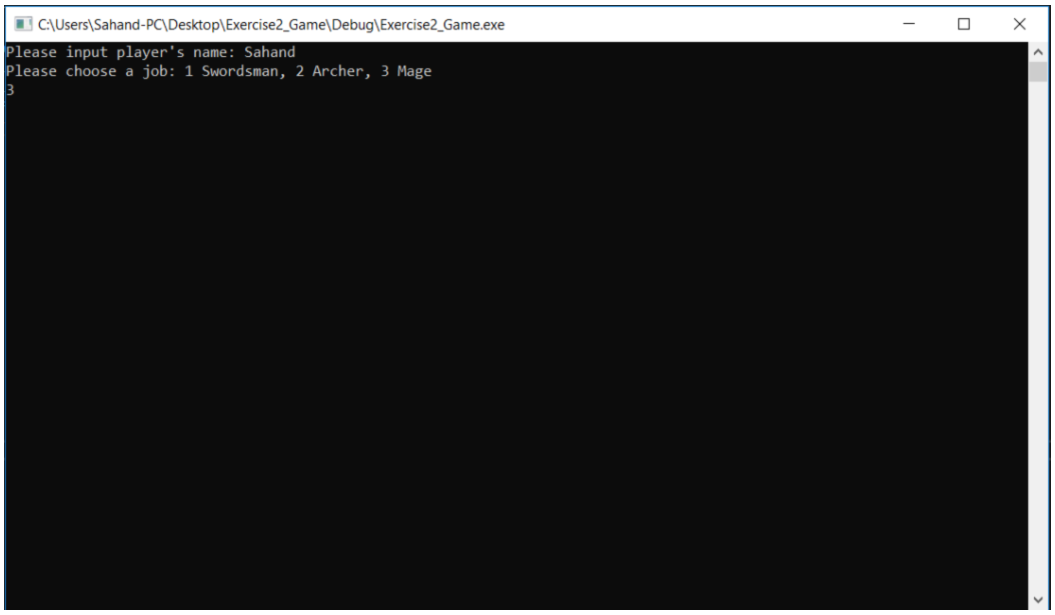
Changes made to the code:
Since we were required to generate enemies with random classes, some changes needed to be made to the given code. Firstly, we should declare and initialize an instance of enemy player object, similar to the human player object. Then, we change all forms of .FunctionName () to ->FunctionName () for enemy, because now enemy is a pointer object. Moreover, we should create a new virtual function for AI (&player) to initialize this function as well. The final change made to the provided code is adding a switch statement. A randomly generated number from one to three decides the case of this switch statement. In each case, a new instance of enemy corresponding to a different character class would be initialized.

Implementation
Header files (.h) ==> container.h, player.h, archer.h, mage.h and swordsman.h
Source files (.cpp) ==> container.cpp, player.cpp, archer.cpp, mage.cpp, swordsman.cpp and main.cpp

Testing

This program has three main parts to be tested:

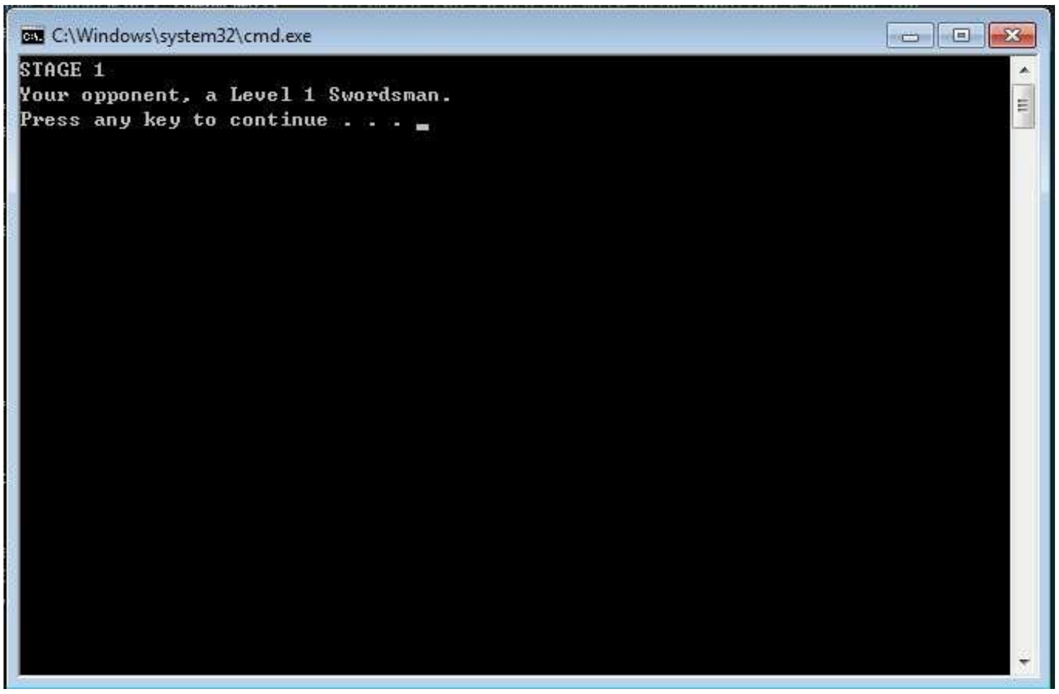
The first part to be tested is the character creation. This is where the user is asked to enter a name for the character and also choose a class out of the three possible choices (Figure 1). Moreover, an enemy of a randomly chosen class should be generated afterwards (Figure 2).



C:\Users\Sahand-PC\Desktop\Exercise2_Game\Debug\Exercise2_Game.exe

```
Please input player's name: Sahand
Please choose a job: 1 Swordsman, 2 Archer, 3 Mage
3
```

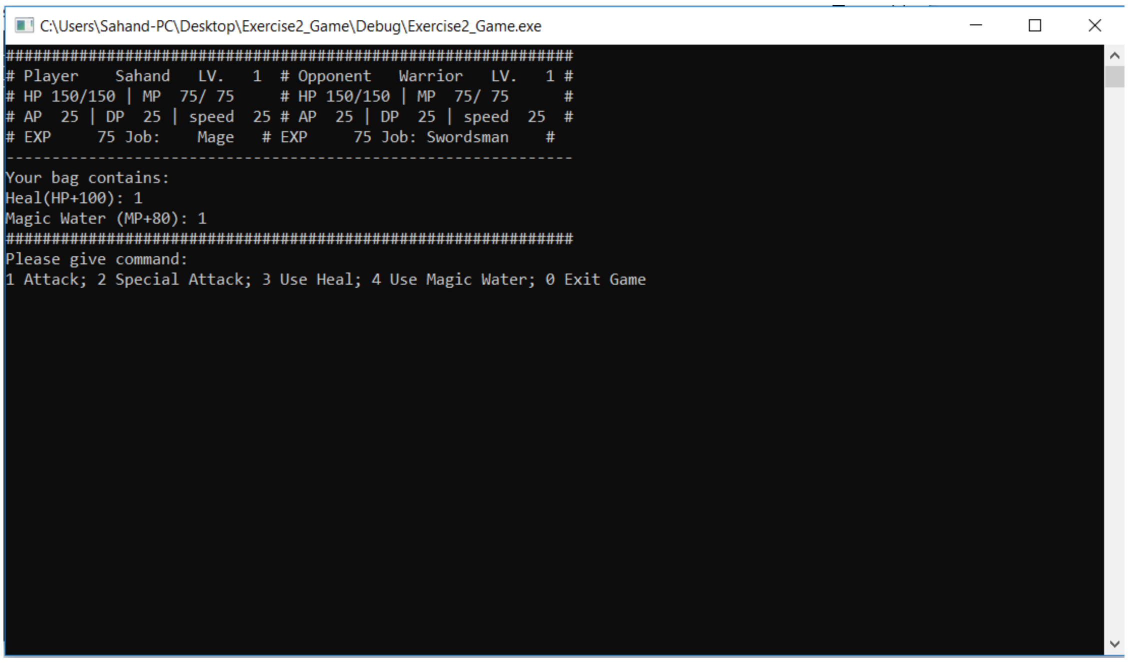
Figure 1: choosing the character's name and class



```
STAGE 1
Your opponent, a Level 1 Swordsman.
Press any key to continue . . .
```

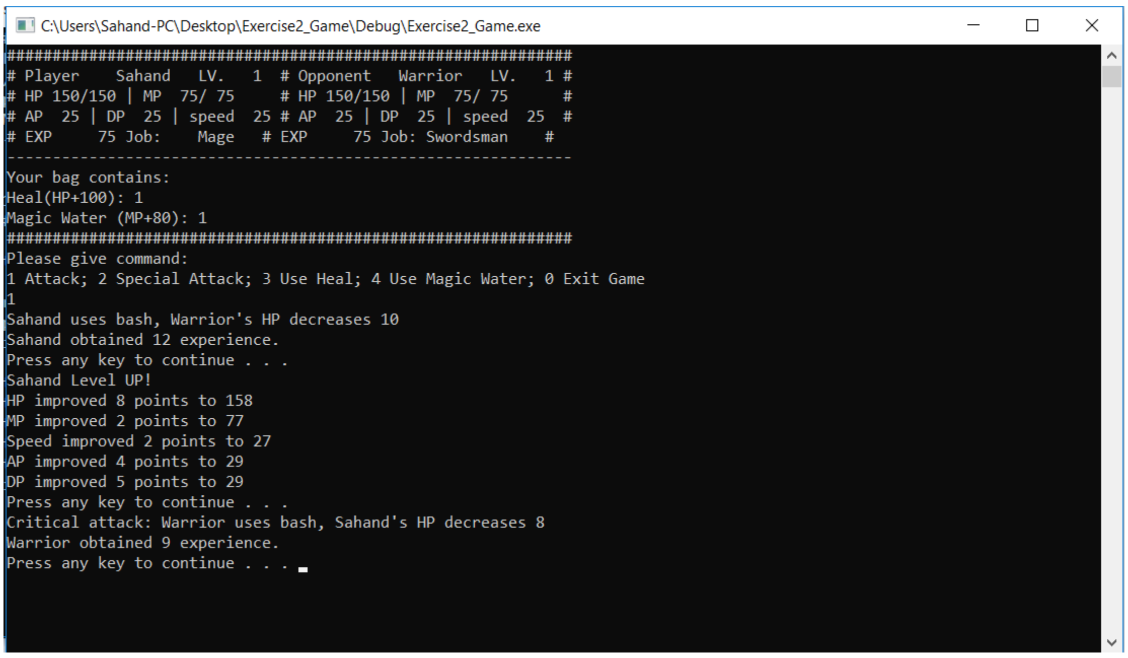
Figure 2: a randomly generated enemy

The second part is the gameplay. The flow of the game should not stop; meaning the player and the enemy should take turns in choosing what they want to do at each stage while the health level, the magic water level, etc. change according to the taken actions. In this part, the user is given 5 options to choose from. Each of them has to be tested in order to ensure full functionality of the game (Figures 3-8).


C:\Users\Sahand-PC\Desktop\Exercise2_Game\Debug\Exercise2_Game.exe

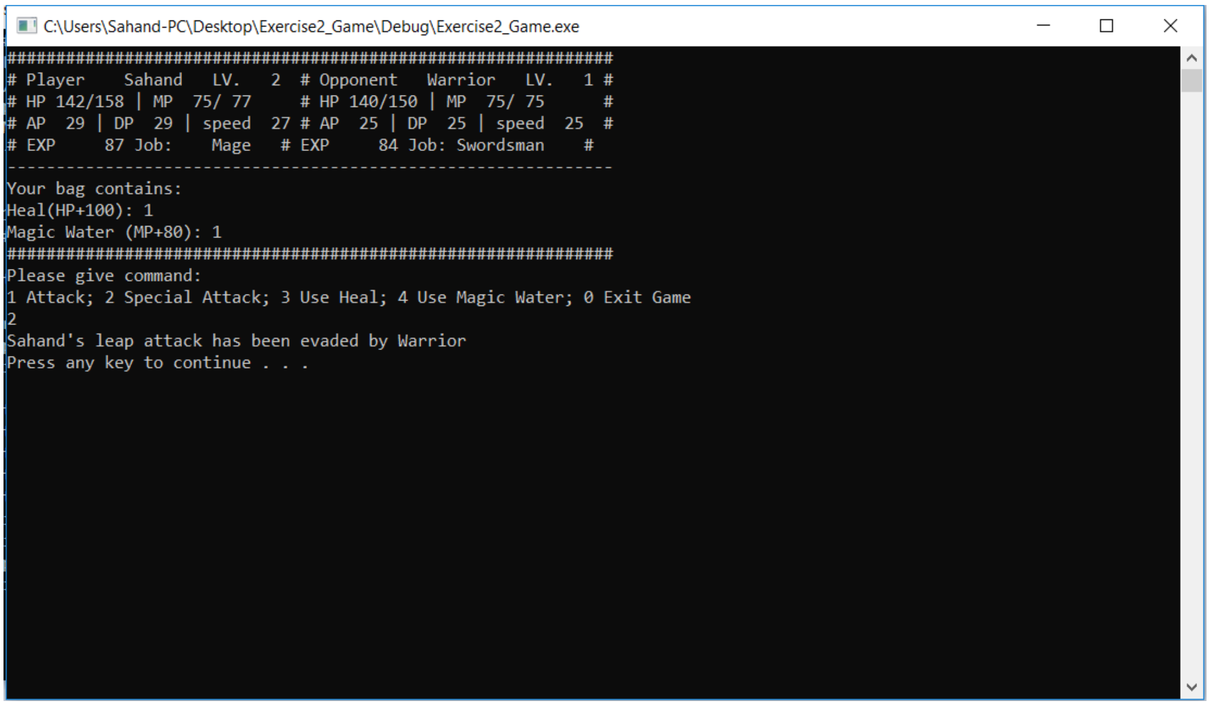
```
##############################################################
# Player    Sahand    LV.    1  # Opponent   Warrior    LV.    1 #
# HP 150/150 | MP  75/ 75      # HP 150/150 | MP  75/ 75      #
# AP  25 | DP  25 | speed  25 # AP  25 | DP  25 | speed  25  #
# EXP      75 Job:     Mage    # EXP      75 Job: Swordsman    #
--------------------------------------------------------------
Your bag contains:
Heal(HP+100): 1
Magic Water (MP+80): 1
##############################################################
Please give command:
1 Attack; 2 Special Attack; 3 Use Heal; 4 Use Magic Water; 0 Exit Game
```

Figure 3: showing the options and attributes to the user


C:\Users\Sahand-PC\Desktop\Exercise2_Game\Debug\Exercise2_Game.exe

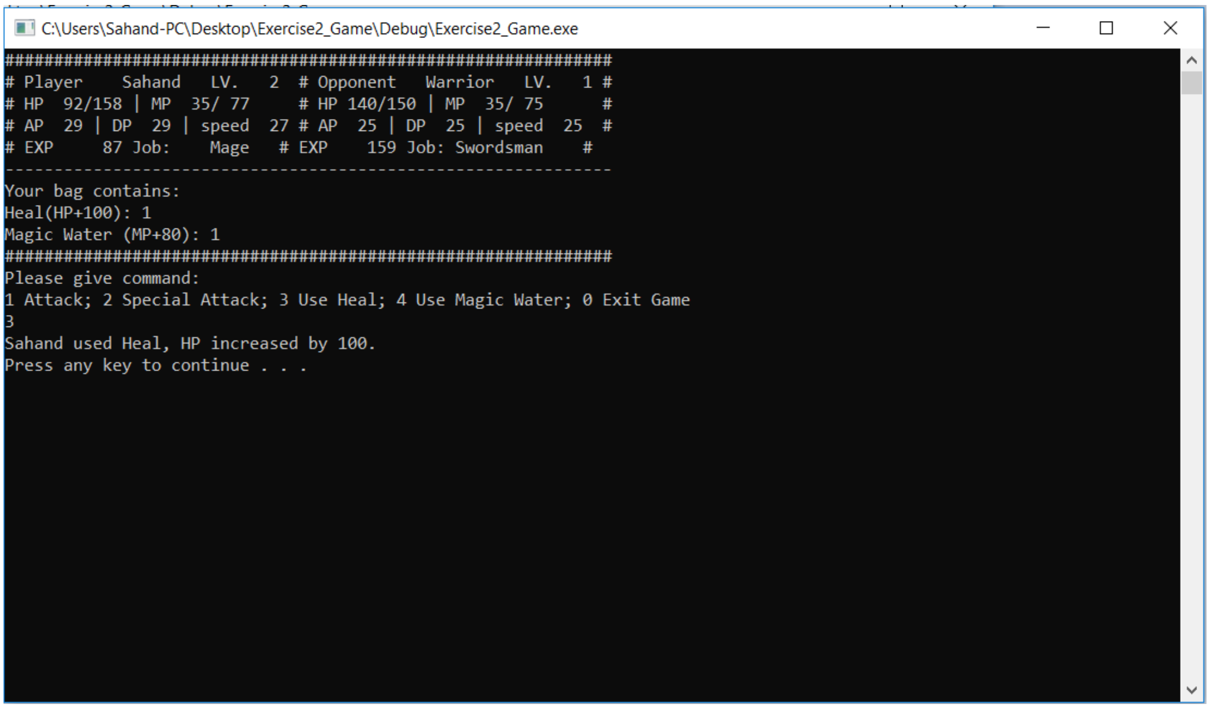
```
##############################################################
# Player    Sahand    LV.    1  # Opponent   Warrior    LV.    1 #
# HP 150/150 | MP  75/ 75      # HP 150/150 | MP  75/ 75      #
# AP  25 | DP  25 | speed  25 # AP  25 | DP  25 | speed  25  #
# EXP      75 Job:     Mage    # EXP      75 Job: Swordsman    #
--------------------------------------------------------------
Your bag contains:
Heal(HP+100): 1
Magic Water (MP+80): 1
##############################################################
Please give command:
1 Attack; 2 Special Attack; 3 Use Heal; 4 Use Magic Water; 0 Exit Game
1
Sahand uses bash, Warrior's HP decreases 10
Sahand obtained 12 experience.
Press any key to continue . . .
Sahand Level UP!
HP improved 8 points to 158
MP improved 2 points to 77
Speed improved 2 points to 27
AP improved 4 points to 29
DP improved 5 points to 29
Press any key to continue . . .
Critical attack: Warrior uses bash, Sahand's HP decreases 8
Warrior obtained 9 experience.
Press any key to continue . . .
```

Figure 4: using the Attack option


C:\Users\Sahand-PC\Desktop\Exercise2_Game\Debug\Exercise2_Game.exe

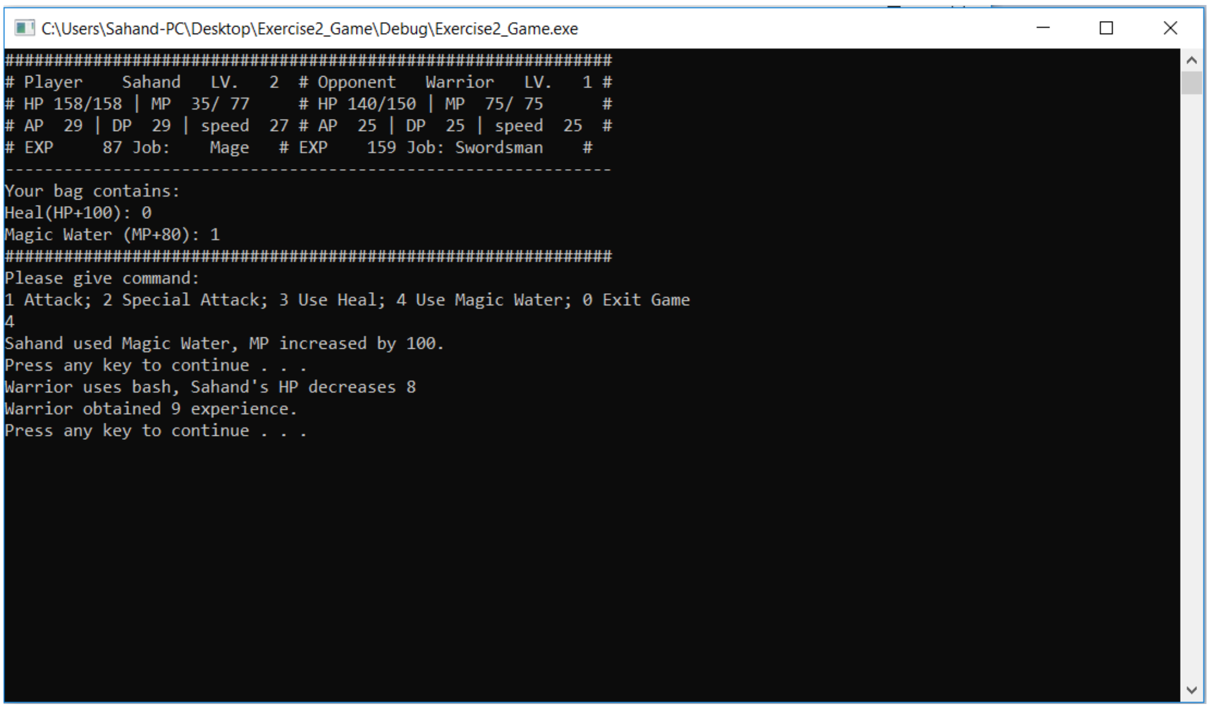
```
##############################################################
# Player    Sahand    LV.   2  # Opponent   Warrior    LV.   1 #
# HP 142/158 | MP  75/ 77     # HP 140/150 | MP  75/ 75      #
# AP  29 | DP  29 | speed  27 # AP  25 | DP  25 | speed  25  #
# EXP     87 Job:    Mage   # EXP     84 Job: Swordsman    #
--------------------------------------------------------------
Your bag contains:
Heal(HP+100): 1
Magic Water (MP+80): 1
##############################################################
Please give command:
1 Attack; 2 Special Attack; 3 Use Heal; 4 Use Magic Water; 0 Exit Game
2
Sahand's leap attack has been evaded by Warrior
Press any key to continue . . .
```

Figure 5: using the Special Attack option


C:\Users\Sahand-PC\Desktop\Exercise2_Game\Debug\Exercise2_Game.exe

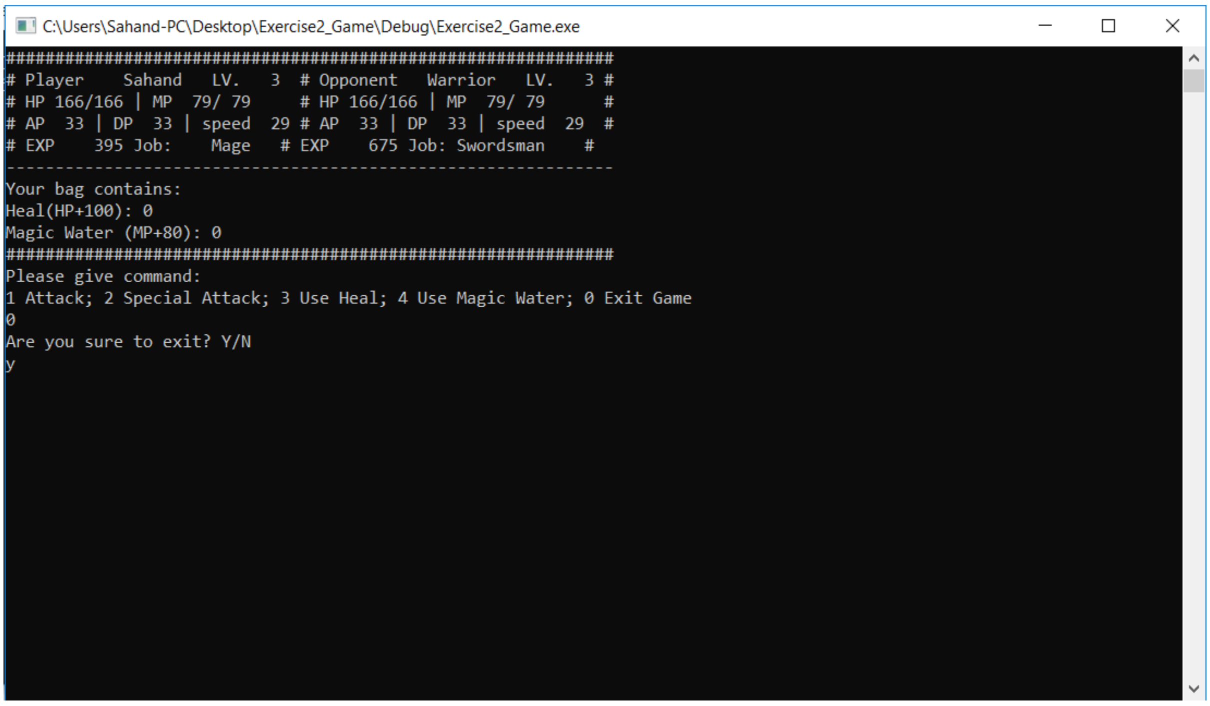
```
##############################################################
# Player    Sahand    LV.   2  # Opponent   Warrior    LV.   1 #
# HP  92/158 | MP  35/ 77     # HP 140/150 | MP  35/ 75      #
# AP  29 | DP  29 | speed  27 # AP  25 | DP  25 | speed  25  #
# EXP     87 Job:    Mage   # EXP    159 Job: Swordsman    #
--------------------------------------------------------------
Your bag contains:
Heal(HP+100): 1
Magic Water (MP+80): 1
##############################################################
Please give command:
1 Attack; 2 Special Attack; 3 Use Heal; 4 Use Magic Water; 0 Exit Game
3
Sahand used Heal, HP increased by 100.
Press any key to continue . . .
```

Figure 6: Using the Use Heal option


C:\Users\Sahand-PC\Desktop\Exercise2_Game\Debug\Exercise2_Game.exe

```
###############################################################
# Player    Sahand   LV.   2  # Opponent   Warrior   LV.   1 #
# HP 158/158 | MP  35/ 77     # HP 140/150 | MP  75/ 75      #
# AP  29 | DP  29 | speed  27 # AP  25 | DP  25 | speed  25  #
# EXP     87 Job:    Mage   # EXP    159 Job: Swordsman    #
---------------------------------------------------------------
Your bag contains:
Heal(HP+100): 0
Magic Water (MP+80): 1
###############################################################
Please give command:
1 Attack; 2 Special Attack; 3 Use Heal; 4 Use Magic Water; 0 Exit Game
4
Sahand used Magic Water, MP increased by 100.
Press any key to continue . . .
Warrior uses bash, Sahand's HP decreases 8
Warrior obtained 9 experience.
Press any key to continue . . .
```

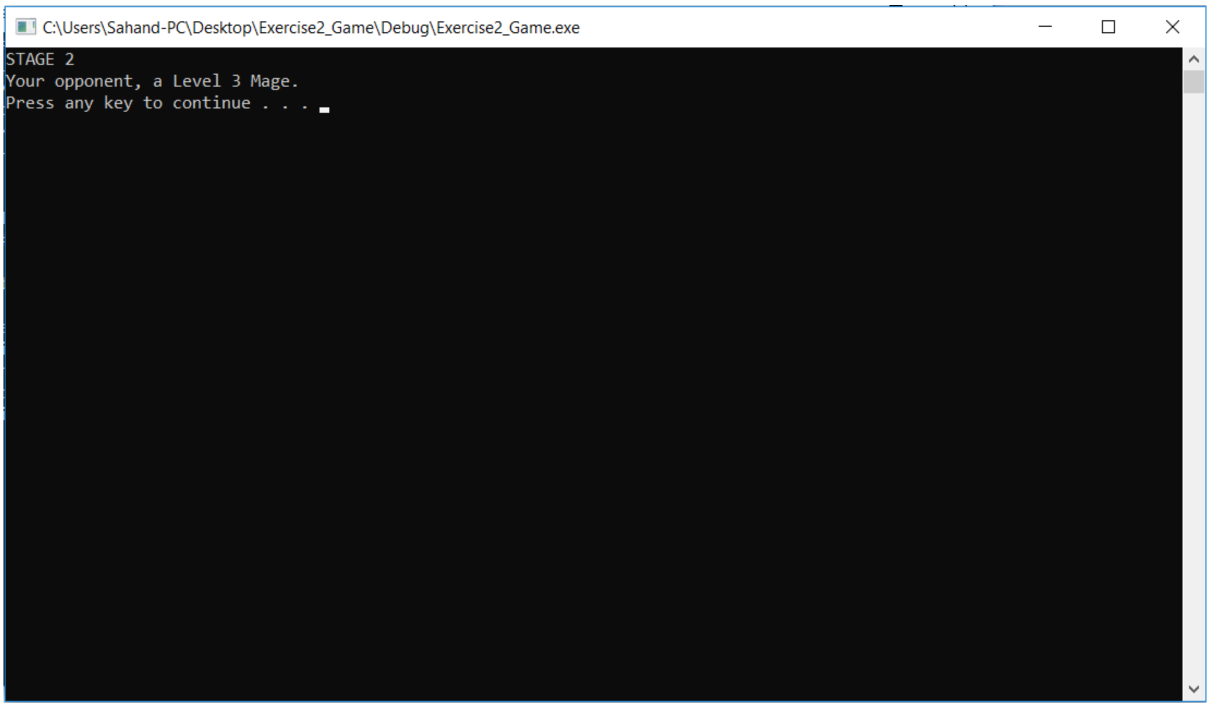
Figure 7: using the Use Magic Water option


C:\Users\Sahand-PC\Desktop\Exercise2_Game\Debug\Exercise2_Game.exe

```
###############################################################
# Player    Sahand   LV.   3  # Opponent   Warrior   LV.   3 #
# HP 166/166 | MP  79/ 79     # HP 166/166 | MP  79/ 79      #
# AP  33 | DP  33 | speed  29 # AP  33 | DP  33 | speed  29  #
# EXP    395 Job:    Mage   # EXP    675 Job: Swordsman    #
---------------------------------------------------------------
Your bag contains:
Heal(HP+100): 0
Magic Water (MP+80): 0
###############################################################
Please give command:
1 Attack; 2 Special Attack; 3 Use Heal; 4 Use Magic Water; 0 Exit Game
0
Are you sure to exit? Y/N
y
```

Figure 8: using the Exit Game option

The third and final part is the ending of the game. In this stage, the user has to be prompted with a message informing them whether they have won and can move on to the next stage or have lost and the game is over (Figure 9).



C:\Users\Sahand-PC\Desktop\Exercise2_Game\Debug\Exercise2_Game.exe

```
STAGE 2
Your opponent, a Level 3 Mage.
Press any key to continue . . .
```

Figure 9: Completion of first stage and progression to the next stage