



Xi'an Jiaotong-Liverpool University
西交利物浦大学

Department of Electrical and Electronic Engineering

CPT106(S2) C++ Programming and Software Eng. II

CPT106 C++ Programming Report

Name :Tiankuo.Jiao

Student ID :1929098

Catalogue

1.Specification (Problem statement)	3
1.1 For exercise 1	3
1.2 For exercise 2	3
2.Analysis	3
2.1 For exercise 1	3
2.1.1 Main menu flow chart.....	4
2.1.2 On an input.....	4
2.1.3 On an output.....	5
2.1.4 Variables.....	5
2.2 For exercise 2	6
2.2.1 Hierarchy chart.....	6
2.2.2 On an input	7
2.2.3 On an output.....	7
3.Design	7
3.1 For exercise 1	7
3.1.1 Specific design steps.....	8
3.2 For exercise 2	9
4.Test	10
4.1 For exercise 1	10
4.2 For exercise 2	14

1. Specification (Problem statement)

1.1 For exercise 1

In this exercise, we need to create a sub-class: iFraction, from the base class, Fraction. iFraction represents the improper fraction. For example:

$1\frac{3}{5}$. Since I did not use operator overloading in Assignment 1, this time I need

to replace the operation with operator overloading. Not only do we have to implement all the functions of Assignment 1, we also need to design an external function **convertF**. Convert a fraction into an improper fraction. For

example, we can convert $\frac{8}{5}$ into $1\frac{3}{5}$.

1.2 For exercise 2

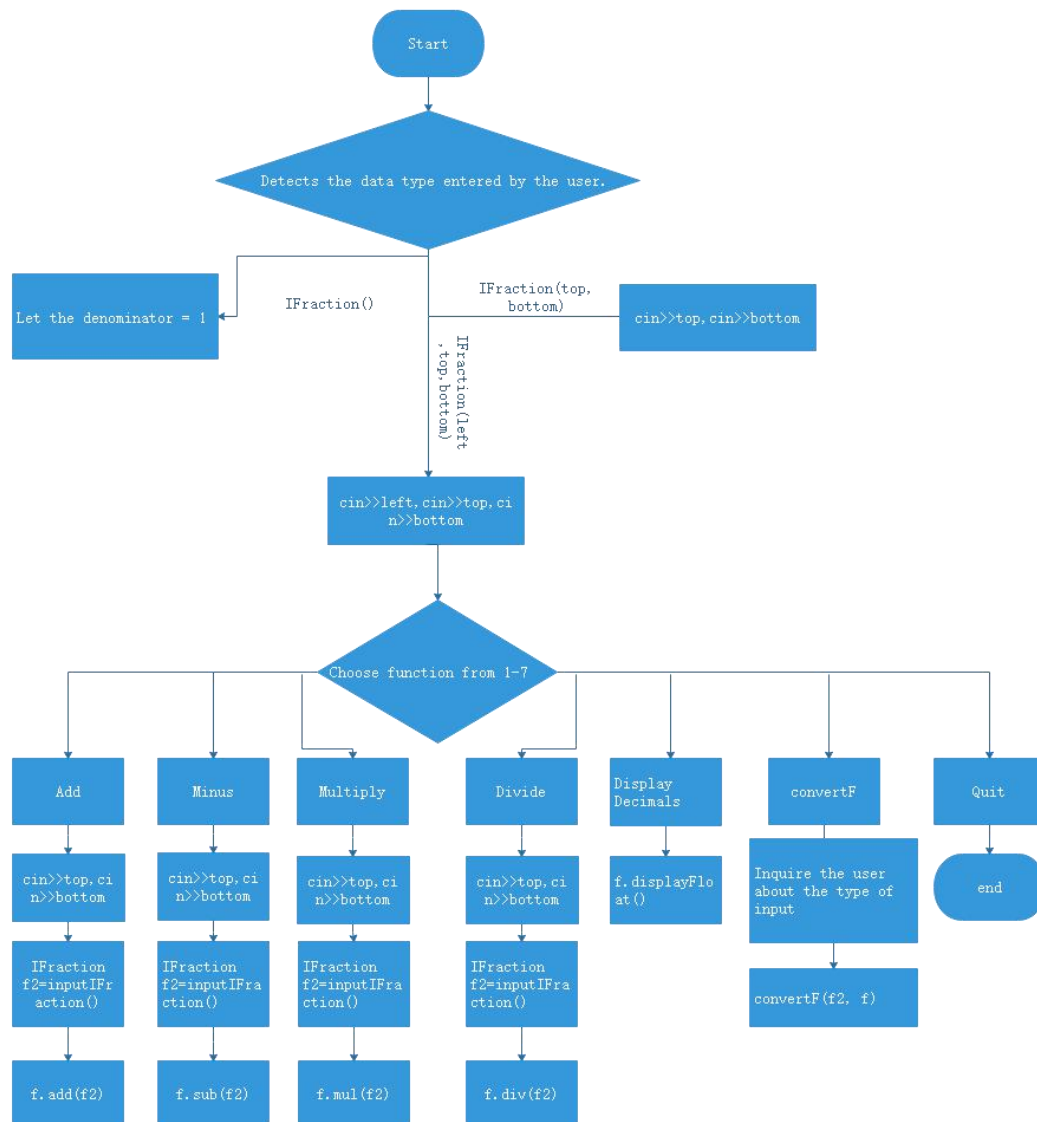
We need to design a computer game. On the basis of filling the blank, we re-completed the role information and added functions. This includes different types of players. Players can choose a job: 1 Swordsman, 2 Archer, 3 Mage. In addition, we designed the Luck value in the game so that if the player's Luck value is high, some attacks will not be effective (missing). In addition, we assign different HP and MP to each occupation according to its characteristics. In the experience of the game, players are better able to make choices based on each occupation. In order to achieve a better game experience, we set up a lucky value. **For Swordsman, we set a 3% lucky value, which means the opponent has a 3% chance of missing ; For Archer, we set the luck value to 5%; For mage we set our luck value to 7%. Moreover, we set a "Boss Dragon". The boss will fight against the brave man in the second game level. Good luck to you!!!**

2. Analysis

2.1 For exercise 1

In order to change the fraction into an improper fraction, we designed the convertF function. In addition, since I did not use operator overloading in the first Assignment, I modified the program in this Assignment.

2.1.1 Main menu flow chart



2.1.2 On an input

Just like the assignment 1, in the beginning, we would like to let the user choose the form of the input fraction. For example:

- 1) IFraction a; represents 0/1
- 2) IFraction b(3,4); represents 3/4
- 3) IFraction c(5); represents 5/1
- 4) IFraction(1,3,5); represents $1\frac{3}{5}$

2.1.3 On an output

When the user input fraction which can be simplified, for example: 3/9. The program will convert it to 1/3, and print the result into the interface.

When the user enters a denominator that contains a "-", the program puts the minus sign on the numerator and prints it out.

When the user choose the convertF function, the program will execute the operator which can convert the fraction to the mixed fraction. Finally, the result will be printed in the screen to the user.

After the user chooses the function, the program needs to execute the operation according to the user's choice. After each operation, the user has the option to convert the fraction to decimal.

When the user chooses to quit, the system will automatically quit.

2.1.4 Variables

We declare "top" and "bottom" in our private class to represent the numerator and denominator of a fraction.

In the common divisor function, we define two variables a and b. In the divisor function, we find the greatest common divisor.

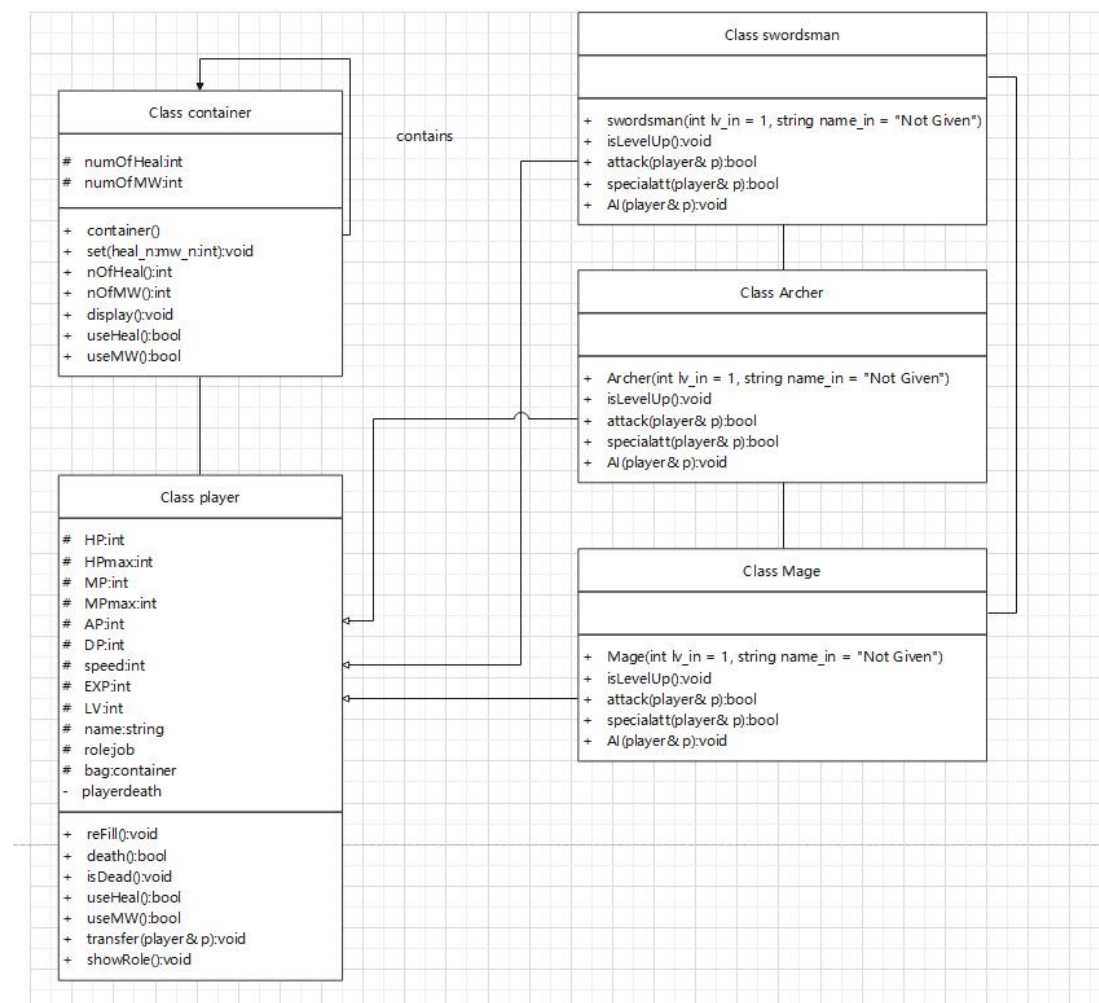
In the add function, we newly define b and t to store common divisors and calculated results, and finally get the final result through common divisors.

We consider that convertF is a friend function of Fraction. It's also a friend function of iFraction. Moreover, Convertf, a friend of a class, has access to all private and protected members of the class. So the variables for convertF are already defined in the class.

2.2 For exercise 2

We first filled in some of the blanks in the source code. Since Archer and Mage are similar to Swordsman, we write the functions of Archer and Mage according to Swordsman. We consider player to be parent class, which defines the common features of a group of objects, and we think archer and mage are sub-class, which has more specific properties and methods. We also designed a hierarchy based on the three roles. According to our design, the game can execute normally.

2.2.1 Hierarchy chart



2.2.2 On an input

We first asked the user to enter a character. The user could choose one of the three roles: 1.Swordman 2.Archer 3. Mage, and then choose the character of his opponent.

After entering the game, we can get information about ourselves and our opponent, including health, MP value, etc. Next, we can select from the functions: 1 Attack; 2 Special Attack; 3 Use Heal; 4 Use Magic Water; 0 Exit Game

Players need to select the functions from the interface. Moreover, heal water should be used when HP value is low.

Use Magic water when our health or MP value is low.

2.2.3 On an output

Once you enter the game, the game will show you the basic information. Include name, level, HP, MP, AP, DP, Speed, EXP, Job, etc.

Once the conditions are satisfied, the player levels up, increasing a series of numerical values.

In the second level, the player will encounter a Boss. Be careful!!!

With every attack, the player can miss, critical attack. Every attack value to an opponent will be displayed on the screen.

If the opponent's HP drops to 0, the player wins, otherwise, the opponent wins.

3.Design

3.1 For exercise1

Since this task was based on Assignment 1, we retained the basic functions of Assignment 1, and we rewrote the operation of Assignment 1 with operator overloading as required. We use operator overloading for addition, subtraction, multiplication and division. **On the basis of Assignment 1, we have added the operation of changing the fraction to an improper fraction.**

3.1.1 Specific design steps:

1) We first declare the numerator and denominator in the private class: top and bottom

Next, we ask the user for input methods, including:

- 1) IFraction a; represents 0/1
- 2) IFraction b(3,4); represents 3/4
- 3) IFraction c(5); represents 5/1

2) In the public class, we perform the operation of placing the minus sign on the numerator. This part of the operation involves the overloading. Operator overloading is the definition of a function that implements the desired operation inside the function. When the operator is used, the compiler automatically calls the function.

3) After placing the minus sign in the numerator, we declare several functions, such as addition, subtraction, multiplication and division, that we need to use later.

4) We set up a function for the greatest common divisor

5) Through the function established above, we establish the divisor function, and calculate on the basis of obtaining the common factor

6) Next, we need to create the addition function, and once we get the result, we also need to simply the result

7) Subtraction is achieved by referring to the addition function by adding a minus sign to the numerator

8) After the user enters the second fraction, we multiply the second fraction by the first fraction

9) For division, after the user enters the fraction, we switch the numerator and denominator of the fraction, referencing the multiplication function

10) For converting ordinary fractions to improper fractions, we designed the convertF function

11) In building the main function, we first need to ask the user to input the numerator and denominator. If we want to do something later, we need to ask the user to enter it a second time. Based on this, we set up a **switch** function to select the function

3.2 For exercise 2

1) We first filled the blanks in the program, and optimized the main function.

2) For Archer and Mage, we defined them as sub-classes. Following the example of Swordman, we wrote header files and program contents of Archer and Mage.

3) For each class, we changed the luck value to 3% for Swordman, 5% for Archer, and 7% for Mage.

4) For Archer, we set HP to 130 and MP to 90

5) For Mage, we set HP to 120 and MP to 120

6) If the player's HP or MP value needs to be replenished in the game, we have implemented the function of supplementing HP and MP through the source code.

7) Once a player's health is reduced to 0, the game is over.

8) We designed a Boss “Dragon” in the second level. The Boss has a high HP and MP, which players should be careful.

4. Test

4.1 For exercise 1

Test 1: Users choose 1: IFraction ()

```
1. IFraction()  
2. IFraction(top, bottom)  
3. IFraction(top)  
4. IFraction(left, top, bottom)  
Enter your choice:1  
current ifraction: 0  
decimals: 0
```

Test 2: The user enters the numerator and the denominator

```
1. IFraction()  
2. IFraction(top, bottom)  
3. IFraction(top)  
4. IFraction(left, top, bottom)  
Enter your choice:2  
Enter Numerator: 1  
Enter Denominator: 2  
current ifraction: 1/2  
decimals: 0.5
```

Test 3:The user enters the numerator

```
1. IFraction()  
2. IFraction(top, bottom)  
3. IFraction(top)  
4. IFraction(left, top, bottom)  
Enter your choice:3  
Enter Numerator: 1  
current ifraction: 1/1  
decimals: 1
```

Test 4: Users inputed the mixed fraction

```
1. IFraction()
2. IFraction(top, bottom)
3. IFraction(top)
4. IFraction(left, top, bottom)
Enter your choice:4
Enter Left: 1
Enter Numerator: 2
Enter Denominator: 3
current ifraction:
           2
    1-----
           3
decimals: 1.66667
```

Test 5: The user chooses to convert "Fraction" into "mixed Fraction"

```
1. operator+
2. operator-
3. operator*
4. operator/
5. Display Decimals
6. convertF
7. Quit
Your Choice: 6
1. Fraction()
2. Fraction(top, bottom)
3. Fraction(top)
Enter your choice:2
Enter Numerator: 13
Enter Denominator: 6
current fraction: 13/6
current ifraction:
           1
    2-----
           6
```

Test 6: The user selects addition

```

1. IFraction()
2. IFraction(top, bottom)
3. IFraction(top)
4. IFraction(left, top, bottom)
Enter your choice: 2
Enter Numerator: 1
Enter Denominator: 2
current ifraction: 1/2
decimals: 0.5
1. operator+
2. operator-
3. operator*
4. operator/
5. Display Decimals
6. convertF
7. Quit
Your Choice: 1
1. IFraction()
2. IFraction(top, bottom)
3. IFraction(top)
4. IFraction(left, top, bottom)
Enter your choice: 2
Enter Numerator: 1
Enter Denominator: 2
current ifraction: 1/1

```

Test 7: The user selects subtraction

```

1. IFraction()
2. IFraction(top, bottom)
3. IFraction(top)
4. IFraction(left, top, bottom)
Enter your choice: 2
Enter Numerator: 1
Enter Denominator: 2
current ifraction: 1/2
decimals: 0.5
1. operator+
2. operator-
3. operator*
4. operator/
5. Display Decimals
6. convertF
7. Quit
Your Choice: 2
1. IFraction()
2. IFraction(top, bottom)
3. IFraction(top)
4. IFraction(left, top, bottom)
Enter your choice: 2
Enter Numerator: 1
Enter Denominator: 3
current ifraction: 1/6

```

Test 8: The user selects multiplication

```

1. IFraction()
2. IFraction(top, bottom)
3. IFraction(top)
4. IFraction(left, top, bottom)
Enter your choice:2
Enter Numerator: 1
Enter Denominator: 2
current ifraction: 1/2
decimals: 0.5
1. operator+
2. operator-
3. operator*
4. operator/
5. Display Decimals
6. convertF
7. Quit
Your Choice: 3
1. IFraction()
2. IFraction(top, bottom)
3. IFraction(top)
4. IFraction(left, top, bottom)
Enter your choice:2
Enter Numerator: 1
Enter Denominator: 3
current ifraction: 1/6

```

Test 9: The user selects the division

```

1. IFraction()
2. IFraction(top, bottom)
3. IFraction(top)
4. IFraction(left, top, bottom)
Enter your choice:2
Enter Numerator: 1
Enter Denominator: 2
current ifraction: 1/2
decimals: 0.5
1. operator+
2. operator-
3. operator*
4. operator/
5. Display Decimals
6. convertF
7. Quit
Your Choice: 4
1. IFraction()
2. IFraction(top, bottom)
3. IFraction(top)
4. IFraction(left, top, bottom)
Enter your choice:2
Enter Numerator: 1
Enter Denominator: 3
current ifraction:

```

$$\begin{array}{r} 1 \\ 1 \overline{) 1} \\ \underline{1} \\ 2 \end{array}$$

Test 10: The user chooses to change the fraction to a decimal

```
current ifraction:
                1
                2-----
                6
decimals: 2.16667
```

For exercise 2

Test 1: Enter the user name and select the occupation

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

Test 2: Choose your opponent's occupation

```
Please choose your opponent a job: 1 Swordsman, 2 Archer, 3 Mage
1
```

Test 3: The game interface

```
#####
# Player      kkk  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: Swordsman # 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
```


Test 4: The user selects normal attack 1

```
#####
# Player      kkk  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: Swordsman # 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
kkk uses bash, Warrior's HP decreases 9
kkk obtained 10 experience.
```

Test 5: When the user upgrades, the corresponding attribute value increases

```
#####
# Player      kkk  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: Swordsman # 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
kkk uses bash, Warrior's HP decreases 9
kkk obtained 10 experience.
请按任意键继续. . .
kkk 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
```

Test 6: Opponent attacks and gains experience

```
Warrior uses leap attack, kkk's HP decreases 50
Warrior obtained 75 experience.
```

Test 7: The user selects a special attack

```
Please give command:
1 Attack; 2 Special Attack; 3 Use Heal; 4 Use Magic Water; 0 Exit Game
2
kkk uses leap attack, Warrior's HP decreases 50
kkk obtained 75 experience.
```

Test 8: Users choose treatment function, replenishing HP value

```
Please give command:
1 Attack; 2 Special Attack; 3 Use Heal; 4 Use Magic Water; 0 Exit Game
3
kkk used Heal, HP increased by 100.
```

Test 9: The user selects 4:Magic water and reverts to MP

```
Please give command:
1 Attack; 2 Special Attack; 3 Use Heal; 4 Use Magic Water; 0 Exit Game
4
kkk used Magic Water, MP increased by 100.
```

Test 10: The user exits and enters Y

```
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
```

Test 11: Both the user and the opponent choose the Archer occupation


```
#####
# Player      kkk    LV.    1  # Opponent  Warrior  LV.    1  #
# HP 130/130 | MP  90/ 90  # HP 130/130 | MP  90/ 90  #
# AP  25 | DP  25 | speed 25 # AP  25 | DP  25 | speed 25 #
# EXP      75 Job:  Archer  # EXP      75 Job:  Archer  #
#####
Your bag contains:
Heal(HP+100): 1
Magic Water (MP+80): 1
#####
```

Test 12: Both the user and opponent choose Mage occupation

```
#####
# Player      kkk    LV.    1  # Opponent  Warrior  LV.    1  #
# HP 120/120 | MP 120/120  # HP 120/120 | MP 120/120  #
# AP  25 | DP  25 | speed 25 # AP  25 | DP  25 | speed 25 #
# EXP      75 Job:   Mage  # EXP      75 Job:   Mage  #
#####
Your bag contains:
Heal(HP+100): 1
Magic Water (MP+80): 1
#####
```

Test 13: User's Victory

```
Please give command:
1 Attack; 2 Special Attack; 3 Use Heal; 4 Use Magic Water; 0 Exit Game
1
kkk uses bash, Warrior's HP decreases 16
kkk obtained 19 experience.
请按任意键继续. . .
Warrior is Dead.
请按任意键继续. . .
YOU WIN
kkk got0 Heal, and 0 Magic Water.
```

Test 14: The opponent evaded the user's attack

```
Please give command:
1 Attack; 2 Special Attack; 3 Use Heal; 4 Use Magic Water; 0 Exit Game
1
miss attack:
```

Test 14: Boss "Dragon"

```
#####
# Player      kkk  LV.  3 # Opponent  dragon  LV.  3 #
# HP 166/166 | MP  79/ 79 # HP 196/196 | MP 134/134 #
# AP  33 | DP  33 | speed 29 # AP  33 | DP  33 | speed 29 #
# EXP   413 Job: Swordsman # EXP   675 Job:   Mage   #
-----
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
```