計算機程式設計 HW12report

103061223 李俊穎

1. Introduction of the problem and your solution

The homework 12 is a board game similar Game of Life. This time we need to design a game's rule by ourselves. However, this time we have been given most of the code. Actually, the problems in this homework are very simple: you need to realize how the codes structures and how they work. There are many concepts contained in hpp. and cpp. such as pointer, constructor, inheritance, virtual member function, and class and vector. Nevertheless, it took me a lot of time to realize the codes because the functions and the pointers are too complex! But the application to it is not hard if you understand what the codes meaning, just need to call the function correctly then this homework can be complete.

2. Implementation details ,Additional features of your program (data structure, flows and algorithms)

In the first step, I adapt the map which given by instructor and redesign the story of the Square

After adapting all Squares, then I use the if-else if to judge the player's position description. If the condition is corresponding, then I call the correct function from the library.

```
if(player[i].Position()->Description()=="You get married with a woman!")//if the Square's Des
{
    if(player[i].Reportmarried().Reportmarry()!=true)//if this player haven't married before,
        player[i].GetMarry();
    else
        cout<<"you have already have a wife so you can't married with another woman."<<endl;
}
```

The other if –else if in my program are do the similar thing as the following. I also design a feature Square is to send the player go back to the start position the following is the code:

```
PlayerGobackstart(map, player[i],Start);//reset the player to the start position
cout << "Player "<<i<<" is at: " <<  player[i].Position()->Description() << endl;
```

The function of it is just pass the start square to set position.

```
void PlayerGobackstart(Map& map, Player& player,Square *Start
{
    player.setPosition(Start);
}
```

In addition, in some condition like get married and have a baby. I write the judge equation to ensure the player won't have more than two wives and have a baby without get married. The following is the solution:

```
if(player[i].Reportmarried().Reportmarry()!=true)//if this player haven't married before,
    player[i].GetMarry();
else
    cout<<"you have already have a wife so you can't married with another woman."<<endl;
```

I directly return the married object in the player class and check if one of element is true. If so, then this player have got married and I don't give him a wife again.

After the player throw the dice and finish his round, I will call the function to report his properties and the total value of it.

```
player[i].ReportValues();//report the property item
cout << "The player "<<i<<" now has ";
cout << player[i].ComputeTotalValue() << endl;//report the total value of player's properties
cout << endl;
```

Finally, I set the do-while loop to end the game. If the dice appear 6 more than six times, then the do-while will make this game stop.

```
}while(endgame<6);//When the number 6 throwed by rolldice appear over six times,then the game over.
```

Then my program will finish!