# C language（Group 14）：Magic tower

## Abstract

"Magic tower" was founded in 2005, is a classic RPG strategy game, even today, there are still tens of thousands of magic tower enthusiasts in China. Through the optimization of the past, it has thousands of different versions. For example, the web version of magic tower (V2.3) allows players to play online with their friends without downloading i.e. expect that we can reproduce the functions in C and implement the basic operational functions (we also write code in C++ if necessary).At the same time, we added some of our own ideas, such as the new magic tower map. We hope that our works can give you the enjoyment of the classic games.

## 1. Introduction & The problem statement

#### 1.1Introduction

Limited to our level and experience, in this project, we designed a 21-layer base magic tower, including the core elements of the game setting of magic tower, fighting between warriors and monsters, interaction with NPCs and friendly props settings. In terms of mechanism, we put the key of a special mechanism door into a higher layer of monster encircles, so as to control players' ability to dig out the resources inside the mechanism door in the later stage and promote the balance of the game. The original plan was to set the hidden door mechanism that could trigger only in the specified location, but due to the difficulty of implementation and the consideration of player experience, the complicated mechanism of the mysterious tower was abandoned by us. Drama respect, namely warrior is done up by demon king plot martial arts, he challenges demon tower from 0 begin layer upon layer, obtained the person of noble help to restore a few actual strength in the way, defeat way small strange and true false boss demon king rescues the princess that is captured by demon king successfully finally. In terms of prop setting, we refer to the prop placement of legitimate games, and discuss to determine the reasonable placement position and corresponding quantity of props such as keys and magic potions. After the team cooperation development, finally completed the project of the 21 - floor magic tower.

Limited to our level and experience, in this project, we designed a 21-layer base magic tower, including the core elements of the game setting of magic tower, fighting between warriors and monsters, interaction with NPCs and friendly props settings. In terms of mechanism, we put the key of a special mechanism door into a higher layer of monster encircles, so as to control players' ability to dig out the resources inside the mechanism door in the later stage and promote the balance of the game. The original plan was to set the hidden door mechanism that could trigger only in the specified location, but due to the difficulty of implementation and the consideration of player experience, the complicated mechanism of the mysterious tower was abandoned by us. Drama respect, namely warrior is done up by demon king plot martial arts, he challenges demon tower from 0 begin layer upon layer, obtained the person of noble help to restore a few actual strength in the way, defeat way small strange and true false boss demon king rescues the princess that is captured by demon king successfully finally. In terms of prop setting, we refer to the prop placement of legitimate games, and discuss to determine the reasonable placement position and corresponding quantity of props such as keys and magic potions. After the team cooperation development, finally completed the project of the 21 - floor magic tower.

#### 1.2Problem statement

1.2.0 New map drawing and transition between floors (ladder), beautify UI interface.

1.2.1 Dialogue between Hero and NPC: how to trigger, continue, and end? How to solve the problem of flickering dialog box when switching lines?

1.2.2 Notes on magic props: for beginners who have never been in contact with magic tower games, how to inform the player of the role of magic props, so as to help the player reasonably plan the route through the pass or the hero's attribute value? Can you display the information of a prop when the mouse moves to its position? Similarly, can you make a monster map so that players can act according to their abilities?

1.2.3 Hero vs monster: the magic tower game is the most important part of the battle, how to design the battle algorithm? Which of the various computational references on the web should we adopt? If none of them fit, design your own. In addition, due to the comparison of the 21-layer magic tower, the monster's strength level is "within reach". Is it appropriate to add "critical strike" for the hero and affect the game balance?

1.2.4 The decision of victory and defeat: when the player defeats the boss monster, should regard as the game victory. However, should you play the plot animation and quit the game when you defeat the boss, or not quit immediately, leaving players to explore and Esc to quit by themselves?

1.2.5 How to use mouse and keyboard to successfully control the game?

### 1.3 Task Division

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Linzhi Huang | Guangxing Chen | Xiaowen Zhang |
| tasks | Engineering structure analysis  Conversion algorithm | Map  Preparation &implementation  Debugging | Project optimization  Code runtime adjustment |

## 2.Analyse

According to the trait of our project, when we begin to design our code，we decided to part the code in to three parts :

•The first period： Resource preparation, map construction and prop placement

in order to allow players to experience the satisfaction of recalling classic games and improve the restoration degree of the game, it is necessary to collect the resource pictures (heroes, props, monsters and obstacles) of the magic tower on the Internet at the early stage. And so on) and carries on the number naming, facilitates the two-dimensional matrix import. At the same time, make TXT dialog text. In 21 map production link, can consider the use of some tools such as "easy X" to reduce the burden of work, reference to the legitimate magic tower game map and props put, increase the rationality of the design.

•The second period: Each function corresponds to the function header file production

Assign the code writing of each algorithm to team members to complete their own parts. Key functions are accomplished in collaboration, such as using keyboard hooks and mice to control the movement of characters or menu cursors, battle boards, store boards, interactive conversations, ladders between floors, and more.

•The third period:

Main function writing (to define all window areas) and code integration. Integrate the assigned tasks into a project and cut down some redundant code. Then it turns to debugging and modification.

## 3. Design

3.1 Designing thought and basic rules

1. to design a successful magic tower，we’d better have a complete design idea. Which means that what characteristic are we about to bring our players with？In which way do we hope our players to success? All above should be considered at the very beginning of the design. For example: if we want players to go the attack route, then we can set up some higher defense and low health monster. On the contrary, if it is a balanced type, put even attack and defense and high health of the monster, often players will take the defensive route. A big no-no is incoherent thinking. If you have a bunch of solid monsters in front of you, causing too much difficulty in the early stage, and a bunch of monsters with high defense and low blood in the final stage, but don't keep up with enough resources to develop the hero attribute value, then this kind of player experience can be imagined. The production of magic tower, the game to grasp the high requirements. This is not a technical problem, but reasonable design ideas, the importance is of no doubt...

* 2.  Boring design to avoid

①The boring design of too many symmetrical maps needs to be avoided. No confusion, obvious routes, symmetrical maps generally have this drawback, if you want to challenge high standards, then this is often a major taboo for makers.

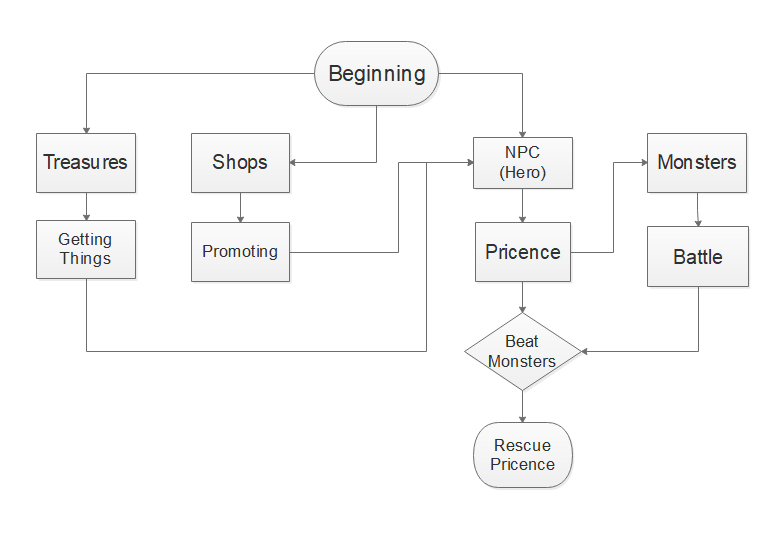
②Too much accumulation of monsters, often allow players to easily find the focus of this layer. The route is more direct and single. This should be avoided even by an ordinary writer.

③Tight key turnover

Difficult works often highlight the turnover of the key. A correct line is often the line that loses a lot of blood but retains the basic health value and consumes the least amount of keys. So, this line, often to set up stronger monsters. The other routes have to put some low monsters, so that the map will be difficult and confusing will be greatly increased.

To this end, especially on our own map, we need to test and test by ourselves, and finally plan a reasonable route. Then set up in the only way to place some strong monsters. At the same time, in the design route irrelevant door, put some intermediate monsters and reasonable subsidies resources, increase the clearance rate.

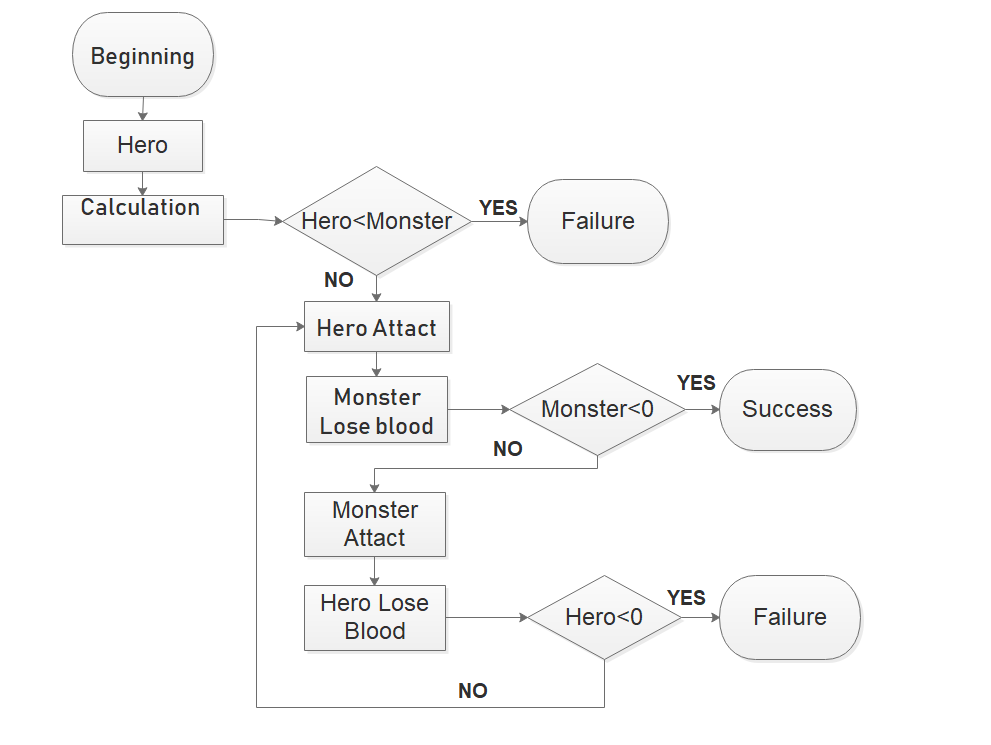
3.2 The total logic for magic tower



(Figure 1 Basic Logic)

3.3exaples for the key functions

3.3.1fighting

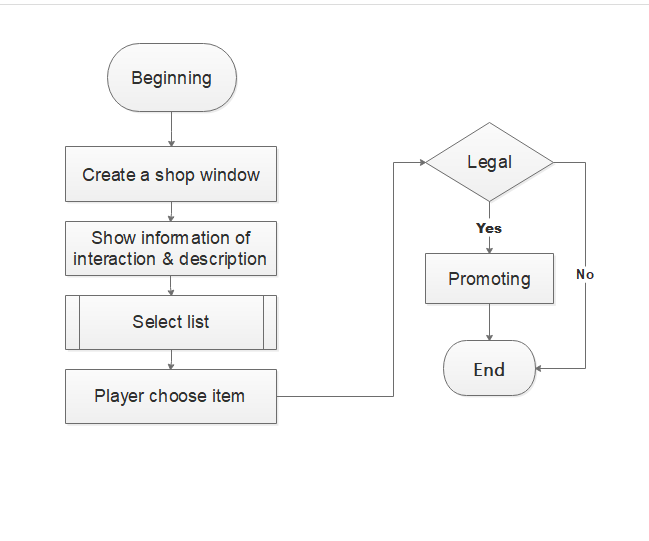


(Figure 2 Fighting Part)

As mentioned above, combat is the focus of our players, the details of which will be shown in the next section 4.The way we've designed the play is that this function is called when the player walks into the trigger play area. Before excitation against there will be a power comparison, at first we have dominated by people with strange value and comparing but found behind will lead to more complex judgement, so we will attack singly, if players attack value is greater than blame the defense value, is a poor damage ability value, where the player is on to the next step, odd players and attack values of armor, if a monster's attack value is greater than the players defensive ability is a poor get monster damage, otherwise the strange ability to damage value as a zero and against the window; Again next, each with their own life values divided by the number of enemy damage ability deserves to win round, if the player hero wins the round number is less than the monster, then inspire against the window, and against the window shows each round of the property value changes (need time control), finally will blame the damage to a player (minus the corresponding health value), to give the player's experience and gold information reflected in the bar, against the end. All previous situations that cause the death of a player's hero do not trigger a combat window. Instead, it only gives a friendly reminder of a lack of strength. After successfully defeating the monster, execute the function of crossing obstacles, similar to opening the door, replace the monster's code name with the code name representing the floor, and replace the location map with the floor map.

Other details participate in the implementation section.

3.3.2shopping

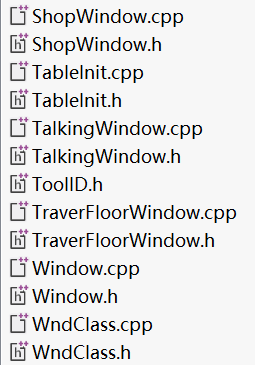
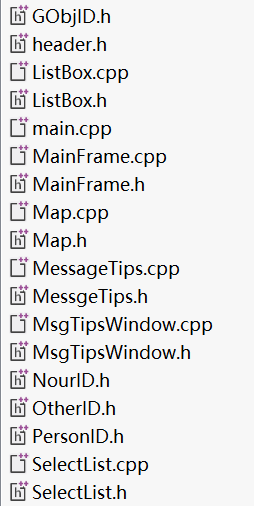
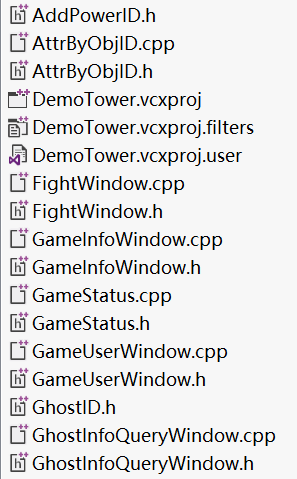


(Figure 3 Shop Part)

The magic tower sets up shops in the level where supplies are scarce. In some floors, there will be two shops. The "currency" of the shops has experience value, gold coin and even key, and the goods are attribute value gain or key. Shops are also important for customs clearance. According to our design, it is impossible for players to defeat the devil only by killing monsters and picking up props, and rational use of different types of shops can make customs clearance twice the result with half the effort. Algorithmic, the store logic chain is relatively clear, each choice is equivalent to making a cycle, complete or cancel the purchase is out of the cycle, as shown in the figure above. Among them, the Select list is a function specially written for the selection operation in the game. The legitimacy judgment is mainly to determine whether the player's currency is enough to buy the corresponding goods and whether the player chooses to quit and give up the purchase。

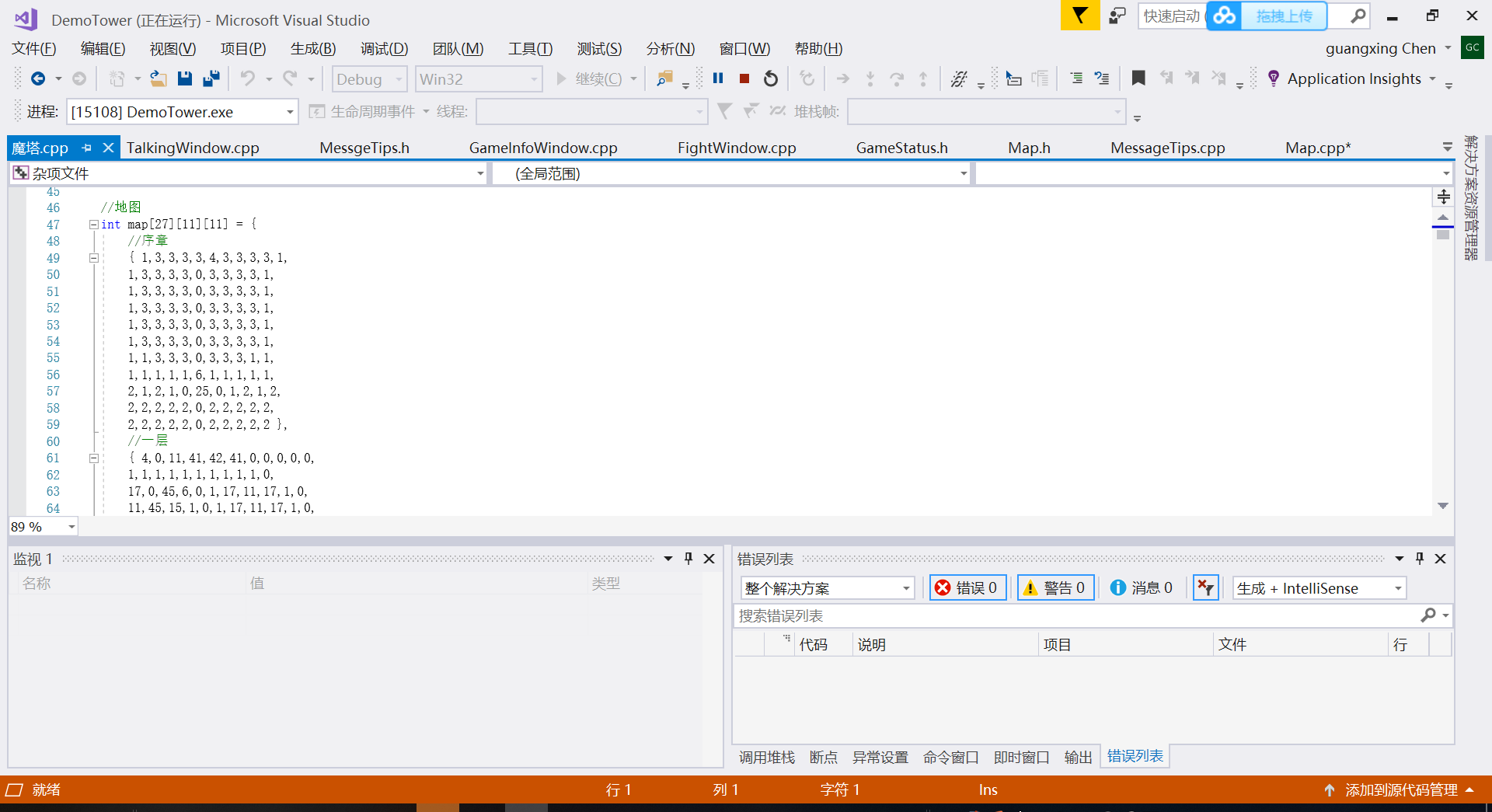
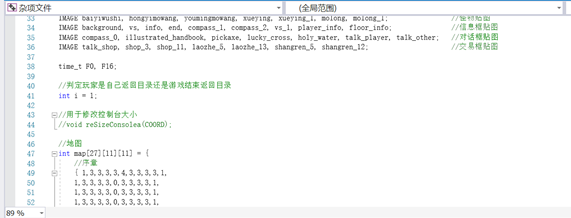
3.4the preview of the whole functions

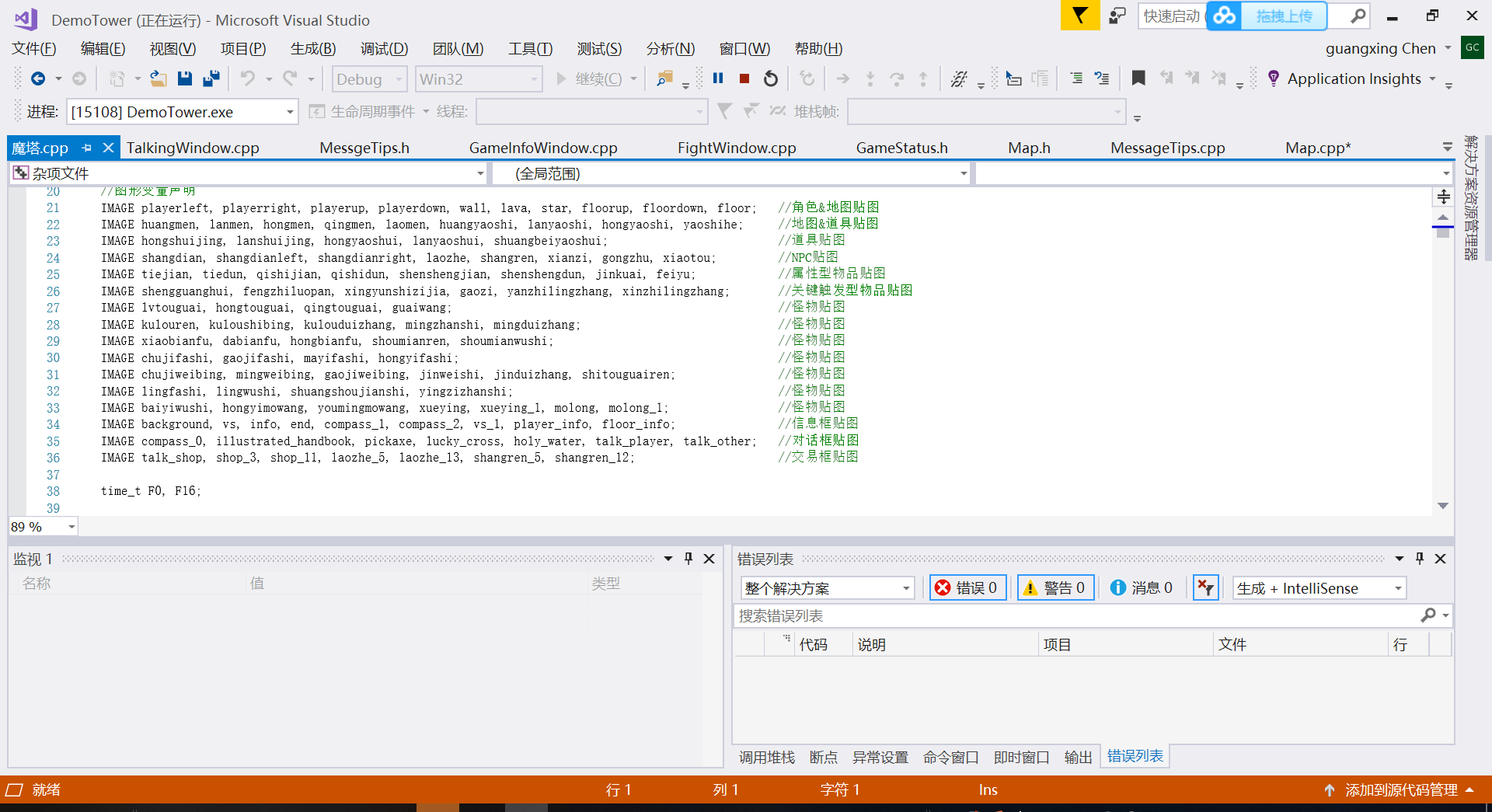
（Figure 4 All functions）



## 4 Implement

In this period it is high time to achieve the functions we mentioned above

* UI interface and map drawing are the same. Here the map as an example of code implementation. Graphical variable declarations that reference the repository. Calculate the pixel point where the image should be, and fill the corresponding digital code into the map matrix, so that each image in the generated window is placed in the specified position. Introduction of pictures it will make a qualitative leap in the artistic effect of UI interface.

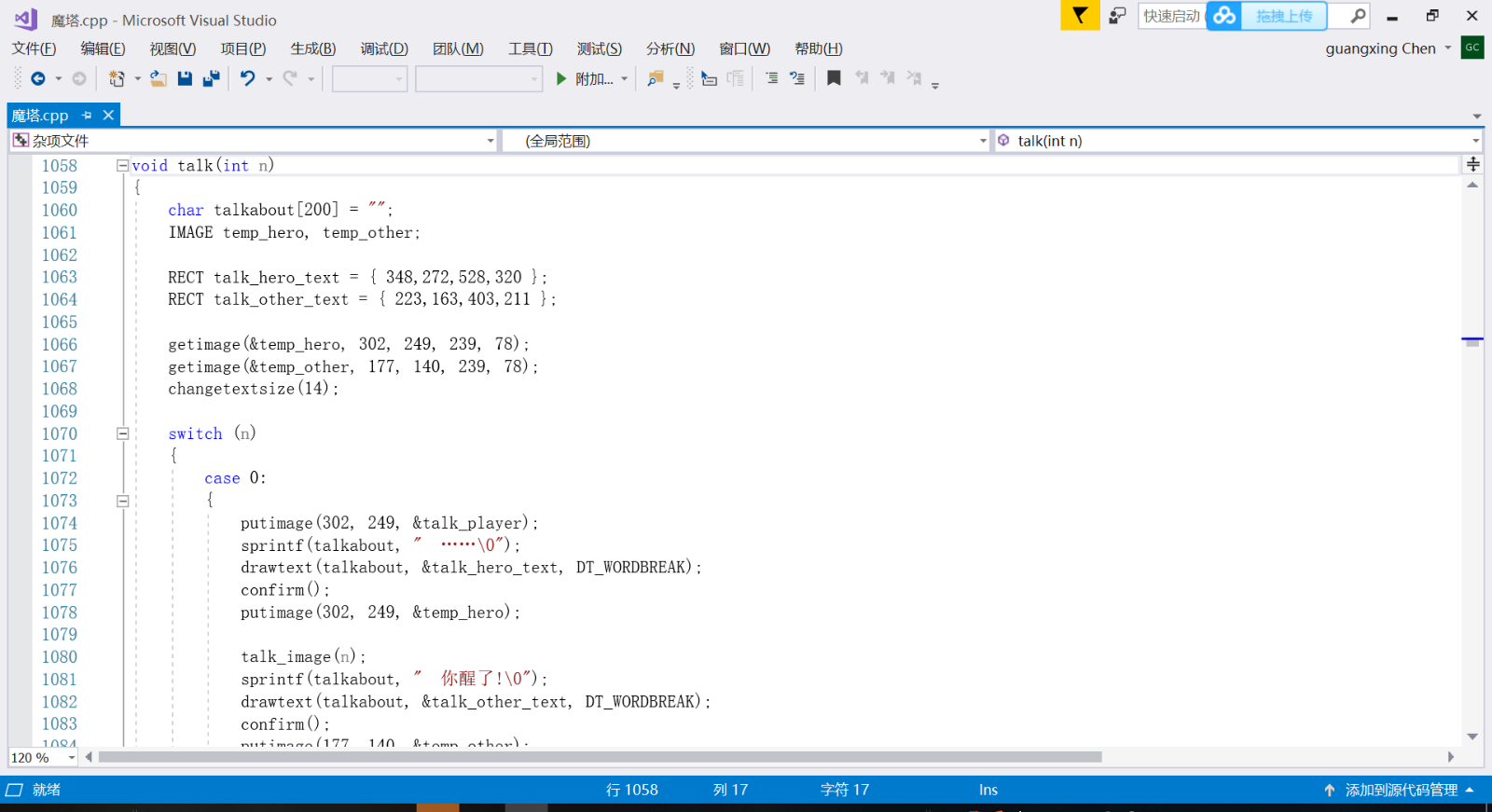


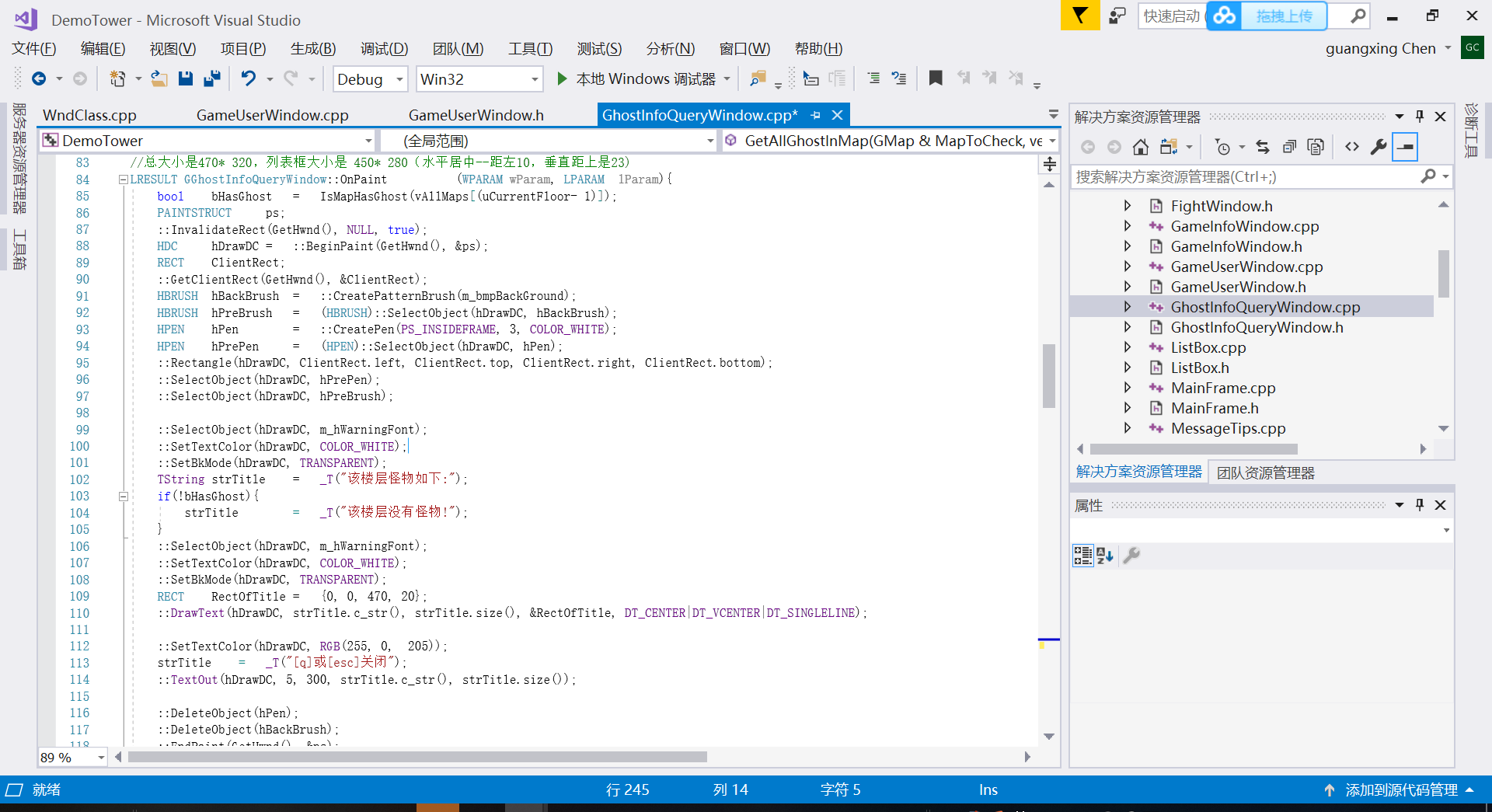


(Figure 5 Maps)

In addition, we establish links between maps, and the excitation region is the point where the ladder is located. Limited by space, it is omitted here.

When we started to write the code, we didn't notice that we had to take a screenshot of the same area of the window image. As a result, the refresh flash was too frequent. We used the get image function to take a screenshot of the same area and paste the window. The following is the code implementation in the second draft of the solution. The final draft code is implemented in the program talking window (final draft in C++).

  
（Figure 6 Talking window）

As for the illustration of monsters and props, we specially set a prop called "eye of heaven". When players get this prop, they can check the information of monsters and props on this floor by pressing a specific key. The annotations and pictures of monsters and props correspond one by one. In this way, although the mouse cannot move to the prop to display the prop notes, it also achieves the target function and increases the interest. We did a similar process for quick access to floors. The following is the implementation method in the final draft (part)

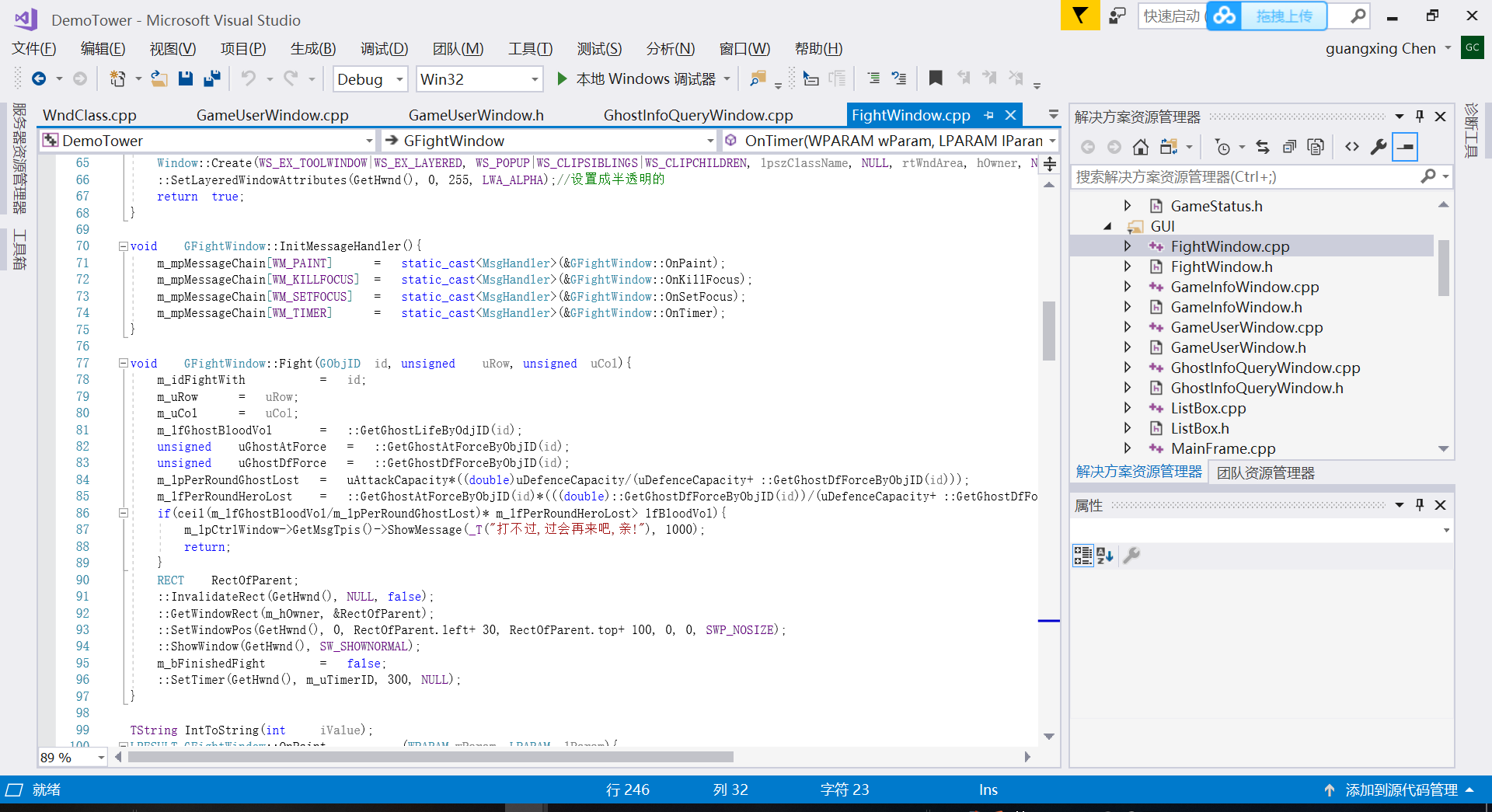
the fighting model，we canceled the design of critical strike，The calculation method adopts the method mentioned in the design. Part of the code is as follows:

Figure7: Fighting Part

On the outcome of the decision, when the hero destroyed the devil, the game has won. But we decided not to end the process immediately, instead allowing the player to continue to explore areas that had not been reached due to the pressure of the task, giving the player more freedom to decide when to exit, which also makes it easier to present all the designer's designs to the player.

As for the character movement control and store menu selection, we realized the control of movement with the direction key, and the menu selection with the keyboard and mouse. As the store menu mouse select a function as follows:

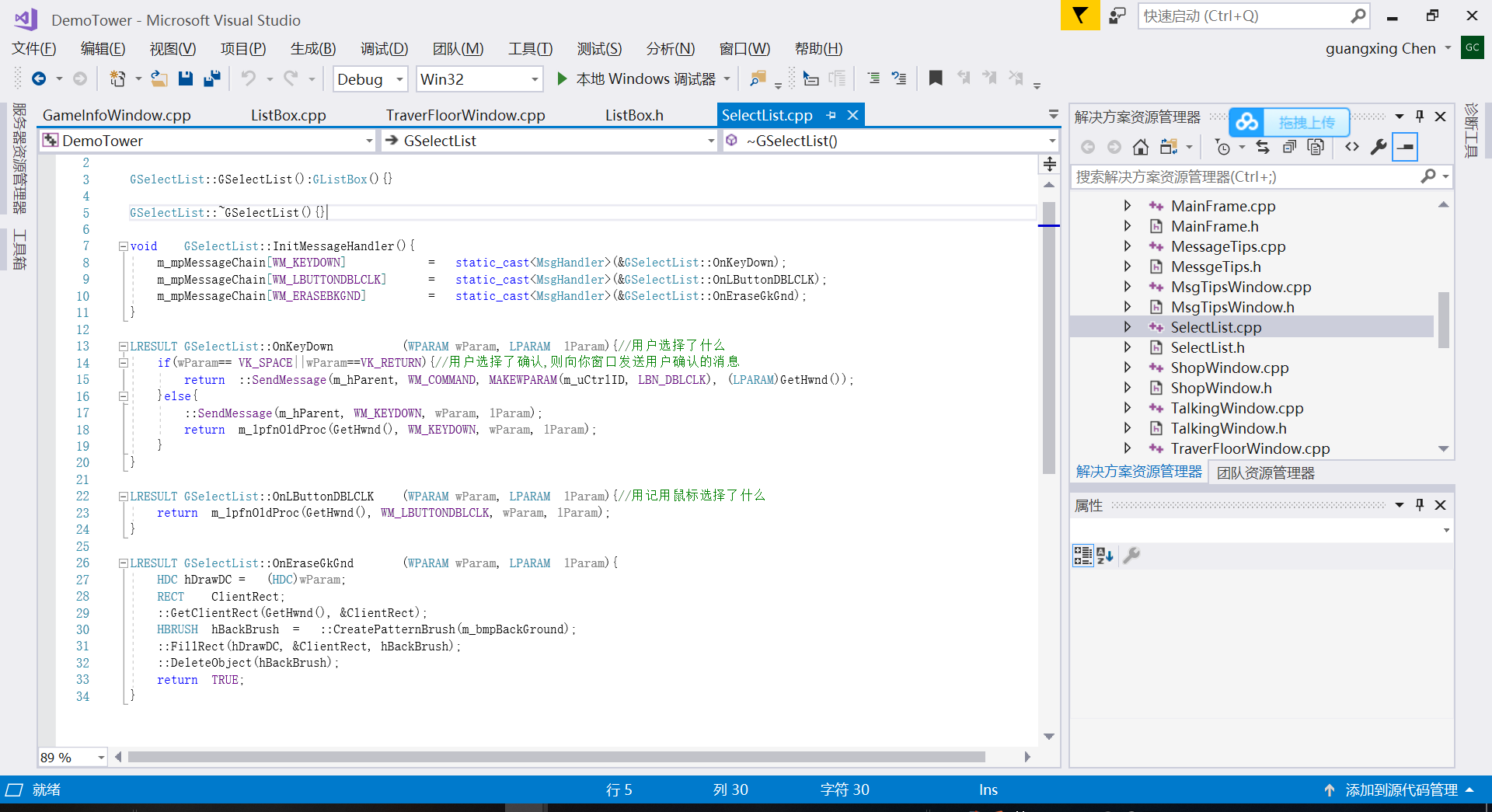


Figure8: Select list

## 5Test

|  |  |  |  |
| --- | --- | --- | --- |
| 调试项目 | 预期结果 | 实际结果 | 备注 |
| Buy nothing | fail | fail | null |
| Control the hero through direction keys | alright | alright | null |
| Check all the dialog box | alright | Screen terribly splash | Problems have been mentioned above. Solution is to change the code |
| Check the battle window 30 times (divided into two situations: insufficient strength can't trigger the battle, sufficient strength can play normally) | alright | Three bugs | Check the code, found that there is a logic error in the judgment of the layer according to the attack, defense and calculation. When the attribute value of the monster and the hero reaches a certain relationship, it will happen that the hero will be negatively drained. It has been modified to refine the hierarchical division of it. |
| shopping | alright | alright | null |
| Check the tools | alright | alright, | Delete the introduction of the tools(keep the data of monsters) |
| Check all the monsters direction | alright | Level 18’s monster is much too hard to defeat. | Lower the level of the monsters on level 18 |

## Result and the conclusion

With the guidance of our senior teachers senior and the cooperation of team members, we completed the "magic tower" project with C language (using C++).The project evokes many elements of the classic game, easy to play but challenging in strategy. We completed the function used to build the magic tower game and realized the basic functions of the magic tower. It can be said that the target has been achieved. We use knowledge in this project, some difficulties and overcome some technologies, such as map design, monster running test in the process of parameter adjustment, even under the level of helpless to give up some original ideas and switch to err on the side of the template, technically encountered setbacks is also abound, such as start even do map, can complete the word game, but fortunately finally come out the result of the overall was a success. In addition to the original maps and routes on several floors, the free exploration after the completion of the ultimate mission, the interactive language of NPC and the setting of special props (eye glasses, four-leaf clover portal) are also the highlights that the basic magic tower does not have. Some improvements have been made to the matchmaking algorithm. Lack of strength will not trigger. Matchmaking is friendlier to novices. All in all, I hope this small game can bring players leisure time to relax and happy. However, due to the strength and time, we definitely have a lot of shortcomings. For example, when players stay in the battle and do not have enough health, money and experience to beat the monsters in the way and move up. Such problems are also expected to be criticized and corrected.

I would like to thank professor Xu and professor AT for their long-term teachings, my seniors for their care and help, and my classmates, especially my partners, for their solidarity and cooperation.