**GROUP 8 PROJECT REPORT: ARCHER BATTLE**

Dan(Boyang) Liu, Yufei Huang, Yuan Huang

Jilin University Honors Program in Science

*Update: July 7, 2019*

# Abstract

The program is a 2.5-D game called *Archer Battle*. But we use Modern OpenGL to give beautiful 3D proformance. Player can operate a archer by mouse and keyboard to fight with the npc archers. Archers can shoot and move freely, and gain the items to strengthen or save them. NPC archer can aim at the other archers precisely and find the items to strengthen them. We use *Breadth First Search* to let the enemies find the items intelligently, and use multiple threads to improve the proformance. This report will concentrate on our program’s introduction, our assgniment, analysis, design, inplementation and test.

# 1 Introduction and problem statement

In this game, Player can operate a archer by mouse and keyboard to fight with the npc archers. Archers can shoot and move freely, and gain the blue stars to give them EXP, strengthen them or the red star to give them HP to them. NPC archer can aim at the other archers precisely and automatically find the items to strengthen them.

In our program, we use Modern OpenGL to build our battlefield, which costs lots of our time. But it’s worthy. We use 3D culcataling and textures to build a beautiful 3D world. And the archers have human’s symbol and have their own movement, to look like human.

We use clever way to generate the map which has stays, walls, red and blue stars randomly generated. Gaining the red stars can recover the HP, and gaining the blue stars can give the EXP. When EXP gets to the limit, the level of the Archer is upgraded. The archer will get few improvement. For example, HP limit, speed, attack, able to dartle,

The most technical part is NPC archers. They need to move in the maps with purpose and cleverly. For that we design to let the NPC use the Breadth First Search to walk to the stars automatically. In that mode, walking can avoid some attack and improve its status. And it can shoot to the player and other archers fiercely by using the difference of the position vectors as the direction.

# 2 Division of labor and contributions

In the beginning of the project, we have a division of labor, each team member has its own equal assignment, difficult or heavy.

Yuan Huang: The generating of the map. Generating walls, stays and the red and blue stars randomly in the map.

Yufei Huang: The drawing of the archer, 3D model, and the movement and free shooting control, generating arrows.

Dan(Boyang) Liu: Writing the clever NPC. Using multiple threads and Breadth First Search based on the OpenGL. Part of the generating of the map, part of the generating arrow and drawing of the archer, mainly for the possibility using those as the bases to create the clever NPC.

**Table 1:** Our Contributions

|  |  |  |  |
| --- | --- | --- | --- |
| name |  | major | main work |
| Dan(Boyang) Liu | 11180306 | physics | All the design, most learning OpenGL and communication of learning, all the implement in codes and the main part of the presentation and report. |
| Yufei Huang | 12170402 | Biology | Getting involved in learning OpenGL and succeed to draw the arrow mode in 2D for later usage. Contribute to the communication. Part of the second Presentation |
| Yuan Huang | 24180601 | Biology | Getting involved in learning. Under the guidance write the txt document for the game map, where 0, 1 stands for stays and walls. Part of the second Presentation |

# 3 Analysis

OpenGL is the only thing we knew that time to create a game and it needs lots of time. So the first state is learning OpenGL. We found the tutorial on the internet, it is [www.learnopengl.com](http://www.learnopengl.com). To get fimilar with the OpenGL base, this project should start with creating the archer and the OpenGL project structure easy to use. And the process is being revealed while learning the OpenGL tutorial. Then the technique of us is improved and the environment is built. To create the next state of the base and improve the coding technique in sequence, this project should choose the generating of the map as the next step. Then at the hardest part in the technique but almost the best state of the coding technique, the project should finish the clever NPC part which contains multiple threads and Breadth First Search. Then it needs to do the HP and EXP system to finish this project. If there is still some time, it can be improved a little more.

## 3.1 3D Space

1. Local coordinates are the coordinates of your object relative to its local origin; they're the coordinates your object begins in.
2. The next step is to transform the local coordinates to world-space coordinates which are coordinates in respect of a larger world. These coordinates are relative to a global origin of the world, together with many other objects also placed relative to the world's origin.
3. Next we transform the world coordinates to view-space coordinates in such a way that each coordinate is as seen from the camera or viewer's point of view.
4. After the coordinates are in view space we want to project them to clip coordinates. Clip coordinates are processed to the -1.0 and 1.0 range and determine which vertices will end up on the screen.
5. And lastly we transform the clip coordinates to screen coordinates in a process we call viewport transform that transforms the coordinates from -1.0 and 1.0 to the coordinate range defined by glViewport. The resulting coordinates are then sent to the rasterizer to turn them into fragments.

It is OpenGL base, so here we attach the explaination in the tutorial.



**Figure 1:** 3D Space

## 3.2 Drawing Archer

From creating a window to succeed to draw the first triangle may be the hardest part in OpenGL, after that there is still a long way to pass to draw our alive 3D archer. Here is no more understratum statement of OpenGL. Just state the main way to draw the alive 3D archer.

The final vertex on the screen is the result of matrix projection \* matrix view \* matrix model \* the original vertex. First the origin vertices are a cube. We use scale, rotation, translation matrixes to create the form and use a translation matrix to put it in the right position of the map.

Mainly, translate the cubes as body, arms, legs, head to the right position of the body, then scale them to make them look like the part of the body. And let them rotate with the time to let it look like moving with the battle. And use a Position vector to create translation matrix to put it in the right position.

Then use view and projection matrixes to put it in the 3D view.

## 3.3 player’s operation

We use OpenGL has callback functions to get the key and censor status. WASD is pressed will change the position and the view matrix, the camera.

The problem is input is in the main function, but for the better structure we build a game class, so we define a Boolean arrow to send the key status, the key is stored as int type in OpenGL, and it’s state is Boolean.

And the censor callback has also to be used in the main function, so we just pass those value.

The for keys we just change the position vertor. For the mouse, we decide to let archer rotate with shooting direction which is also the mouse’s direction, so we need to write a rotation matrix by ourselves, OpenGL doesn’t give that function which can generate matrix by direction because it’s more 2D.

## 3.4 Generating the map

This project uses a txt document where 0 and 1 stands for the stays and the walls. Using c++ stream reading the file and store the date to a c++ vector of a integer vector. The integer vectors have the raw’s data and its vector stands for the column. Then generating the stays and the walls to the very position.

## 3.5 Generating stars

At the stays, we randomly choose to generate stars.

## 3.6 enemies’ movement

The enemies movement have something similar to the players’. The biggest problem is that they should be enough intelligent so that they can find the shortest cut to approach the stars. Thus we use *Breadth First Search*.

## 3.7 Threads

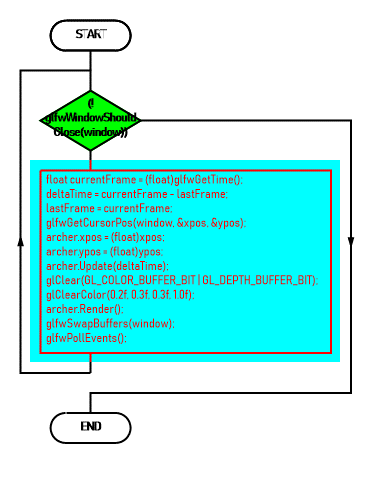
Using Breadth First Search sometimes may be too slow, the rendering can’t wait, need to create a new Thread for each archer finding ways. So we just detach the thread and simply make sure the values are not changed while the finding ways thread is running.

## 3.7 Death of enemies and players

When the health is empty, the archer will die. We actually can’t afford to make a animation of the dying, it’s totally not necessary. We just let its status renew, position return.

# 4 Design

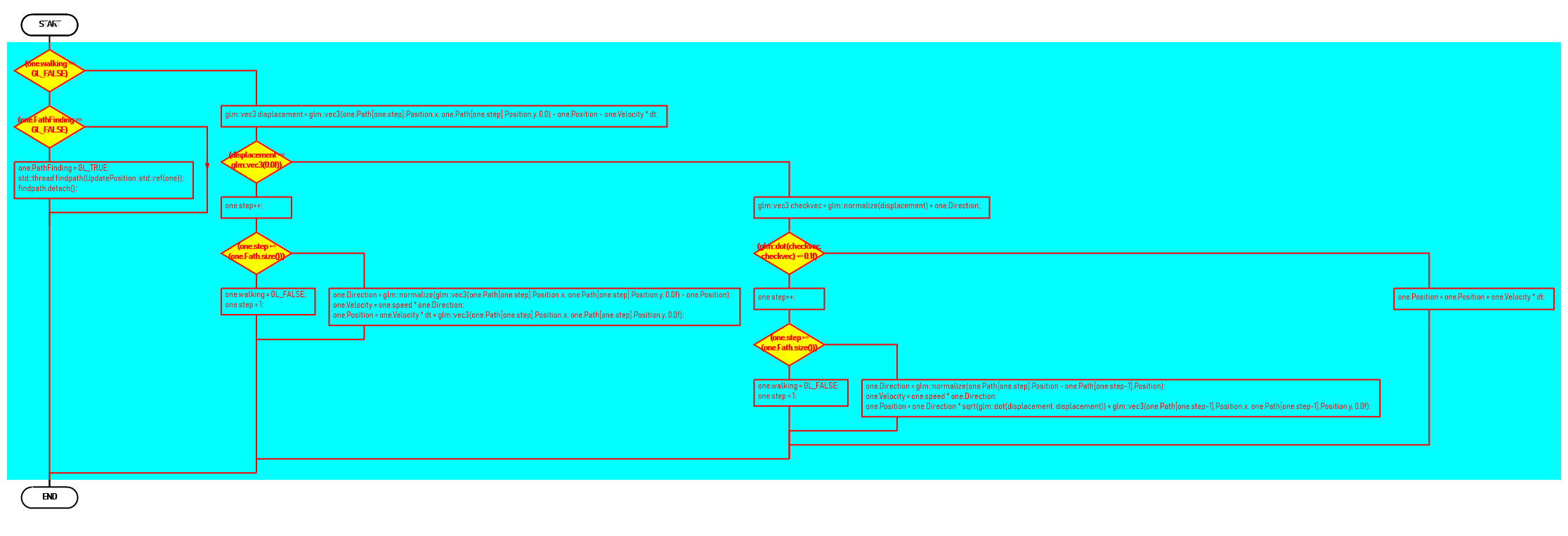
## 4.1 main fuction

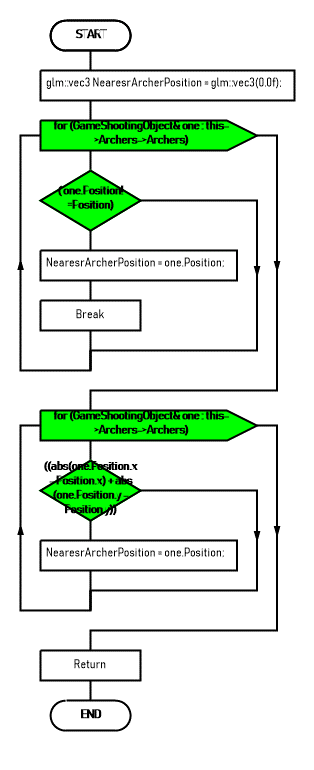
Main function is where the rendering loop exists and it calls the game class’s object archer. **Figure 2:** FlowChart of main function

## 4.2 function: operate

**Figure 3:** FlowChart of function: operate

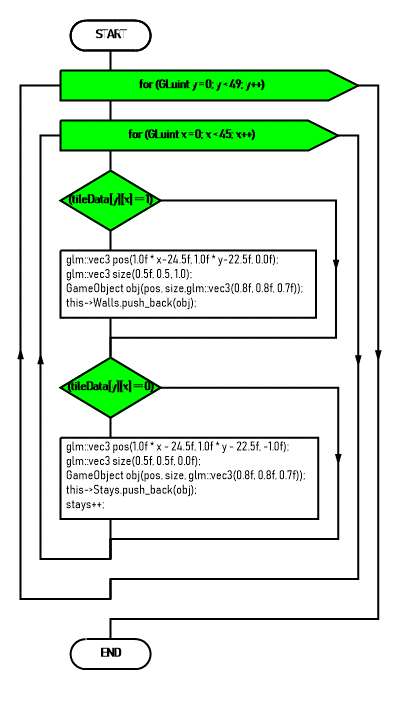
## 4.3 movement of enemy

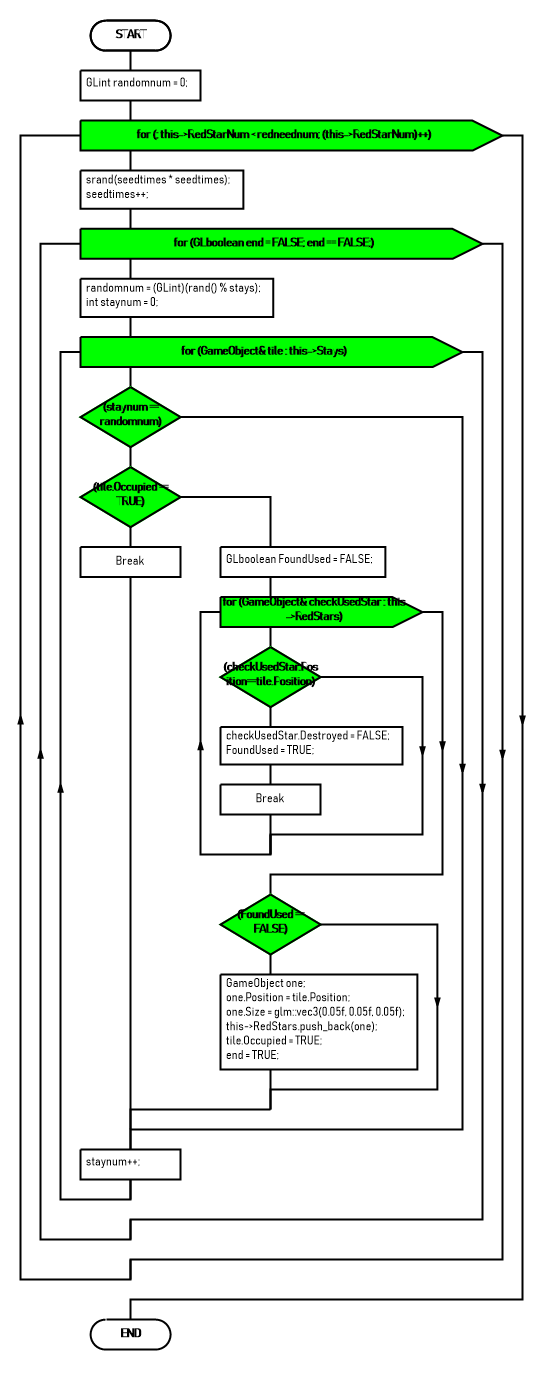
For enemies we use BFS to find a path, then try to let it follow the path. First, make it can stop to find the way and go to follow the path. So we define a status called walking. While not walking, it is finding way, and while walking, it is follow the path. Then check if it reach the path stay, if it’s true, make it head for the next point until the last point to find the next way.



**Figure 4,5:** FlowChart of NPC moving

## 4.4 Generating the map





**Figure 6,7:** FlowChart of generating the map

# 5 Implementation

In this part, we show part of our codes of several functions. The followings are the contents of our Implementation.

*1.main function*

*2.Breadth First Search*

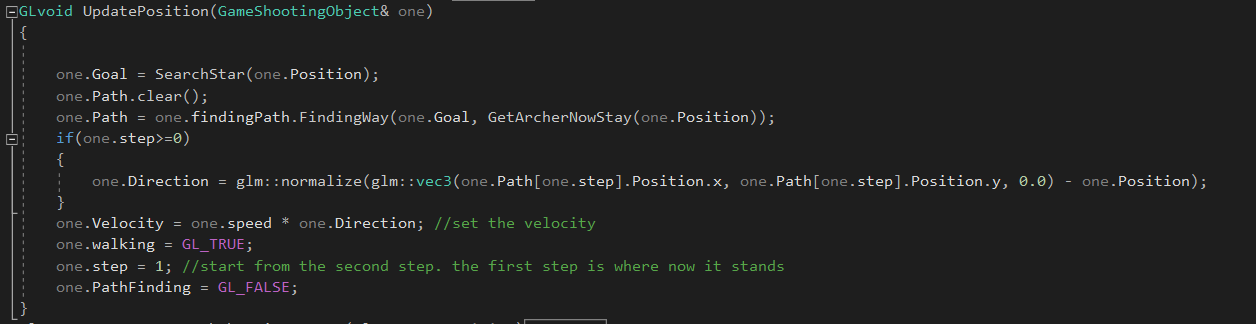
*3.movement of the NPC*

*4.generating of the map*

*5.drawing of the archar*

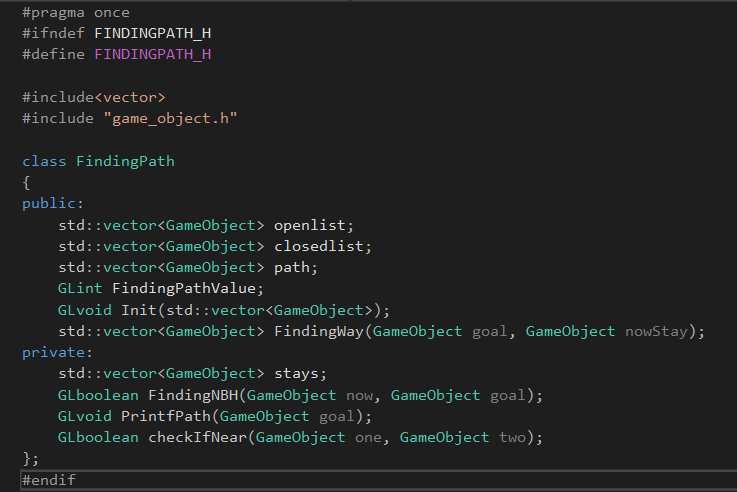
**5.1 *main function***

Rendering loop

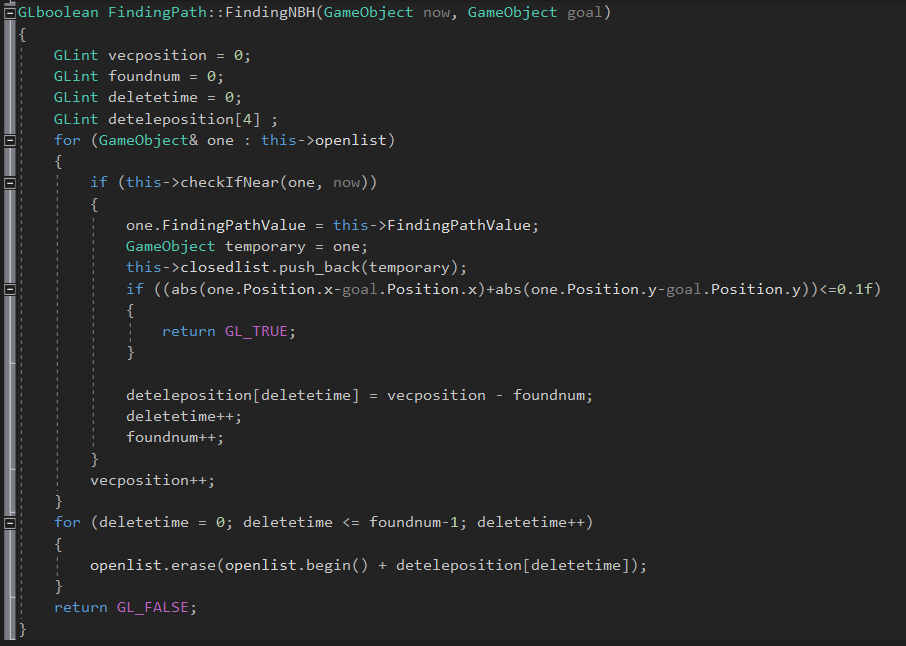


**5.2 *Breadth First Search***

In this part, we use lotd of vectors which are convenient.

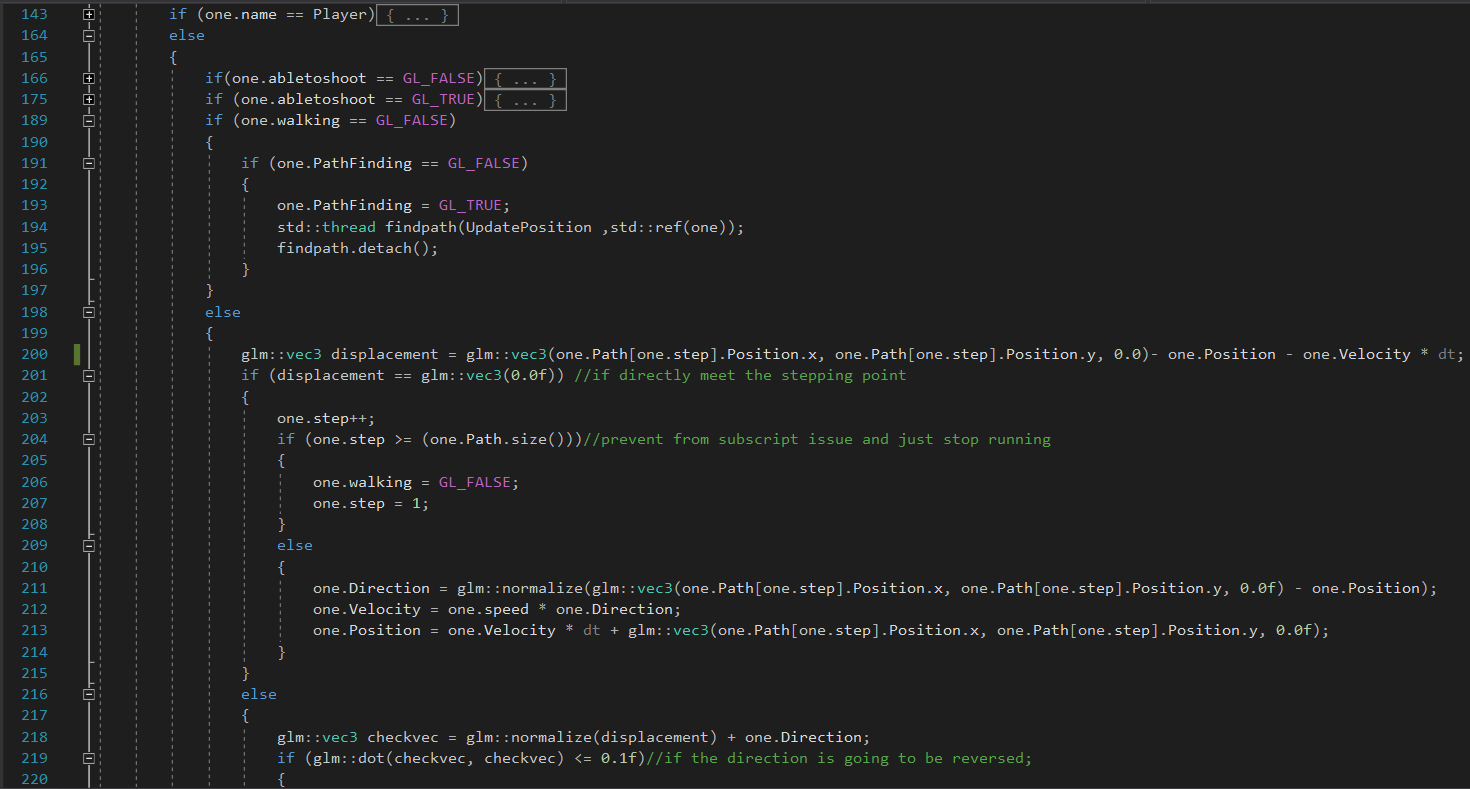


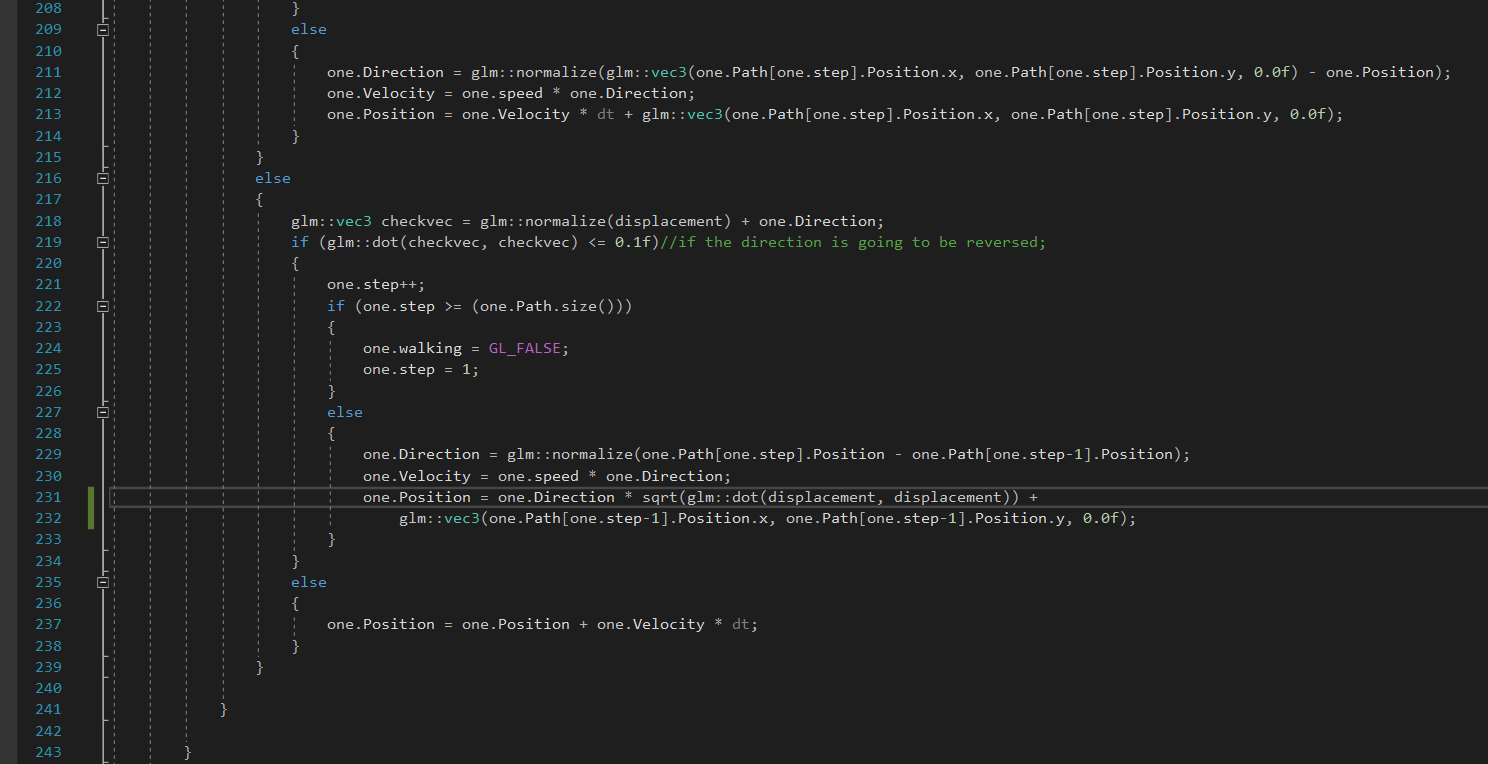


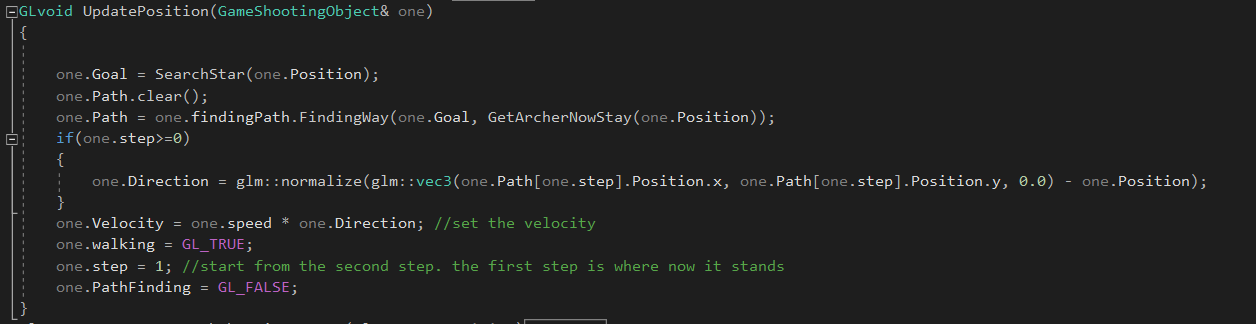




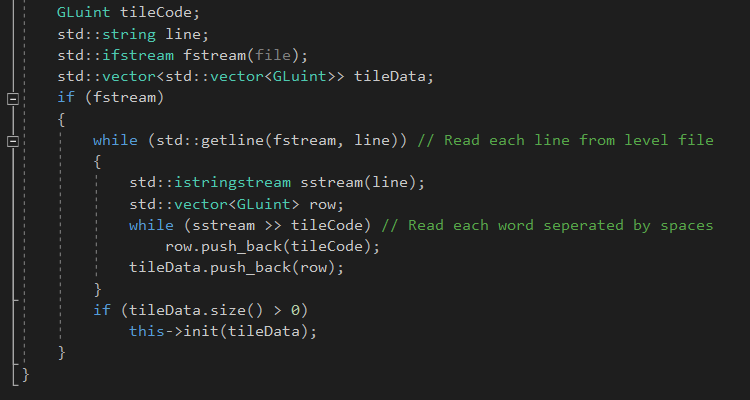
**5.3 *Movement of the NPC***

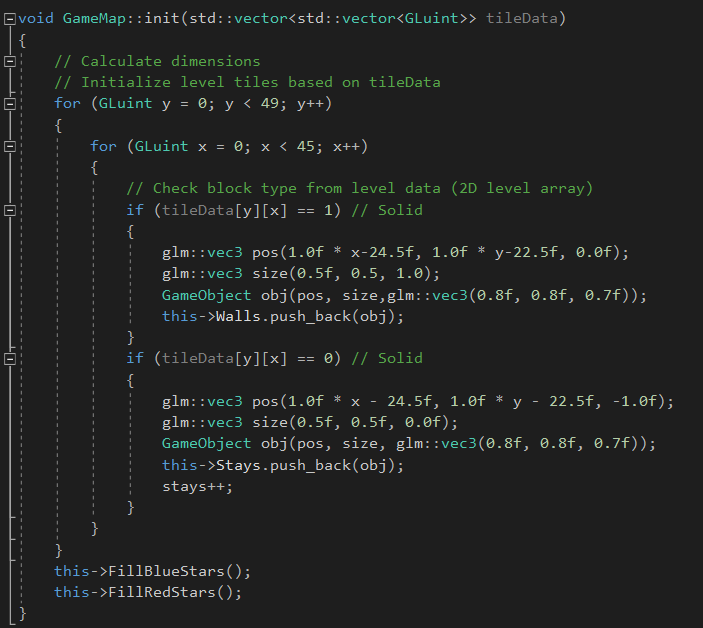
******

******

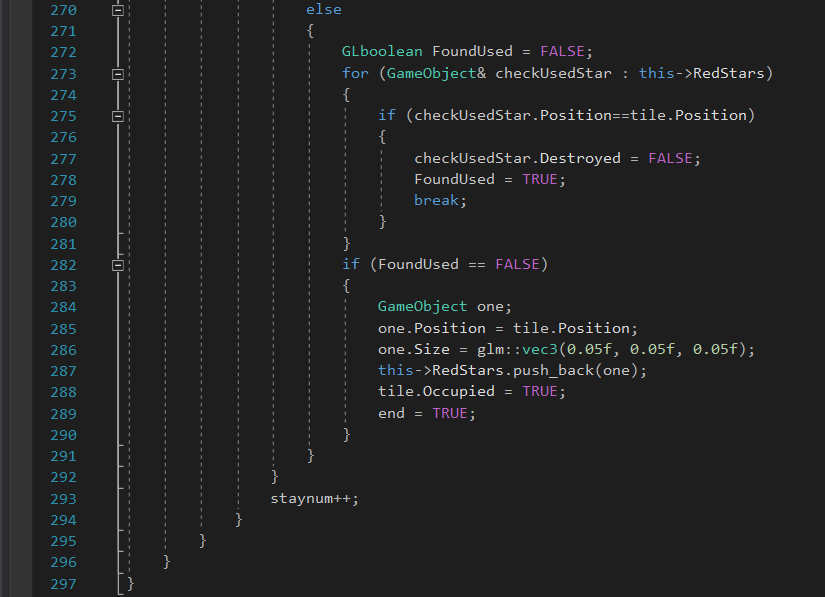
******

**5.4 *Generating of the map***



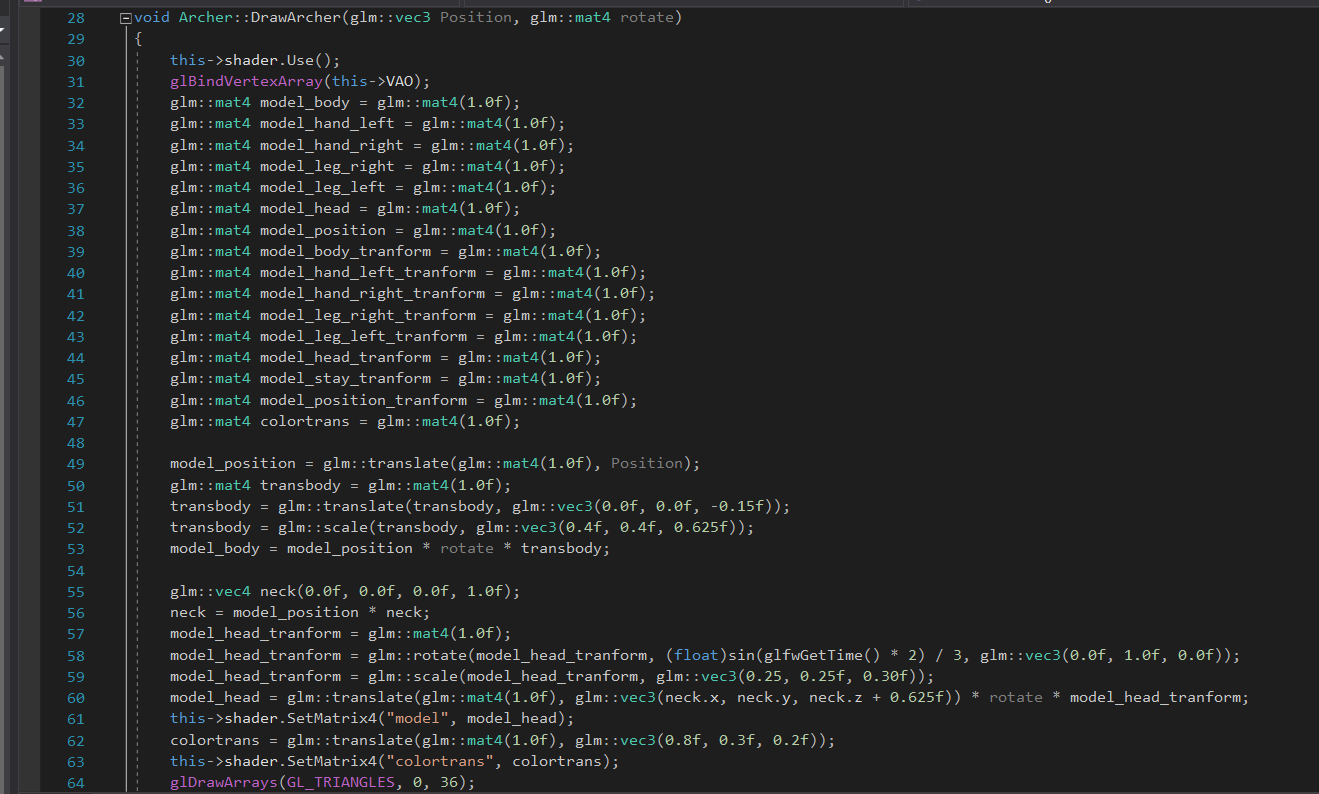


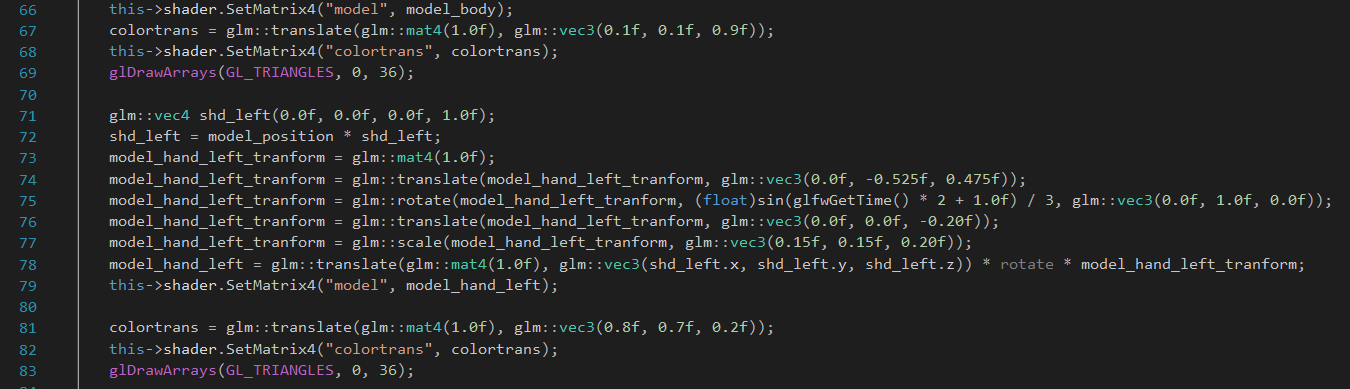




**5.5 *The drawing of the archer***

Just show a little part as example.





# Test

The fighting is interesting, NPC is very aggressive. The environment is beautiful. Everythind is sample but functional.