GUI:

gameWindow:

Dealing with the game GUI. Doing the square refresh and the data update. The GUI can be divided into three parts: game, data and help information. The game part involves 64 squares, which handle the game processing. The data part handles the data about the game like the score. The help information give the cost of each terrain square.

mainWindow:

Dealing with the login part. There are two pushButtons, one is start, which will construct a gameInstance object when clicked, the other is tutorial, which will some help information about how to play the game. What’s more, while the window is open, the BGM is playing until start a game.

Game:

gameInstance:

The control of the whole game. Initially, the constructor will construct every relevant part about the game, like the randomly generated map, the location of every pieces, and initial the timer, whose interval will be set to 200 ms, then play the BGM. During the game, each timer interval, a function named main\_thread will be called. In the main\_thread, firstly check the winning, then check the lost. In principle, the priority of winning is higher than yield. If the game is still, then call another function named routine. In the routine, firstly check userInput. Then move the hero. Meanwhile, randomly move the monster. The detail of each functions will be illustrated later.

Win:

The requirement to win is that hero reach the end point.

Lose:

The condition when hero lose is that monster catch hero.

Random\_move:

The movement of monster is not absolutely random. In most situation, monster will move to hero in the relatively “shortest” way. Because the pathfinding algorithm of monster is same as hero’s, which means it will also take the cost of terrain in count, which is unnecessary for monster. In a special situation(20%), monster will move in a absolutely random direction.

Piece:

As the abstract base class, piece class has the fundamental property of different entity in the game: x, y and image. The image is the path to corresponding image.

terrain:

As the base class of all terrain, additional property is cost of each kind of terrain.

Character:

The base class of all character, which is changeable during the game.

Pathfinding:

A star:

As an efficient pathfinding algorithm, A star can find the shortest path between two points, which is the abstraction of two piece in the game. On the base of Dijkstra algorithm, we add a simple heuristic function.

Graph:

We abstract the whole map of the game as a graph. Each piece is a node, and the cost to go into the terrain is the weight of the directed edge goes into the node.

Art style:

We choose the style of pixel art, and chiptune as our style, in the nostalgia of FC.