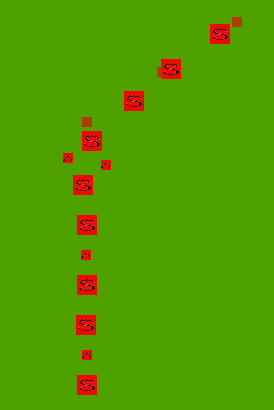
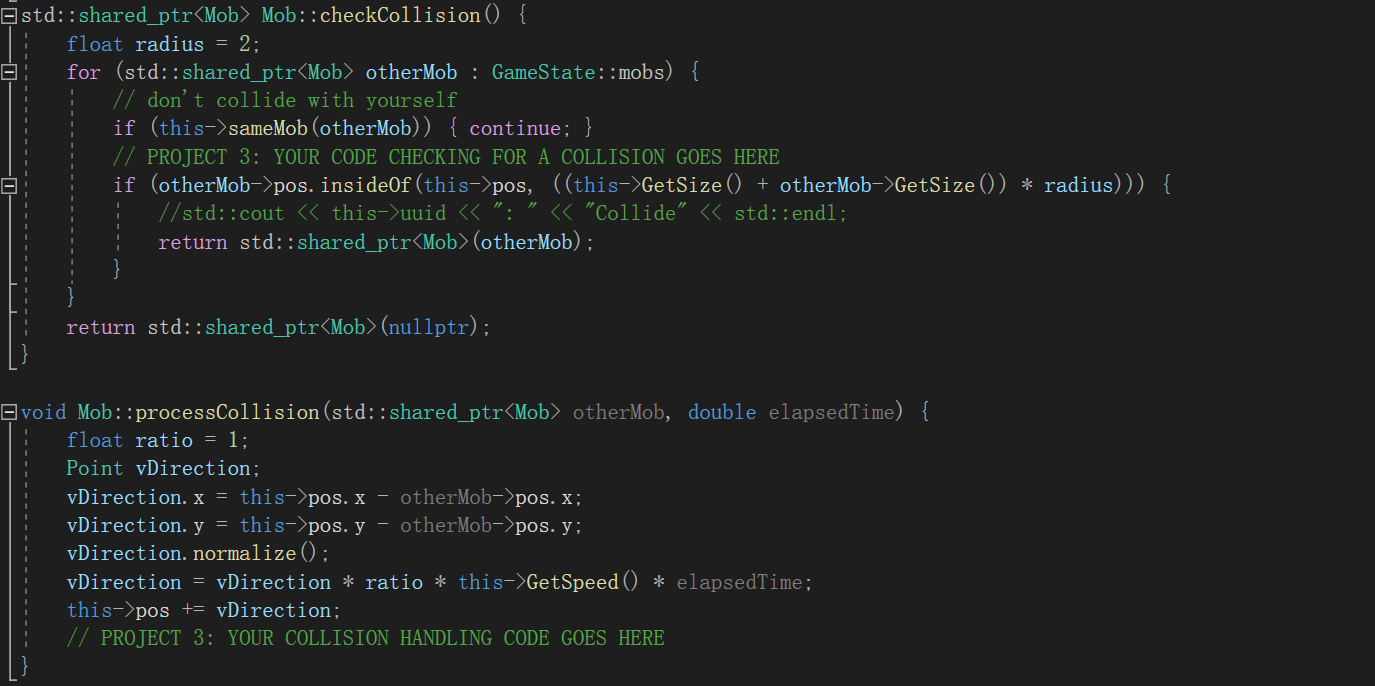
### Mob Collision Logic

In this assignment, there are three types of collision: mob-mob, mob-building and mob-river. In the provided code, there are two functions that opened for me to implement the code. For checkCollision() function, I used the insideOf() function to detect if another mob is in range.

On the collision process function, the algorithm is simple. After knowing that the collision process function will be called after the movement of mobs, I added another transform onto the mob object, which is in the opposite direction of collision.



The code I can write is limited by the structure of class, so there are actually some disadvantages about this algorithm. First, the processCollision() function can only process the collision with one mob every frame, but the actual situation will be more complicated. There will be a lot of mobs that collide with each other.

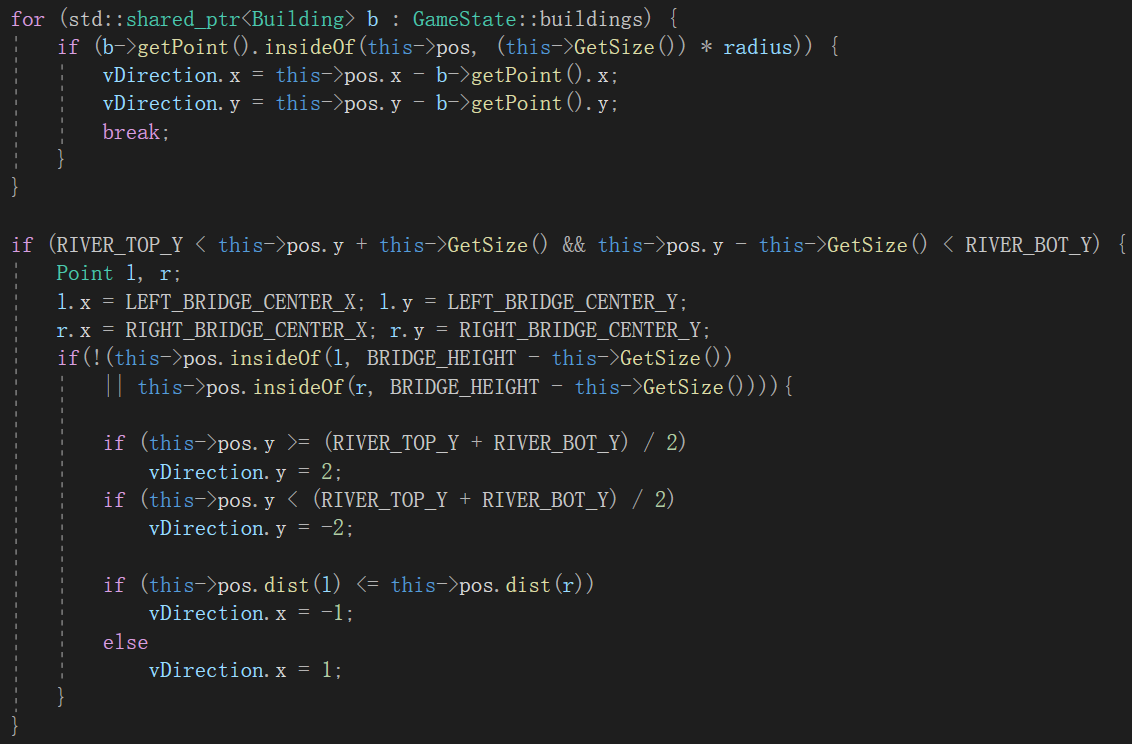
To improve this, if I’m allowed to change the structure, I would merge check and process into one function, so that I can calculate a direction based on multiple collisions.

Also, the assignment requires mobs to have difference in mass which can make mobs with different sizes can be pushed in different degrees. I did this by using speed of mobs. Lighter the mobs, faster they can go, and the force will be stronger pushing them out of the collision.

### Building & River Collision Logic

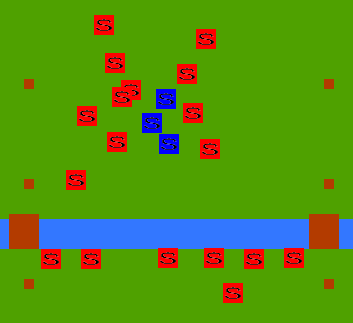
Because of the limitation in the code structure, mobs and buildings are different types, they have different behaviors. What’s more, the river is static written in the structure, there are only parameters as APIs to implement the collision.

So, to check and process the collision happens between mobs, buildings and the river, I created a new function called checkAndProcessCollisionBuilding(). This function is invoked after the mob collision process.



The core algorithm is the same with mob collision processing, but because buildings are static, the collision will be calculated separately in different cases like towers and rivers. There will never be a situation that a mob collide with both.

For the river, mobs will consider if it is inside of the river first, then they will look at the situation that they are on a bridge. The basic logic is easy to achieve.



### Collision During attack

As I noticed when playing the game, I found that when mobs began to fight, they seem to forget the collision with everything. In this game, there are two types of state in mobs: move and attack. They are totally separate and have different logic. In the movement processing function, it will consider collision, but not for attack state. So I added the processing function to the attack state.

