

To begin with, we started working on the design of the clans in our game class. We created a vector called "clan design" and copied all the data except for the commas. Then, we obtained instance pointers for the fence, townhall, and cannon. We looped over them using the "clan design" vector, assigning the number zero to represent empty spaces. We already set a grass background for the game and added it to the scene.

Next, we created a function to randomly generate troops in the empty spaces (represented by zeros). We also changed the zeros inside the fence to the number 4. We set a count in the troop formation function to limit the number of troops to 5 in level one. As the levels progress, the number of troops will increase.

In the "move troops" function, we made the troops target the townhall and destroy anything they collide with after 5 seconds. We handled the mouse press function within the game, as it was easier since we already had the same instance in the game, rather than passing it as a parameter to the cannon. We also implemented a "check collision" function to determine if a troop collides with the fence, townhall, cannon, or bullet, and remove the corresponding item from the scene. If a troop collides with a bullet, both the bullet and the troop are deleted from the scene.

To add a time limit to each level, we created a timer that increases as the levels progress. The player has to keep their townhall standing for a longer time as they progress through the game. Initially, the timer is set to one minute for the first level. We updated the timer in a function called "update timer" and established a signal connection in the game's constructor. Additionally, we added a start button as a variable in the game and reset the timer in a function called "start game" when the start button is enabled.

We also created a separate class for the bullet to handle its movement and direction when the mouse is pressed. The "target" function calculates the target position by subtracting the cannon position, obtaining the differences in the x and y coordinates, and dividing them by the length. We implemented another class for health, which is instantiated in each class to determine when the health decreases or increases. If the health reaches zero, the corresponding item is destroyed; otherwise, the worker class is called. Although we haven't handled the worker class in milestone one, we plan to do so soon.

Additionally, we created a class for the townhall, which includes the image and an upgrade mechanism. Players can earn points upon winning, which can be used to upgrade items. Consequently, we added a "fence1" class to handle the upgrade, health, and image for the fence. We also implemented a cannon class to manage the image, health, and upgrade functionality. Lastly, we created the troops class and implemented the "stop" function to control their speed.