A screenshot of a computer program

Description automatically generated

A diagram of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

A white rectangular object with black lines

Description automatically generated

At the start of the application, there is a login window(finalized yet not implemented and will be done in class presentation). After that you get the main menu where you have the options, setting the difficulty, volume and showing the about, an exit button, and a start button. On start the story dialogues open and the audio works, after the dialogues are over the user is presented with a choice of one of three weapons, showing their damage and speed. The castle, bullet, defence unit, fences, enemies, builders, score, and health bar classes all inherit from the Object class since all of them have a changing attribute. It is necessary to implement all these classes since each class has its separate QGraphicsPixmapitem image, which will appear/disappear, and for the animations of the moving classes, like the builders, and enemies. Bullet aggregates with defence unit because the bullet will be instantiated in the defence unit. The scenes and game classes are the main classes where the action happens. The castle, cannon, fences, and troops classes aggregate with them. The spawn enemy function in the defence unit class spawns the enemies using a Qtimer every 3 seconds. The player wins each level when the countDownTimer instance reaches to a certain time indicating the level to be over. This is necessary so that when the timer reaches zero, the player wins the whole game all at once. The sound effects, voice overs, and theme music are all added through out the game. The bullet class includes the damage it does as well as an equation to move the bullet using a unit vector so as not to change the speed of the bullet if the mouse moves further away. Inside the defence unit class, the bullet is spawned. The fence class includes the health and image of the fence as well as the scaling of it. The main window opens the weapon select dialog and on choosing a weapon the game opens, similarly on transitions between levels, a dialog opens indicating winning the previous level then when pressed next, the weapon select dialog opens. The enemies class sets the health and the damage done by the troop. There are health and damage attributes in multiple classes to ease the transition between levels since these are the attributes that will affect the conditions of winning or losing in levels. Also, in the enemy class, the image attribute is changed every 50 ms to animate the movement of the enemy using a QTimer. The enemy positions are randomized to be outside the clan using a do while loop. The troops move using the A\* algorithm to be able to find the shortest path between their position and the castle, avoiding any fences and the defence unit in the way. If there is a collision with a fence or the castle, their health decrease by the amount of damage that enemy has, and the enemy alternates between a still animation and a one where its attacking. The health of the enemies decrease upon colliding with a bullet, in the bullet class, a function is called on collision that makes the bullet disappear, and if the troop’s health reaches zero, it would also disappears. When the castle’s health reaches zero, the player loses. There are 6 levels to the game. At the beginning, if no difficulty is set, it is set by default to easy, and the defence unit is by default the cannon. Each level has a different theme (different images, different sound effects, also different enemies and varying difficulties). If the player wins, the ending of the story comes appearing. If the player loses, an ending with a game over sound effect applies, and an option to restart or exit appear. Forward declaration was used to solve the issue of circular dependencies, as many classes needed to be included in each other, which appears when 2 classes are inter-included and are necessary for more efficient access(for example, the forward declaration of fences in enemies and builders).

The builders work in the same way as the enemies, going to any damaged fence when there exists one, and then goes to it, heals it, and then gets to the builder hut after it is done, if no other damaged fences are there. On collisions with enemies, the builders get instantly deleted from the scene, and nothing affects the enemy.   
In level 5, the game shows that you have defeated the boss, but it is just a setup for the boss fight included as level 6, where it is like the past levels, but have an instance of the final boss, which moves from left to right on the screen and shoots magic that target the castle and severely damage any fence in the way while damaging the castle by 1 at each hit.  
Also, across the map, health boosters for the castle, damage boosters for the bullets, and multishot boost for the bullets all appear at random times in every level, and on collision with a bullet, each booster does the thing the name mentions it does.