ELEC-A7150 - C++ Programming
Project plan
Towerfall: Ascension

Perttu Yli-Opas
Joonatan Bergholm
Essi Rantanen
Jaakko Visti

November 20, 2017

1 Overview

The plan is make Towerfall: Ascension like game with online multiplayer. Players will shoot lasers that bounce from the walls at each other, trying to be the last man standing. Amount of bounces that the lasers can bounce will most likely be limited to a smallish number so that the play area won't be filled with them. Naturally the player characters will die if they are hit by a laser. The map will be a procedurally generated 2D cave with lots of walls.

Players can jump around the cave as well as use a jetpack to outposition their enemies. Players will generate ammo for their weapons and fuel for their jetpack slowly over time to ensure the game will end and that all positions on the map remain reachable. Some time-triggered events will be added to prevent the players from sitting in safe places, which could result in a stalemate. These events may include some kind of deadly energy field that decreases the play area.

2 Libraries & tools

The Qt library will be used for handling graphics and connectivity. QT C++ API is used, with .ui widget configuration files for window configuration. Git will be used for version control.

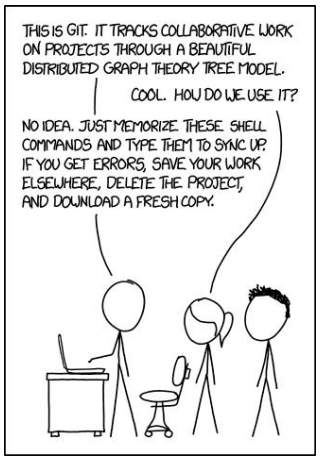

Figure 1: We will use git in this project

3 Architecture

The server and client will be implemented separately. The connectivity will be implemented using TCP sockets.

3.1 Server

The server window will show connected players and can be used to start the game once all players have connected. It will also show some statistics, at least points of the players when the game is running. Number of players can be different, probably limited to a number less than ten. The structure of the server is shown in figure 1.

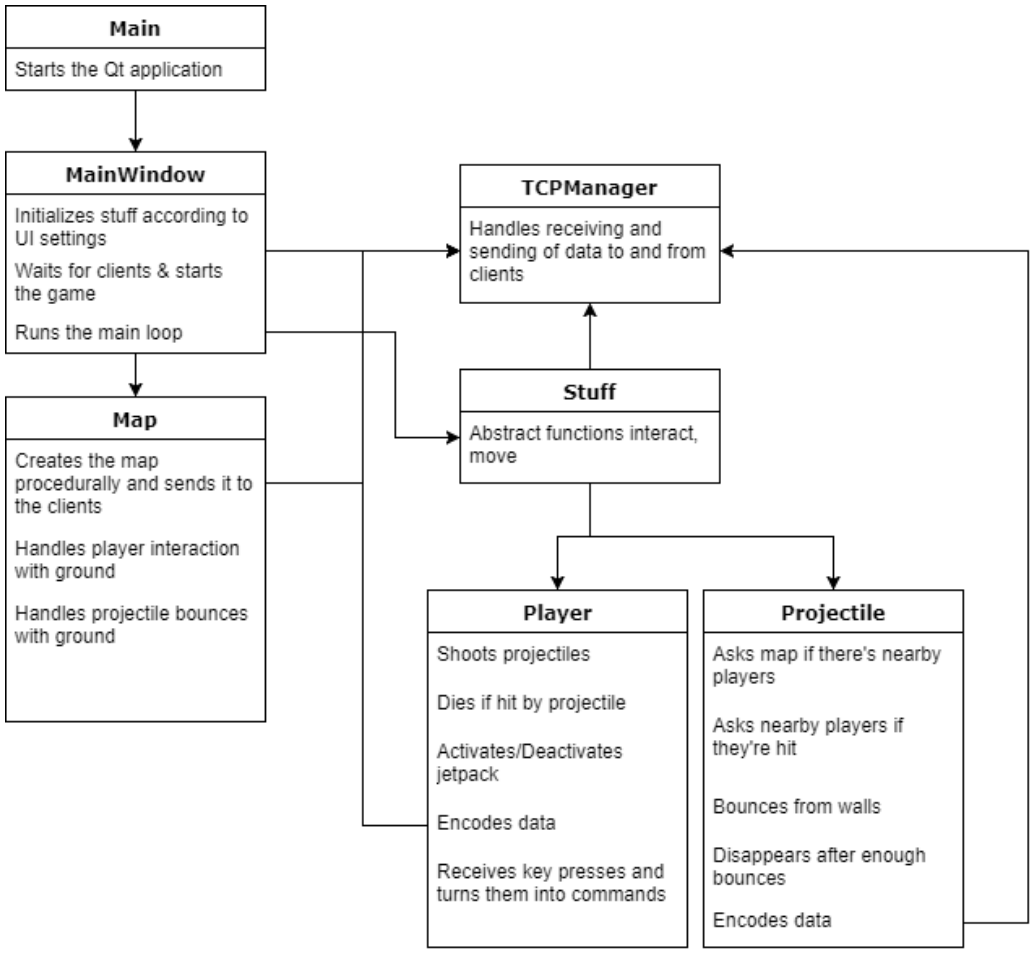

Figure 2: Draft of server-side structure

3.2 Client

The client will be used for playing. All players will be playing on different computers, being connected through internet. When starting, the client needs to ask for server IP address, which will be visible in the server window. The structure of the client is shown in figure 3.

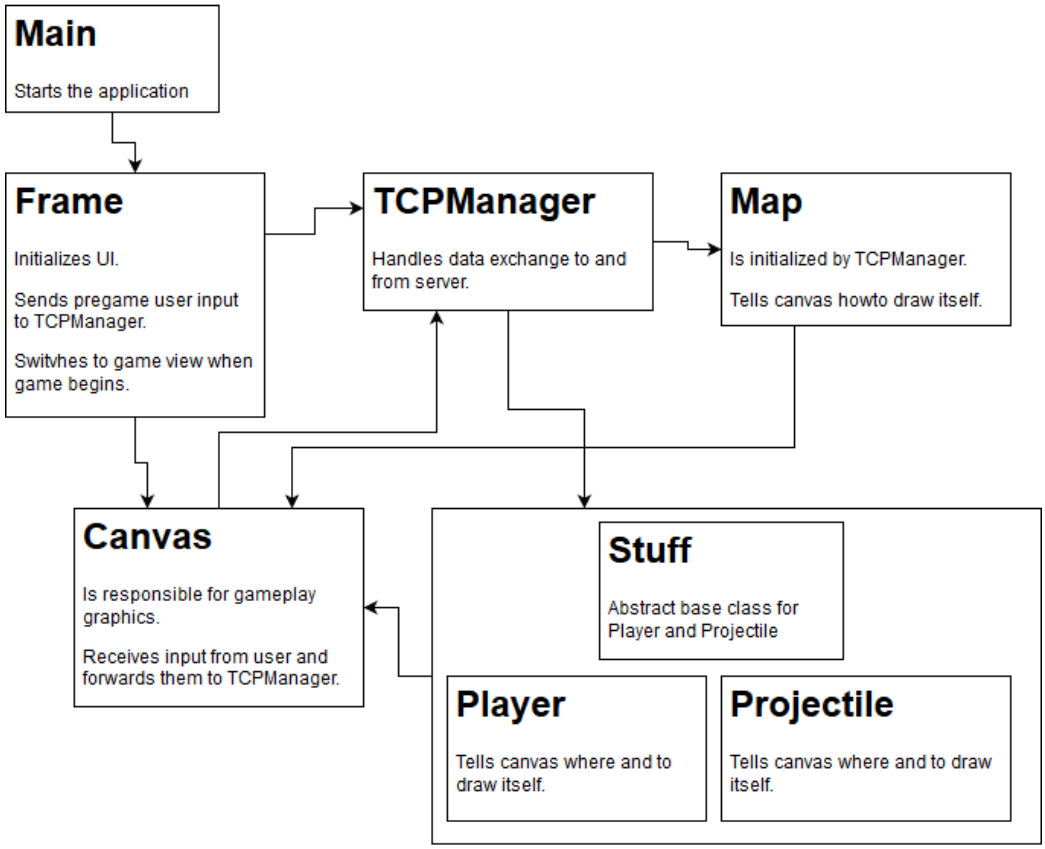

Figure 3: Draft of client-side structure

4 Roles

Work has been divided into four parts and project members have been given them based on their previous coding experience. Joonatan's job is to program the basic structure of the client side, while Perttu will handle the server-side

structure (figure 1). Jaakko will implement the class Stuff and it's child classes (Stuff will be inherited by all non-static battlefield objects). Thus Essi will be left with classes Projectile and Wall as well as the construction of the map.

5 Learning goals

Due to different amounts of experience between the group, we will learn different aspects. Perttu and Joonatan will concentrate more on project structure efficiency and the management of a project with multiple programmers. Jaakko and Essi will need time for learning the libraries and tools used, but will learn also quite a bit about the structure of a real-life application.