COP290: Design Practices
Software Requirement Document
Pocket Tanks

Vaibhav Vashisht 2016CSJ0002
Pratik Parmar 2016CSJ0049

1 Introduction

1.1 Purpose

The purpose of this application is to create a open source version of ”Pocket Tank web game”.

1.2 Project Scope

Objective behind developing this game is to provide more features to the players without the need to get deluxe version of the game and also give an improved version will chatting facility.

2 Product Features

The Software consist of following functionality.

1. **Offline Gaming** Pocket Tank will be played on one computer.
2. **Server Side** Server will be created so that 2 players can play this game from their computer.
3. **Chat System** Chatting Facility for players to communicate during their match.

3 Operating Environment

Pocket Tank will work on any operating system but the server need to have all the required software dependencies working on it.

4 Software Requirement

As this is a web based application so only the server computer will need to have required software.

1. **Database** Firebase, MongoDB or MySQL,Preferably MySQL will be used.

2. **Node Js and npm** For creating server.
3. **Web Browser** Chrome or Firefox, to run the game.

5 Technical Process

Following would be the Languages/tools would be used to develop this game

1. **HTML5** It will provide basic structure to frontend of Pocket Tanks.
2. **Styling**
   CSS3 and Bootstrap
   These will provide a better presentation for structure of HTML and graphics.
3. **Graphics Rendering**
   HTML Canvas or CreateJs(EaselJs and TweenJs)
   These will generate graphics and animination like Terrain, Tanks etc for the game .
4. **Music and Sound**
   CreateJs(SoundJs)
   For providing music and sound to the game we will use SoundJs. Each kind of event in the game will have a different sound which will be played when that event occurs. For Music in game, It will run through out the game unless the play mute it. User will have the option to adjust volume of the music and sound.
5. **Functionality**
   Javascript AngularJs
   These will handle the working of the game. Movement of graphical component will be handled by these and all the frontend work like DOM handling etc will be covered.
6. **Server-side Programming with TCP**
   NodeJs and ExpressJs
   Backend will be handled by NodeJs and ExpressJs. It will create a server and handle all requests or control demanded by the client. Database will also be handled by this.

7. **Database**
   FireBase,MongoDB or MySQL
   MySQL will be used for data storage. As it is a Relational Database management system data will be stored in form of Tables.
   Player Id will the primary key for database.

8. **Communition between Server and Client**
   socket.io
   For Real time communication between server and client socket.io will be used.
   It will also handle texting in the game and transfer data between client and the server.

9. **Video and Audio Transmission**
   WebRTC and PeerJs
   WebRTC is also used for Real time communication between client and server. It allows audio and video communication to work inside web pages by allowing direct peer to peer communication.
   For Peer to Peer communication, PeerJS will be used which will create a PeerServer, on that server we can have a secure video and audio transmission. PeerJS simplifies WebRTC peer to peer data,video and audio calls.