ICS 168 - Sprint 2A - Design Document

Team: CodeBombers, Students: @cayuso - Christopher Ayuso, @whitnet - Whitney Tran, @mengfanw - Mengfan Wang, @jhsosa - Joshua Sosa, @dtwilli1 - Daniel Williams

1) Your game will need to have login and persistence of user information. What kind of information, besides user login and password, should you store? Game scores? Levels achieved?

Login:
- -IP Address
- -ServerPort
- -Username
- -Password

Stored Information:
- -Rounds Won
- -Rounds Lost
- -Players Killed

2) What screens will your game have (besides login and main game play, if any other)?

-Login Screen
-Chatroom Screen
-Lobby Screen
-Main Game Screen

3) And most importantly, how will your multiuser game look like? What interactions are possible? How do players compete or cooperate? What kind of network interactions will you expect to have? E.g. send username+password, receive game score, move left, move right, etc.

Login Screen

-Clients enter Ip address, Port, username, password to connect to the main server

Interactions

-Logging in with previous account username
-Logging in with new account username
-Invalid Password Message
-Invalid Server Message

Chatroom Screen

-Log out Button
-Visual of available lobbies to join
-Chatbox with usernames and messages
-Join/Create Lobby Box

Interactions

-Creating/Joining Lobbies
-Chatting with other connected players in chat room
-Logging out

Lobby Screen

-Player names in lobby
-Lobby host indicator
-Chatbox
-Start Button
-Log out Button

Interactions

-Host starting main game once 2 or more players are in the lobby
-Chatting with other players in the lobby
-Logging Out

Main Game Screen

-Canvas game
-Chatbox
-Round Won Visual of each player
-Log out Button

Interactions

-Game Mechanics
-Chatting to players in game
-Logging out

1) Your choice of game

- BomberMan (**Software Stack** - Web Browser JavaScript, HTML, C++ Server)

2) Your plans for how the game is played, what are the game objects and their allowed movements and behaviors

- Players move around the map and are allowed to set bombs.
- Bombs are able to break certain blocks but not walls.
- Players can acquire pickups that modify play. (Extra Bombs, Speed Increase/Decrease, Bigger Explosion)
- Players can kill other players and themselves with bombs.

3) How do you plan on making it a multiuser game? Give a brief description of the high level architecture. In this section, you might also want to include your ideas for scoreboards, databases or files to persist information.

Players compete to survive via placing bombs to break down walls to uncover power ups that will give them a competitive edge against opponents. Players can gain power ups like extra bombs to place, increased walking speed, or increased explosion range. With these power ups, players can strategically attempt to kill other players. Scores will be kept in the form of rounds won rather than players killed. This would be persistent information for the server to keep and display in the form of a leaderboard. We can also implement persistent information of statistics such as bombs placed, players killed, or walls broken down.

4) Specifically, talk about the architecture of the networking aspect. Some things that you might talk about in this section include TCP/UDP, bidirectional communication between client and server.

We are using a Javascript and HTML based client. As well as a C++ based server. Our communication between client and server is TCP. We are planning to use a hybrid client/server architecture. Some game logic will be implemented on the client and the server. With this implemented we will send relevant information from the client to the server to update statuses and movement. The server will then relay this to other clients. We will have a database in the form of a file upkept by the server that holds the username and passwords. This file will also keep track of rounds won in the session and the overall rounds won for the user.

Client events sent to the server:

-Directions Change
-SetPlayerDirectionMovement
-StopPlayerDirectionMovement
-GameStart/Over Event
-BombPlacedEvent
-TimeStamps
-PlayerDeathEvent

Server events sent to Clients:

-Blocks Destroyed
-Player Position
-Bomb Positions
-TimeStamps
-Bomb Animation Stages
-Randomized Item Locations
-Round