Sprint #: 2
Game Name: Bomberman
Team: Arielle Chongco, Jasmine Hunter, Luke Lohden, Jaysen Gan, Matthew Ruiz, Akihiro Izumi
Date: 11 April 2016

Work Completed Summary

- Added new art assets to make the game unique from the ones that came with the default game we started with.
- Implemented main menu
- Coded a 'Login' Server
  - Handles login authentication
  - Saves account details in database
  - Lists active games for clients to connect to
- Implemented client to client game connection to run a multiplayer game
- Added game sounds to actions within the game

Actions to start doing:

- Introduction of artificial lag
- Implementation of a chat system

Actions to keep doing:

- Keep up regular contact through Facebook
  - The team coordinates efforts through Facebook to keep everyone up-to-date with tasks assigned and what tasks still need to be completed
- Keep meeting up on Mondays and Wednesdays from 2:50 PM to 4:00 PM, Thursdays from 2:00 PM to 6:00 PM
  - This is a good method of confirming details with each other in person as well as setting the goals for everyone for the rest of the week
- Keep updating Trello
  - This is another method we use to keep track of tasks completed as well as tasks we need to complete and who is working on them.

Challenges:

- Getting the Unity sockets to connect to our Python login server. The LLAPI socket functions to communicate do not work with programs other than unity. We had to look up how socket communication worked with typical C# code. Managed to resolve issue eventually however another issue came up.
- Odd issue between communicating with Python login server and client where the client cannot re-establish connection after the first successful connection. The first connection is always successful. Connections afterwards are unable to access the socket.
- Delay in implementing the login server

- Game server issues with connecting multiple clients. It turned out, after much testing, the connection will not work with public IP addresses, but it will if both devices use private IPs on the same network.

Technical Plans

- In order to Sync the games with the server we plan on using local extrapolation only on the player's position, everything else will be handled authoritatively by the server in lock-step synchronization.
- Currently bomb placement is primarily run on the client side and expects the server to receive the information at the correct time. In order to handle this issue, we might move responsibility to the server so the player requests to place a bomb and then the server will place it for them then update the game state for all players.
- Implement scoring system to base win and lose conditions
- The next sprint requires implementing and handling lag. We will use the Clumsy 2.0 latency simulator recommended for the next Sprint.