**Final Design Notes**

As noted in the project proposal, the aim of this project is to design a “bullet hell” type shooter game. Therefore, in contrast to other shooter games, the focus of my game is on dodging an overwhelming number of bullets to defeat enemies and progress in the game. This project was heavily influenced by similar games, such as those from the Touhou series. Following those games, my game involves various different types of enemies and bullets/projectiles, as well as power-ups to strengthen the player. The player starts with 3 lives and 3 bombs, which can be used to destroy all onscreen enemies, and they can gain lives and bombs through power-ups. In addition, like the Touhou games, the player can level up their weapon by collecting power-ups to become stronger, as well. When the player is hit, they will lose a life and subsequently go down a weapon level. To add complexity to this mechanic, my game also implements a feature where each of the player’s bullets becomes stronger based on how many enemies he has killed. This power level will also go down by one when the player is hit. The weapon level, on the other hand, just increases the range of the player’s bullets. At the end of the game, the player will face a boss who shoots various bullets and moves in a pattern. When the player dies or beats the game, s/he can enter his/her name, which will be recorded along with the score and be added to a leaderboard of high scores. A feature that my game implements that other similar games do not, however, is cooperative multiplayer with two players. This feature is available both as local multiplayer and as server-based multiplayer. Different clients can connect to one server and work together in the game to beat it. I implemented this feature to make the game more entertaining since the player can play with friends, and because I noticed that this feature was noticeably lacking among games of its kind.

In regard to the user interface, upon opening the game, the user will first encounter a splash screen that allows s/he to access the different aspects of the game. The instructions page shows the player how to play the game and gives information about controls/features. The leaderboard page displays high scores of players who have played the game, allowing players to display their prowess at the game. The main menu also gives access to the different game modes: single player, local two player, and server-based multiplayer. Upon entering the game, there are sound effects and music in order to make the game more immersive. In addition, many of the sprites for the player and enemies are animated as well, making the game look better graphically. The moving clouds in the background serve a similar purpose by representing the idea that the player is flying through the air. Finally, when an enemy dies, there is an animated explosion, making their defeat more realistic and satisfying.

For the actual code, I used mainly object-oriented programming. I made separate objects for the player sprite, different bullet types, enemies, powerups, the explosion animation, and the background clouds. User input is taken through keyboard and mouse, and, in the server-based multiplayer mode, it is sent from the clients to the server, which updates the game, and then sends the information back to the clients. I made animations for various sprites such as the player sprites and enemy sprites by appending frames of gifs to a list and cycling through them. For the gameplay, there are separate functions to represent the actions of the enemies and what happens when the player hits the enemies. There are also event functions for moving the player, getting powerups, and other game features such as using a bomb. On the server-side, there are several update functions that send data to the client, and there are corresponding network functions on the client. Finally, there are separate draw functions and helper functions to draw the various game screens, such as the 1 player screen, 2 player screen, main menu, high scores, and instructions. There are screens for game over and for winning a game too. All of this is framed in a run function similar to in the Tkinter framework.