Shaelynn Li

Dr. Crawshaw

COMP 4911

February 27, 2020

Computer Network Project Proposal: Competitive 2048

Competitive 2048 is, as its name may suggest, a version of the popular mobile game, 2048, where players begin with a 4 by 4 block grid containing one initial block with an initial value of either 2 or 4. The player can swipe the board in the four standard directions to shift the contents of the grid in that direction, and with each swipe, another block of either value 2 or 4 is generated in a random location. Blocks of the same value will combine and sum to a different value when colliding, culminating in the titular 2048 block (and beyond). Distinct from the single player, standard 2048 game, however, Competitive 2048 will involve two or more players, and will be turn-based, with all players playing with the same, identical game board, which will be displayed across all clients and updated concurrently as the game progresses. During each round, each player can perform a limited number of swipes while the other players wait. The intent of the game is to achieve the highest possible score, while also attempting to outmaneuver the other players and force them to cause a game over. To further differ from standard 2048, and to inject more interest into the game, some additional actions will be added in the form of various power-ups, such as the ability to force an opponent to skip or repeat a turn, or to clear the entire board of all high/low-value blocks.

The game will be programmed in Python and will use a client-server model to achieve its multiplayer capabilities, with one player being the host server of the game, and all others connecting as clients. The process of creation will proceed in stages: the first stage will be to create a fully-functioning, single player version of the game; the second will be to extend it for multiple players; and the third will be to incorporate the power-up elements. Since Competitive 2048 needs to track and display several different values to all players, such as player scores and the status of the game board, it will be necessary to ensure that all of these values remain consistent across all clients, and that these values are not lost in transit. This may entail a sizeable amount of data being transferred over the network with each game move, as well as complications in ensuring that the data is transferred without loss. I anticipate that these two aspects may be some of the more significant challenges to this project, and may require research into how other similar multiplayer games achieve these aspects effectively.

For the aesthetics and game aspects of Competitive 2048, I intend to use a Python programming library such as Pygame to achieve an appealing, windowed visual appearance. Python’s *socket* module provides a possible way to achieve the game’s network communication aspects; similarly, PodSixNet is a Python networking library designed for writing multiplayer games, which also exists as a possibility to accomplish Competitive 2048’s multiplayer capacities. The final version of Competitive 2048 will include all of its program files and a Design Document outlining the overall design of the program and documenting the specifics of the code, as well as any potential improvements to various aspects of the game, or future expansions in the game’s capabilities.

Possible Resources:

1. Pygame: <https://www.pygame.org/docs/>
   1. Open source Python programming language library designed for creating games.
2. PodSixNet: <https://github.com/chr15m/PodSixNet/>
   1. Python programming language designed for usage with multiplayer Python games
3. Tutorial for developing multiplayer online games with Python: <https://techwithtim.net/tutorials/python-online-game-tutorial/>
   1. This tutorial contains information on the various aspects involved in designing a multiplayer online games with Python.