**Requirements Specification: Grid Runner**

**Problem Definition:**

We are working on creating a modified Splix.io replica in the console. We intend to make the final version to have a fixed sized-grid multiplayer functionality. The game’s basic functionality is to allow the player expands their territory by directing their movement path with WASD controls (up, down, left, right respectively). A user captures territory by connect their path back to their previously acquired area. To simplify the conquer algorithm only the cells that are a part of another user’s uncaptured path will be automatically acquired while the area in between the path and previously conquered territory will remain unchanged. These functionalities will expand to multiplayer with the added requirement of being able to destroy another player by cutting into their unfinalized path.

**Requirements:**

* Player loses when they run into their own unconquered path
* Player can acquire territory by moving across unconquered blocks and then over their previously acquired territory
* Player loses if they run outside the grid (world not endless)
* Player acquired territory, player’s current position, and player unfinalized path is clearly visible with appropriate color and intensity settings in the console
* When a player loses all of their territory is lost and gets marked as neutral
* A player can defeat another player when they cross the unconquered path of the other player

**Assumptions:**

* User has internet access (without restrictions)
* User is executing the program on a windows machine

**Data Structures:**

Clients:

* We will use a 2D vector to store *GridPoint* objects that contain information about each grid position (such as captured, path, player position, and player ownership properties)
* We will use a map to store *Player* objects that contain information about each players’ name, color and path state (allows easy retrievals by ID of the person)
* We will store client’s current path state in a hash of pairs<int,int> (to quickly retrieve presence of a grid point in the path)

Server:

* We will use a map to store *Player* objects that contain information about each players’ name, color and path state (allows easy retrievals by ID of the person)
* We will store clients’ current path state in a hash of pairs<int,int> (to quickly retrieve presence information of a grid point in the path)

**Functions** (names may differ in actual program):

Drawing:

* DrawGrid(): will redraw the entire grid into the state that resembles the current state of the game (which cells owned by who, where are players’ paths, which cells are still uncaptured)
* SetXY(): will set cursor position to proper location in console
* SetColor(): will correctly set the color of the cursor in console
* Clear(): will clear the console screen to prepare redrawing of the grid
* DestroyPlayerCells(): conquered cells set to neutral, new spawn point set
* AddPlayerCells(): adds current path to conquered cells, clears path data structure
* MovePlayer(): moves player cell to a new location, with option to add cell to path

Compute:

* NextGridPosition(): calculates the next grid position (-1, -1 if outside grid) based on the players current input

Inputs: playerDirection, curX, curY, gridSizeX, gridsizeY

Output: nextX, nextY

* MovePlayers(): on server this function performs changes in server player objects position
* and then checks the next action for the player (capture, die, move on owned land, move on other’s land)
* CheckPathCompletion(): checks if a player has completed its path and captured territory (this runs for every player before checking for conflict), sets player message option to be path complete “PC”.
* CheckConflict(): checks the entire grid for players on each other’s paths or one’s own path (if in conflict marks the players option to be Death “D”)

**Pseudocode:**

CheckConflict()

**for** each player

**for** each other players

**if** in any players path

mark other player for death

CheckPathCompletion()

**for** each player

**if** path is not empty and currently on owned territory

mark current player for territory capture

add score to player (amount of elements in path)

NextGridPoint(ref curX, ref curY)

**If** direction is up

curY + 1

**else if** direction is down

curY - 1

**else if** direction is right

curX + 1

**else** direction is left

curX - 1

**if** curX < 0 or curX > grid width or curY < 0 or curY > 0

curX = -1

curY = -1

MovePlayers()

**for** each player

Find next grid point (function above)

**for** each player

**if** not dead because outside the grid

Check path completion (function above)

**for** each player

**if** not outside grid or completed a path

check conflicts (finction above)