**Project Proposal: Grid Runner**

The grid runner project is essentially a console replica of the popular online game called Splix.io. We are planning to begin by creating a simple single player platform in which the user can move a rectangular grid on the console. In order to be able to read in user input in real time and without lag there will be a need to use many C++ functionalities along with inline assembler portions that demonstrate our knowledge of the classes content. The final version of the project is expected to be multiplayer capable with assembler portions handling most of the algorithmic work in filling in the proper cell values and checking various conditions.

The major problems for us in this project consist of three main themes. The first is displaying the current state of the grid on the console screen. Because of the need to use a large chunk of C++ code, the assembler code to print characters, change text color, and move to the appropriate section of scree at the right time will need to be properly linked to the .cpp files. The second problem is to get a consistent message system running between the game server and all of the game clients in order to make the game multiplayer. Solving this problem will consist of exploring the Winsock API and properly setting up connections and listening to sockets so that each player can send their inputs to everyone else. The final and most challenging part of this project is replicating the game functionality as seen in Splix.io. We will have to focus on two primary algorithms. One to fill in the captured area one the user has managed to draw a path outside of his territory and an algorithm to destroy people when another player has crossed their unclosed path. We currently planning on implementing a timing system in which a user will move to the next cell at certain intervals (unlike continuous movement in splix.io). This will make it easier to fill in the captured areas and determine if a player needs to be destroyed.

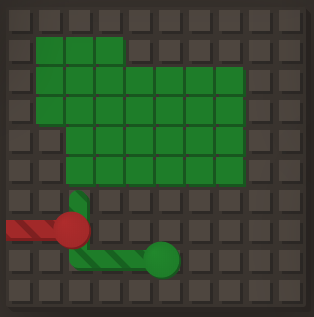


Figure . Algorithm to destroy players

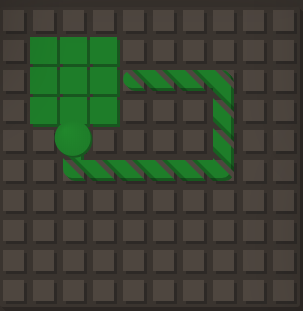


Figure . Algorithm to fill in captured area

Inline assembly code will be are primary concern and the focus of our presentation however there are plans to play a couple demonstration games with the whole class if the multiplayer functionality of the game works.