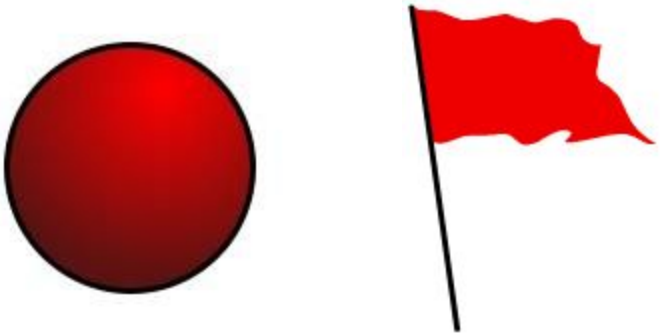
Project #5

Super Flag

Team 22

11 AM Section



Developed By:

Trevor Parchem - (tparch3@uic.edu)

Simon Acosta - (sacost6@uic.edu)

Sean Barber - (sbarbe2@uic.edu)

David Woloch - (dwoloc2@uic.edu)

Purpose:

Super Flag is an adaptation to the popular online multiplayer capture the flag game “TagPro”. Each client will connect to the server and be put into a map where players are circle balls and can move around with WASD and attempt to collect flags that help them score points while fending off defenders. The players will be able to bounce off the walls as they are constrained to the size of the map. The ability for power-ups has been added, these when collected increase the speed of the player. Created using Phaser, Socket.io, Express, and Javascript.

High-Level:

- Server:
  - Express
    - Setup server
    - Routing of webpage
  - Socket.io
    - Accept and reject connections
    - Send out messages to connected clients
    - Receive messages from connected clients
  - Node.js
    - Express and Socket.io are both modules extending Node.js
- Client:
  - Phaser
    - A game platform for WebGL
    - Display graphics

        - Physics engine
    - Socket.io
        - Send out messages to a connected server
        - Receive messages from a connected server
- Frontend:
    - HTML

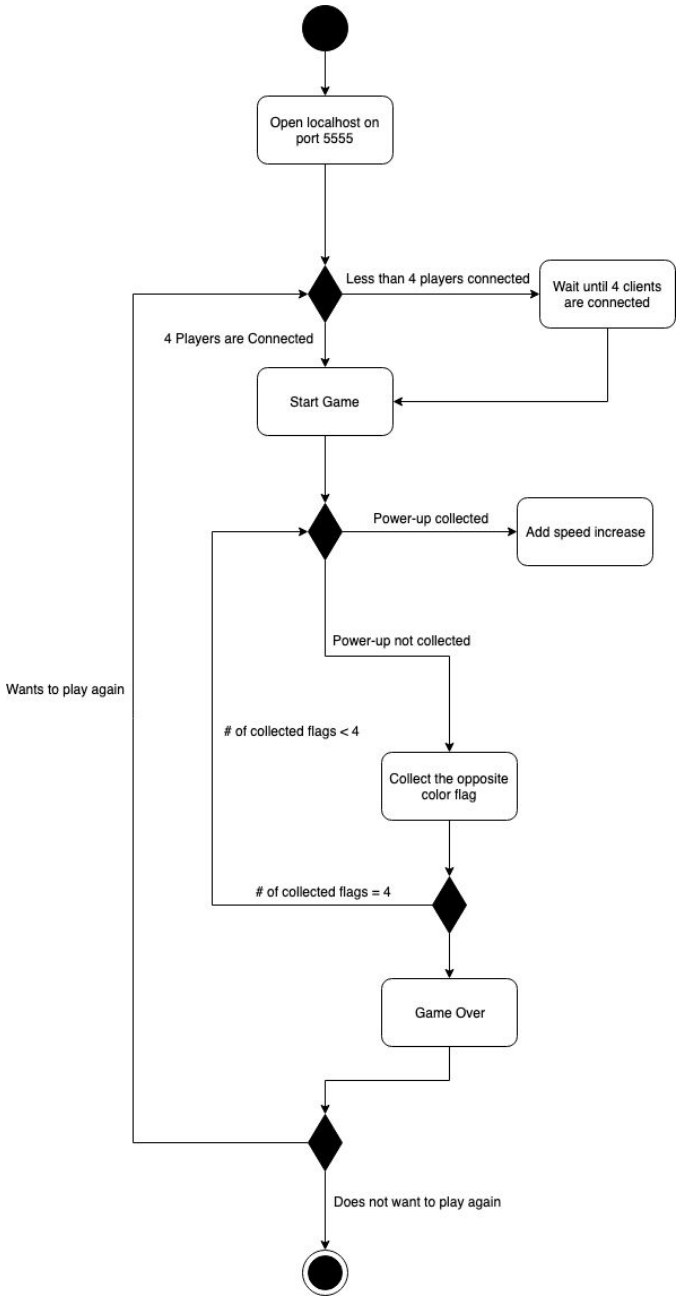
      Implementation Details: A body containing the three scripts we are using, most importantly game.js. Phaser does all the work for updating a canvas within the game.js code.

Low-Level:

- Server
    - Create a server and listen on port 5555
    - Route pathnames using express.io
    - Have clients connect
        - If 4 clients are connected, do not allow any new connections
        - Else allow a connection and set up a new player
        - Receive messages about the flag status, player movement, and playing again
    - Send messages to clients regarding the player position, the updated flag position, and player score
- Client
    - Create a connection with the server
    - Create an instance of Phaser using a config file
    - Create the gameplay scene using a background of 800,600

- Enable collisions with the world boundaries so players cannot leave the screen
- Create a group of physics objects that are the other players
- Use cursors provided from Phaser to get keyboard input
    - Use the w, a, s, d keys to control the player movement
    - Use the spacebar once a game has finished to possibly replay
- Connection to server
    - Get the player list, if a new player has connected, flag locations, power-up locations, score, and client status
        - Client status contains whether they want to play again and whether the game should start
- Display the scoreboard, players, flag location
    - If someone wins, display the winner, then ask for replay by pressing spacebar
- Using Phaser update
    - Emit player location
    - Update player location on the map and their velocity
- Create a player based on the number of players on the server currently

Model:
Activity Diagram



Model:
UML Diagram