Battle Royale

Submit Assignment

Due Apr 21 by 11:59pm **Points** 125 **Submitting** a file upload **File Types** zip

Introduction

The first projects have given you an opportunity to develop skills in architecting and coding a game, along with learning (probably) a new language and development platform (web). This project's purpose is to put all of that to work in a sophisticated game; this is intended to be the biggest project of the semester.

Warning: The assignment description is open for (small) changes throughout development as you work on the project and discover changes that need to take place. The description is not-yet complete as of March 12th, we will make discoveries along the way that necessitate changes to the design. That sounds an awful lot like real software development, now that I think about it!

Assignment

Write a Battle Royale style game detailed at the following link: **link**

Technical Requirements

- Must be a Node.js server-based solution.
- Keyboard configuration options
  - Keyboard commands

    The interface for this must present a screen where the name of the game function is displayed and to the right of it, the key combination. Using the mouse, the user can select the action, then some visual will change to indicate it is possible to now select a new key combination, then the user presses the new key combination (doesn't have to be a combination, could be a single key) and that immediately becomes the new keyboard shortcut for that game function.

These configuration settings must be (at least) persisted to the browser's local storage.

- You must create a particle system that allows the game code to start an effect and have that system provide the complete management of all active effects. In other words, your game code will have a single particle system "manager", for the lack of a better term. Then, during gameplay, when an effect is needed, the game code invokes an effect, using the particle system, and it goes from there. Code for invoking an effect might look like..

  ```
  ParticleSystem.createFireEffect({
    position: { x: 100, y: 100 },
    duration: 1000, // milliseconds
    ... // ...won't be a parameter, that is filler for whatever else is needed for the effect
  });
  //
  // This particle system will also have .update(elapsedTime) and .render functions.  This way
  // your gameplay code doesn't have to do anything other than create an effect and let the
  // particle "engine" take care of everything.
  //
  ```

  - (*tasteful*) Particle effects are expected for the following items:
    - When a player is eliminated; by another player or by shield.
    - When a projectile hits another player.
    - When a projectile hits a building.
    - Randomly located around the inner/closing edge of the shield.
- Networking
  - Must use client prediction with server reconciliation.
  - Must use entity interpolation.
  - Must send individualized game updates to each player in the game based on their spatial location.
- User Registration/Login
  - Use a very, very simple password hashing scheme, doesn't need to be cryptograhically secure; but you can make it that way if you like (it isn't that hard actually).

  - Do not use a DB like MongoDB or otherwise, use a simple flat file for storage. I ask this not because it is the best idea, but because I want the server to "just work" when I run it on my computer without having to set up a DB or DB connection. I also want it to always work in the future, so no external DB connection either that is already setup.
- High scores (scoring) must be persisted to the server.
  - Same as above, use a simple flat file and read into memory.
- Sound
  - Sounds effects for the firing of weapons, explosions, ...
  - (optional) Music playing in the background.
- Health bar displayed above the self player's character, but not other player's characters.
- Game View
  - Player view is a subset of the overall gameplay arena.
  - Buildings are always displayed.
  - Mini-map on player view that shows their location in the overall arena.
  - Player sees a representation of their field-of-view (FOV).
  - Player can only see other players and power-ups within their FOV.
  - Player can not see other players and power-ups fully obscured by a building, even within their FOV.
  - ? Player can adjust their FOV to be shorter but wider, or longer but narrower.
  - Any explosion that occurs within the players spatial surrounding is visible and audible.
  - Game status is displayed in the upper right-hand section of the screen.
  - Game messages are displayed in the status region of the display.

User Interface

- Menu system
  - Join Game Lobby
  - Create New User
  - Options
  - High Scores
  - Credits

Art & Sound Assets

The game may be themed in any way, but the art assets must meet the game design and technical requirements. Some possible sources for art assets can be found at the following links:

- Open Game Art: https://opengameart.org/ (https://opengameart.org/)
- Game Art 2d: https://www.gameart2d.com/freebies.html (https://www.gameart2d.com/freebies.html)
- Space Shooter: https://opengameart.org/content/space-shooter-redux (https://opengameart.org/content/space-shooter-redux)
- 2d Characters: https://opengameart.org/content/700-sprites (https://opengameart.org/content/700-sprites)
- 2d Characters: https://opengameart.org/content/platformer-sprites (https://opengameart.org/content/platformer-sprites)
- Structures: https://opengameart.org/content/colony-sim-extended-version (https://opengameart.org/content/colony-sim-extended-version)

Battle Royale

Criteria	Ratings		Pts
Node.js server-based solution *All files served to the browser from a server.*	10.0 pts Full Marks	0.0 pts No Marks	10.0 pts
User configuration *Configurable controls and game options, persisted to local storage (or server if you want).*	15.0 pts Full Marks	0.0 pts No Marks	15.0 pts
Sound	5.0 pts Full Marks	0.0 pts No Marks	5.0 pts
Game scoring, with high scores persisted to server *This also includes the user interface to view and sort the scores.*	10.0 pts Full Marks	0.0 pts No Marks	10.0 pts
Overall game and menu interface/presentation	15.0 pts Full Marks	0.0 pts No Marks	15.0 pts
Particle system & effects ** Player eliminated * Projectile hits another player * Projectile hits a building * Randomly around the inner/closing edge of the shield*	20.0 pts Full Marks	0.0 pts No Marks	20.0 pts
Networking - Client prediction with server reconciliation	15.0 pts Full Marks	0.0 pts No Marks	15.0 pts
Networking - Entity interpolation	15.0 pts Full Marks	0.0 pts No Marks	15.0 pts
Networking - Individualized game updates *Each client in the game is sent a customized update based on their spatial location in the arena.*	10.0 pts Full Marks	0.0 pts No Marks	10.0 pts
Player registration/login	10.0 pts Full Marks	0.0 pts No Marks	10.0 pts

Criteria	Ratings	Pts
		Total Points: 125.0