Project Name:

Asteroids.io

Project Group #1 Members:

- Jack Schiavo - Back-End Programmer
- Caroline Greer - Gameplay Programmer
- Raghav Gupta - Front-End Programmer

Project Location

https://gitlab.cs.umd.edu/cmsc436spring2019/cmsc436-jschiav2/tree/master/semesterProjectGroup1/Asteroids.io-master

Description of Project

Our project, Asteroids.io, is a multiplayer adaptation of the Atari arcade classic for the mobile platform. In the original game, a single player is assailed by dangerous computer controlled UFOs in an asteroid field, where if the player contacts either the asteroids or the UFOs' lasers, they will lose a life. In this new version of the game, two rival players can face off against one another in space, firing lasers at one another amidst flying space debris. Instead of dying when asteroids hit your player's spaceship, the spaceship spins, stunned and unable to shoot, leaving them vulnerable for their opponent to fire upon! After a user shoots their opponent 10 times, they are declared the victor in the game.

Throughout the game, asteroids are randomly generated outside the field of view. These spinning hunks of rock fly through the play area and can be destroyed by the players' lasers. When hit by the lasers, larger asteroids split into smaller component asteroids and small asteroids disappear entirely.

Most of the gameplay elements, such as the sprites, animations, background, text, collision physics, and music, are handled using Apple's SpriteKit.

The project features original artwork, as well as atmospheric particle effects, to give the scenery a fun space-themed look. Lasers are red rectangles which stand out against the violet starscape behind them. Music is heard while the game is in session and other sound effects are played to enhance the gameplay experience.

Some animations are also featured to beautify the game. Asteroids spin as they hurtle across the play area, making them look more chaotic on the screen. Some gameplay-relevant animations are also added, such as a blink and spin animation for the spaceships. If a player is hit by an asteroid, it sends their spaceship spinning and unable to fire. Finally, if a player hits their opponent, the opponent blinks in and out of existence and is invulnerable temporarily to indicate that they were indeed hit.

The multiplayer is handled by an authoritative server utilizing Socket.io and Node.Js. This allows for real time communication between devices, as well as keeping the game fun and fair. The server can handle multiple games at once by organizing matches into different rooms so that no client receives the wrong messages. Each player in a game sends the server its actions each frame (every 1/10 seconds); the server then updates the player's state and broadcasts the information to all clients in the room. By sending only the actions we can validate whether it is a legal move or not, mitigating unexpected results. The server also handles synchronizing object instantiation and deletion on a client's request. Asteroids are procedurally generated by the server at random intervals and broadcast to the clients with the information for direction and position of the asteroid. Upon any type of collision in either client's game the server will verify the hit and broadcast an updated game state. In addition to localhost, our multiplayer server is deployed on heroku which allows us to play our game wherever we want.

Features of the Project

- Live multiplayer with synchronized gameplay
- Scales to support multiple games at once
- Animations, such as a blink and spin animation
- Server created in node.js
- Live server running on Heroku handles long-distance multiplayer
- Original artwork by our team
- Original virtual joysticks for movement and firing
- Particle effects when objects are destroyed
- Sound effects when the lasers are fired and objects are destroyed
- Background Music
- Main Menu

Evaluation of Goals Met

Our team was able to meet our basic, minimum viable product goals. We have a fully synchronized and functional game with sprites, sounds, and animations that is obviously inspired by the original asteroids concept. We did not finish our stretch goals of including power ups and achievements, but this could be done with the skills we have developed over the course of this project and would be a good future goal.

Assets Used

All artwork (asteroids, spaceships, background) was created by our group for this project.

Particle effects were generated using Xcode's SpriteKit particle effect templates. We used the official apple tutorial here

Sounds were located on Freesound.org and freesoundeffects.com. All of the sounds are Creative Commons Licensed and available to the public for free use.

For both the explosions -
https://www.freesoundeffects.com/free-sounds/explosion-10070/

Background music -
https://freesound.org/search/?q=space&f=&s=score+desc&advanced=0&g=1

Laser -
https://www.freesoundeffects.com/free-sounds/missile-10071/

Project Setup (From the ReadMe):

Clone the project and follow the steps below to setup

Setting up the server

Make sure you have you node and npm installed on your machine.

1. **Navigate to `Asteroids.io/server/` in the terminal**
2. **Run `npm install` to get dependencies**

Starting the server on localhost

- Run `npm start` to start the server
- Or run `nodemon server.js` if you want the server to automatically restart whenever changes are made to the server code

Setting up the app

Make sure you have Carthage installed on your machine

1. Navigate to `Asteroids.io/app/Asteroids.io/`
2. Run `carthage bootstrap` to install frameworks

Running the app with localhost

Switch the target scheme in Xcode to Asteroids.io-Dev and run

Running the app with live server

Switch the target scheme in Xcode to Asteroids.io