Game Concept/Summary

My game is a 2d asteroid shooter. Players must dodge asteroids whilst attempting to shoot and destroy the asteroids for points. The aim is to achieve the highest score possible. The player gains points for shooting asteroids, and loses points for colliding with them. The player will advance to the next level when all the asteroids on the current level are destroyed. Levels increase in difficulty by adding more asteroids.

Controls:

Spacebar to shoot

Left arrow key to rotate left

right arrow key to rotate right

up arrow key to move forward

player slows down and stops automatically

Game Structure



Start of game. Three large asteroids. These decompose into two smaller asteroids each



The different asteroid sizes after they have been shot



Level 2 starts after all asteroids cleared in first level. The process repeats infinitely

Implementation Specification

The game runs using only the phaser.js file and one other js file; main.js. All objects are created and defined through main and all the logic and behaviour is handled here as well.

First a new game is made. The game takes the usual phaser.game parameters and also calls the functions preload, create and update. I have used preload to load my images into the game before they are assigned to variables etc so they have enough time to load properly before startup. I have created class-scope variables for my bullet group, asteroid group, ship, movement keys, scores etc. sticking to js convention of declaring variables in groups and before they are needed, as to improve readability.

I have used create to assign backgrounds, sprite images, populate the game with asteroids, add arcade physics, create interactive keys and initialize the scores. Update handles changes in movement, asteroid numbers, score and changes level. I have used groups for a data structure for my bullets and asteroids, as I can loop through each using foreach as well as add and change the groups easily.

All the sprites are given arcade physics which allow me to check for collisions between each other as well as define max velocities and drag coefficients etc. The starting asteroids are given random locations to spawn on the screen and also random velocities. Subsequent asteroids spawn where their ‘parent’ was destroyed, their velocities are still randomized however.

Firebullet takes no parameters, it finds the first bullet in the bullet group that does not exist, and resets its attributes, I make it appear alittle in front of the player ship and give it a velocity dependent on it’s rotation, which itself is dependent on the rotation of the player.

screenWrap takes a sprite as a parameter, it checks if the sprite is out of the gameWidth and height, if it is, it makes the player appear on the opposite side to which it is currently occupying.

I have used asteroids.forEach with a parameter of an anonymous function. The function serves to run a screenWrap check on the asteroid, it also checks for collsions between the asteroid and the player and bullets. It also updates the score and highscore.

* Provide a clear explanation of how the data structures/objects and code used within the game are organised and operate
  + **Code Structure**: Document the overall flow of the code and where it is located. Identify each of the functions that you have created and provide an informal specification of each including:
    - Purpose
    - Parameters

Critical review

The implementation is good because the way I have handled things is simple and easy to understand. The logic is kept from being complex for the most part (changing levels is abit of a mathematical trickery however). Where possible I have used phaser’s libraries to simplify collision handling and sprite manipulation which has greatly reduced the size of the code I have produced.

The variable names are sensibly named and this makes the code more readable. Groups and singular are separated by plurality, and are named as uniformly as possible, e.g. score, highScore.

Executing a preload, create, update cycle is a good, well known practice in javascript which again makes the code easier to pick up and understand, as well as edit and improve.

Assets

All assets are open source and found/created by <http://kenney.nl/assets/space-shooter-redux>

References

Any code I have added or implemented from other sources can be found here:

1. Base for game-Phaser examples <https://phaser.io/examples/v2/arcade-physics/asteroids-movement>
2. How to use various methods- Phaser Docs <https://phaser.io/docs/2.4.4/index>
3. How to random spawn asteroids-ZekeChan <http://www.zekechan.net/asteroids-html5-game-tutorial-3/>
4. How to add score – coding geeks <https://www.youtube.com/watch?v=eDNWzpIphFo>
5. Checking for collisions – Phaser tutorials <https://phaser.io/tutorials/making-your-first-phaser-game/part5>
6. Where I got my assets – kenney <http://kenney.nl/assets/space-shooter-redux>