Asteroids JS

js-asteroids (as the project is called on Github) is a game where most of the code is written but needs a few tweaks to make it work.

Running the Game

To run the game you need to have Python3 installed and have the prerequisites for the game installed. The only one that needs installing is Flask.

```
pip3 install flask
```

To run the game cd into the directory

```
cd <path_to_download>/js-asteroids
python3 main.py debug
```

The debug flag is optional and will allow hot reloading of any code changes rather than requiring you to restart the whole app.

To stop the app use:

```
ctrl^c
```

(which is the control key and the letter c held at the same time).

Once the app is running you can navigate in a browser to:

http://localhost:5000

And you should see a menu screen where if you select start you will see a blank screen (this is expected). To navigate the menu use ‘w’ and ‘s’ to move up and down and then ‘e’ to select an option.

If you make a change and nothign shows then it might be worth checking the console (right click and inspect element, then select the console window) and see if there are any errors. If there are then it will tell you where the error occurred (file and line number).

Exercise 1

We first need to tell the asteroid and the player object to draw themselves and pass in the drawing context so the game knows where to draw the game objects.

We need to open up the file game.js (which is where the game is run from).

In here there is a function called GameDraw - locate this and insert the following lines in the appropriate section. To draw the asteroid insert:

```
asteroid.draw(ctx);
```

And to draw the player we need to insert the code:

```
player.draw(ctx);
```

Now refresh the page and start a new game - it should show a player character and an asteroid.

Exercise 2
Next up we need to be able to move the player left and right. This is done in the file ‘player.js’ in the function PlayerUpdate().

There are some handlers already for when a key is pressed. We need to change the location of the player (which is stored in this.loc.x). There is a speed variable on the player already called this.speed. So we need to change the code to alter the loc.x variable.

eg.

```
        this.loc.x += <the value to change by>
```

Now refresh the page (or restart the app if you didn’t run it with the ‘debug’ switch) and you should now be able to move the player.

Exercise 3
You’ll probably notice that even when we collide with the asteroid nothing happens and it passes straight through the player’s ship.

To do this we need to edit the update function in game.js and add the following code:

```
if( rect_collides( asteroid, player ) ) {
        player.reset();
        asteroid.reset();
}
```

This code checks if the asteroid is touching the player and moves the player back to the start location when it does.

Exercise 4
The game.js file contains the details of all of the parts which load up the player and this includes creating asteroids to throw at the player. Have a go at creating more asteroids to make the game a bit harder.

If you want to have a go at randomising the speed of the asteroids then try looking in asteroid.js and inserting the following code into the function AsteroidReset().

```
 getRandomInt( min, max);
```

Which returns an integer (whole number) value between the min and max values.

Excercise 5

Now have a go at making the game either a bit harder or expanding - perhaps adding a score or timer into the mix. If you want to ask any questions about how to do things you are unsure about then put your hand up and I'll come over.

Further Work

If you would like to customise the space craft then either edit the existing one (in the 'img' folder - called 'player.png') or create a new image of 40x40 pixels and drop it into the folder renaming it to player.png.