Asteroid JS

In this set of exercises we are going to look at how to move using trigonometry. This allows us to move all over the screen and rotating allowing us to move forward and background.

The first thing we need to do is alter index.html and change any reference to player.js to player_rot.js (careful there are two places it needs to be changed).

Exercise 1
The first thing we need to do is to allow the player to rotate - this is contained in the same place that we moved last time (PlayerUpdate()). We now have a rotation stored in this.rot. So we need to react to the keys ‘A’ and ‘D’ and update the rotation accordingly.

```
this.rot += this.rot_speed;
```

Try inserting this in the key handlers and refreshing the page - see what happens.

Exercise 2
We now need to use Sin and Cos functions to allow movement based on our direction of travel. This needs to happen inside the key handlers for the ‘W’ and ‘S’. Because the direction of increasing Y values is inverted compared to a regular graph you may find that the direction of travel is not what you would expect. Try inserting the following code into one of the key handlers.

```
this.loc.x += this.speed * Math.sin(rot_rad);
this.loc.y += this.speed * Math.cos(rot_rad);
```

Experiment a bit until you get the results you’d expect. Remember that the code will need to be after the conversion to radians (stored as variable rot_rad).

Rest of the Session
For the remainder of the session we are going to look at different things to improve with the code and really make it your own. Like maybe adding a score or transforming the way that the asteroids move - maybe you don’t want them to move for instance. We can explore some of the possibilities together and I’ll run through a few things on the board for anyone who doesn’t know what they want to do.