**Shark Attack– Project One**

My first dev project for the General Assembly Software Engineering Immersive course, and my first ever project using JavaScript.

***Goal and Timeframe:***

To build a functioning browser game with vanilla JavaScript in 7 days.

**Deployment:**

The game has been deployed and is available [here](https://florastocks.github.io/spaceinvaders-p1/).

**Technologies used**

* **HTML**
* **CSS**
* **JavaScript**
* **GitHub**
* **Git**
* **Google Fonts**

**Brief:**

A picture containing background pattern

Description automatically generatedShark attack is an underwater spin on the classic 80’s arcade game - Space Invaders. A player moves left and right across the screen, aiming and shooting the invading aliens, each alien shot is added to their score. The player must shoot every alien before they reach them, and avoid being hit by the incoming alien shots.

**Controls:**

* Click start button to start
* Use the left and right arrow keys to move crab left and right, respectively
* Use the space bar to fire at the sharks

**Process:**

**Day 1- Research, Pseudo coding, Drawing up the Game**

Shape

Description automatically generatedA picture containing graphical user interface

Description automatically generatedBeing my first project, I did not have a plan on how it was that I would undertake creating this game. I started the project by playing the game and mapping out a plan of all the different functionalities that I would need in the game. I created a wireframe to help envision how I wished the game to appear. And wrote pseudocode of my MVP and the different functions I would need, in order to be able to deliver the game in the time was given.

**Day 2 - Creation of the Grid, Start Button, Score, Lives Display, Player Movements**

I created the game grid by setting the width to 15 and made a cell count of the width times width. To create the grid, I used the cell count and a for-loop to create divs, and then pushed these divs to an empty array and append them to the grid div in my HTML. Each cell is given an index number, which allows me to determine where on the board the sharks and player start, and when the game begins, where they currently are, as well as whether they have been successfully hit.

I created the player by adding a player class to a cell, and created a keyDown event listener that allows the player to move left and right when the corresponding keys are pressed, by using a modulus (%) and the width variable, the if statements create a boundary between which the crab can freely move left and right without coming off the grid or throwing an error.

I created the starting button and added in the score and lives display. I did the majority of the CSS, using flexbox.

Text

Description automatically generatedText

Description automatically generated

**Day 3 - Add in the Sharks, Adding the Sharks Movements, CSS, Photoshop**

I focussed on the enemy sharks, adding them to the grid by creating three invaders’ arrays (one for each row) containing the index values of the cells on the grid. Then concatenated the three row arrays together to create an enemy starting position array.

I then worked on the shark’s movement logic, that moves them to the right, down, left, down, and so on. I created a setInterval of 0.75 seconds, which moves the sharks across and down the grid until they reach the bottom.  By using a modulus (%) and the width variable, if statements create a boundary between which the crab can freely move left and right without coming off the grid or throwing an error.

Throughout the day I added in CSS, such as the sharks and the bullets, making them central to their cells, and not letting them repeat within the cell.

**Day 4 - Player Firing, setIntervals, Removal of Shark and Bullet if hit, CSS**

I wrote functionality for the player firing, the lasers in the game are on timers. I added in an event listener to pick up when the space bar is hit. When it is hit, an interval begins which moves a bullet up the board. When hit, the shark is spliced from the enemy array, and points are added to the score. The bullets are removed from the board when they reach the top, if they have not hit a shark. The bullets are also removed when they hit a shark.

**Day 5 - Shark Firing, setIntervals, Removal of life if Crab was hit, CSS**

I used math.random() to select a random shark, from which a bullet will be dropped, and then created an interval that will make these bullets continuously move down the board, until reaching the bottom, or hitting the crab, at which point the class of the bullet is removed and the interval ended.

**Challenges:**

As this was my first coding project, there were a few challenges I came across:

* I struggled getting the lives to be correctly removed from the player.
* I had issues getting the sharks to move left, after moving down on the grid.

**Bugs:**

When the Player hits the space bar continuously in quick succession, the interval of the sharks moving across the board is sometimes disrupted, making them move fractionally slower.

**Future Improvements:**

* Add an instruction page, for Users to read before beginning the game.
* Create pop up page for winning and loosing

**Key learnings:**

* How to use Flex-box well
* Manipulating DOM elements
* working with setIntervals