

ReadMe Template Worksheet

Your ReadMes are the key deliverable that engineers will want to see as part of your job search. Each engineering team will look at different aspects of your ReadMe and repo. Some will go into the code itself and explore. Others will just want to see the showcase code snippets in the main ReadMe. Some will read the full thing to understand your approach, others will skim to specific sections.

It's crucial that you cover all the different sections below to ensure that you've got the information for all engineers that check these out.

We regularly have employers discuss the importance of the ReadMes in what they're looking for and why they interview the grads that they do - so don't underestimate the importance of writing strong ReadMes!

For any pair or group project, you cannot share a ReadMe. These must be written independently to ensure that the engineers reading this understands **your** specific experience and approach.

It's a good idea to start your ReadMes during the planning stage as this is the best way to get ahead and save time when it comes to finalising your first draft post-project. As you plan each aspect of your project, note down what you intend to do with screenshots of your plan and anything else you think would be useful, then when you execute this part in your code, you can adjust this part of your ReadMe as needed depending on how your process changed, or if it went as planned then you can leave it as it is.

Make a copy of this document for each project you use throughout the course and fill in each section.

Once the content has been finalised here, you can then quickly upload these onto your GitHub repository later.

Things To Consider:

- + That there are **no spelling mistakes in your ReadMe** - if you see a spelling error highlighted below, edit this.
 - + Some engineers will reject applicants if their ReadMes are full of mistakes. From their perspective, if your ReadMes are full of mistakes, what is your code like...
- + That your **technologies are capitalised correctly** - i.e JavaScript, jQuery, MongoDB
- + That your **formatting is consistent** throughout - headers, indentation, full stops in bullets etc
- + Any **hyperlink included works**

- + That you **include images throughout** - code snippets, pictures of your planning stage, screenshots of the final project.
 - + These can be still screenshots or gifs
 - + This breaks up the text in your ReadMe and helps to keep the reader engaged
- + That your ReadMe **sounds like you** - see this as an opportunity to showcase who you are to the engineering community and prospective employers.
 - + Think back to the Personal Brand session and how employers want to **get a sense of who you are**. The content you write should sound as if you're talking through your experience.

ReadMe Sections

Description

Our first project of the course commenced during week 3 where I recreated the arcade game, Space Invaders using HTML, CSS and JavaScript. The game involves moving your player left and right using the arrow keys on your keyboard, shooting at your enemy; a grid of aliens who slowly move down the screen to earn points. Your player can lose lives by being hit by the alien's missiles. Your player needs to kill all aliens before they reach you otherwise it is game over.

Deployment link

<https://maiadee.github.io/project-1-spaceinvaders/>

Getting Started/Code Installation

Explain how the reader accesses your code. Include a step by step approach.

Timeframe & Working Team (Solo/Pair/Group)

We have been given up to 6 days to work on this project individually

Technologies Used

HTML, CSS and JavaScript

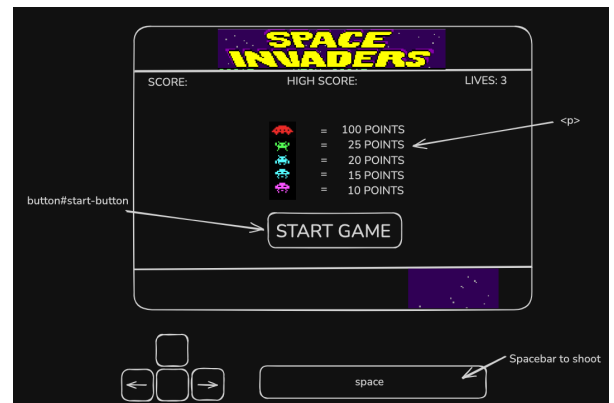
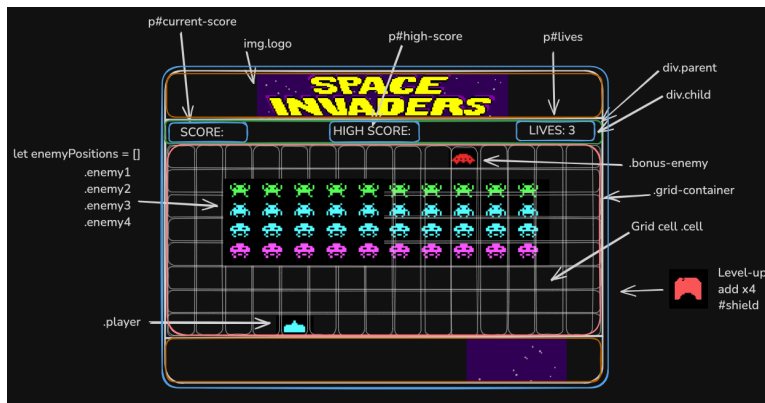
Brief

We were asked to make a game in the browser using DOM manipulation techniques.

The game must have a win/lose condition and separate HTML, CSS, JavaScript and JavaScript data files

Planning

- I created a wireframe prior to starting any pseudocode. This allowed me to break up the elements of my game into sections, looking at where my parent and child elements would be



- I used pseudocode to break down each section of my game further, looking at

Build/Code Process

Sunday

- created my HTML start page - wanted a base to work from
- Game container with 4 divs - logo/results-display/grid-container/footer
- Used flex box to position within game-container and to position elements within each individual div
- Found my assets and saved to folder
- Added start button to load page that will trigger many of my other functions

Monday

- created grid using JavaScript
- Gave each div in my grid a class of cell and pushed them into cellsArray within my function
- Created CSS property with class of alien and player with background-image of my assets so I could add and remove these from my array indices to simulate movement
- Added class of player to my players start position
- Created my block of aliens using for loops - one per row
- Pushed each indices into my alienIndices array and gave each cell class of alien
- Created a function to allow my player to move left/right using arrow keys - added a 'keydown' event listener to my document which calls this function - chose keydown so could continuously move my player left and right
- within this function I have nested an if statement stating that if space bar is pressed my player will shoot - a function I will add later on
- Began process of making my alien array move
- Broke it down to a move right function and a move left function
- Focused on getting them all to move together, once that was working I could look at stopping the aliens at either wall
- By EOD my aliens would move right but they would wrap the screen - same for moving left

Tuesday

- focused on finishing my alien movement
- Tried a few ways to do this - realised I wasn't targeting my alien array I was targeting the DOM element (div) as I had pushed my aliens into cellsArray[i] rather than into my alienIndices array
- Tried using a forEach loop instead of a for loop, this wasn't updating the index of my aliens
- Set Interval of 1 second - use some() to iterate through the whole array and check if any of the indices have hit the right side wall. If yes use - use a for loop give each index a new index (+= grid columns)

```
if (alienIndices.some((alien) => alien % gridColumns === gridColumns - 1)) {
  clearInterval(rightInterval);

  for (let i = 0; i < alienIndices.length; i++) {
    alienIndices[i] += gridColumns;
  }
}
```

- If they have not touched the wall, use another for loop to give each index a new index of +1 as shown in the code below -

```
for ([let i = 0; i < alienIndices.length; i++]) {
  const alienCurrentIndex = alienIndices[i];
  const alienNextIndex = alienCurrentIndex + 1;
  alienIndices[i] = alienNextIndex;
}
```

- Used same logic for moving left and called that function after my aliens hit right wall and moved down a row - same for the reverse
- Created CSS property with a class of .bonus-enemy
- Used a set interval to remove and add the class of bonus-enemy and update the current position each interval
- Used the same logic to stop the enemy at the left wall
- Used a set timeout to restart that movement every 8 seconds
- Made my player shoot using the space bar - called that function within my player move function
- Created CSS property with class of player-bullet to add and remove from my cellsArray indices
- Set interval of 200 seconds adding and removing the class
- With the function I'm checking if the bullet has hit an alien using classList.contains to check and classList.remove to remove the bullet and clearing the interval to stop the bullet from moving on
- Tomorrow I will need to figure out how to permanently remove the alien class from that cell once this happens
- I also need to add another if statement within that function to determine whether I have hit the bonus enemy

Wednesday

- I started Wednesday by debugging my enemy shoot and player shoot functions as there were both working but I was getting classless undefined in my console for lines in both functions after the game had been played for a few seconds
- The issue for playerShoot was that I did not have an if statement for what to do when the bullet left the grid, this is when I was receiving errors in my console.
- I needed to add an if position to check whether the bullet's next position would be less than 0, off the grid. If so, clearInterval
- The same went for the enemy shoot function. I had an if statement to check whether the enemy bullet was off the grid but it was in the wrong order in my function. I needed to check the index position before my interval begins.
- My most challenging task of the day was figuring out how to make the alien permanently disappear when they are hit by my players bullet. They currently disappear when hit and reappear when the interval repeats.
- The best logic I had for this was to remove the aliens from the array permanently when hit.
- I broke this down into first finding the index of that alien and then removing from the array using splice() as shown below -

```
const alienIndex = alienIndices.indexOf(bulletPosition);
if (alienIndex !== -1) {
  |  alienIndices.splice(alienIndex, 1);
}
```

- I needed to decrease my lives every time my player was hit by a bullet, which I nested inside my enemy shoot function
- I cached my lives span so I could update the HTML and created a variable of lives which I set to 3
- Using an if statement to check whether the bullet current index contained a class of player
- I then decrease lives using `—` and updated the innerHTML with the new number
- I repeated this process for the score, nesting it in my player shoot function and adding 20 to the score variable
- Tomorrow I need to complete my reset game function and display win/lose on the screen dependant on game results
- To start this off I added a reset button to my footer but as I only want to see this once the player wins or loses I added the class of hide-button to the reset button before the gameStart function is called, the same class I added to my start button inside my gameStart function when the button is clicked

Thursday

- This morning I wanted to tackle my final CSS tweaks. Before the start button is clicked there is no space displayed where the grid will be - I wanted to create that space on my landing page with the start button in the centre of it.
- To do this I played around with the height and width properties of my grid container in CSS.
- I then needed to centre my start button within this - I did this using position: relative and moved it 200px to the right
- I noticed once my game was played for a while I had a couple of bugs. It appeared to be when all my aliens had been killed
- I had to spend a fair amount of time on this trialing different lines of code and different orders of my code.
- My console was searching for indices outside of the cellsArray,
I realised I needed to remove a line of code and place an if statement at the top of my enemyShoot function to stop the function from running if there were no aliens
- I did the same for the moveBullet timer within this same function. To stop the game from searching for bullets that were outside of the grid in this code I am checking that the bullet has moved either beyond the last cell in my grid or before the first cell in my grid. If either are true the interval is clear and I'm ensuring no code beyond this in my function is executed

```
if (startingCellIndex >= cellsArray.length || startingCellIndex < 0) {
  clearInterval(moveBullet);
  return;
}
```

- I then focused on my reset function.
- Within this I reset my score and lives by updating the innerHTML and reset gameActive to true
- I then added an event listener to my reset game button and called the reset game function in this
- I found when I did reset my game, I had duplicates of my player, aliens and bonus enemy
- Upon reflection I realised I had to make sure I was setting my player back to the start position and clearing all of my intervals in my reset game function

- I also needed to reset my cells and alien arrays back to empty and clear by grid container - I did all these within my createGrid function so that they were all cleared before my grid was created
- I finished tweaking my CSS by making sure my game was centred within the browser page using flex box and giving the body a black background
- This meant I needed to play around with the width of my game container and grid container ensuring everything was contained within their parent elements

Challenges

- One thing I found challenging was choosing which iteration method to use and how. I knew this was a weaker spot of mine and doing this project allowed me to build on that knowledge and become more confident with iteration methods
- Using the right syntax when targeting indices in arrays, especially when it came to finding out current and next positions of these indices. It was a case of confusing the difference between targeting the aliens array rather than the whole cells array. I gave my arrays more defined names such as changing aliens to alienIndices, this made my code more readable when understanding the process I was going through
- Creating my reset function was a challenge as I had so many timers within my code, some nested within others. I now know that I need to remember every single timer needs to be cleared individually for the game to reset.

Key Learnings/Takeaways

- I feel more confident using arrays and loops and iterating through them.
- I really enjoyed the Pseudocode process and found it extremely helpful to break things down. However I feel that as the project went on there were things I had not thought about within my pseudocode that tripped me up. I think next time I would benefit from more detail within my pseudocode and breaking down the project further. I also feel a personal timeline would help me to manage my project better

Future Improvements

- With more time I would like to store my current score within a highscore variable that I can update
- I would also like to give each row of aliens a different number of points when hit by the player
- The most challenging feature I would like to add would be the 4 protective elements as these would take up 4 cells each and also breakdown each time a missile hits them

Attributes/Assets

<https://www.deviantart.com/scpthunder/art/Pixel-rocket-358437926>

<https://www.pngall.com/space-invaders-png/>

<https://www.pngkey.com/maxpic/u2w7w7a9y3y3i1e6/>

[https://vsbattles.fandom.com/wiki/Space_Invaders_\(Characters\)](https://vsbattles.fandom.com/wiki/Space_Invaders_(Characters))

<https://dosgames.com/game/space-invaders-/>