**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 employer partners 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. Keep your copy in your Outcomes folder, so the team is able to add feedback into your docs.**

Once the content has been finalised by the Outcomes team, you can then 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**

**Instructions**

*Here, give a short description of the project. It can be a couple of sentences where you discuss the point in time during the course that you completed it, the topic of the project and potentially the tech stack.*

**Insert your Description here:**

Frogger is an arcade game where the objective is to navigate through traffic without being hit by obstacles. If you are fortunate enough to reach the final destination, where the winners are, you will score 100 points for each successful attempt, until you exhaust your lives.

**Deployment link**

**Instructions**

*Here include the information on where the deployed project can be found. If login details are needed to access the full project, make sure you include them.*

*If you have not yet deployed your project, you can add this in later.*

**Insert your Deployment link here:**

<https://gibacarnieli.github.io/Frogger/>

**Getting Started/Code Installation**

**Instructions**

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

**Insert your Getting Started/Code Installation here:**

**Timeframe & Working Team (Solo/Pair/Group)**

**Instructions**

*Share the timeframe given for the project and whether you worked independently, in a pair, or in a group.*

*If you worked in a pair or group, include the names of the people you collaborated with. As a bonus, you can also provide links to their GitHub repo.*

**Insert your Timeframe & Working Team here:**

I had one week to build this grid-based game using JavaScript, HTML, and CSS. I chose to create the standard version of Frogger without altering the characters of the game. I worked on it independently, and my version of the game doesn't have levels; instead, it relies on accumulating points.

To navigate through the game, you must use the keyboard arrow buttons while crossing the street.

**Technologies Used**

**Instructions**

*List every technology you used to complete the project. This can be in one long list, or broken down into categories (Back End, Front End, Development Tools).*

**Insert your Technologies Used here:**

**HTML**

* Grid with 100 cells, 10 rows and 10 columns.
* ‘Start’ button for game start and the audio start.
* Audio Element for background music.

**CSS**

* Grid using flex-box

**JavaScript**

* keyUp event to move the characters.
* Interval to move obstacles.
* Click events to start the game.

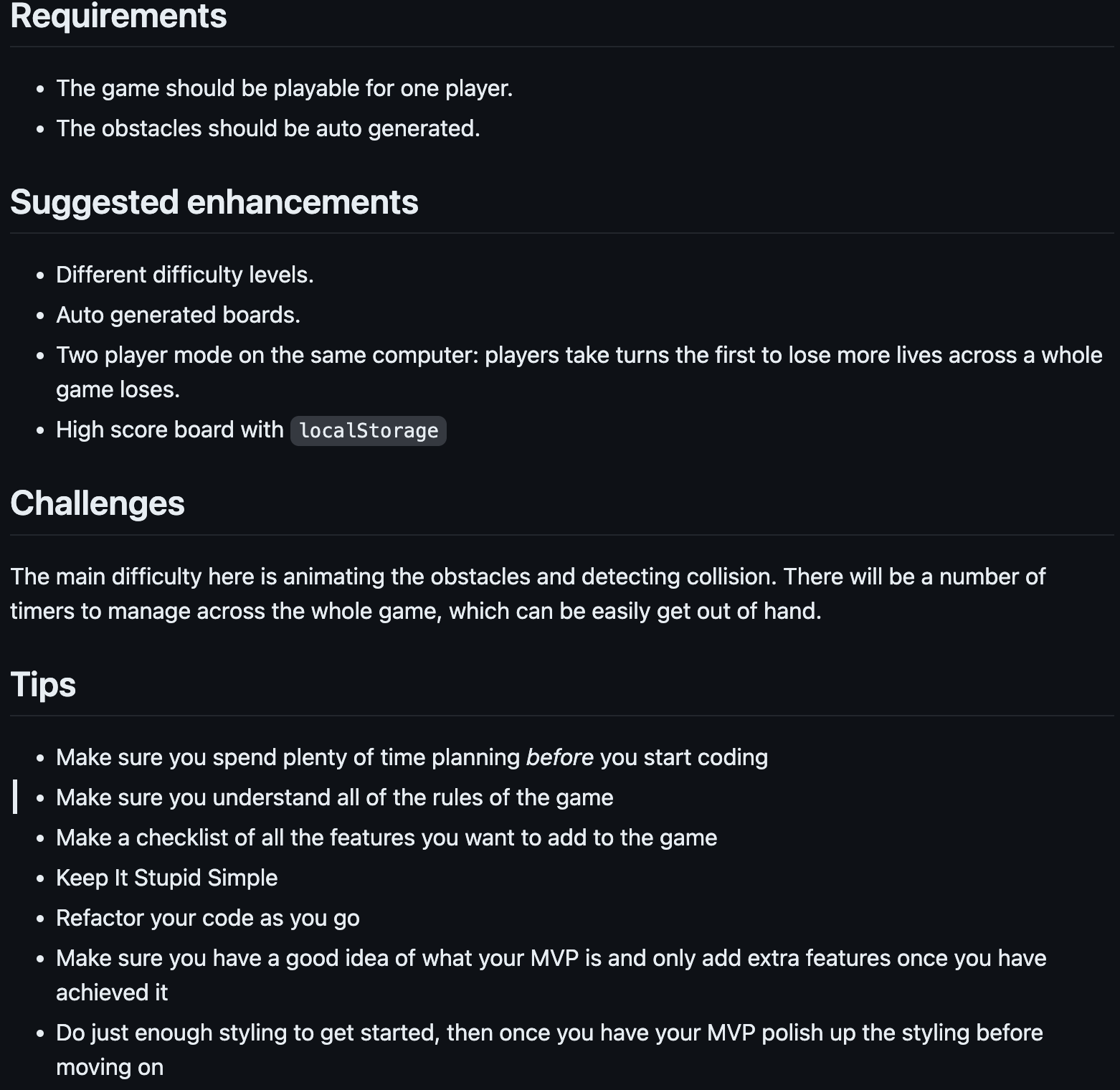
**Brief**

**Instructions**

*Include the brief set by your instructional team here. This sets the context of the project you were working towards and mimics briefs you will be set later in your future roles.*

*This can either be in bullets or in a paragraph.*

**Insert your Brief here:**



The brief from GA was to build a game, able to move around the grid and able to score points. Could it be 1 player or more. We had a list of alternatives and I chose the frogger.

The requirements to the game should be playable for one player at least, and have obstacles that will be auto generated.

The main difficulty here was to animate the obstacles and detect the collisions. I was able to do it using the tools that I had learned in Javascript to detect and also change the directions while you are playing.

**Planning**

**Instructions**

*The planning stage is important, as all projects in your future roles will have detailed plans before any coding happens. It is a great experience to share with potential engineer employers, as this reflects real engineering team practices.*

*Start by explaining the initial steps you took in the project.*

* ***Did you do any sketches****? If so, discuss this and include images.*
* ***Any wireframes of the front end and UI****? You did? Then explain this and include images.*
* ***Any ERDs****? Same here, explain and include images.*
* ***Use a project management tool to plan the sprint****? If so, talk through this - what tool did you use? How you allocated tickets/responsibilities, sprint timeline etc. Also include screenshots of this.*
* ***Any pseudocode****?*
* ***If it was a group or pair project*** *- Discuss who was designated which tasks. This is very important, as engineers want to understand who owned the different code elements when looking at a group project.*

*For each project, review the above bullets and discuss every step you took in the planning stage, including the relevant images.*

*Not every project will include the above, but it’s important to discuss any of the bullets that you did implement.*

**Insert your Planning here:**

***This was my wireframe from before starting my project, the only difference between this wireframe and the final project is that I added more obstacles****.*

*A screen shot of a computer screen

Description automatically generated*  
I also created pseudocode to get an idea of potential issues I might encounter or to ensure the smooth functioning of my project. It proved helpful when I needed to revisit my work and understand what was missing for everything to function as it should.

Pseudocode

! Elements

current-score (this will be the span tracking current score made)

progress ( i will try add a level count))

amount-life-remaining (span that tracks the amount left life)

button (start the game)

obstacle for the frogger pass

transition for the elements that need to have as obstacle

! Variables

score amount (number) - every time he passes a level, he get 100 points

lives - will have just 3 lives per game (every time something hit, he loses a life)

completion (number) - the final score from the user (alert window)

! Grid

\* Making a grid in JS

This function will create all the grid cells and append them to the existing grid

Set width and height of the div cells, grid will 6 rows for the levels

will be a grid without cell number visible, for track if he still can go up, left, down

obstacle in 3 different lines are my plan, for be more interesting

i will also add 2 lines that he can wait for the best moment to cross.

**Build/Code Process**

**Instructions**

*The Build/Code Process will be the longest section of your ReadMe and will be most insightful to the engineers that review them. This is where you will discuss the steps you took to code the project.*

*You want to see your ReadMes as a way to walk the engineers through your approach and problem solving from the start of the project through to the end.*

*You'll need to include a minimum of 3-4 code snippets, highlighting code you're particularly proud of and these code snippets will have descriptions on what you did, how and why to set the context of the snippet you include. These explanations are important for the engineers, as they will want to understand what you did and the reasoning behind the steps you took.*

*You don't need to document every single thing you coded, but walk them through the key sections of the project build.*

*For any group project, you will just focus on your contributions.*

*Some people will document the build/code process by discussing the key stages they worked on. Others will do a day by day guide. It’s entirely up to you how you structure this, as long as you discuss all the key things above.*

**Insert your Build/Code Process here:**

Day 1 – I began by contemplating how to approach my project and worked on creating my wireframe. I searched for and downloaded the sound and images I wanted for my background while also identifying potential challenges that I might encounter during the project development.

Day 2 – I initiated the development of my grid using flexbox and incorporated the frog and roads to get an idea of how my layout would look. By the end of the day, I also implemented the keyUp function to enable the movement of the frog within the grid.

A screen shot of a computer code

Description automatically generated  
  
Day 3 – I introduced the obstacles and began contemplating the movement using setInterval for speed control. Initially, I implemented a regular movement, but with a bit of assistance, I incorporated random movement for the obstacles, which significantly enhanced the overall appearance and gameplay of the game.

Day 4 – I incorporated background music, addressed a few bugs in my game, and introduced a level system. In this setup, the user has only one life and needs to complete five levels to emerge victorious in the game.

A screen shot of a computer program

Description automatically generated

Day 5 – I completed my game to prepare for presentation and corrected the collision detection, resolving an issue in detecting when the cars approached from the sides to hit my frog.

A screen shot of a computer program

Description automatically generated

**Challenges**

**Instructions**

*Challenges are great for showing your learning journey and problem solving, and this is a section that many engineers will check out. Every day of your engineering career you’ll encounter challenges, this is part of your growth and development. It’s the challenges you encounter that helps you become a stronger and more competent engineer.*

*Here you will detail any particular challenges you encountered as you were coding the project.*

*Questions to answer here:*

* *What technical challenges did you come across?*
* *Why were these challenges?*
* *What problem solving did you do to rectify them?*
* *Team dynamics/ Project management*
* *Tools/Tech you used*

**Insert your Challenges here:**

I had many challenges during my project. Till the last minute before delivering the project to my lectures, I had to deal with some issues with my code, but I didn’t stop to think that I would find a solution for my problems, and I did.

I couldn’t see where I was doing wrong, so I decided to print a few parts of my code and then I saw that I was calling the same function at different times and with different names.

I printed to be better visually as my code wasn’t long so I could do it. I searched for functions that I could use in my code, and I found it.

**Wins**

**Instructions**

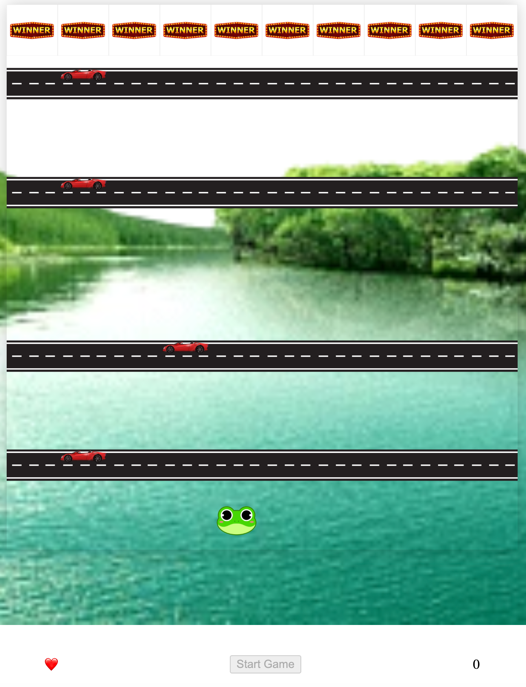
*The Wins section is your opportunity to highlight the aspects of your project you are most proud of. See this as your chance to showcase these parts of your projects to the engineers reading your ReadMes.*

*Things you could discuss here:*

* *Interesting problem solving you did*
* *Strong sections of code*
* *Collaboration with other team members*
* *Visual design of the project*

**Insert your Wins here:**

My achievement in this project was resolving all the problems I encountered in one night. I didn't believe it would be possible, not even for a second. + Tell us how you did it.



I also enjoyed the part of my code where the car should pass, start, and finish, and the ability to have a random movement instead of a linear one.

A computer screen shot of text

Description automatically generated

**Key Learnings/Takeaways**

**Instructions**

*This section is one of the other most important parts of your ReadMe from an engineers’ perspective and helps to differentiate each of you from your classmates and team members.*

*Engineers love to understand what you learn from each project and how it has shaped you as an engineer.*

*See this as your opportunity to show the engineers how your skills grew during each project sprint.*

*Things you could discuss here:*

* *What Technologies/Tools do you now feel more confident with? Tell them specifically what you learnt about these.*
* *What engineering processes did you become more comfortable with? Standups? Pair programming? Project management? Tell them what you learnt from these processes?*

**Insert your Key Learnings/Takeaways here:**

I learned how to create a loop for continuous music playback.

I learned how to use the play() function to control audio.

I gained proficiency in using Flexbox.

I learned how to implement randomness for moving the obstacles.

**Bugs**

**Instructions**

*If you have any bugs in your project, it’s important that you flag them in your ReadMe. This helps the engineers reviewing your projects to understand that you are aware that there are issues - if you don’t flag these, then they won’t have that visibility that you know these problems are in your code and it can result in them not having a full understanding of your technical knowledge.*

*In either sentences or bullets, explain what the bugs are.*

*If you have no bugs, you can leave this section blank.*

**Insert your Bugs here:**

I didn't observe any bugs but I would like to continue working on it to enhance the visual appeal of my game.

**Future Improvements**

**Instructions**

*It’s common to get to the end of your project and have ideas on what you would do if you have more time, as well as how you might improve it.*

*If you do, you should detail this here. It’s great to give that context on potential future improvements, to share your creative or technical ideas with the engineers reading your ReadMes.*

*In either sentences or bullets, explain what your future improvements would be.*

**Insert your Future Improvements here:**

I would like to add more functionalities to my game, such as pop-ups, crash sounds, and victory sounds.