**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:**

Hungry Bao! is a game based on the game Frogger. You must take the Bao to the finish line without hitting any obstacles. Should not hit the sauce dish, also not get hit by the steamer or picked from the chopstick.

**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://yingjod.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 spent 5 days and worked independently to build a grid-based game using JavaScript, HTML and CSS. In my version of Frogger, I replaced the frog to the Bao, who must navigate the course using the keyboard arrow buttons while avoiding all obstacles. The game has two different level, once hit the finish line the next round would be more difficult.

**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**

* Header with name of the game.
* Division for information bar for level and lives and score.
* Section for the Grid
* Audio element for the background music.
* Button for the game start and mute the background music.

**CSS**

* CSS to design the background and organize all the items in the page.
* Use the transform and -webkit-transform property to flip the image.

**JavaScript**

* keyUp event to move the characters.
* setInterval to move obstacles.
* clearIntervel to remove obstacles.
* Play and Mute background music.
* Click events to start the game.
* Grid with Cells including 7 rows and 7 columns in a total of 49 cells.

**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:**

**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.*

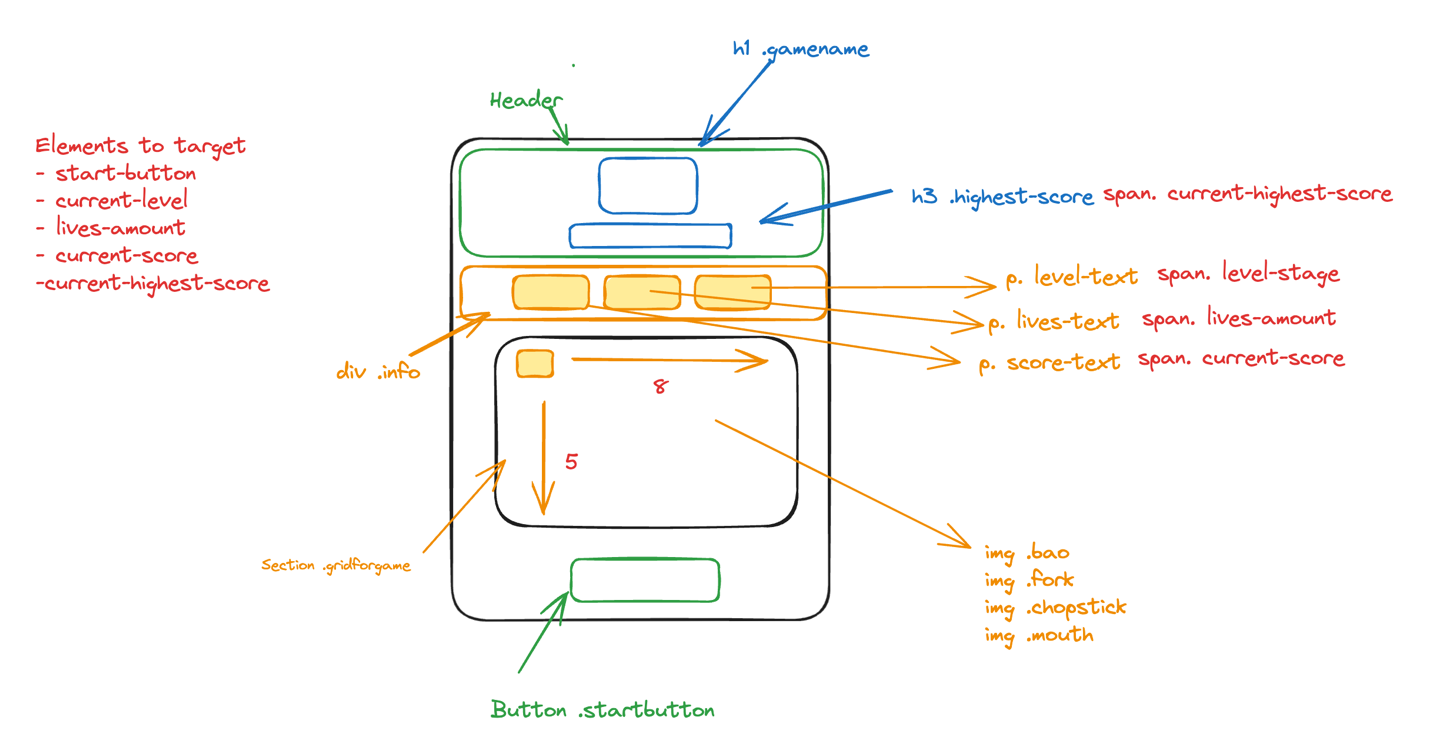
* ***Did you do any sketches****? If so, discuss this and include images.*
* ***Any wireframes of the front end and UI****? You did? Then explain this and include images.*
* ***Any ERDs****? Same here, explain and include images.*
* ***Use a project management tool to plan the sprint****? If so, talk 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:**

I drew the wireframe for the game and wrote the pseudocode before start to code.

**

**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 drew the wireframe for the game and wrote the pseudocode. build up the HTML for the most of content in the games. And write CSS to style the User Interface. Also, searched for the image and music I needed to use for the game.

Day 2 - use JS to build the grid on the page and add the Bao to the game, set keyUp function to move the Bao around. Also set up the start button to achieve the game.

Day 3 - I added the obstacles and made them move at different speeds. From right to left and left to right, created the and added the other non-moving obstacles sauce dish. And set up the score counting.

Day 4 – set up lives counting today, which was a bit miserable, different collision should use different function. Set up 3 types of function for any possibility for collision. Added pop-up page for end of the game. Added background music to the game.

Day 5 – set up the second level of the game and improve the User Interface. Organized the code nicely.

**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:**

* Make the lives decrease when the collision happens, not matter is the Bao hit the enemy or Bao be hit by enemies, must set up different function for different situation.
* Disabling the other two surfers while the first one was active.
* Getting the right sound effect to stop playing after 4 seconds when winning or game over.
* Making the obstacles move at a higher speed after each round.

**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:**

* First time to work a project from 0 to 1, that is a good experience putting what you learn into practical use.
* Ask for help is important when stuck on problem.

**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:**

* Learned how to use setInterval () and clearIntervel().
* Learned how to use CSS to flip the image.
* Learned how to create a grib by using HTML/CSS/JS
* Learned how to use play() and mute() functions to control the audio.
* Learned how to use function correctly to the exact part to make the code tidy and clean. DRY the code.

**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:**

* When the lives turn to zero, the alert will not pop-up immediately.

**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:**

* Set up the win status for the game.
* Add more different music for different events happen.
* Set up a pop-up for construction the rule of the game when the page loaded.