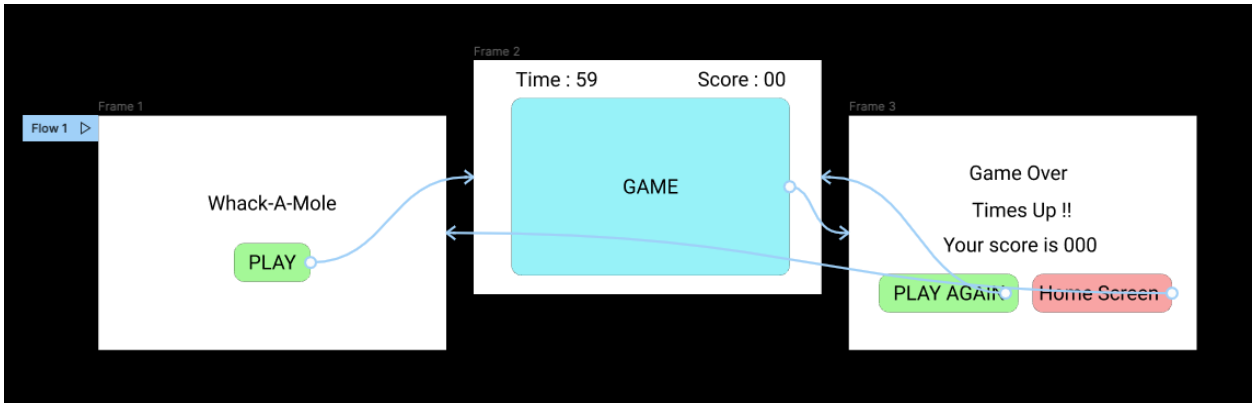
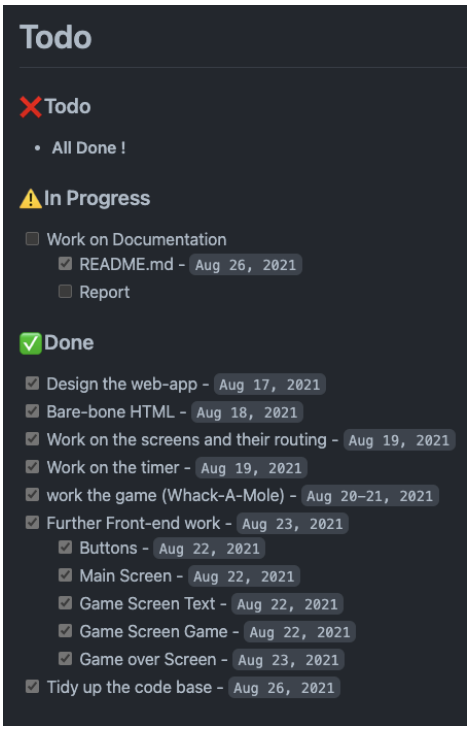
REPORT

Introduction

I chose to build a web-app for the game, the tools I used to build it are HTML, CSS and JavaScript (the basic web development tools). I designed a prototype in figma first and then proceeded on building the app. Information regarding the Design can be found here



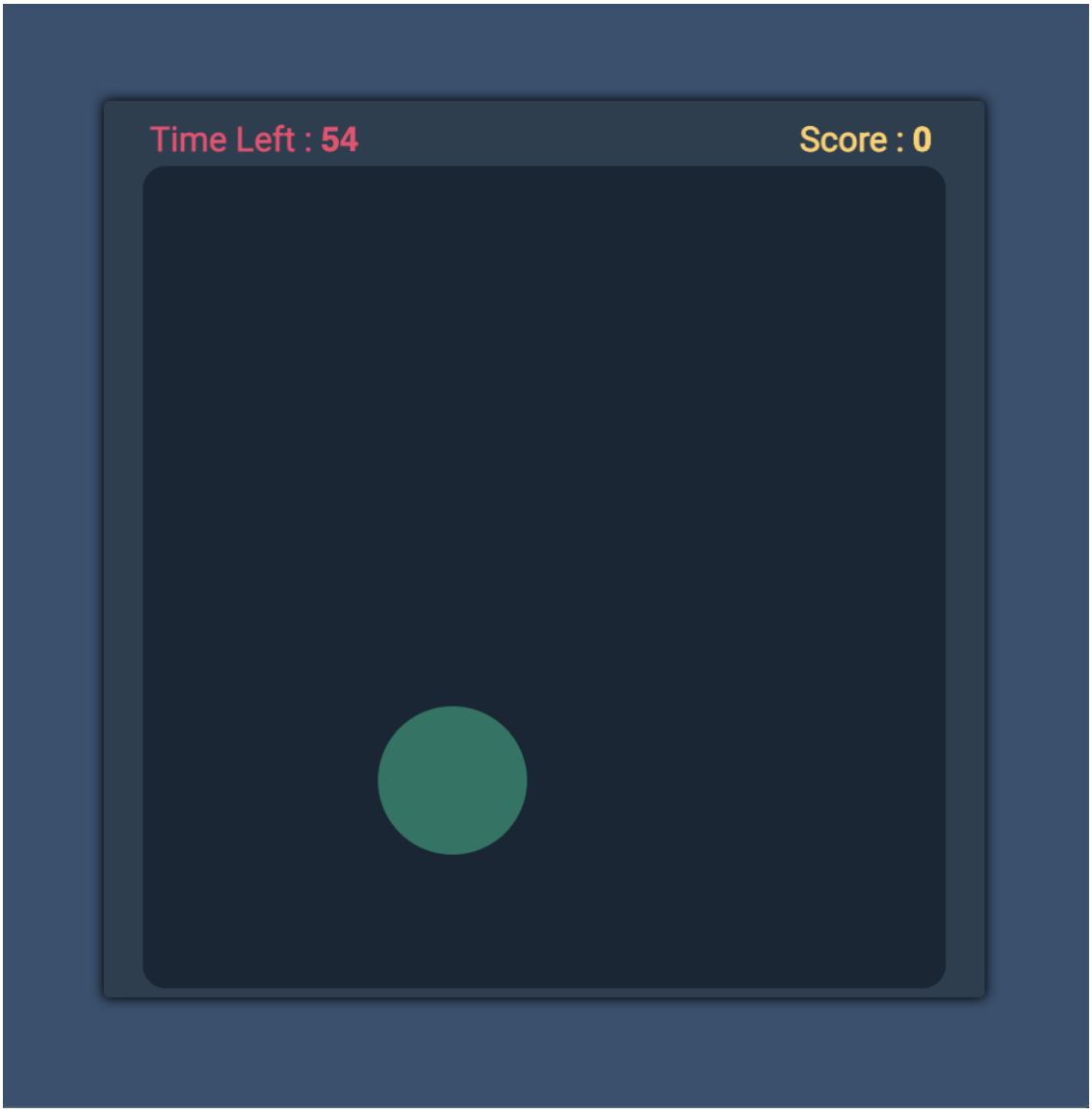
I also kept track of my progress using a todo list, which can be found here. I broke the complex task of building the whole app into smaller do-able tasks and continued to work on one after the other. The detailed working of the system to be reported is provided below.



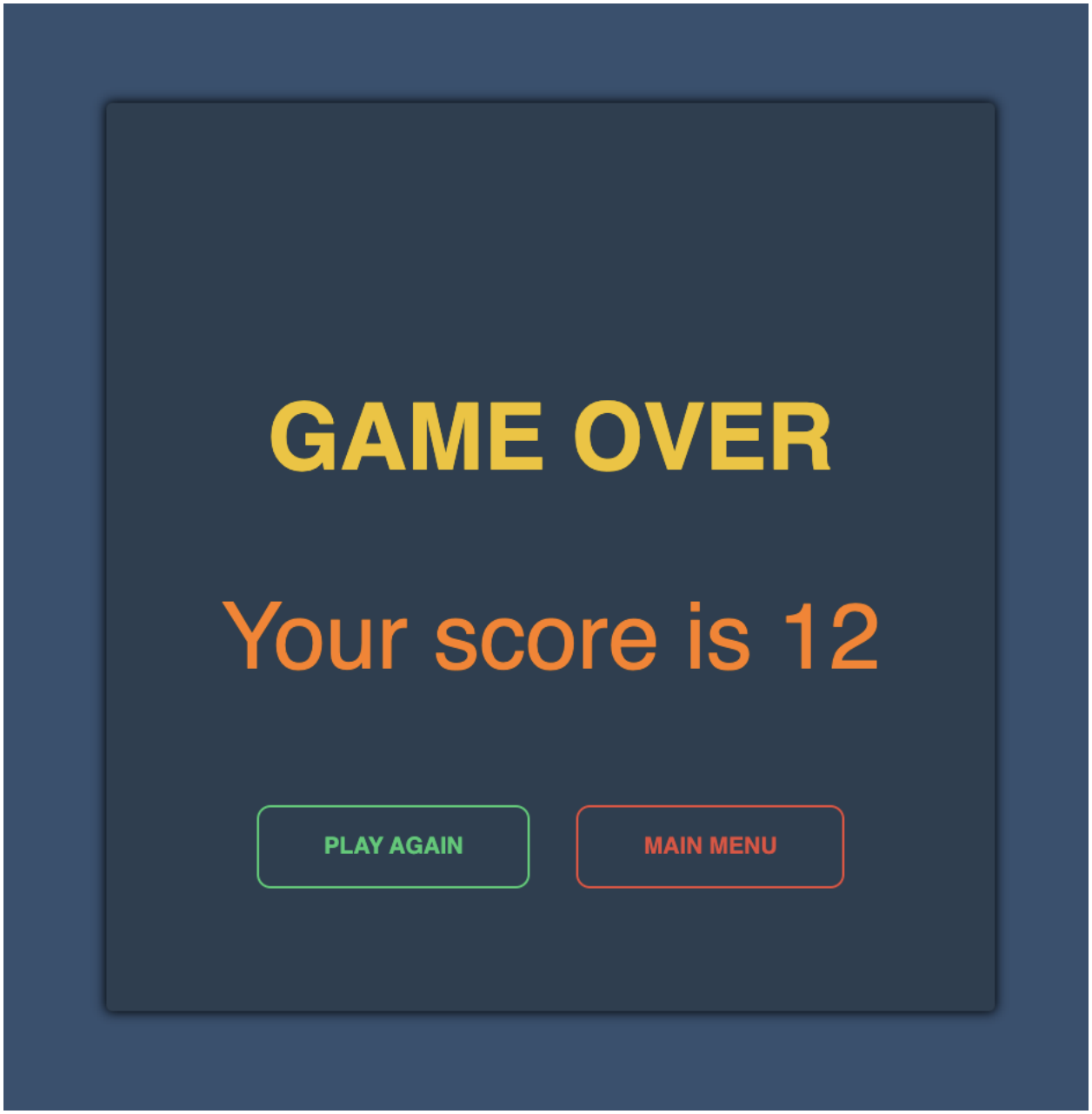
The web-app is made up of three screens

a. **Main Screen -** contains some text and the play button

b. **Game Screen** - contains Timer, Score and Game



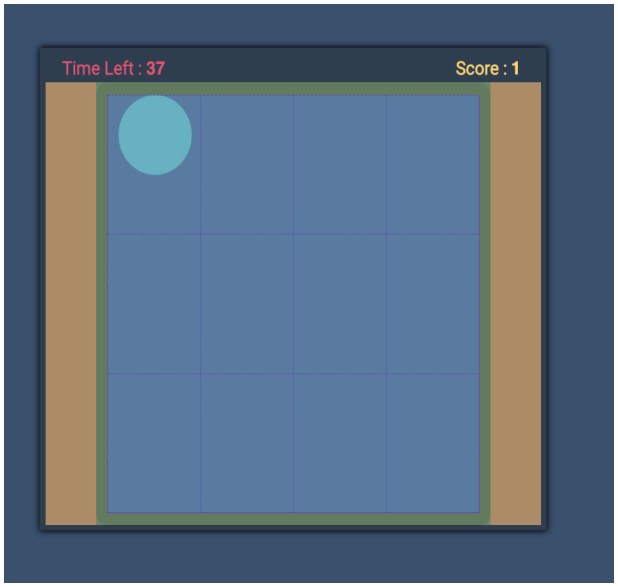
c. **End Screen** - contains the score and two buttons one to play again and one to return to the main screen.



Detailed working of the system

The System can be divided into two parts: the game and the screen routing.

a. **Game -** The game runs till the time left becomes zero, It uses a timer built with javascript functions like setTimeout() and setInterval (the related code can be found in script.js) and score increases every time a circle is clicked using eventListeners and an increment function. Now the game is made up of a 4 x 3 grid, each section of the grid represents a position where the circle could spawn. The circles are made visible by adding a class called “up” and vanish when the class is removed. I use a function called spawnCircle() that picks a random position and a random time within an interval (1 second) using other helper functions and then adds the class ‘up’ to make them visible. I also use another function to make sure that the random circle to be spawned is not the same as the previous position. This happens till the time left becomes zero.



b. **Screen Routing -** the screen routing is quite simple, the screens are div components, whenever a button that is supposed to render another screen is pressed I use a function that adds ‘hide’ class to the current div and removes ‘hide’ from the to be rendered screen’s classlist.

The code is documented with comments in the script.js file, please consider to better understand the working of the system.