

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from this bar, containing the date.

5/29/2022

# Tiktok Youth Camp

Project A – Hangman Game

Several thin, curved lines in dark blue and light grey originate from the bottom left and sweep upwards and to the right.

Htet Htet June Han

SINGAPORE INSTITUTE OF TECHNOLOGY  
COMPUTING SCIENCE

1901838

[1901838@SIT.SINGAPORETECH.EDU.SG](mailto:1901838@SIT.SINGAPORETECH.EDU.SG)

[JULIALIU08@GMAIL.COM](mailto:JULIALIU08@GMAIL.COM)

## Contents

Introduction .....	2
Planning Documentation.....	2
Project Preparation (24 <sup>th</sup> May - 26 <sup>th</sup> May) .....	2
Project Production (26 <sup>th</sup> May - 28 <sup>th</sup> May) .....	2
Project Build, Project Deployment and Project Report (28 <sup>th</sup> May - 29 <sup>th</sup> May) .....	2
Technical Documentation .....	3
Project Documentation.....	6
Computer Demonstration (Screenshot) .....	6
Mobile Demonstration (Screenshot).....	7
Tablet Demonstration (Screenshot) .....	7
Project Retrospective .....	8
What went well in the project? .....	8
What could be improved? .....	8
What will we commit to improve in the next project? .....	8

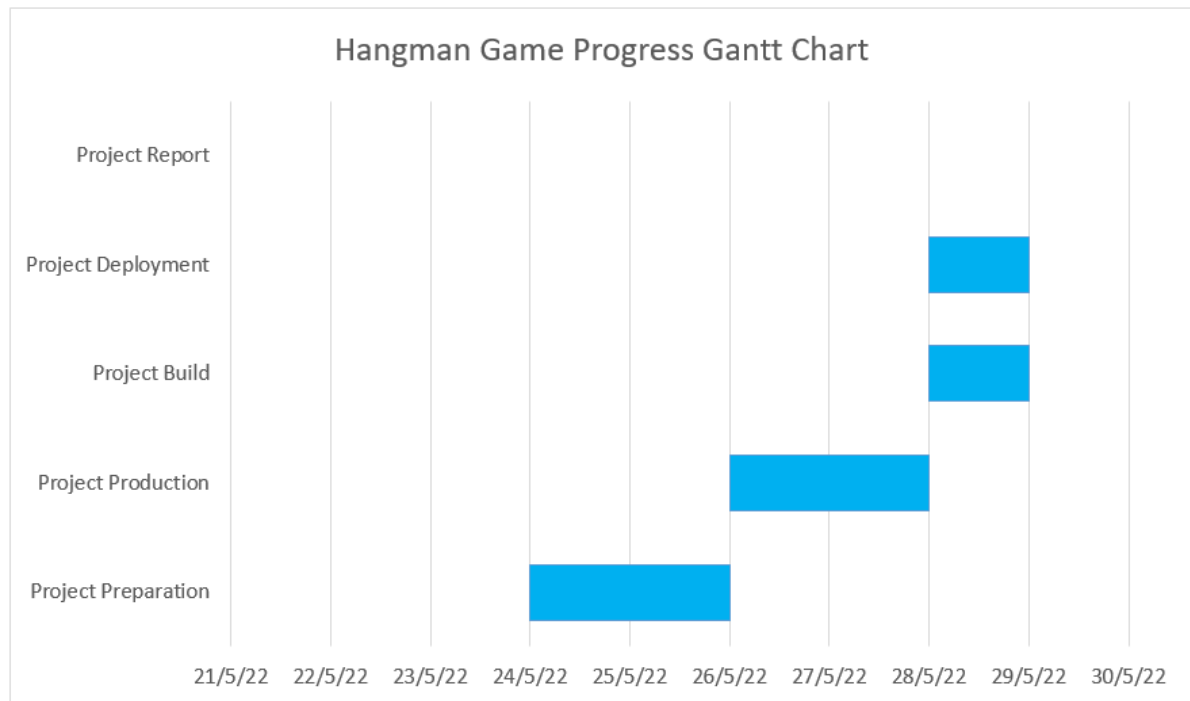
## Introduction

I have chosen Project A, to build a playable Hangman Game that is supported on mobile, tablet and desktop viewport.

React was chosen to build this project, as it is an efficient framework to build the progressive web app compatible with various devices.

Fruits was chosen as the game category for guessing in this Hangman Game.

## Planning Documentation



### Project Preparation (24<sup>th</sup> May - 26<sup>th</sup> May)

As it is my first time working with react, I explored react tools for creation of the project.

### Project Production (26<sup>th</sup> May - 28<sup>th</sup> May)

This is the period I started building the Hangman game on my local machine.

### Project Build, Project Deployment and Project Report (28<sup>th</sup> May - 29<sup>th</sup> May)

As I require demonstration videos for its ability as a progressive web application, I built the project and deployed the project to the GitHub pages.

The original code has been uploaded to GitHub as well.

Project report is written on 29<sup>th</sup> May.

## Technical Documentation

The project was built using React, as it is simple and efficient to use for building progressive web applications, apart from React being one of the technologies focused during the Youth Camp. As it would also be my first time using React, I thought it would also be a learning opportunity.

Moreover, as the game needs to be able to work on computer, mobile and tablet, using React functions with hooks to maintain the state would be more efficient to build the game with during the limited period of time.

Below is the main code snippet used for the Hangman game.

```
import React, { useState, useEffect } from 'react'
import { wordGenerator } from './Wordlist'

import try1 from './images/try1.png'
import try2 from './images/try2.png'
import try3 from './images/try3.png'
import try4 from './images/try4.png'
import try5 from './images/try5.png'
import try6 from './images/try6.png'
import try7 from './images/try7.png'
import try8 from './images/try8.png'

const Hangman = ({title, totalTries, stickmanImages}) => {
  //Number of tries by user
  const [tryNum, setTryNum] = useState(0)
  //Guessed letter by user
  const [guess, setGuess] = useState([])
  //User Guessing Progress to render for page with ' _ '
  const [userGuessProgress, setUserGuessProgress] = useState([])
  //Answer for the Game
  const [answer, setAnswer] = useState(() => wordGenerator())

  useEffect( () => {
    const guessedWord = () => {
      setUserGuessProgress(answer.split("").map(letter =>
(guess.includes(letter)? " "+letter+" ": " _ ")))
    }

    guessedWord()
  },[answer, guess])
}
```

```
const guessHandler = (e) => {  
  let letter = e.target.value;  
  setGuess([...guess, letter]) //Add the input letter to the state list  
  answer.includes(letter)? setTryNum(tryNum) : setTryNum(tryNum + 1)  
}
```

```
const generateButtons = () =>{  
  return "abcdefghijklmnopqrstuvwxyz".split("").map(letter => (  
    <button  
      key = {letter}  
      value = {letter}  
      className='btn btn-dark btn-primary m-1'  
      type='button'  
      onClick={e => guessHandler(e)}  
      disabled={guess.includes(letter)}  
    >  
      {letter}  
    </button>  
  ))  
}
```

```
const resetButton = (e) => {  
  setTryNum(0)  
  setGuess([])  
  setUserGuessProgress([])  
  setAnswer(() => wordGenerator())  
}
```

```
const gameOver = tryNum >= totalTries-1;  
let gameButtons = generateButtons();  
let winner = userGuessProgress.includes("_ ")? false : true
```

```
if (winner) {  
  gameButtons = "You Win!"  
}
```

```
if (gameOver){  
  gameButtons = "You Lost!"  
  setTryNum(totalTries)  
}
```

```
return (
```

```

<div className='container'>
  <h1>{title}</h1>

  <div className='text-center'>
    <img src = {stickmanImages[tryNum]} className= 'image' alt = 'Hanged
Stickman' />
  </div>

  <div className="text-center">
    <h3>Guess the name of the fruit:</h3>
    <h4>
      {gameOver ? answer: userGuessProgress}
    </h4>
    <>
      {gameButtons}
    </>
    <br />
    <button type = "button" className="btn btn-primary m-4" onClick={{e} =>
resetButton(e)}>
      Reset
    </button>
  </div>

  <h4 className="float-bottom text-center">Wrong Guesses: {tryNum} of
{totalTries}</h4>
</div>
)
}

Hangman.defaultProps = {
  title: 'Welcome to June\'s Hangman Game',
  totalTries : 8,
  stickmanImages : [try1, try2, try3, try4, try5, try6, try7, try8]
}

export default Hangman

```

I built the game using functions with hooks, and as the game needs to be compatible with various devices, i.e. Computer, mobile and tablet. UseState hook aided in maintaining the state or the variables during the rendering of the game and the useEffect hook helped in attempting to utilise the latest values of the variables without loading the game again to render the latest version of the variables for the game due to user response.

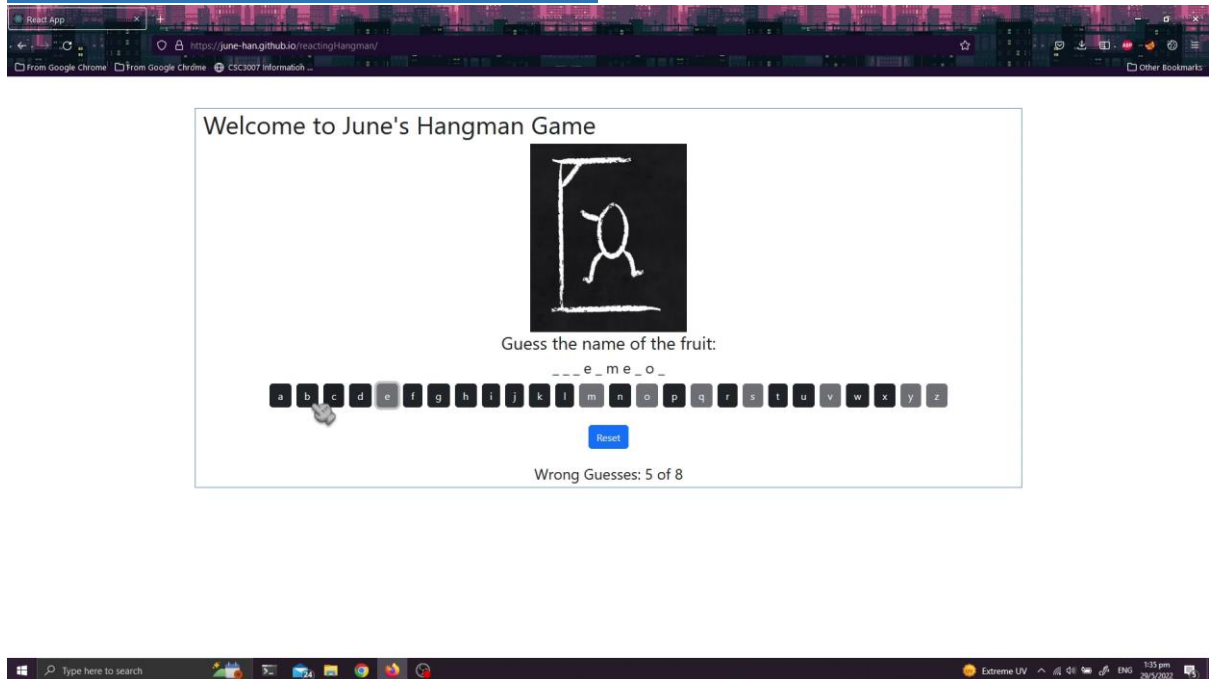
# Project Documentation

Project demonstration on different devices has been hosted here on Youtube:

<https://youtu.be/SKevVOWiZIU>

The video showcase the gameplay on computer and tablet, including how the game is supposed to be when the user wins or loses. The video will be attached with the submission as well.

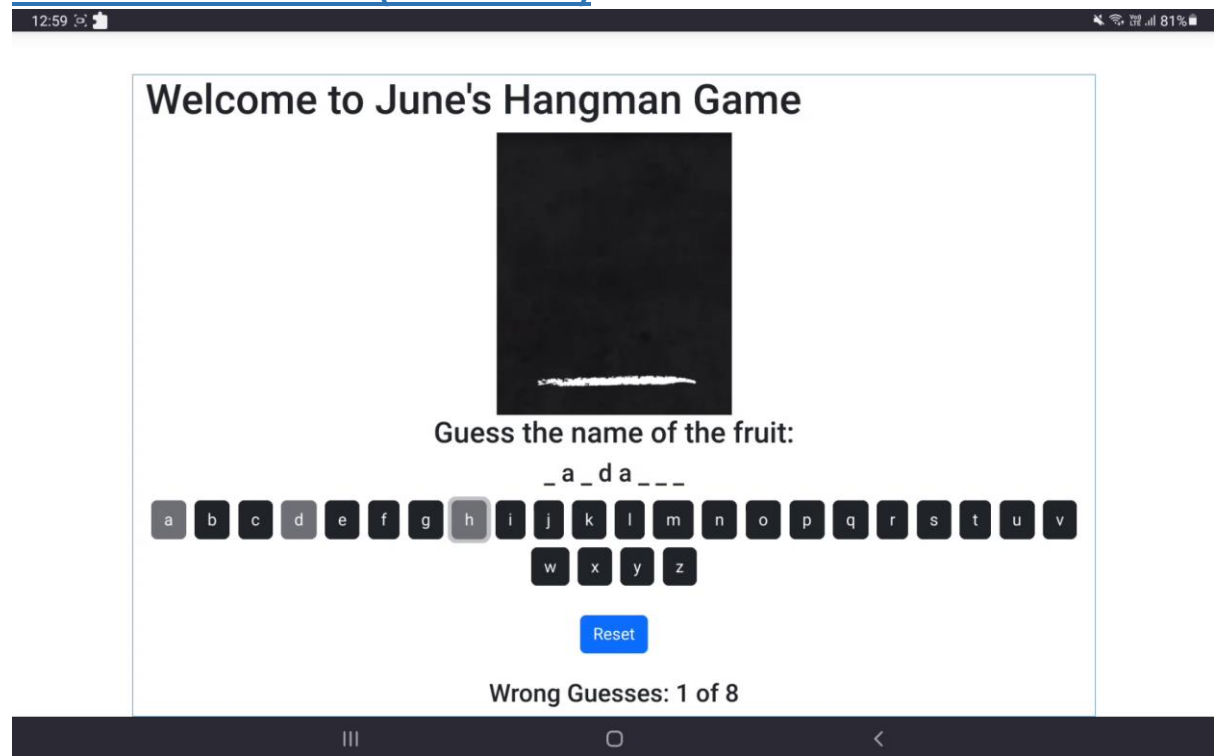
## Computer Demonstration (Screenshot)



## Mobile Demonstration (Screenshot)



## Tablet Demonstration (Screenshot)





## Project Retrospective

### What went well in the project?

As I had to complete the project by myself, I had chosen Project A, to be able to complete the project on time during the short duration as I juggle together with my school work. There are a lot of rough edges to improve for the project, but I am glad that the project went smoothly from production, built, deployment to demonstration.

### What could be improved?

There are quite a few things to improve for the project, such as the way the words are stored and retrieved with a function. I could implement as part of the state for a component or have a json database, then it would be easier to have a separate word database for the game without interfering with the main code.

Hangman.js could be further separated into different components, as it is currently the component with the longest code.

### What will we commit to improve in the next project?

New additions would be the bonus points stated in the project outline, which I could progress on when I come back to this project, and also to implement a scoreboard with username based.

I could maybe play around with more animations or attempt to build more complex games for next project.