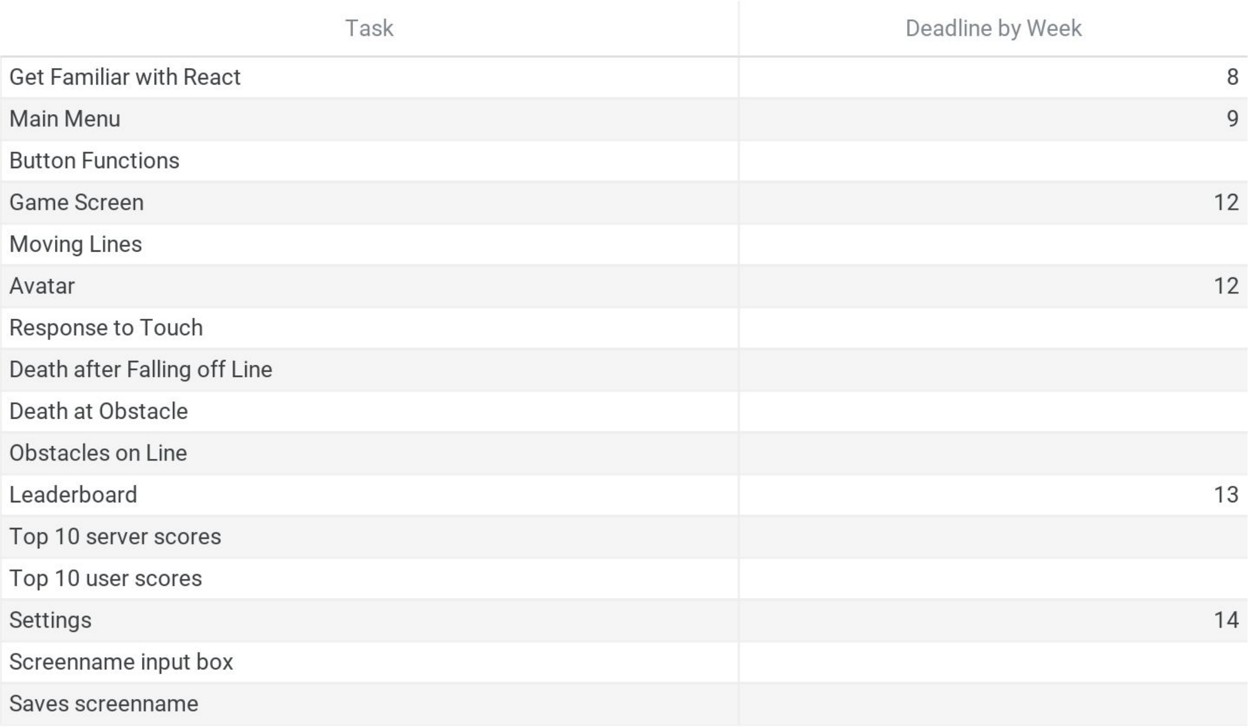
4.0 Software Development Plan

4.1 Plan Introduction

This plan outlines the development of LineGame, an application-based game developed for iOS where users are tasked with maneuvering from line to line to avoid obstacles and increase their score. The goal is to create an easy to learn but hard to master game that rivals Flappy Bird or Temple Run. Development will be based in React with a majority of the development time going to creating, implementing, and refining the game and its auxiliary screens. We also plan to implement a highscore database further into the development of the game. The completion checkpoints for our sub-tasks are as follows:



Task	Deadline by Week
Get Familiar with React	8
Main Menu	9
Button Functions	
Game Screen	12
Moving Lines	
Avatar	12
Response to Touch	
Death after Falling off Line	
Death at Obstacle	
Obstacles on Line	
Leaderboard	13
Top 10 server scores	
Top 10 user scores	
Settings	14
Screenname input box	
Saves screenname	

4.1.1 Project Deliverables

Project Proposal Presentation and Document [02]:

description: introduces project purpose and target audience

justification: identifies which computer science skills will be applied

Requirements Specification [05]:

component breakdown: outlines CSCI's CSCs and their CSUs and modules

functional requirement: details component's necessary actions

performance requirement: standards for actions

project environment requirement: lists necessary hardware, software, resources

Initial Development Schedule [05]:

rough draft of task schedule + GANTT chart

Software Development Plan [07]:

deliverables: describes what is to be given to customer by deadline

resources: hardware and software elements needed to complete project in all aspects of development and execution

project organization: describes group dynamic and responsibilities

schedule: GANTT chart

task/resource: assigns tasks to person and elements

Software Design Description Document (Architecture) [11]: outline of software that specifies interface and connects components

Software Design Description Document (Complete) [12]: detailed description of software that breaks each component's interface into modules, functions, classes, etc

Test and Integration Plan [14]: unit tests that verify each and every action for every component

User's Manual Final Updates [16]: details on how to install, uninstall, start and use every feature of software

4.2 Project Resources kevin

4.2.1 Hardware Resources kevin

4.2.2 Software Resources kevin

4.3 Project Organization

As a two-person group, we will each be responsible for our assigned tasks as assigned in the task/resource table. This includes designing, implementing, and thoroughly testing each component of said task. Once each larger subsection is complete, we will individually work on the aesthetic components.

4.4 Project Schedule

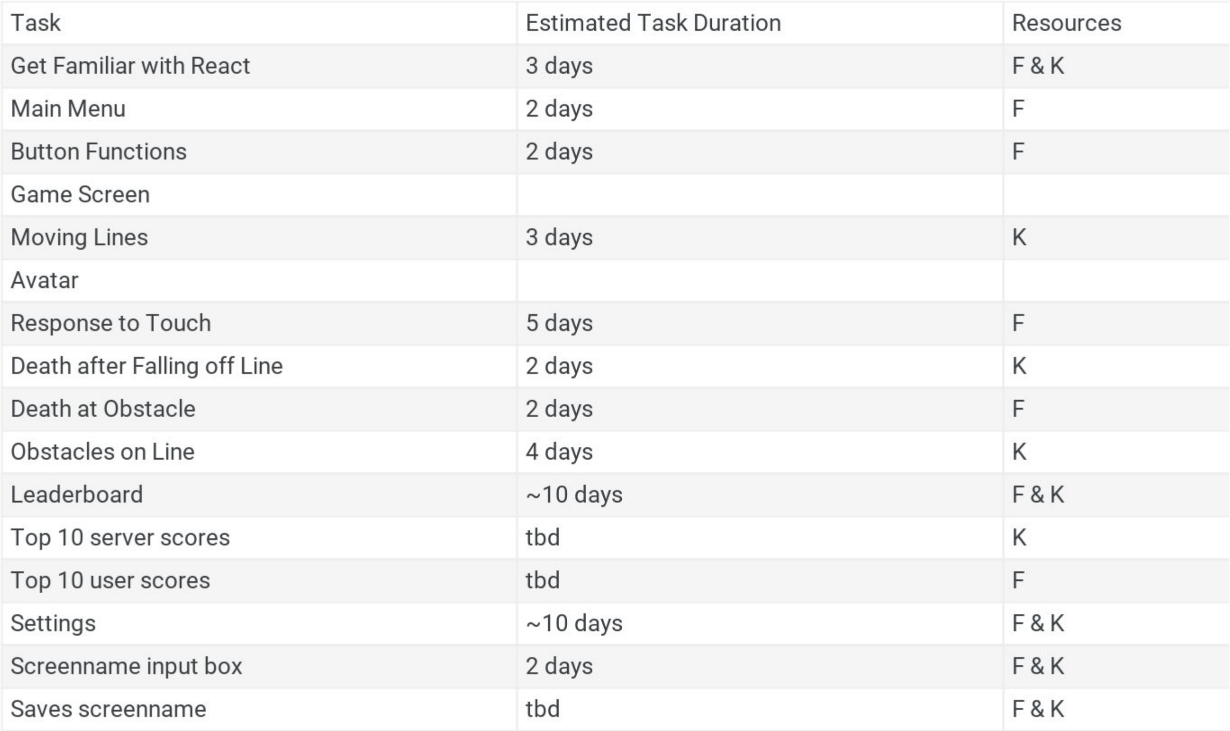
The flow of development will be broken down by visible attributes and their functions in order of access. We will develop individual screens first, then connect them by their planned cues (buttons and game events).

4.4.1 PERT / GANTT Chart

[in github repo]

4.4.2 Task / Resource Table

Each task will be done in React on our own computers and iOS devices, which requires no hardware/software resource sharing.



Task	Estimated Task Duration	Resources
Get Familiar with React	3 days	F & K
Main Menu	2 days	F
Button Functions	2 days	F
Game Screen		
Moving Lines	3 days	K
Avatar		
Response to Touch	5 days	F
Death after Falling off Line	2 days	K
Death at Obstacle	2 days	F
Obstacles on Line	4 days	K
Leaderboard	~10 days	F & K
Top 10 server scores	tbd	K
Top 10 user scores	tbd	F
Settings	~10 days	F & K
Screenname input box	2 days	F & K
Saves screenname	tbd	F & K