

Fall 2019: Senior Capstone Technology Review

Isak Foshay: Wednesday 27th May, 2020

CONTENTS

I	Group Info	2
II	Objective	2
III	Goal	2
IV	Options	2
IV-A	Controls	2
IV-A1	Description	2
IV-A2	HammerJS	2
IV-A3	Dragula	2
IV-A4	JQuery Touch Punch	2
IV-B	Level/Lesson Design	2
IV-B1	Description	2
IV-B2	Zoom	3
IV-B3	Microsoft Whiteboard	3
IV-B4	Openboard	3
IV-C	Movidius USB Stick Implementation	3
IV-C1	Description	3
IV-C2	TensorFlow	3
IV-C3	PyTorch	3
IV-C4	Spark	3
V	Selections	4
V-A	Controls Selection	4
V-B	Level/Lesson Design Selection	4
V-C	Movidius USB Stick Implementation Selection	4

Abstract

A review on what technologies may be applicable to parts of the "Programming for Kids" project that I will be working on. The project will serve user's phones computer science lessons over WIFI via an Intel NUC. As a team we will be collaborating on each levels design and implementation. The culmination of the lessons will be the curriculum for teaching machine learning.

I. GROUP INFO

There are no predefined rolls in our project group as our client prefers a collaborative approach. We are group 67 and our project name is "Get Good, Kids!". So far my role in the project has been to facilitate communication between our team and the client as well as to schedule meetings between our client and the team. The team will work on the lessons in a collaborative way in order to maximize the potential for creative solutions to the challenge of conveying complicated concepts and topics towards a younger inexperienced audience.

II. OBJECTIVE

The objective of the project is to elevate highschooler's computer science understanding in a way that does not explicitly require a teacher's guidance. This will be done through a web based application on the phone framed in the context of a game in order to promote interest from highschool age students.

III. GOAL

The project's goal is to create a web based application in order to concretely and elegantly teach kids the computer science tools and techniques in order for the student's to get future careers in computer science. A sub goal is to be able to teach teachers as well via the same app framework with potential supplemental resources in order to better prepare teachers to be a potential mentor for students to turn to with questions.

IV. OPTIONS

A. Controls

1) *Description:* The Controls aspect of the project will be one of the most integral pieces we need to get right in our project. The Controls will be relevant, but not limited, to moving a character, selecting menu items, dragging components, and navigating menus. If the controls are not optimal it doesn't matter how quality the levels are. People will not want to use the content because the user experience is so poor.

2) *HammerJS:* HammerJS is a JavaScript library which allows for easier implementation of the addition of touch and swipe handling within a browser based environment. HammerJS is one of the most recommended online that I had found and boasts a minimal file size of only 7.34KiB.

3) *Dragula:* Dragula is another JavaScript library that aids in swipe handling in a browser environment. A feature that Dragula boasts that HammerJS lacks is it's shadow drop feature. This shows a user where the object will be placed when the user drops the object. Dragula also supports the JavaScript REACT framework which we will be using for the user interface.

4) *JQuery Touch Punch:* JQuery Touch Punch is an addon for the JQuery framework. The module is a super light 584 bytes. Touch Punch enables the use of dragging elements with the JQuery framework. Touch Punch also contains the feature of elements knowing when an object has been dropped on top of it.

B. Level/Lesson Design

1) *Description:* For the lessons and levels being created we, as a team, will need to sit down and come up with lessons/levels in order to teach the user computer science topics. This will be done on a level by level basis and will

change with various factors such as different team members abilities in the specific area to who has more creativity that session. Our client has made it clear that he does not want a traditional team in which each person has his or her own job. Instead, our client has made it clear that he much prefers a collaborative approach in design and implementation of the project. This leads to the situation in which there will need to be lots of meetings for everyone to be able to stay on the same page regarding level/lesson development.

2) *Zoom*: One way to potentially boost our productivity in the meetings is to record them via Zoom. This would allow team members who aren't present to view what was said and also allow for going back and reviewing potentially skipped over / missed ideas. Zoom allows for meeting recordings as well as meetings in general in addition to it's transcription feature for ease of documentation.

3) *Microsoft Whiteboard*: Microsoft Whiteboard is another technology we can potentially use for jotting down potentials ideas and such. This would allow us to also take screenshots of ideas for easy readability later. Microsoft Whiteboard also allows for remote collaboration of multiple teammates. This would allow for remote only meetings that could be more convenient in the winter months when meeting in person could be a challenge.

4) *Openboard*: Openboard is very similar to Microsoft whiteboard but is open source so may be an alternative solution in case MS whiteboard is unavailable or doesn't fit our needs. Openboard also works on linux when MS Whiteboard does not. Openboard is a lightweight program that can be used on a bare bones setup that would be convenient to bring to meetings.

C. Movidius USB Stick Implementation

1) *Description*: A Movidius USB Stick is a Neural Compute Stick made by Intel. It is traditionally plugged in to a laptop or pc and is used by machine learning programs to boost the speed of traditionally taxing calculations for the average processor which the compute stick does faster with more energy efficiency. The client would like us to incorporate this USB stick into an app for a phone and it is currently unclear as to how he thinks that this can be done / envisions it being done, as there is no USB port on a phone. In order to teach machine learning the use of python will be all but mandatory. This leaves us a couple of options to use alongside of python in order to teach the machine learning concepts.

2) *TensorFlow*: One way to implement this piece of hardware into our project would be to use TensorFlow which has support for the compute stick. This would be done by having the stick plugged in to a small 4 inch computer that a user would then connect to via the phone app in order to interact with the stick. TensorFlow also boasts a wide array of data visualization components in order to more easily visualize what our data sets are looking like at any given time.

3) *PyTorch*: A second option would be PyTorch. This is another python based machine learning resource that could be used in TensorFlow's place. Pytorch has a more robust set of prewritten optimization algorithms. However, Pytorch lacks in it's visualization components as opposed to TensorFlow.

4) *Spark*: Spark is a machine learning framework not based on python. It works with SQL databases which caters more towards an audience focused on statistical computation. Spark is usable in python but is also able to be used in the statistical software R. For students who are interested in statistical computation learning how to use Spark would be beneficial in it's flexibility towards other programs.

V. SELECTIONS

A. Controls Selection

For the control related to moving different programming blocks we will be using Dragula. This is for its user friendliness and ability to easily copy programming blocks to different elements on the screen. For different potential game actions we will be using HammerJS. This is for its custom gesture recognition as well as its ability to minimize accidental up scrolls on the page that could possibly refresh the page which would cause troubles inside the application. JQuery Touch Punch is light weight. But, the fact that we are using REACT means that it wouldn't truly be 584 bytes and would need the entire JQuery framework installed along with it. The functions and additions that Touch Punch are not robust enough for what the project requires.

B. Level/Lesson Design Selection

We will definitely be using Zoom in order to record meetings, project meetings to remote parties, as well as documenting our different ideas. If Zoom is unable to properly capture a whiteboard during meetings we will use Openboard as well in order to more accurately portray what we are writing during meetings. When doing remote meetings we will be using Microsoft Whiteboard for its collaboration features in order to all work on the document at the same time. The remote collaborative editing feature will also allow for more consistent meet times for the design of different lessons.

C. Movidius USB Stick Implementation Selection

TensorFlow is the framework with more user control-ability with its lower level functions. This may provide for more user control in an experienced user, this doesn't pair well with the project's target audience of highschoolers who have little to no programming experience. That is why we will be using pyTorch for its more accessible library functions. Spark doesn't have quite enough user friendly library elements in order to justify its compatibility with multiple programs.