Yes We Code!

**Author Name:** Alana Allbritton

**Professor:** Christopher Kreider

**Executive Summary:** This project helps build STEM foundations by making code education fun. *Yes We Code* is an app available for Android devices that introduces students and the young at heart to the technical world of programming. With an animated character as a guide, the user will be taken through a module for coding basics containing visuals for stronger retention. Afterwards the user will have access to two different games where they get to put there coding skills to the test!

# Introduction

As technology continues to advance, so should the mechanics needed to teach students STEM-related skills. One way to solve this problem is to begin teaching skills like programming at a younger age. Therefore, I would like to create an application that includes a coding basics module and coding games that reinforce what the user has been taught.

## Project Background

Youth STEM education should be affordable, accessible, and entertaining. While some companies and organizations do provide programming education, it does not always cover these three traits. Even in my own experience, I was never exposed to coding until college and therefore had a very difficult learning curve. Now that I do have experience in Computer Science and Youth Development, it would be appropriate to create a project connecting youth with coding opportunities.

## Project Description

The end product of this project should be a working application that anyone grade 6 and older could test and understand its purpose: to teach coding through animated visuals and games. The concept is limitless, but due to the amount of time given, I am confident I could produce at least one interactive module, and two games in relation.

# Proposed Solution

## High Level Plan

I currently plan on writing most of my code through the IDEs *Eclipse* and *Android Studio*. *Eclipse* will be used when writing out my basic code and tests while *Android Studio* will help combine my Java and XML code together to create a running mobile application. I will also need a source for a majority of icons, images, and characters as a base for my animations and user interface design. As for my diagrams and mockups, I will hand-draw a couple for drafts, and then use technical tools such as *Balsamiq* and *Draw.io* to make official presentations. Progress will be logged in an *Excel* spreadsheet.

# Project Schedule

## Work Breakdown Structure

I have divided my project into six major steps. The first key step is creating the diagrams and mockups. The mockups will show the basic button and areas of input for the user for each page. The diagrams are to present the interaction between files within *Android Studio*. The next step is creating the basic code for text within the module, gaming mechanics, and simple animations. If everything goes according to plan, I will continue in creating more advanced animations (simple animations combined, adding sound/video). The hardest step is integration: adding the module, games and other navigation pages together to complete the application. The last step is testing and editing all in preparation for the final presentation.

## Project Calendar

#### Intermediate Milestone #1 (10/24/19)

* Presentation of Mockups and Diagrams: By the time of the presentation, I should be ready to focus mainly on coding.

#### Intermediate Milestone #2 (11/14/19)

* Integration Completed: The module, games, and other interface pages should all be accessible in mobile application form.

#### Weekly Update #1 (9/24/19)

* Research and Practice Advanced Android Development Coding Techniques (5 hours)
* Start Writing Module lessons (2-3 hours)

#### Weekly Update #2 (10/1/19)

* Phone/Tablet Mockups (3-4 hours)
* Class Diagrams (2 hours)

#### Weekly Update #3 (10/8/19)

* Critique Phone/Tablet Mockups & Class Diagrams (3 hours)
* Start finding icons, images, etc. for visuals/animations (4 hours)

#### Weekly Update #4 (10/22/19)

* Code moving text (3 hours)
* Start coding “code block” gaming mechanic (3 hours)
* Practice coding animations (2 hours)
* Prepare for Milestone Presentation (2 hours)

#### Weekly Update #5 (10/28/19)

* Integrate game mechanic into games (4 hours)
* Practice timing moving text with animations (4 hours)

#### Weekly Update #6 (11/5/19)

* Start advanced animations/visuals or continue critiquing current animations (3 hours)
* Start coding in-between user interface pages (i.e. Home, Exit, Pause, etc.) (2 hours)
* Start integrating game, module, and UI pages together (3 hours)

#### Weekly Update #7 (11/12/19)

* Continue integrating game, module, and UI pages (5-6 hours)
* Prepare for Milestone Presentation (2 hours)
* If possible, start writing test code (2 hours)

#### Weekly Update #8

* Proofread module (2 hours)
* Test application as a whole (running APK file) (3 hours)

#### Weekly Update #9

* Continue proofreading and testing (4 hours)
* Prepare for Final Demonstrations (3-4 hours)

# Project Deliverables

As proof of design, I will submit a class diagram for my Java code and tablet/phone mockups. My end product will be an APK file (the application itself), but I will provide access to an archive with the files used for the APK.