Release 0

**General**

Team 5 Members: *Frankie Situ, Illia Zahrebelnsiyi, Josiah Lee, Muzi Jiang, Murtaza Mushtaq*

**Introduction**

*This is the first project-planning document (Release 0) about our project to create a Chip-8 emulator and a visualizer. In addition, we will also develop two Chip-8 games and a tool to help us develop the emulator and games.*

**Project Organization**

* Back-end Developers and Testers (Emulator): Josiah & Frankie
* Back-end Developer and Tester (Visualizer): Murtaza
* Game Developer and Tester: Muzi
* Tools Manager and Tester: Illia

*The Assistant Project Manager changes every two weeks. The Project Manager changes every two weeks. The current Assistant Project Manager is promoted to be the new Project Manager for the next two weeks.*

**Risk Analysis**

* Not much risk is involved.
* Team members might become sick and miss out on meetings.
* Team members might forget to work on the project.
* Team members might forget to push their work

*If such risks were to happen, other team members are able to easily cover for the missing team member.*

**Hardware and Software Requirements**

* A web browser that runs JavaScript will be needed to run the emulator.
* Github will be our main software for database repository collection, version control, and source code management.
* Team members will code in JavaScript using an editor like Notepad++ or Visual Studio Code.
* Although not required, team members may use others plugins, libraries, or tools.

*For the repository, we were looking for something that is simple and easy to interact with. Github is a tool that allows us to centralize our code and make constant changes. We ultimately decided to use this version control system.*

**Communication Tools and Techniques**

*We will be using the following tools to communicate and share ideas:*

* Canvas Group Discussions
* Facebook Messenger
* GitHub

**Work Breakdown**

* Emulator
  + CPU
    - Memory: Chip-8 has 4096 bytes of memory.
    - Registers: Chip 8-has 16 registers: 15 8-bit registers, and 1 special 8-bit register.
    - Opcodes: Chip-8 has 35 opcodes that are two bytes long (16 bits).
    - PC (Program Counter): Shows which instructions we are currently executing.
  + Stack
    - The stack is used to remember the current location when a subroutine is called.
  + Inputs
    - A method that allows users to interact with the system through keyboard inputs
  + Graphics and Sound
    - A method for the emulator to draw on our display/screen
  + Timers
    - There are two timers that count down at 60 Hz.
* Visualizer
  + This shows the memory, the registers, and the instructions currently being executed. Programs can be paused and run backwards or forwards one instruction at a time.
* Games
  + Two simple and fun games
* Tools
  + A text editor plug-in

**Weekly Schedule**

*Our team members have agreed that we will meet every week on Friday after our CMPT 276 class.*

*In these meetings, we will review the work that we have done from the previous week and discuss topics such as:*

1. Did we accomplish the tasks that we set out to do?
2. What can we do better next week?

*We will also plan for the coming weeks. For example,*

1. Tasks that we will assign to each team member
2. Deadlines for the tasks
3. Problems that we might encounter in the coming week
4. Solutions for the above problems

**Project Schedule**

Week One - (Jan 7 ~ 13)

*This week will be spent planning.*

1. Where and what time are we going to meet up?
2. How are we going to approach this project?
3. What research do we need to do?
4. What roles do we assign to each member?
5. What do we know and what we don’t know?

Week Two - (Jan 14 ~ 20)

*Week two will be focused on research.*

1. What are the specifications of the Chip-8 Emulator?
2. What tools will we need?
3. Get comfortable with JavaScript

Week Three - (Jan 21 ~ 27)

*In week three, we will break down the different tasks, assign these tasks to our members, and start implementing some of the methods.*

1. Create a data structure specifying all the data we need for our CPU
   1. Memory
   2. Registers
   3. Program Counter
   4. Stack
   5. Inputs
2. Implement methods that runs our program
   1. Fetch method
      1. Fetches an opcode
   2. Decode method
      1. Decodes the opcode
   3. Execute method
      1. Execute the opcode
3. *Implement the two timers ( sound timer and delay timer )* 
   1. *60 Hz*
4. *Implement a method that creates a frame on our webpage that displays black and white pixels*

Week Four - (Jan 28 ~ Feb 3)

*Week four will be focused on implementing as well. This week, we will be implementing the 35 opcodes.*

1. Identify the opcode
2. Decode the instruction and specify what our system will do
3. Execute the instruction

Week Five - (Feb 4 ~ Feb 10)

*This week, we will be implementing the graphics, user inputs, and user interactions with our emulator.*

1. Implement a method that allows our system to draw to our display
2. Implement a method that allows users to interact with the system through inputs

Week Six - (Feb 11 ~ 17)

*Week six will be focused on debugging and testing. Each member will be creating automatic tests for their own parts.*

1. Implement an automated testing method
2. Fix all bugs

Week Seven - (Feb 18 ~ 24)

*We will be implementing the visualizer this week as well as a method that shows memory, registers, and executed instructions. This allows programs to be paused and run backwards and forwards one instruction at a time.*

1. Clarify the specifications
2. Plan for implementation

Week Eight - (Feb 25 ~ Mar 3)

*In week eight, we will create a Chip-8 tool that will be useful for developers.*

1. Brainstorm for ideas
2. Pick the best one
3. Design and implement

Week Nine - (Mar 4 ~ 10)

*This week we will be brainstorm for simple and fun games.*

1. Pick the two best ideas
2. Design and clarify specifications
3. Start implementing

Week Ten - (Mar 11 ~ 17)

*We will be making the finishing touches to the two games for this week.*

1. Testing and debugging
2. Discuss and refine the gameplay experience

Week Eleven - (Mar 18 ~ 24)

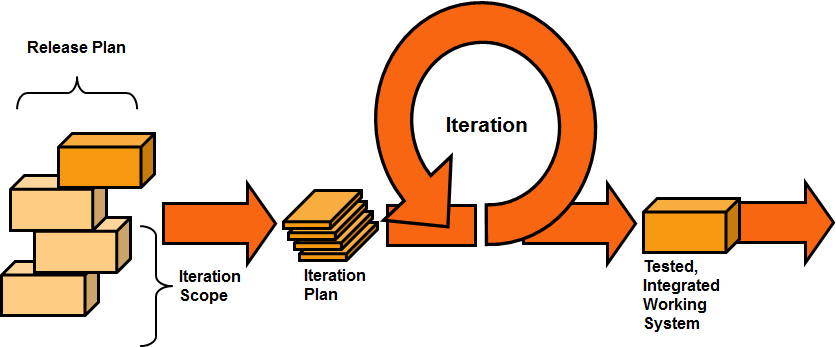
*This week we will be doing the finishing touches and prepare to wrap everything up for the final submission.*

**Reporting Mechanisms**

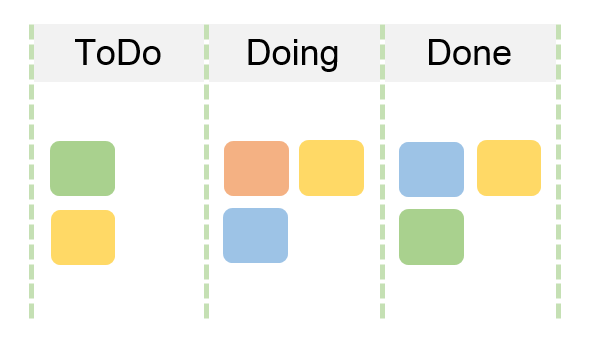
*Team members will regularly post code and other documentations on the Github repository. Team members will also report on their progresses during the weekly meetings.*

**Software Methodology**

*We will be implementing the “Agile Methodology” along with “Kanban Method” for this project. These are great methods that ensure that we are keeping track of the team’s progress. These methods also allow us to focus, prioritize on important tasks, and keep our workflow constant without cluttering and cramming.*



Refer to citation page for reference and credit



Refer to citation page for reference and credit

**Testing, Quality Assurance Tools, and Methods**

*Each person will be testing their own code and tasks for bugs and for quality assurance. We will run our emulator on different web browsers to make sure our emulator is as robust as possible. In addition, different Chip-8 games and programs will be run on our emulator to make sure that it works well. To make sure all runs smoothly, we will create automated testing methods and use the Visualizer to analyze and fix bugs.*

**Citation:**

“What Is Agile Software Development?” *What Is Application Lifecycle Management - Learn the Tools & Methods*, Inflectra, www.inflectra.com/methodologies/agile-development.aspx.

Admin. “What Is Kanban? Comprehensive Overview of the Kanban Method.” *Digité Blog*, Digité Blog, www.digite.com/kanban/what-is-kanban/.

“Multigesture.net.” *Multigesturenet*, www.multigesture.net/articles/how-to-write-an-emulator-chip-8-interpreter/.