CMPT276 - Team 1’s Product Document - Release 1

*Instructions for automated tests, chip-8 usage, and tool usage are located in INSTRUCTIONS.md*

Administrative Details:

* We plan to have meetings every Wednesday from 12:30 - 2:30 after class, and are all also available Monday from 12:30 - 2:30 for an emergency meeting if need be.
* We will be using Slack for our online communication method, with separate channels on Slack for each part of the project to keep discussions organized.
* We have a Github repository setup for our version control system.
* The Emulator, Visualizer, and Tool (a Chip8 Sprite-Editor) will be created in Javascript, while the two games (Pong and Tetris) will be created using Assembly Language.

Role Breakdown:

Emulator: Josh T. & Adam

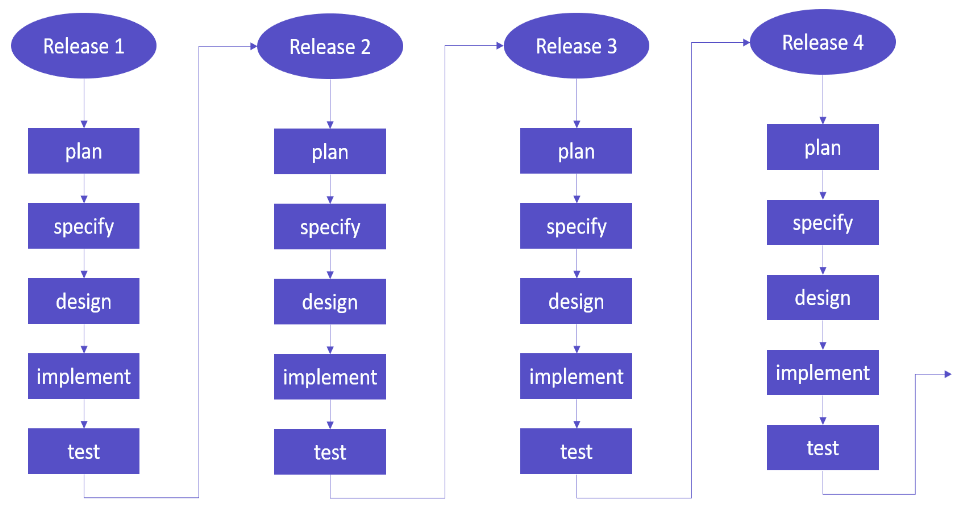
Visualizer: Adam

Tool: Sarb

Game 1: Josh H. & Sarb

Game 2: Brandon & Sarb

Software Methodology:



* We will be using an Incremental Development model.
* The planning and specifying phase will occur during meetings, to ensure that we are aware of and understand the requirements for the next release.
* The designing phase will occur in subgroups with those working on the emulator and visualizer in one group, and with those working on the game and tool in another group.
* Implementation will be largely done individually with the help of groupmates.
* The testing phase will mostly be done by unit testing, with one large automated test about a week before release.

Testing & Quality Assurance

* Instructions to run automated tests are in INSTRUCTIONS.MD
* Unit Testing:
  + For every function that we write for the emulator, visualizer, and tool, we will create a separate function to test that the original function does what it is intended to do, and add the test function to an automated testing file.
* System Testing:
  + Use system testing for simpler features such as the tool.
* Acceptance Testing:
  + Having peers test the emulator by playing the game later on in the development near releases 3 and 4

Detailed Use Cases for Release 2

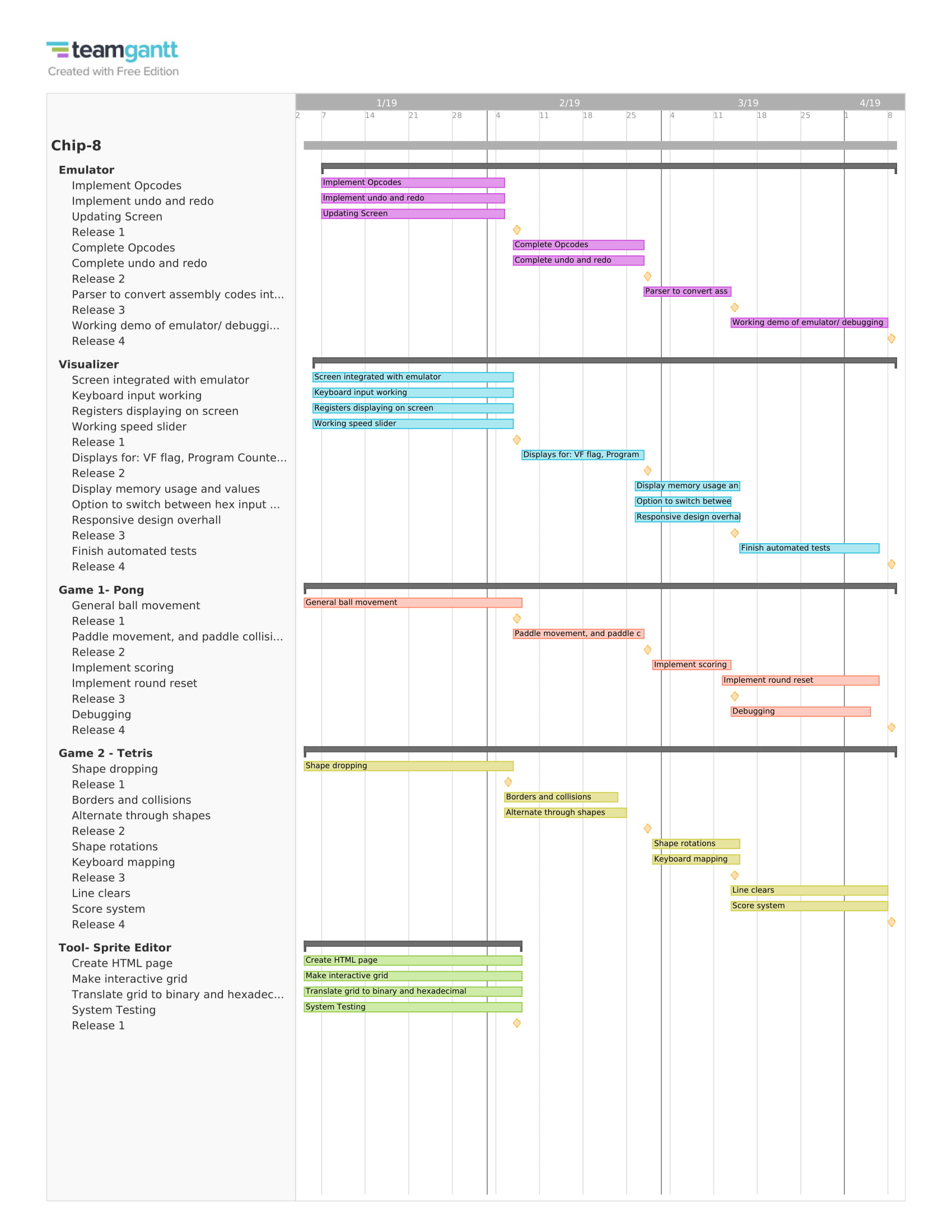
* Emulator
  + Will have automated testing showing all opcodes working properly when turn on the program.
  + Will fix most of the bugs in the emulator.
  + Will have an option to switch from hexadecimal to binary codes for the emulator to run, in case program is written in binary.
  + Codes that are put into the emulator text field will be able to work correctly for games, or programs, written in hexadecimal opcodes, as the developers’ intention (binary codes might be able to as work properly as hexadecimal codes).
* Visualizer
  + Will display register values
    - This includes the registers from V0-VF, the delay timer, the instruction register, and the program counter
  + The visualizer is for anyone who wants to analyze memory management, usually for those writing a Chip8 program or ROM
  + Text based input that integrates with emulator
  + UI for stepping forwards and backwards in the code
  + Toggle to switch from hex code input to assembly
  + key-button inputs working on click just like their keyboard counterparts
  + Timers and instruction history display
* Game 1: Pong
  + For release 2, the paddle will be drawn. Users will be able to move the paddle using keyboard inputs, and the paddle will interact with the ball properly (the ball will “bounce” off of the paddle).
* Game 2: Tetris
  + Will have basic side to side movement for blocks as well as rotation of blocks
  + Will have the “instant drop” feature
* Tool
  + Interact with grid to change boxes in order to create a sprite
  + Reset the grid to create another sprite
  + Translate button working. Displays the sprite translated into binary and hexadecimal.

Detailed Work Breakdown

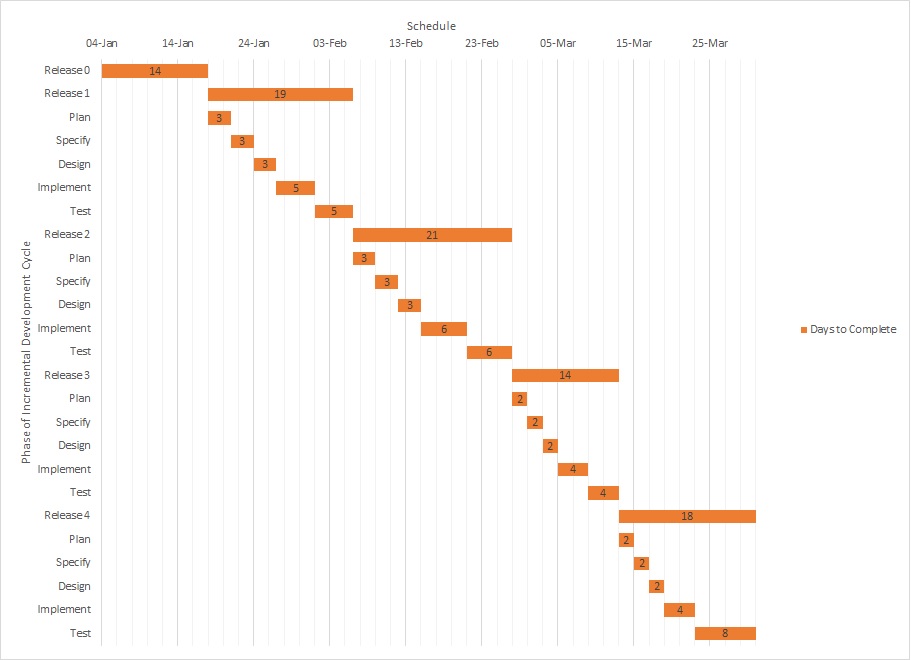
* Emulator (Total estimate: 60 hours (split roughly 5 hours/week over 12 weeks))
  + Release 1 (est. 20 hours)
    - A buggy working emulator.
  + Release 2 (est 15 hours)
    - Expect to have most of the bugs fixed from a lot of testings.
  + Release 3 (est. 15 hours)
    - Focus on polishing and optimizing the emulator.
  + Release 4 (est. 10 hours)
    - Fully functional emulator.
* Visualizer (Total estimate: 24 hours (split roughly 2 hours/week over 12 weeks))
  + Release 1(est. 8 hours)
    - All the registers and memory displayed alongside the chip-8 screen.
  + Release 2 (est. 6 hours)
    - Option to load program from hex or from mnemonic.
    - File selection button to load programs from file
  + Release 3 (est. 5 hours)
    - Controls menu to customize input keys.
  + Release 4 (est. 5 hours)
    - Rigorously tested and debugged Visualizer.
* Game 1 - Pong (Total estimate: 60 hours (split roughly 5 hours/week over 12 weeks))
  + Release 1 (est. 10 hours):
    - The ball is drawn to the screen, and is able to bounce off the walls and not go “out-of-bounds”
  + Release 2 (est. 15 hours):
    - The user is able move the paddles through keyboard input
    - Ball has proper collision with the paddle (bounces off).
  + Release 3 (est. 20 hours):
    - Implementation of the scoring system, including round “resets” for when a point is scored.
  + Release 4 (est. 15 hours):
    - Additional Testing and Optimization
* Game 2 - Tetris (Total estimate: 60 hours (split roughly 5 hours/week over 12 weeks))
* Release 1 (est. 20 hours)
  + Basic shape movement/physics
  + Collisions between borders and blocks
  + Stacking and spawning of different blocks
* Release 2 (est. 20 hours)
  + Shape rotations
  + User input
* Release 3 (est. 10 hours)
  + Line clears
  + Scoring system
* Release 4 (est. 10 hours)
  + Potential additional features (power-ups) if time permits, otherwise, optimization of code
* Tool - Sprite Editor (est. 15 hours to complete)
* Release 1 (est. 20 hours)
  + Interacting with grid to change boxes in order to create a sprite
  + Reset the grid to create another sprite
  + Translate button working. Displays the sprite translated into binary hexadecimal
  + System Testing

Detailed Work Breakdown:

We implemented the detailed work breakdown in a gantt chart. JPG image is provided or access the pdf with this link: <https://drive.google.com/file/d/1NmXA6pGnwslELsDF75-gQvDHEhW4qD5N/view?usp=sharing>



Detailed Schedule (with due dates)

* The detailed work breakdown shows what specific pieces of the project will be done for each release, the following schedule breaks down each release based on our software methodology (incremental development), to plan how we will complete each part:
* For each release, we have scheduled out how much time will be spent on the Plan, Specify, Design, Implement, and Test aspects of the Incremental Development Cycle.

Changes since Release 0:

* Software Methodology:
  + Explicitly stated that we are using the Incremental Development Model
  + Added details, explaining how each phase of the incremental development cycle will be approached / how we will use the model.
* Adjusted testing and quality assurance section. Decided to work in JavaScript instead of Jest
* specified detailed use cases for release 2 for each major component.
* Adjusted detailed work breakdown section
  + For each major component of the project (Emulator, visualizer, game 1: pong, game2: tetris, tool):
    - Add details about what will be completed for each release
    - Add time estimates for how long it will take to complete each part of the releases
  + For Tetris:
    - Added shape collisions, borders, and spawning to goals for release 1
    - Moved shape rotations, and spawning of different blocks for release 2
    - Moved line clears to release 3
* Added Gantt chart for work breakdown
* Added Gantt chart for schedule (Incremental Development Cycle)

Features Completed/Not Completed for Release 1

* Emulator
  + Planned features include:
    - Implementation of most of the opcodes, with exception of some
    - Implemented undo and redo features
    - Implemented screen function features
* Visualizer
  + Planned display elements not completed for Release 1:
    - Register I
    - Timers
    - Stack
    - Memory
    - VF flag

**Cause of delay**: Emulator debugging took priority.

* + Planned features Completed:
    - Screen interfaced with emulator
    - Text field for code which can be loaded into the emulator
    - Functioning step forwards and undo buttons
    - Working speed slider
    - Foreground and background colour options
    - Keyboard layout display
    - Display for V registers
    - Working play/pause button
* Game 1 - Pong
  + Features specified for release 1 that are completed:
    - General ball movement is complete, with proper collisions with the wall
  + Unplanned features for release 1 that are completed:
    - Implement a delay feature to smoothen out the ball movement (Note: this feature works using an external emulator, but not on our groups emulator).
  + All features that were planned to be implemented were implemented.
* Game 2 - Tetris
  + Features specified for release 1 are complete and include:
    - Shape movement and physics (falling blocks)
  + Unplanned features for release 1 are complete and include:
    - Shape/border collisions
    - Stacking and spawning of different blocks
  + One feature that was planned for release 1 but was not implemented was the shape rotations due to the agreement to prioritize user input for the shape rotations instead of having the shapes constantly rotate while falling.
* Tool - Sprite Editor
  + Features specified for release 1 that are completed:
    - Interact with grid to change boxes in order to create a sprite
    - Reset the grid to create another sprite
  + Unplanned features for release 1 that are completed:
    - Translate button working. Displays the sprite translated into binary and hexadecimal.
    - Ignore lines that are not apart of sprite (ignore empty lines)