MAE Operation & Execution Modes

This document outlines the different modes for how the MAE should operate.

# Operation Modes

Operation modes describe how MAE is used by the end user.

## Interactive Mode

In this mode, the console next to the MAE is unlocked and has full debugging capabilities to the MAE, and all the mechanical parts are used. The camera feeds could also be viewed live from the console screen or be displayed on special screens on the wall of the STEAM center. In interactive mode, jobs that are queued online are not executed, and the online console becomes read only, which provides in-person users, prompt, uninterrupted access to the MAE while they are present.

## Interpretive Mode

In this mode, instead of executing a list of instructions, the console allows the end user to directly enter instructions and they will be executed immediately after being entered, enabling the end user to visualize specific operations.

## Sandbox Mode

Sandbox mode has a user interface which the user can use to build and run programs on the MAE, in addition to the other debugging tools available in interactive mode. This will include step by step instructions in which the user can build programs

## Unattended Mode

Unattended mode is the most different mode when compared to Interactive mode. In unattended mode, the console is disabled and is replaced with a virtual queue system. Programs are added to the queue from the website at any time and are processed while MAE is in unattended mode. When a program reaches the front of the queue, the MAE is reset, and the cameras start recording, and mechanical mode execution is started. Once the program finishes, the recordings end, and all memory blocks are zeroed. Then, the MAE repeats this process with the next program in the queue, if any. If there is no item in the queue, the MAE will pause until unattended mode is disabled or another program is added to the queue.

This method of an unattended mode is beneficial for a variety of reasons. For instance, unattended mode could be enabled after business hours, which would prevent online operations from stopping in-person users from using the MAE and would also eliminate the privacy concerns of recording inside of the STEAM center.

# Execution Modes

Execution modes describe how the MAE’s virtual machine executes the program.

## Mechanical Mode

Mechanical mode is the default mode of MAE, as it was originally envisioned by Mr. Ben-Yaakov, where the memory blocks move around the machine and all mechanical functionality is enabled. Execution itself is handled by the virtual machine which runs in software, and the virtual machine interfaces with the various hardware components of MAE to provide a visual representation of the execution. In this mode of operation, the speed of execution is dependent on the time it takes for the hardware to move all of the necessary parts around to visualize an instruction.

## Fast Mode

Fast mode is like Mechanical mode, but all mechanical functionality is disabled, and execution is only handled in software to increase the speed of execution. This mode could be enabled manually by the user or could be automatically employed for certain tedious processes, such as loading the program into memory before the program is executed, or for zeroing all the memory blocks. Since fast mode is difficult to visualize for the end user, it could be visualized by adding strings of LED lights along the tracks that blocks usually move, and then these LEDs could be briefly turned on and off depending on what operations were happening within the virtual machine. Since this mode of operation is all solid state, the speed of execution is variable.

## Virtual Mode

Virtual mode is an even faster version of Fast Mode, where programs are run only within the virtual machine, and there is no output on the MAE of any kind. Right now, this mode will only be used internally to evaluate if programs which are submitted to the unattended mode queue to see if the program will take too long to run. In virtual mode, every instruction has a hardcoded weight value that helps evaluate how long it would take for the instruction to be executed, which will be used to calculate an ETA of how long it would take for a program to execute.