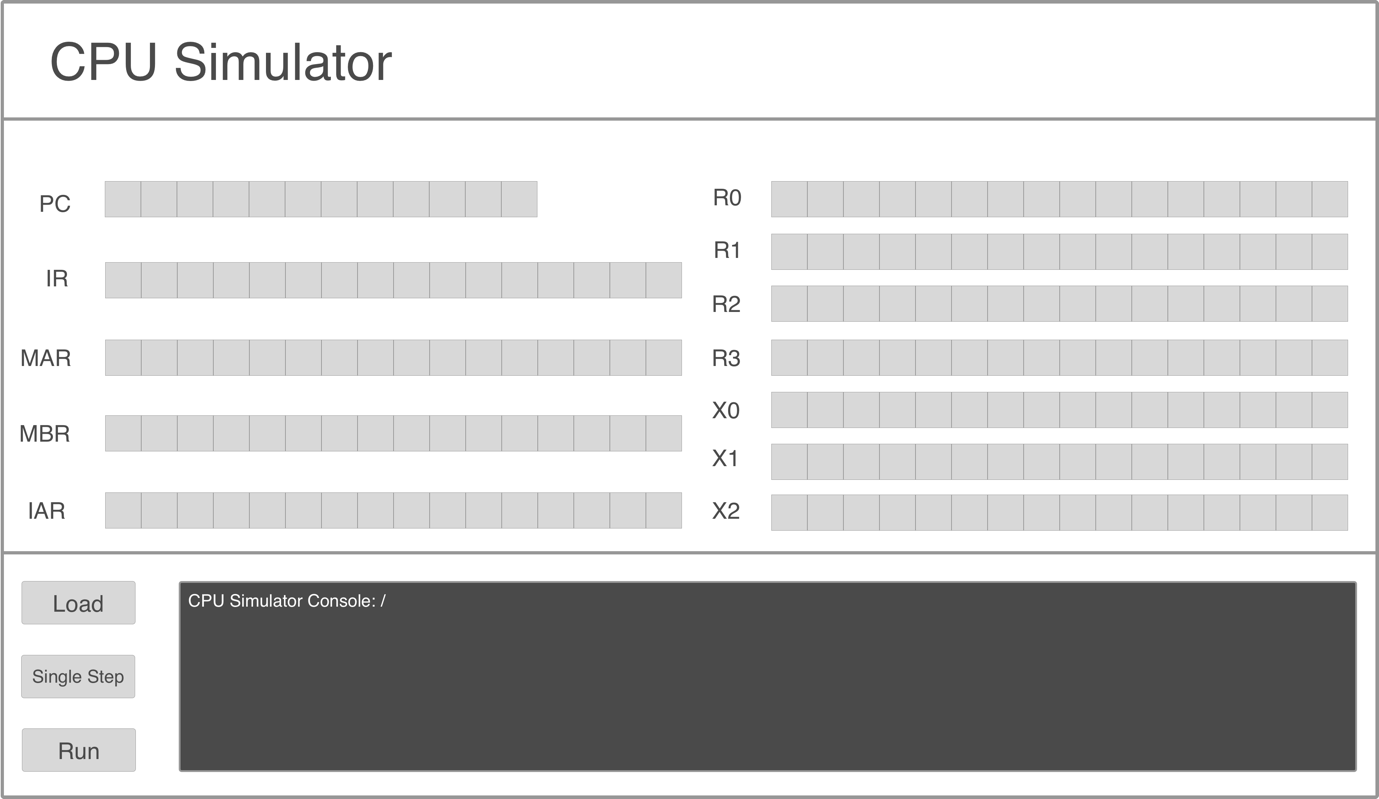
|  |
| --- |
| CSCI 6461 Computer Simulator |
| User Guide |
| Lulwah AlKulaib Amirah Abdulrahman Yichen Zhou Han Wang |
|  |
|  |

**Phase I: Basic Machine**



2

1

3

Figure 1: GUI Panel Display

* When the Jar file runs the GUI in ‘Figure 1’ will be displayed.
* The GUI consists of three parts:
  1. Each of the registers in the first column exists to hold certain information:
     + PC: displays the current instruction that is being executed.
     + IR: displays the currently executing register.
     + MAR: displays the memory address of the data to be fetched.
     + MBR: displays the data fetched from and to the memory.
     + IAR: displays the address of the current executing register.
  2. Registers R0-R3 are General Purpose Registers, Registers X0-X2 are Indexed Registers.
  3. CPU Simulator Console: is the control area for the user to load files, run the program and monitor the output. It consists of the following:
     + Load Button: Loads a text file of instructions into memory.
     + Single Step Button: Loads memory contents one instruction at a time, and changing register values accordingly
     + Run Button: Loads all instructions in memory sequentially into registers and displays values on the console (For now. It will later perform instruction execution in coming phases).
     + Simulator Console: A console to display messages and output to ensure correctness.

**Phase II: Memory and Cache Design**

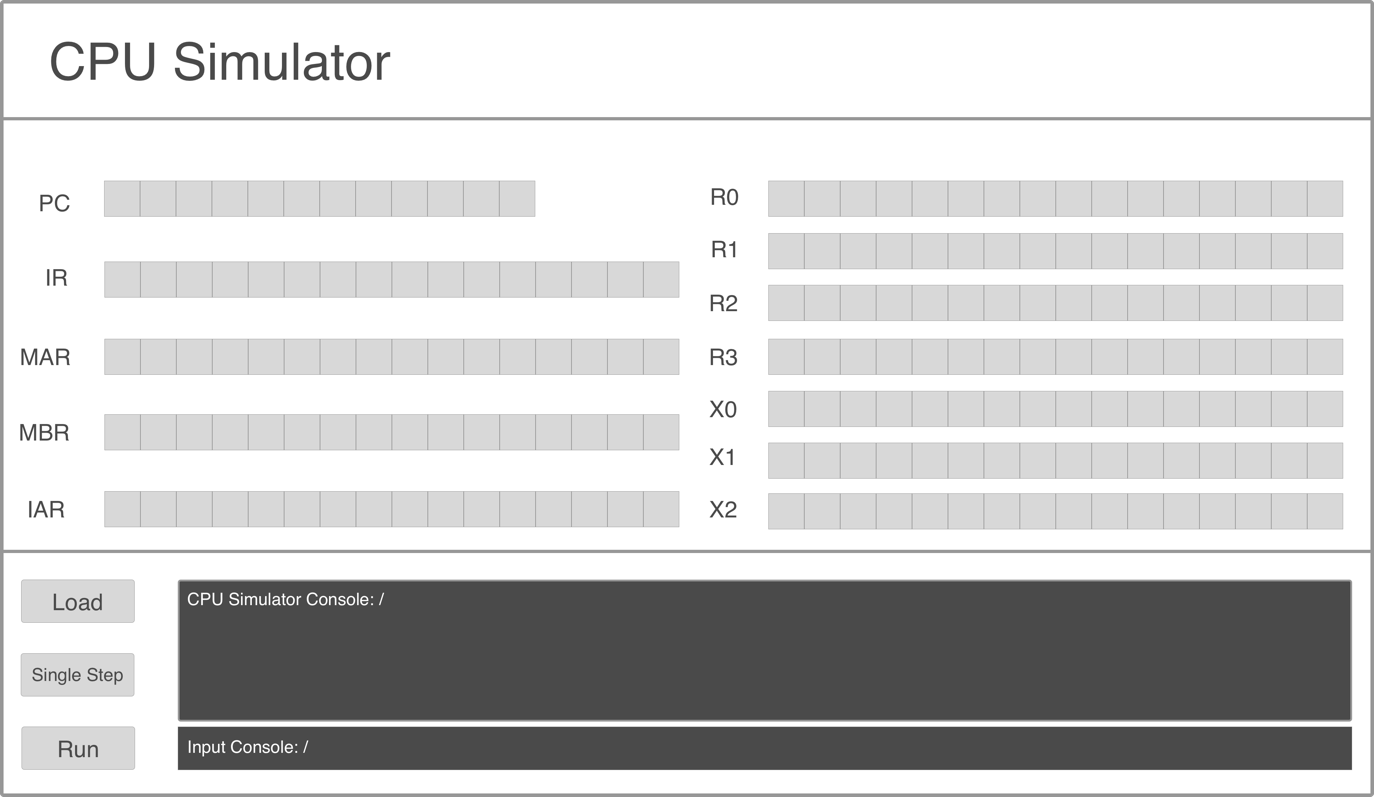


Figure-2: GUI Panel Phase 2

* When the jar file runs, the GUI Panel – Phase 2 will be displayed.
* Components added in Phase 2:
  1. New “Program 1” choice in “Load Button Section”.
  2. New “Individual “OPCode” choice in “Load Button Section”.
  3. New Input Console, used as input interface for user 1 to input data.
* How to run Program 1?
* Choose Program 1 in Load Button Section to initialize Program 1.
* Input at least one number as instructed in Simulator Console showed in the Input Console, then press “Enter” to load numbers in Register and Cache. Invalid input(s), such as string, character and symbol will be automatically removed.
* Input only one number to search the closest one in the previous inputted number(s). Invalid input(s), such as string, character and symbol will be automatically removed.
* Find the result in the Simulator Console.
* How to run Individual OPCode?
* Choose Individual OPCode in Load Button Section to initialize program.
* Input 16-bits instruction in the Input Console, then press “Enter” to load instruction.
* Because it is an individual instruction, program may crash based on un-prepared condition – overflow, underflow, etc.

**Phase III: Executing Program 2**

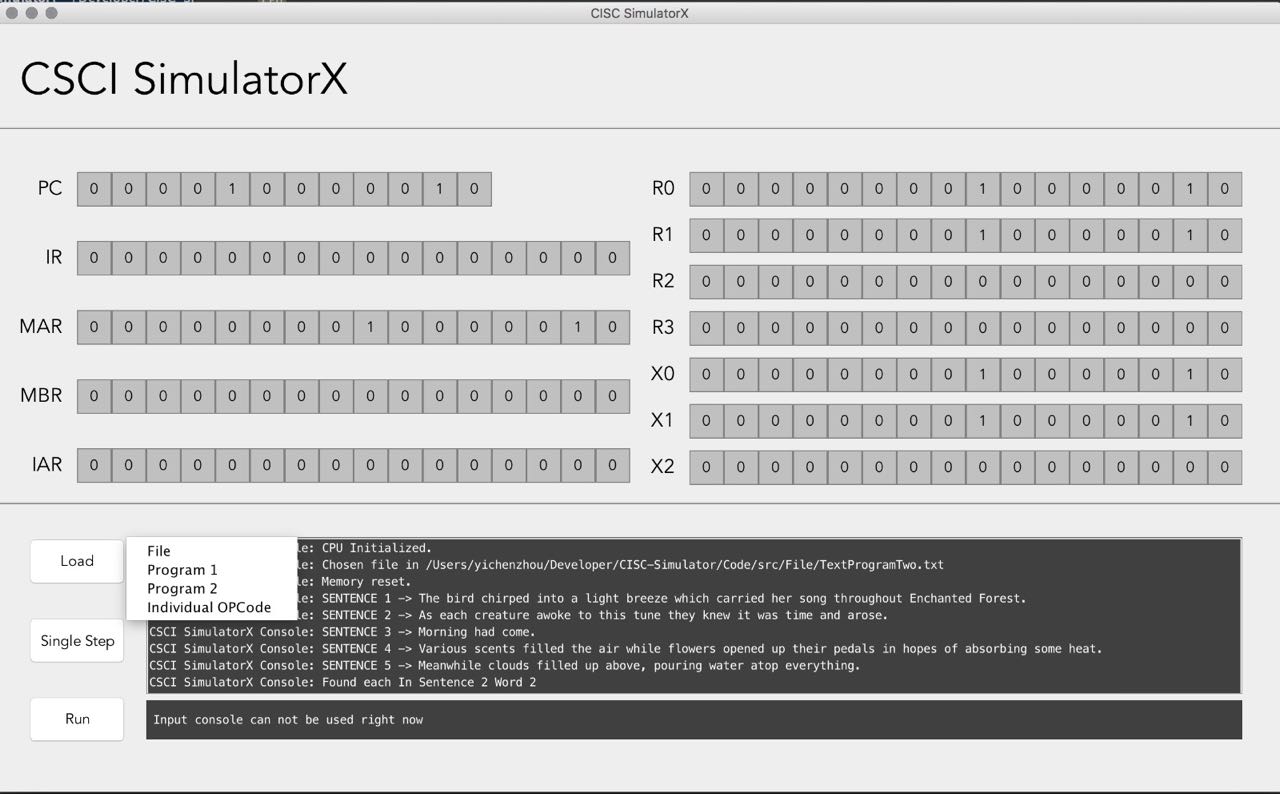


Figure 3: Step 1 - Loading Program 2

* When the jar file runs, the GUI Panel – Phase 3 will be displayed.
* Components added in Phase 3:
  + New “Program 2” choice in “Load Button Section”.
* How to run Program 2?
* Choose Program 2 in Load Button Section.
* Choose program 2 file from file explorer.
* Once the program is loaded into memory, user can use input console to choose a word to search for.
* User then inputs a word from the displayed sentences, press enter.
* Program will return location of that word.

|  |  |  |  |
| --- | --- | --- | --- |
| Version No. | Version Date | Author | Description |
| 0.1 | 02-05-2017 | Han Wang, Lulwah AlKulaib | Phase I: Basic Machine |
| 0.5 | 03-04-2017 | Yichen Zhou | Phase II: Memory and Cache Design |
| 1.0 | 03-32-2017 | Lulwah AlKulaib | Phase III: Executing Program 2 |
|  |  |  |  |