Jenny Johns
Arnold Esguerra
Michael Hanna
Final Project - Computer Systems Instructions

How to access on the web

1. If you are under the age of 13, make sure you have your parents permission to access the internet.
2. Open up a browser, like Google Chrome, Mozilla Firefox, or Safari. If you are using Internet Explorer (I.E.) or Microsoft Edge, use that to download one of the above browsers, uninstall I.E./Edge immediately after and begin enjoying life more.
3. Go to: https://cache-simulator-jam.herokuapp.com/
4. Example runthrough can be found through the YouTube video: https://youtu.be/IKHT7Ju0y4E

How to run on local host:

- OS: Linux: Ubuntu, Windows, Mac, etc.
- Requirements:
    - MongoDB
        - Download MongoDB: https://docs.mongodb.com/manual/administration/install-community/
        - Make sure to add the binaries to the system PATH
    - Node JS
        - Download Node: https://nodejs.org/en/download/
    - Angular
        - Install angular by typing in the command line: npm install -g @angular/cli
    - Git
        - Download Git: https://git-scm.com/download/win
- Download the github repository by cloning: https://github.com/Summer18CS5600/finalproject-team-space-jam.git
- Enter the repository through command line

Linux/Ubuntu/Mac:

In the terminal...

- After installing mongo type “sudo service mongod start” (or on Mac, just “sudo mongod”) begin the service.
- Install dependencies by going into the project folder by typing “npm install”
- Build the project by typing “sudo ng build”
- Start the web server by typing “npm start”

- Start the back-end server by entering “sudo nodemon server.js”
- Go into a browser and go to url: http://localhost:4200

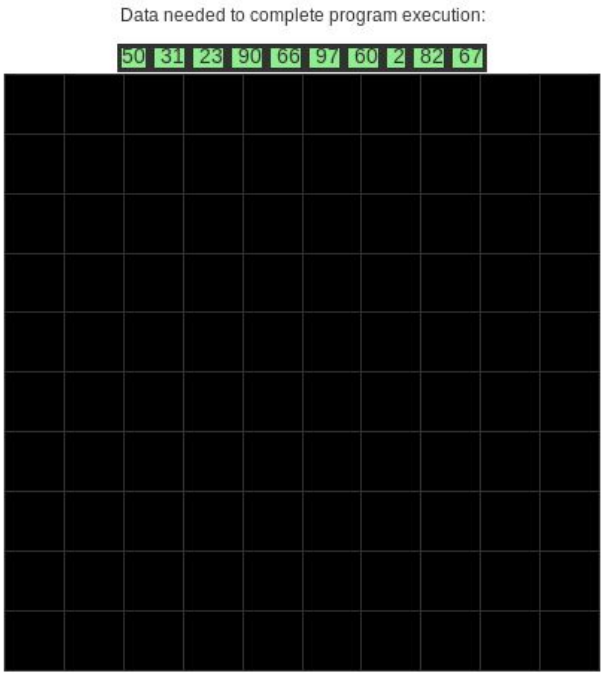
Windows:

- Build the project by running: npm install
- From within the repository start Mongo, by typing the command : mongod
- Open another command prompt and direct back to the same repository from there, and build the webpack by typing: npm start
- In another terminal, and directed back to the repository, start the server by typing: node server.js
- In your web browser, go to the url: http://localhost:4200
- If you’ve added code and want to push to github:
    - First build the project for a production environment by typing the command: ng build --prod
    - Then commit and push all changes to github:
        - git commit -m “your message”
        - git push
    - Changes from the master branch of the github repository are automatically deployed on Heroku, so the changes should be seen after the site it done rebuilding the project

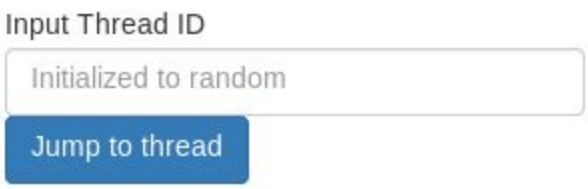
How to access it online/play the game:

- After reaching the game on the Heroku URL: https://cache-simulator-jam.herokuapp.com/
- Begin by initializing the memory region, by typing an alphanumeric ID in the textfield and clicking on the button ‘Access Memory’.
    - If multiple users enter the same ID, they can simultaneously access the same memory region

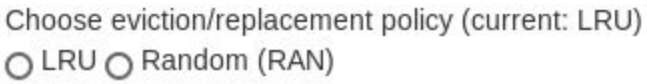
The memory region will be displayed as such:



- If you want to enter a specific thread, enter one in the thread id field, otherwise one will be randomly generated for you.



- **The goal is to access all the data needed to complete your program (as displayed by the line of numbers on top of the memory region, initially in green).**
    - This data is determined by the thread id.
    - If it is completed, it will turn red and you can move onto the next number.
    - The data must be access **in order!**
    - At any time, feel free to jump to a different memory region or thread id and come back and the state will remain as it was.
- The cache eviction/replacement policy is defaulted to LRU, or least recently used.
    - The players can change the eviction/replacement policy by choosing the radio button of the one they would like to use.



- Also displayed is the current cache.
    - This shows the current cache lines in play within the memory access region
    - Depending on the eviction/replacement policy chosen, this will update accordingly throughout the game

        - Ex: If using LRU: After bringing up 4 cache lines, the next cache line brought in will then cause the ‘oldest’ cache line to be evicted from the current cache.
    - It has a cache line and when it was last used. Each time you click on the same memory region, there is a counter that increments. The higher the number, the younger it is. The lower the number, the older it is.

The Current Cache:

Line Number: 16 **Last Used :** 11	64	65	66	67
Line Number: 21 **Last Used :** 12	84	85	86	87
Line Number: 24 **Last Used :** 13	96	97	98	99
Line Number: 19 **Last Used :** 9	76	77	78	79

- There is also an event log displayed on the page.
    - For every move made (shows a running account of the last ten), it details whether it is a hit or miss.
        - Ex: When just beginning the game, the first access will result in a cold miss.

Most Recent (at top of list):
CONFLICT MISS: The Cache Line #4 was replaced by Cache Line #24
CONFLICT MISS: The Cache Line #15 was replaced by Cache Line #21
Cache was hit by the cache line #16!
CONFLICT MISS: The Cache Line #22 was replaced by Cache Line #16
CONFLICT MISS: The Cache Line #23 was replaced by Cache Line #19
CONFLICT MISS: The Cache Line #19 was replaced by Cache Line #4
CONFLICT MISS: The Cache Line #16 was replaced by Cache Line #15
Cache was hit by the cache line #22!
CONFLICT MISS: The Cache Line #24 was replaced by Cache Line #22
Cache Line 23 caused a cold miss!