GLASS

Final Report

Team:

Tommy Galletta

Alexander Lockard

Faculty Advisor:

Dr. Stansifer

Final Release

The final version of our tool for this project, version 0.3.0a, is available on the download page of our project website

Milestone Six Task Matrix

Task	Completion %	Tommy	Xander
Polish implemented features	90%	45%	45%
Finalize user documentation	100%	80%	20%
Conduct evaluation and analyze results	100%	65%	35%
Test/demo of the entire system	100%	0%	100%
Create user/developer manual	100%	50%	50%
Create demo video	100%	100%	0%

Task Discussion

Polish implemented features

- Added several new features to the interpretation script system
- Reworked project website
- Many changes made and bugs fixed both for main system and GUI

Task Discussion (cont.)

Finalize user documentation

- Documentation updated to reflect all of the new additions and changes to GLASS
  - Documentation now contains over 20 individual subsections!
- Documentation for older versions of GLASS moved to an archive accessible on the project website

Task Discussion (cont.)

Conduct evaluation and analyze results

- Conducted user demos with 7 Florida Tech students of varying academic backgrounds
  - Each user performed a series of tasks, and the results were recorded and analyzed

Thank you to our testers!

- Erin Brasher
- Emma Conti
- Christopher DeMuro
- Evan Gunderson
- Christopher Hinton
- Ty Mercer
- Timothy Shane

You all helped to make this tool the best it can be!

Evaluation Results

- Syntax definitions, interpretation scripts, and documentation were all rated approximately a 9/10
- GUI scored 5.3/10, mostly due to uncaught bugs
- Overall rating for the tool was an 8/10

Task Discussion (cont.)

Create user/developer manual

- Created developer manual with information about how to extend the GLASS system

Create demo video

- Created a video showcasing the main features of GLASS and how to make an example syntax definition and interpretation script

Task Discussion (cont.)

Test/demo of the entire system

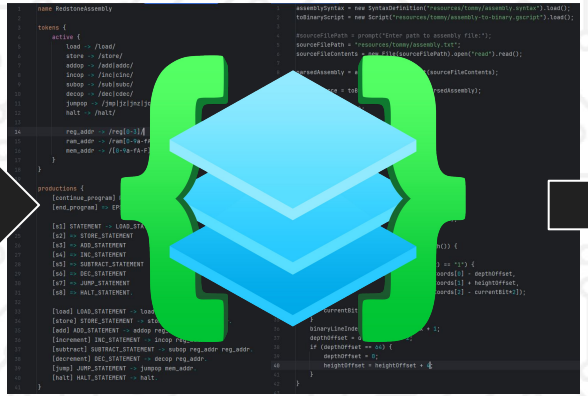
- Created demo of an example syntax definition, an example source file, and an example interpretation script

Speaking of which...

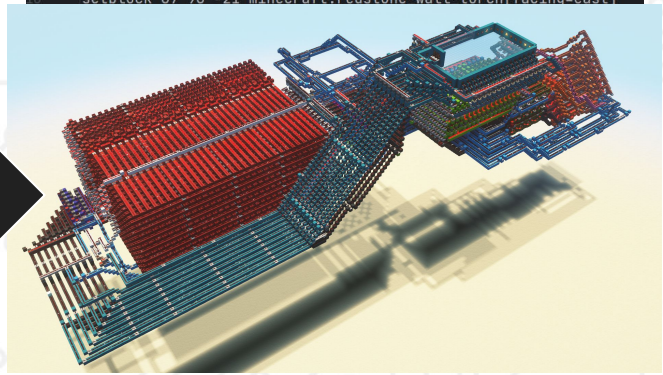
Demo Time!

Bonus: Practical Application

```
load ram0 reg0
add reg2 reg1
cinc reg3
dec reg0
jc 01h
load ram1 reg1
store reg2 ram2
store reg3 ram3
halt
```



```
fill 71 98 -3 9 110 -33 air replace minecraft:redstone_wall_torc
setblock 71 98 -9 minecraft:redstone_wall_torch[facing=east]
setblock 69 98 -7 minecraft:redstone_wall_torch[facing=east]
setblock 69 98 -19 minecraft:redstone_wall_torch[facing=east]
setblock 69 98 -25 minecraft:redstone_wall_torch[facing=east]
setblock 67 98 -7 minecraft:redstone_wall_torch[facing=east]
setblock 67 98 -15 minecraft:redstone_wall_torch[facing=east]
setblock 67 98 -17 minecraft:redstone_wall_torch[facing=east]
setblock 67 98 -19 minecraft:redstone_wall_torch[facing=east]
setblock 67 98 -21 minecraft:redstone_wall_torch[facing=east]
```



Lessons Learned

- Importance of an agile development process
- Importance of code modularity
- Understanding project scope
- Importance of user feedback
- Version control and task accountability

Faculty Advisor Feedback

- The amount of flexibility the interpretation script provides is good
- The project website looks great
- Happy to see original use cases being tackled
- Happy with our commitment to the project overall

Questions?