Intro to Scripting Semester Long Project Reflection Write-Up

Game-Time Limiter

Eric Ragle

**What Went Well**

Even though the project is not yet user friendly, it accomplishes its basic task: closing an application after a specified amount of time. It also displays a warning message that the time is almost out after a specified amount of time. It successfully uses Bash, Python, Batch, a Windows Environment, a Linux environment, a consistent coding style, (very basic) logging, comments, script reliability, script robustness, socket communication, and proper error handling. It successfully integrates multiple languages into a workflow.

**Challenges Faced and How They Were Overcome**

Overall, the most notable challenge I encountered was the Python Flask server. I have never had any experience setting up anything like this. I spent hours reading documentation and watching tutorial videos. The only reason this was challenging was my lack of knowledge. I have learned much from being forced to implement this. I also had some trouble figuring out how a multi-language workflow works (calling scripts of different languages, etc). This was for similar reasons, I had just not had much or any experience prior to this project. Creating a single script that launches all of the other ones in a clean, structured way (launch.py) was quintessential for this project, and it was one of the last things I did. I also initially wanted to have launch.py be an .exe, because this would make it much easier for a user with less technical knowledge to run my scripts. However, whenever I used PyInstaller to do this, once the .exe was launched it would create hundreds of processes that could not be closed and it would use up all of my resources until I restarted my computer. I figured, for now, the Python launch script would work fine. I did not have enough time to troubleshoot this.

**Lessons Learned**

I learned so much from this project. I learned about sockets, Python libraries, PATH, setting up servers, clients, logging, error handling, and multi-language workflows. I feel like I have a new knowledge and increased ability to make scripts for my own use in the future. I am excited to continue this project after the semester. I feel like I need to finish it and learn more.

**Future Improvements**

I feel the biggest drawback with my project is the lack of user-friendliness. I want to incorporate a GUI screen that, when launched, allows you to select from a list of currently running applications instead of having to go into the Bash script and manually input your process name. The GUI would also allow the user to input the allotted time and warning time in seconds, also preventing them from having to access the code. When all things are configured, the user can hit a ‘launch’ button. This would all be able to be launched from an .exe, instead of having to go into a command line and open the Python launch script. I also want the .exe to check for dependencies and install them for the user if they do not currently have everything needed.

**Design Changes, What They Were, Why They Occurred, The Details, Etc.**

There is now a single script that launches everything. This makes it so much easier to keep track of what is currently running. It is also much easier and faster to use.