3-2 Milestone Two: Enhancement One: Software Design and Engineering

Vy Huynh

Southern New Hampshire University

CS499 Computer Science Capstone

Sherri Maciosek

November 15, 2024

The artifact that I started the first enhancement with was my clock project from CS210. In the clock project, the user is asked to input a time such as 23:00:12, the program will then return the user the time in military time, and AM/PM time. The user will then be able to add 1 hour, add 1 min, or 1 second to the clock, in which the clock will return the new time in AM/PM and military time. I selected this project as part of my portfolio because I felt it was a very fun project to code, and it was also one of the first projects that I coded in. So I knew that my code was not polished and had countless issues and edge cases that went unchanged and unaccounted for. My plan for enhancement for this project is first rewrite the program in python instead of its original language of python. Next, I would like to make the clock run by itself instead of it not running. The user would still be able to add 1 hour, 1 min, or 1 second to the clock, and it won't stop the clock but instead keep it running. The user will also have the ability to manually input how many hours, minutes, and seconds they would like to change the clock by, instead of just adding 1 hour, 1 minute, and 1 second each time. First, I will study and fully understand the working of the C++ file beforehand. After understanding how my code works, I'll figure out how to get a live running clock on python. Next I will convert all the code that is C++ and convert them to python code. Finally, I will add the last extra function of the code, which allows the user to choose to manually enter the number of hour/minute/second they would like to change the clock by. All these improvements overall improve the experience of the clock as it allows for automations where users aren't expected to manually input the time, and it allows for more freedom as users can also enter custom hours they would like to change the clock by.

From the original planned enhancements that I planned in module one which are: translate code from C++ to python, make the clock run in real-time, and allow users to manually change the clock. All these enhancements were met and completed by me, the code work as

intended and the enhancement overall made the code more usable and efficient. My original course outcome plan that I planned to meet for this artifact are “Design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts.” and “Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals.” I believe that I met both course outcome goals with the enhancement that I have made to my project.

Reflecting on my experience for this project, I had a lot of fun coding for this project. I did quite a bit of research because the first issue that I ran into is that I did not know how to set up an actual running clock in Python. This was quite a quick research though, as a quick Google showed many Stack Overflow that showed me the solution. The second challenge that I faced is that I did know how to get input, usually you can just use *and = input(‘name?’)* and get an input but since the clock is constantly running, using ‘input’ is going to stop the clock from running, for this I did a short research and that is when I found ‘terminos’ and ‘tty’ to use to not disrupt the input. Finally, the last problem that I ran into and stumped me for quite a while is that even after using Terminos and TTY. I basically had the same issues as if I use input where my clock would not update until I send it an input, so basically I was back to square one. This research was a lot more difficult as it led me down a rabbit hole, even some Reddit posts told me to use ‘Curses’. I thought about using Curses so I did some research. However, I learned that Curses are not easy to learn and I wouldn’t even have time to learn Curses and completely rewrite my code in a few days. I decided to do some more research and finally I bumped into a Stack Overflow post that a user asked “How to check if stdin has some data?” Upon reading this post it was

exactly what I needed and after looking into the post some more I was able to find a GitHub with a set of code that I tested out, and it allowed my clock to run in real-time while also being able to receive input at any time. With that issue solved, it basically solved all my problems and my code was finally completed.

Reference

262588213843476. (2012, February 24). *Python function to get keypresses from the terminal*. Gist. https://gist.github.com/jasonrdsouza/1901709

262588213843476. (2013, March 5). *Linux python asynchronous input from stdin using select()*. Gist. https://gist.github.com/davstott/5086945

Curses Programming with Python. (n.d.). Python Documentation. https://docs.python.org/3/howto/curses.html

How do I get the current time in Python? (n.d.). Stack Overflow. https://stackoverflow.com/questions/415511/how-do-i-get-the-current-time-in-python

How to get user input during a while loop without blocking. (2009, August 11). Stack Overflow. https://stackoverflow.com/questions/1258566/how-to-get-user-input-during-a-while-loop-without-blocking

mlzboy. (2010, September 21). *How do I check if stdin has some data?* Stack Overflow. https://stackoverflow.com/questions/3762881/how-do-i-check-if-stdin-has-some-data