Final Report

EECE3093

Tyler Brunelle

**Overview and Justification**

I selected an independent project, something I have wanted to develop from the ground up for a while, a sports betting tracker. There is no good way to view your betting history across multiple sportsbooks simultaneously. Most serious betters track their bets in a spreadsheet, so I wanted to emulate that while putting a computer science spin on it.

The name of my project is “Bet Tracker”, here is a link to my public repository: <https://github.com/tybrun/bet-tracker>

I am a pretty inexperienced programmer. None of my co-ops have put me in the position to write much code, I haven’t done enough learning outside of the classroom to better myself as a computer scientist. So, I wanted to use this project as my first attempt to build a Graphical User Interface (GUI). I wanted to design a window that a user can interact with, as opposed to interacting with just a terminal. I chose Python, as it is the most versatile and well-known programming language. I figured there had to be some good libraries to help me build this project. I settled on Tkinter, as I read good reviews regarding its ease of use.

I considered making a personal budgeting application, as my primary budget app, Mint, was shut down recently. But I would like to save that one for when I have more knowledge and experience, and I can deliver a truly good product.

**Project Context**

This project came to mind after I turned 21 and was able to gamble. I have never cared for gambling, always seeming to lose more than I win, that is the whole point after all. But, my friends and roommates enjoy sports betting, and I love sports. However, I struggle with money, so I don’t want to throw it down the drain. I also am a very analytical thinker; I like seeing the stats and metrics.

I looked around for some sort of existing software that bettors use to keep track of their betting history. Something that could show their net gain (profit/loss), their total number of bets, their history over certain timespans, their betting habits, and statistics of that nature. But nothing exists, bettors rely on homemade Excel workbooks or nothing at all.

If done well, this project could fill that space, providing something for bettors to use to keep themselves accountable, potentially preventing themselves from going off the deep end, getting addicted to betting or causing financial problems.

I wanted this project to be a lot grander than it was, with all the data being stored in an SQL database. I have been taking Database Management I & II for the last two semesters, so I thought this project might help me bridge the gap between my typical computer science work and database management. That ended up getting cut, as I ran out of time.

**Project Governance**

I created a GitHub Project to communicate with myself throughout my project, utilizing the Agile development strategy. I planned on completing the work throughout 7 cycles, or iterations. Each iteration was between 3-5 days, as I started the project very late. I communicated with myself by writing After Action Reports (AARs) on occasion. Those contained notes on tasks I completed or was working on, notes for my future self, or ideas for my next workday.

**Task Description**

GitHub Project Creation: The first step, try to outline the project, or at least the first iteration or two.

Repository Creation: This was my first assigned task, creating the repository. Creating the repo itself, a README, and some of the basics.

Research: Figure out how I was going to create this project. My research focused on if a Python and SQL connection was possible and feasible, how I was going to create a GUI, and which GUI library I was going to use. I found a few good websites and videos to emulate for my own needs.

Create Basic GUI: The task was to create something small and simple so I could start filling out the rest of the project. I struggled with this part of the process, taking up the vast majority of my time. The abundance of time spent on creating the GUI caused me to cut many of the fun features I was hoping to include in this project. The SQL database, more metrics, plots, user logins, etc. Once I chose Tkinter, I felt like I did enough research to accomplish this task in minimal time. But everything I did seemed to go wrong. I started building the project the wrong way, then corrected it to another wrong way, and did that dance a few times until I finally stumbled on something that worked. I also spent too much time with little details that did not matter. Like colors and fonts, stuff that should come later, I was far too concerned with at the beginning, wasting a lot of time.

Creating Statistics: Programming the non-GUI stuff was easy, granted there was not much of it and it was nothing complex. I used an Excel file to store the betting history, then read that into a Pandas DataFrame to play around with in the program. I am familiar with both of those which made this part easy. Given more time, I would have switched the Excel file to a SQL database, just to see how that worked. I am sure it may not be the most optimal solution, but I imagine it would be fun to try.

**Build Evidence**

I did not collect enough of this throughout the project, as I struggled to produce many working builds. When I did produce something good, I did not think to document it.

Working window. Got something to show up on the screen, did not have any functionality.

A screenshot of a computer

Description automatically generated

Something closer to my mockups. The background colors hear tripped me up for a while. Same goes for the size of the buttons, as I could not figure out how the grid method worked with the buttons.

A screenshot of a computer menu

Description automatically generated

Started to figure out the grid method a bit more, but was still confused by a lot of the tkinter methods. The color behind the buttons when using a grid layout, I never figured out. I could not change it from gray as seen below to any other color, no matter what I tried. Wasted a lot of time on that.

A screenshot of a computer

Description automatically generated

**Submitted Artifacts**

See my repository for my code and documentation, but much of this is not applicable to my project as I did not opt to contribute to an open-source project.

**QA Strategy and Evidence**

Again, this was not me contributing to an open-source project, but I did not do a good job documenting my Quality Assurance strategy. In fact, there is no documentation other than my old code.

**Plan Updates**

Initial Plan: Bet tracking application programmed in Python using Tkinter library to build a GUI. Data to be stored in a MySQL Server. Potentially create an interactive dashboard for view metrics, similar to Tableau. Complete work over one month, spending around 6-10 hours/week working on the project for a total of 24-40 hours. Keep workload balanced over the month, not working 8-hour days, but contributing a little each day.

Outcome: Bet tracking application programmed in Python using Tkinter library to build a GUI. Data stored in an Excel file. Completed work over the month, with the majority of work being completed over the final few days, for a total of 39 hours, 26 coming in the final week.

This project should have been started much earlier. I continued to push it to the side. I initially felt like there was a lot of time, and it wasn’t the most pressing assignment on my plate. However, in February and March, I began to feel overwhelmed by how big of a task it would be. That lead to a big mental block in starting the project, which I didn’t start until the end of March/beginning of April. Given more time, this project could have lived up to the ideas I had. However, with the limited time, many changes were made to the plan.

I switched from a MySQL Sever to an Excel file as it was easier, it was something I had done before, and would help me complete the project on time.

I did not stick to my already short timeline as this month was busy with my other courses, my work, and extracurriculars. I keep prioritizing other tasks over this assignment, instead of finding the time to contribute to this project. That led to me putting in much of the work in a one week span as opposed to the entire month.

Experiences and Recommendations:

Challenges:

Finding good resources

I struggled to find good resources, particularly for tkinter. That library did not have the best documentation or easy-to-use instructions. That led me to take way too long to figure out how to work around problems. Maybe other libraries have better resources, but I should have done more thorough research beforehand. Perhaps building a few smaller, simpler projects before starting this one.

Mental Block

When I had free time early in the semester, I pushed this assignment off as it was due at the end of the year. It didn’t seem super urgent, I could go have a fun weekend instead of working on it. Towards the middle of the semester, once I had my idea, I kept making it larger than it was. Building it up to be an incredibly difficult task that would take forever to accomplish. So, I made excuses, worked on other assignments, and found other ways to spend my time instead of starting on this project. Once I started, it didn’t feel as bad. Once I came up with my GitHub Project, laid out my tasks, and built my plan of attack it felt much more digestible. Starting on that earlier would have helped me realize it is doable.

Knowing my Limits

I should have known that my project wouldn’t be perfect. I am rusty coding-wise and have not made anything like this before, it probably won’t be the prettiest or most perfect project. I set some standards too high initially, causing me to get stuck on silly little things like fonts and colors. I spent hours trying to figure out when I could have made much more progress and implemented more cool features. For future projects, I should start small, build from there, then make it pretty at the end.

Learning:

GUIs

Tkinter was a good introduction to these kinds of libraries. I didn’t love it, but at least now I know what to look for and what not to look for. It was nice to finally play with something like that, see how it works, what is possible and what isn’t. I feel more comfortable making GUIs now, so I’ll try to work that in to future projects.

Future

I want to keep exploring this project. Maybe a rework into a website would be cool, learn how to make something like this in HTML and CSS. Maybe that will be more flexible than Tkinter, I have to imagine it will.