**Ethan Coyle and Jonathan Hogan**

**Instructor- Dr.Saikat**

**CMPS 4143 Comput. Methods**

**Project Final Report**

Project Name (*or a top page*)

1. **Introduction (*More details; not just like your proposal*)**
   1. **Goal:**

The Goal that we ad in mind whenever we first started this project is to create a more in depth program simulation that could be useful related to real world events. By creating a Lottery Simulator that simulates exactly like in real world, not only us, but anyone else will be able to see exactly how everything works when it comes to playing the lottery and see everything that goes into the lottery when they decide to go down to the store and pick out a ticket.

With this hypothesis in mind, we decided that we were going to create a simulation based off of this and see it first hand in action. The Benefit of doing out program would help people gain a more in depth insight to the inner working of this massively played game from the confort of their own home without spending a single dime. Also, by creating this project, we gain a more knowledgeable outlook ourselves as to tkinter works in python as well as seeing the odds of winning the grand prize in the lottery and how they stack up to more than the odds of change.

* 1. **Problem Statement:**

**Why is it hard?**

This program is not necessarily hard. Although, to create an interactive simulator that will model those of a real life lottery game with the lottery numbers and the winning number and have all of the inner workings be accurate as well with the number of balls, the range of the lottery ball numbers and those generated is one of the key aspects of getting one part of the true to life simulation. Secondly, Since Neither Jon nor myself have worked on creating a program utilings a graphical user interface, that presented a giant obstacle for us to try and over come throughout the assembly part of our program.

**How hard is it?**

This project on a scale of 1-10 in complexity would probably be ranked as a 7 in complexity. The reason behind this assessment is that some part of creating a working simulation that simulates a lottery similar to the one that is being implemented everywhere is tedious and there can be no overlapping of the winning numbers and the simulator must accurately utilize the numbered balls accurately as well.

1. **Contributions: (*More details; not just like your proposal*)**

* WE split up our program initially into two parts for each of us to work equivalently Part 1 was setting up the graphical user interface and all the associated components.Part 2 was to create the interface to display the numbers whenever the buttons were pressed and updated the numbers inside of the labels accordingly. The Contributions Made by Jon on this project was to set up the number generator using the definitions and to ensure that the numbers would not repeat and be updated every time the reste button or generate buttons were pressed. After the buttons and gui interface was implemented, ensured that the displays displayed true numbers within the range of a real life simulation. My Contributions for this project was to implement the gui interface and set up the labels and buttons to interact with jons number generator for both of the winning numbers and the user drawing. I was also in charge of setting up the placement of everything inside of the gui window to create a user friendly display in which the user can easily see everything that is going on in the window and fully interact with everything in the window.

1. Related Work (*More details; not just like your proposal*)

Base off of everything that we researched, there has been a lot of programs that are being designed in python. However, the scale of this programs are merely random number generators that are designed to make it easier for the user to potentially catch a match. That is not how the lottery works in real life and we didn’t find hardly anything about anyone using a graphical user interface to create a whole lottery simulator so we decided to give it a go. While out program might be on the cusp of being somewhat small. What we have created has opend the possibilities of so much more. Not only for that fact that we were able to create a working simulation but with this basis in mind, we plan to potentially in the future devise and work on a next step for out program that will add to our program where we make this program even more immersive in such like a game in an app store based on what we have only , with that we have both been talking about trying to take our program to the next level there and devise a way to determine if there is a way to determine is we can 100 percent guess the random numbers. This topic has not had much research on it and it would be merely academic curiosity if we could devise this in our next stage but it has been a topic that both Jon and I have been curious about because there are a few articles stating that it might be potentially possible. But our program adds to what others have done by creating a very well structured and immersive interface in which the user can easily maneuver around and see how the lottery is played versus a bar bones example that isn’t a true to life immersive experience that we found in 90 percent of the past topics done on this topic.

1. Methodology (*More details; not just like your proposal*)
   1. **What approach am I going to try?**

at first, we approached this program with an open mind. We had to do research into seeing how the lottery game actually works on the large scale from the drawing numbers and the user drawings. We then determined the range of numbers of each of the balls both drawn and winning numbers.the technology that we used was using the tkinter graphical user interface in python where a window is created and all of our items are contained within. There was not really any formulas that we used aside from us having to make sure none of the drawn numbers were repeated from the random number generator. In conclusion, after our program was done we realized how astronomically low the chances of all the numbers on the winning numbers and the user numbers matching up especially with the range of numbers in which each of the balls range.

* 1. Why do I think it will work well?

WE originally thought that the going would be pretty rough especially deailing with creating a program in a language which neither one of us had much experience with at first, and also with creating our simulation using a graphical user interface in that language as well we expected to have problems all throughout.

* 1. Details of your used method; possibly flowchart or figure to illustrate your workflow

The methods that we used in creating our simulation was all pretty much wrapped around the tkinter interface in python. We had to create a whole interface that would allow the user to click an option on the screen and a result would happen inside of the labels. This part posed a very challngeing aspect of this where whenever the buttons for the options were pressed, the random number generator would have to update with every single one of the drawing numbers in the list. Also, we had to maintain the integrity of the drawing numbers where there were no repeats in any of the labels for the drawing because in a real world lottery game, there are no repeated numbers when a drawing is drawn.

* 1. What steps (task list) are required?

The steps required for this was first, to set up the graphical user interface. The user interface was the backbone to our program. Without this, we could have created a simulation for the lottery but it would be very mundane and boring and would not accurately depict what we wanted it to display and interact with the user. After the window was created, we needed to create buttons and labels that would interact with one another which would have the buttons as the options for the user to pick from and the labels would hold all the values similar in a lottery ticket. The next step was to implement the random number generator where the numbers that were generated would be placed inside of the labels and would refresh to new set everytime the buttons were pressed as well. This posed to be very challenging because everything needed to be updated accordingly whenever the buttons were pressed in a linear manner(the button is pressed, the label updates) After all of this was doen working, we just had to clean up the program and add a background for out program to make it more visually appealing.

* 1. Which of these steps is particularly hard?

The hardest part of this was getting the buttons, labels and the numbers to work well together and update accordingly without any delay and properly disaply to the screen. The part of this we had when creating this project was the labels. At first, we had issues with the placement of the labels not being where we wanted them to be. The next problem we had was whenever the buttons were pressed, the display values were not working. We got the numbers to work but they all just printed out the same number for all of the labels. Finally, after working with the number generator, we finally got the numbers to update properly and properly integrated with the labels and update whenever we pressed a button to update them.

All of our steps in the end have worked out the writing of the program turned out to be more challenging on getting everything to integrate properly and led to a successful build. With the simulation compiling and doing everything that we were wanting it to do, we can conclude that given our proposal for the program, our simulation was a success and it does model the variations that are shown in a real world lottery.

1. **Results and discussion**

True to our proposal, our biggest issue overall was dealing with tkinter GUI.

* 1. We experienced a lot of difficulties with this graphical user interface. The first issue we had was creating the labels and buttons. While this was practically new to us, learning the labels and tkinter buttons was problematic because at first, nothing was working. We finally figured out how to get the buttons to work and update the results in the labels which showed the drawing balls and we finally figured out how to get the labels to actually display the results whenever the buttons were being pressed
  2. Another issues we experienced was the random selection of balls we couldn’t quite work out why they were repeated numbers sometimes which in the real lottery, the numbers that are drawn never repeat so that posed a temporary challenge to both of us. We finally figured out how to get it where none of the values were repeated in either the drawn numbers or the winning numbers
  3. Thirdly, out main issue was the placement of the labels and buttons, we had to work out the placement issues to properly display in the position that we wanted them to be by specifying the x-axis and the y-axis rather than padding to left or right.

1. **Conclusion:**
   1. Our final analysis from accomplishing we would say that our project from start to finish was a success. We successfully created a program that is efficient and accurately simulates how a real life lottery program would work. From the numbers set to run to the updated number of balls and the range of those numbers in a real world scenario, to the drawing and comparison to the winning number, our program, despite minor challenges presented a successful outcome. A person can try multiple times to try to match any of the numbers from the generated ticket and the winning number and see just how astronomically low the chances are that all the numbers will match up.
   2. **What are the future directions; scopes of this project/work**

For future direction on our program, both of us are wanting to possibly extend this project out now that we have successfully created a life simulation and used a graphical user interface with out lottery simulation to potentially develop this into a full fledged game or some other sort of program in which can be modelled inside of an app store where anyone is able to download it and get hands on experience with this program

1. **Reference:**

Grauschopf, Sandra. “Can Software Help You Predict the Lottery?” *LiveAbout*, LiveAbout, 29 Jan. 2021, <https://www.liveabout.com/can-this-software-help-you-> predict-the-lottery-4174474.

“How to Write a Simple Lottery Program in Python.” *Develop Paper*, 10 May 2020, https://developpaper.com/how-to-write-a-simple-lottery-program-in-python/.

Lottery, Texas. “$253 Million.” *Texas Lottery | Home*, https://www.texaslottery.com/.

Mohammadi, A., and I. Nakhaei Kamal Abadi. “Heuristic Algorithm for Solving the Integer Programming of the Lottery Problem.” *Scientia Iranica*, No Longer Published by Elsevier, 15 May 2012, https://www.sciencedirect.com/science/article/pii/S1026309812000909.

“Python 3.10.0 Documentation.” *3.10.0 Documentation*, https://docs.python.org/3/index.html.

Sohail, and Admin. “Basics in Python with a Simple Lottery Number Game.” *Helical IT Solutions Pvt Ltd*, 31 Jan. 2019, <https://helicaltech.com/basics-python-lottery-> game/.

“Tkinter - Python Interface to TCL/TK¶.” *Tkinter - Python Interface to Tcl/Tk - Python 3.10.0 Documentation*, https://docs.python.org/3/library/tkinter.html.