**Adrian’s CS151 Project 2:**

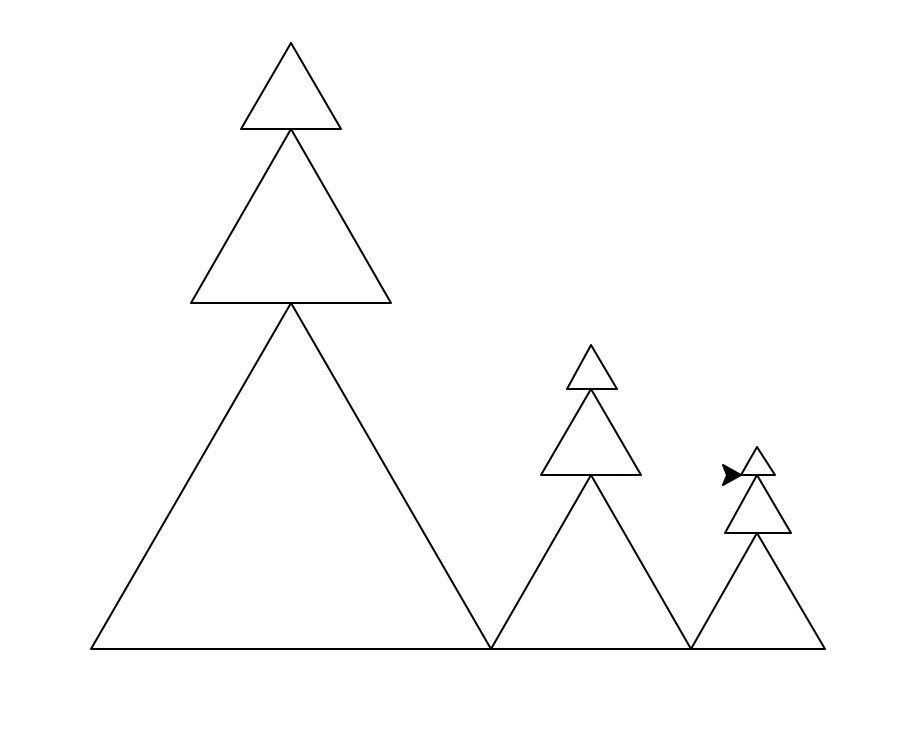
**A shape collection**

# Abstract

For this project, we had to create a Python file containing functions that produced different shapes and shape conglomerates using feasible commands from Python’s pre-installed library called turtle. The Python file created would then serve as a library for a main file to draw shapes and scenes on the turtle's window using my named turtle. During the lab, that was a precursor to this project, I had to create two, triangle-drawing functions that helped in the process of making a final triangle stack function that used my named turtle to draw triangle stacks in different locations of the turtle window. During the project, I created and copied from the lab several functions to draw simple shapes (rectangles, curved lines and circles, red-colored hearts, and triangles). I then used these functions to create more complex shapes (a red-colored diamond, playing cards consisting of diamonds or hearts within a rectangle, and a green playing board consisting of two half circles and a rectangle). To create and then use these functions both in my lab file and my main code file (which called functions from my shape library file), I used several Python turtle commands to create a turtle window to use my turtle to draw on, lift my turtle’s pen up and place it down, make it go to a specific spot on the turtle window, make the turtle turn a certain number of degrees from its current position, make the turtle face a certain direction, make it go forward or backward, and make it fill in created shapes. In combination with these commands, I used loops in order to limit the number of lines of code needed. However, this was not always possible. I also used a scaling parameter to fit the different shapes within the same scene. In addition, I used penup and hide turtle functions in order to not obscure my final gaming scene with my turtle’s pen. Besides coding, I also needed to use some basic algebra to help calculate where I wanted to place different shapes in relation to others. Ultimately, all of the lines of code I created were able to produce the images I expected.

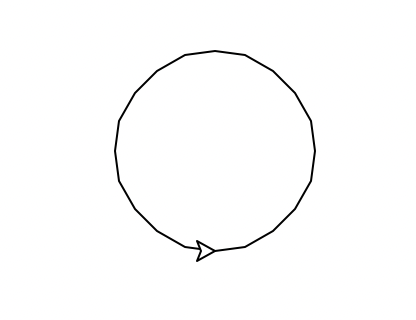
# Results

First pattern



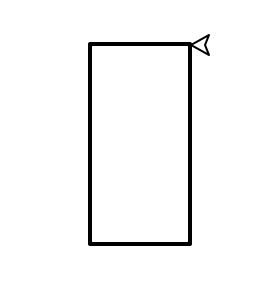
For my lab, I created three triangle-stack “trees” using a turtle with Python in Visual Studio Code with the turtle beginning to draw in the center of the screen and then moving from the center of the ending location (top triangle of largest tree) to different x and y positions to draw the second and third triangle stacks beside the first triangle stack. The main component of my code used for this image was a triangle stack function that called code from a previous triangle function and used goto commands to create three triangles that made up the stack. The triangle function called various codes from a goto function and a different triangle function and added a scaling parameter in order to draw triangles in different locations of the turtle window with different sizes. The goto function consisted of a goto command in addition to a penup and pendown command to reduce lines of code and so that when goto was called a line would not be drawn by the turtle while it moved to a new location. The triangle function consisted of a loop of forward and left turns that made the different sides of the triangle.

First gaming shape



For my first gaming shape, I created a circle using a turtle with Python in Visual Studio Code with the turtle beginning to draw in the center of the screen. The main component of my code was a function that used turtle’s circle creating function and took in various parameters (x, y, radius, extent, tilt, outside\_color, and inside\_color) to create curved lines (extent of less than 360˚) or circles with different edge and fill colors at any location on the turtle window and with any radius.

Second gaming shape



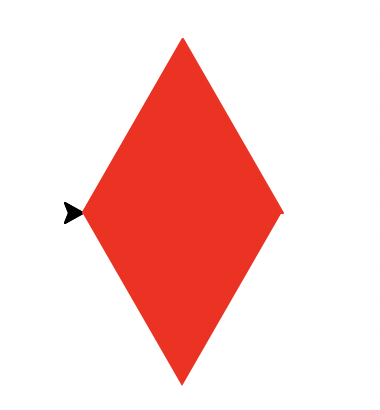
For my second gaming shape, I created a rectangle using a turtle with Python in Visual Studio Code with the turtle beginning to draw in the center of the screen. The main component of my code was a function that took in various parameters (x, y, width, height, scale, outside\_color, inside\_color) and used a loop function to create the four sides of the rectangle. Although the image has a black edge and white fill color in addition to these side sizes, the parameters allow for all of these aspects to be changed.

Third gaming shape



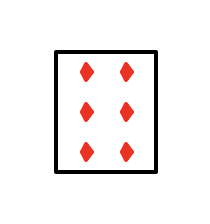
For my third gaming shape, I created a red colored heart using a turtle with Python in Visual Studio Code with the turtle beginning to draw in the center of the screen. To create this shape, I used a function that consisted of a left turn to make my turtle face the correct initial direction. Then in order of lines of code, the function consisted of a forward command to draw the straight portion of the left side of the heart, a loop of right turn and forward commands to draw the curved part of the left side of the heart, a left turn command to head in the right direction to draw the right side of the heart, and finally the same loop and forward command to draw the right side of the heart. The function also took in several parameters (x, y, and scale) to change the size of the heart and its placement on the turtle window.

First gaming-themed compound shape



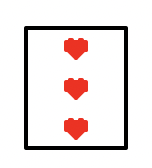
For my first gaming-themed compound shape, I created a red-colored diamond shape using a turtle with Python in Visual Studio Code. To create this compound shape, I created a function that took in the same parameters as the triangle function used in the lab to help move the diamond around on the turtle window and scale its size. The function consisted of a block of code to change the pen and fill color for the diamond and a block of code that called from the triangle function twice with a command to turn 60˚ in between them so that the triangle bases would be facing each other rather than the triangles being stacked on top of one another.

Second gaming-themed compound shape



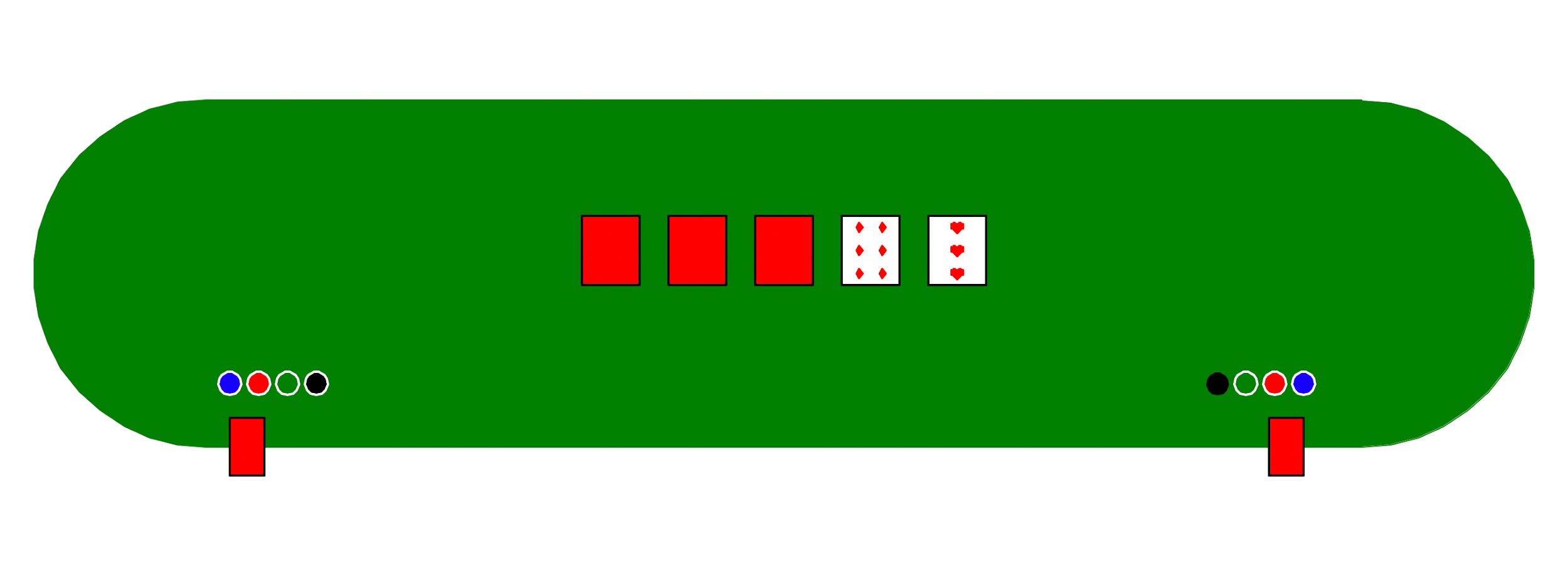
For my second gaming-themed compound shape, I created a six of diamonds playing card using a turtle with Python in Visual Studio Code. To create this compound shape, I created a function that took in the same parameters as both the rectangle creating and diamond creating function and called from both these functions with different x and y values to fit the diamonds within the card.

Third gaming-themed compound shape



For my third gaming-themed compound shape, I created a three of hearts playing card using a turtle with Python in Visual Studio Code. To create this compound shape, I created a function that took in the same parameters as both the rectangle creating and heart creating function and called from both these functions with different x and y values to fit the hearts within the card.

Gaming-themed scene



After creating the above shape and compound shape functions in the shape library file and creating functions and code in my main file to call from my shape library file and draw them, I created a gaming-themed scene of a poker card-game position. First, I called one board function, five card functions, eight circle functions, one, heart card function, and one diamond card function in the main code. By commenting out different functions, conducting calculations, and doing some final tweaks, I debugged drawing errors and ended up with the above scene. Then, I created a gaming1 function, and moved all of my main code for the above scene to within that function. I finally named all needed parameters, and called the gaming1 function in my main code with zeros as parameters because they were already set within the function.

# Reflection

During this project, I expanded my knowledge about Python turtle commands and using different Python files to draw scenes from the previous project. I also improved and practiced my organization of functions and code, which helped a lot with the more complex functions that I used during this project in the process of debugging and critiquing. Finally, I learned how to use the terminal to help move files from my desktop to my project folder within google drive, which made organizing files faster than clicking and dragging within finder.

# Acknowledgements

1. I used this article to help create a function for the heart shape (https://medium.com/analytics-vidhya/draw-heart-with-python-using-turtle-7bd8b9ef31d9)
2. I consulted lecture notes to help write loops and remember various commands that I used.