Eduardo Granados

Bullseye Game

The library used for this program was Turtle. The idea behind this program was because I enjoy playing video games, so I thought about a simple game that places an marker on a target at random. At first this seemed like it would be somewhat some challenging, because I wanted to have the target moving, and have the user click on the target and see where it lands. Although it got complicated, and there were somethings that the library didn’t offer, and to get some of the values I needed, that math couldn’t be performed or I just didn’t know how implement it.

One way this program can be used is like the pinball game that used to be on Windows machine. Something very simple and easy to get the mind off things. Especially during these troubling times. Another use for this is with machine learning. Maybe a model could be trained to shoot at the target, and the more its trained the accurate it gets.

|  |  |
| --- | --- |
| for i in range(160, 0,-16):  rings.penup()  if count != 11:  rings.begin\_fill()  rings.pendown()  rings.penup()  rings.goto(0, -i)  rings.pendown()  if count == 1 or count == 2:  rings.color("black", ringColors[count])  elif count ==3 or count == 4:  rings.color("white", ringColors[count])  else:  rings.color("black", ringColors[count])  rings.circle(i)  rings.penup()  rings.home()  rings.end\_fill()  count += 1 | ringColors = {  1: "white",  2: "white",  3: "black",  4: "black",  5: "blue",  6: "blue",  7: "red",  8: "red",  9: "yellow",  10: "yellow"  } |

The piece of code above is what creates the target outline. At the top we begin with a **for loop** which is the radius of the circles. Next we doing into a **if statement**, and controls what color each ring should be filled and its outline color. This is done by using a counter, and we start on the outside or the biggest circle and work in. Since the first ring it worth 1 point, we can assign the value to the color. So when we reach a certain value, make it that color and get the color from a dictionary named **ringColors**.

Documentation link:

https://docs.python.org/3.3/library/turtle.html?highlight=turtle