CS2513 Assignment 2
Submit via Canvas by 2359hrs 22nd November 2024
Assignment is worth 5 marks

Task:
Create a simple game. The game should start by creating a blank canvas in a tinter frame. Add a blank canvas and add a mouse event handler to this. When a user clicks on this canvas, an oval should be drawn at a random position within the area of the canvas. This should happen at a random time (determined using sleep and random (see below). The amount of time should be between 5 and 10 seconds.

Once the oval appears, the user should click on the oval as quickly as possible.

If the user manages to click on the oval, your program should print ‘hit’ to the console. If they miss you should print ‘miss’. You should also track the time at which the oval appears, and the time at which the user clicks. Print the difference between these to the console also.

Your code should be implemented using the principles of OOP that we have studied in lectures.

Helpful Facts:
Drawing shapes on a tinter canvas:
You can add a canvas as a widget to a tkinterframe. Furthermore, you can click on the canvas to generate a click event that can be handled.

```
root = tk.Tk()
myCanvas = tk.Canvas(root, bg="white", height=300, width=300)
myCanvas.bind("<Button-1>", mouse_function)
```

Drawing an oval is done as follows:

```
myCanvas.create_oval(randx, randy, randx + 100, randy + 100, fill=‘red')
```

This creates an oval within a bounding box. The top left x-coordinate is given by x0, while x1 is the bottom right x-coordinate. We choose to make a circle, so we specify that the bottom x and y coordinates are 100 beyond where we were with the x0, y0 values.

In this case, you would add a function, mouse_function that takes an event. You can access the x and y components of the event.

```
def mouse_function(event):
print(event.x, event.y)
```

Random Numbers:
You can generate a random number using the random module. You call the random.randint(v1, v2) method where v1 and v2 are the upper and lower bounds of the range from which the number is randomly chosen.

Getting current time:
Using the datetime module, you can get the current time using currenttime = datetime.now()

You can get the time taken for something by subtracting two such values:

```
currenttime1 = datetime.now()
currenttime2 = datetime.now()
print(currenttime2 - currenttime1)
```

Putting your code to sleep:
You can have your code's execution pause for a predetermined amount of time by using the time module. When you time.sleep(10) your code will pause its execution for 10 seconds.

Is the point in the circle:
You can have a rough estimate by determining if the mouse event is within the bounding box.

A more accurate version is as follows: we can determine if a point is in a circle if it the distance from the centre of the circle is less than the radius of the circle. We calculate the centre of the circle as:

xcent = x0 + (x1 - x0)/2, ycent = y0 + (y1 - y0)/2

The radius is given by:

radius = (x1 - x0)/2

And distance between two points is sqrt((x1-x0)^2 + (y1-y0)^2) (you can import math for these functions).

Notes:

1. All work must be your own. You are bound by the plagiarism slide shown in Lecture 1. If in doubt ask Cathal. You must write all of the code – code generated by AI, borrowed from someone else, downloaded from the web, etc, will not be accepted as your own.