**Name: Session:**

**Programming I**

**Alternative Graphics Packages**

**Lab Exercise 1.8.2025**

When it comes to graphics, Pygame is not the only game in town. On the server you will find an alternative graphics module (graphics.py). In order to use this module, remember that you need to place it in the same folder as the source code that wishes to use it.

Here is a simple little program that you can use to test it out.

from graphics import \*

def main():

win = GraphWin(‘Circle’, 640, 480)

shape = Circle(Point(200, 200), 30)

shape.setOutline("red")

shape.setFill("red")

shape.draw(win)

for i in range(10):

p = win.getMouse()

c = shape.getCenter()

dx = p.getX() - c.getX()

dy = p.getY() - c.getY()

shape.move(dx, dy)

win.close()

main()

Notice that this program defines a function main() and then calls that function. You should get used to having a function called “main”.

1. Alter the program above to make it do the following:
   1. Make it draw squares instead of circles
   2. Have each successive click draw an additional square on the screen (rather that moving the existing one)
2. An archery target consists of a central circle of yellow surrounded by concentric rings of red, blue, black, and white. Each ring has the same “width” which is the same as the radius of the yellow circle. Write a program that draws such a target.

Now check out this program:

# Description: Interactive graphics program to draw a triangle

from graphics import \*

def main():

win = GraphWin("Draw a Triangle", 640, 480)

win.setCoords(0.0, 0.0, 10.0, 10.0)

message = Text(Point(5, 0.5), "Click on three points")

message.draw(win)

# Get and draw three vertices of triangle

p1 = win.getMouse()

p1.draw(win)

p2 = win.getMouse()

p2.draw(win)

p3 = win.getMouse()

p3.draw(win)

# Use Polygon object to draw the triangle

triangle = Polygon(p1,p2,p3)

triangle.setFill("peachpuff")

triangle.setOutline("cyan")

triangle.draw(win)

# Wait for another click to exit

message.setText("Click anywhere to quit.")

win.getMouse()

win.close()

main()

1. Write a program that allows the user to draw a line segment and then displays some graphical and textual information about the line segment.

**Input**: 2 mouse clicks for the endpoints of the line segment

**Output:** Draw the midpoint of the line segment in cyan

Draw the line

Print the length and slope of the line

**Formulas:**



1. Write a program that displays information about a rectangle drawn by the user.

**Input:** 2 mouse clicks for the opposite corners of the rectangle

**Output:** Draw the rectangle

Print the perimeter and area of the rectangle

**Formulas:**



Finally, check this out:

# Program to convert Celsius to Fahrenheit using a simple

# graphical interface.

from graphics import \*

def main():

win = GraphWin("Celsius Converter", 400, 300)

win.setCoords(0.0, 0.0, 3.0, 4.0)

# Draw the interface

Text(Point(1,3), " Celsius Temperature:").draw(win)

Text(Point(1,1), "Fahrenheit Temperature:").draw(win)

input = Entry(Point(2,3), 5)

input.setText("0.0")

input.draw(win)

output = Text(Point(2,1),"")

output.draw(win)

button = Text(Point(1.5,2.0),"Convert It")

button.draw(win)

Rectangle(Point(1,1.5), Point(2,2.5)).draw(win)

# wait for a mouse click

win.getMouse()

# convert input

celsius = eval(input.getText())

fahrenheit = 9.0/5.0 \* celsius + 32

# display output and change button

output.setText("%0.1f" % fahrenheit)

button.setText("Quit")

# wait for click and then quit

win.getMouse()

win.close()

main()