

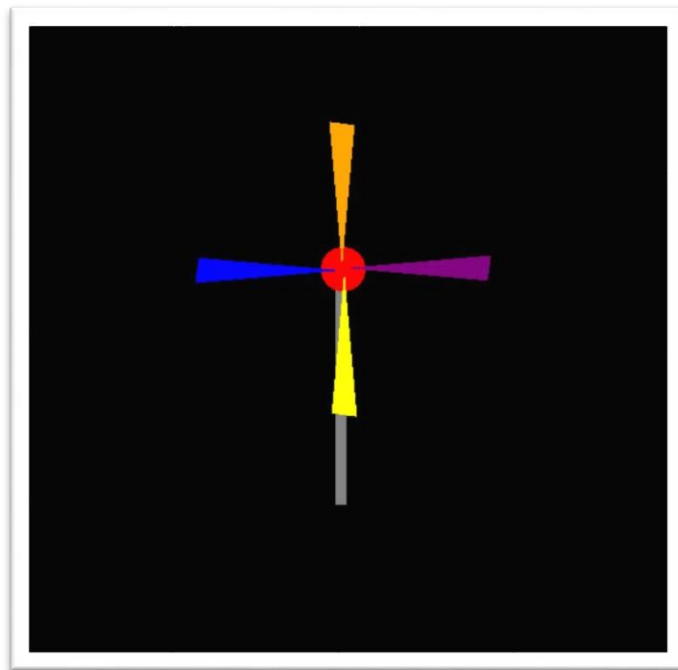
## 2. Windmill project in pyleap

Date: 17/01/2024

### Explanation

This Python code utilizes the `pyleap` library to create a graphical animation representing a windmill. The graphical elements, including a background rectangle, text, and various shapes to depict the windmill components, are defined. These components include a rectangular body, a circular midpoint, and four triangular handles of different colors. The handles are positioned at different angles to simulate rotation. The windmill's rotation speed is read from a file specified by `speed\_file\_path`. The `read\_speed` function attempts to retrieve the speed from the file, ensuring it falls within a valid range of 1 (Low) to 10 (High).

The main animation function, `windmill`, repeatedly draws the graphical elements and updates the rotation angles of the handles based on the read rotation speed. The animation loop is established with the `repeat` function, and the entire application is executed with `run()`. In essence, this code creates a visually appealing windmill animation where the rotation speed can be dynamically adjusted through an external file.



### Steps

1. Importing Modules

```
from pyleap import *  
import time
```

The code imports necessary modules from the pyleap library for graphical elements and animation. The time module is also imported, though it's not explicitly used in the provided code.

## 2. File Path and Speed Limit

**speed\_file\_path = ' '**

This variable holds the path to a file (speed.txt) that presumably contains the rotation speed of the windmill. The windmill's rotation speed is expected to be an integer between 1 and 10.

## 3. Reading Windmill Rotation Speed from File

**def read\_speed():**

**# ...**

This function attempts to read the windmill rotation speed from the specified file (speed.txt). It checks if the value is within the allowed range (1 to 10) and handles various error cases.

## 4. Drawing Windmill and Updating Rotation

**def windmill(dt):**

**# ...**

This function is responsible for drawing the graphical elements of the windmill and updating the rotation of the handles based on the windmill's rotation speed.

## 5. Animation Loop

**repeat(windmill)**

This line repeats the windmill function, creating an animation loop.

## 6. Running the Application

**run()**

Finally, this line executes the pileap application, running the graphical animation.

### Code:

```
# Importing necessary modules from the pyleap library
from pyleap import *

# Setting the size of the window
window.set_size(600, 600)

# Creating graphical elements for the windmill project
bg = Rectangle(0, 0, 600, 600, 'black') # Background rectangle
#txt = Text("Windmill project in pyleap", 130, 540, 20, "black") # Text
element

body = Rectangle(285, 160, 10, 200, 'gray') # Body of the windmill
mid = Circle(292, 372, 20, 'red') # Circle representing the midpoint of the
windmill

# Handles of the windmill
handle1 = Triangle(293, 374, 392, 460, 403, 440, 'purple')
handle1.rotation = 0
handle1.set_anchor(292, 372)

handle2 = Triangle(293, 374, 392, 460, 403, 440, 'orange')
handle2.rotation = 90
handle2.set_anchor(292, 372)

handle3 = Triangle(293, 374, 392, 460, 403, 440, 'blue')
handle3.rotation = 180
handle3.set_anchor(292, 372)

handle4 = Triangle(293, 374, 392, 460, 403, 440, 'yellow')
handle4.rotation = 270
handle4.set_anchor(292, 372)

# Path to the file containing windmill rotation speed
speed_file_path = 'C:/Users/z004tu0x/Desktop/IT_Group/2.
Graphical_Wind_Turbine/speed.txt'

# Function to read windmill rotation speed from the file
def read_speed():
    try:
        with open(speed_file_path, 'r') as file:
            speed_str = file.readline().strip()
            if speed_str:
                speed = int(speed_str)
                if 1 <= speed <= 10:
                    return speed
            else:
                print('Please make sure to enter the Speed limit between
1 - Low and 10 - High')
```

```

        exit()

    else:
        print("Error: Empty line or non-integer value in the file.")
        return None
except Exception as e:
    print(f"Error reading speed from file: {e}")
    return None

# Function to draw the windmill and update rotation based on speed
def windmill(dt):
    bg.draw()
    #txt.draw()
    body.draw()
    mid.draw()
    handle1.draw()
    handle2.draw()
    handle3.draw()
    handle4.draw()

    speed = read_speed()

    if speed is not None:
        # Update handle rotation based on speed
        handle1.rotation += -speed
        handle2.rotation += -speed
        handle3.rotation += -speed
        handle4.rotation += -speed

# Repeating the windmill function to create animation
repeat(windmill)

# Running the pyleap application
run()

```