ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ № 1

за курсом “Комп’ютерна графіка та анімація”

студента групи ПЗ-24у-1

Кондрачука Олександра Вадимовича

2024/2025 н.р.

1. **Постановка задачі**

Розробити прикладну програму (мова розробки за бажанням), яка працює з графічною бібліотекою (вибір за бажанням). Використовуючи графічні примітиви (не менше трьох різних примітивів) даної бібліотеки побудувати статичну картинку. На дану картинку додати динамічний об'єкт (примітив або комбінація примітивів). Передбачити анімацію даного елемента зображення наступним чином (одне на вибір):

- анімація реалізовується шляхом перерисовки всього вікна доданку;

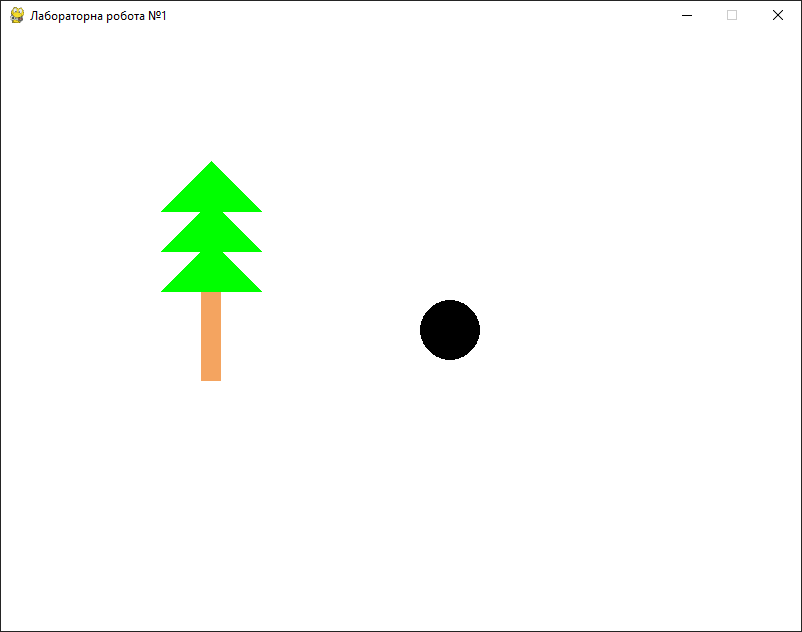
- анімація реалізовується шляхом перерисовки тільки тієї частини екрану, де розташовано динамічний об'єкт.

1. **Опис розв’язку**

Створив програмне вікно за допомогою мови програмування python та графічної бібліотеки pygame.

За допомогою pygame.draw.polygon створив 3 трикутника та за допомогою pygame.draw.rect створив прямокутник. Потім об’єднав ці фігури у ялинку підставивши потрібні координати. За допомогою pygame.draw.cicle створив коло та за допомогою sin та cos переміщав коло, додавши функцію плавності переміщення.

1. **Тестування програми**

****

1. **Додаток**

import pygame  
import sys  
import math  
  
SCREEN\_WIDTH, SCREEN\_HEIGHT = 800, 600  
FPS = 60  
  
WHITE = (255, 255, 255)  
BLUE = (0, 0, 255)  
RED = (255, 0, 0)  
GREEN = (0, 255, 0)  
BLACK = (0, 0, 0)  
BROWN = (244, 164, 96)  
  
CIRCLE\_RADIUS = 30  
CIRCLE\_MOVE\_CENTER = [400, 300]  
CIRCLE\_MOVE\_RADIUS = [50, 50]  
  
pygame.init()  
  
screen = pygame.display.set\_mode((SCREEN\_WIDTH, SCREEN\_HEIGHT))  
pygame.display.set\_caption("Лабораторна робота №1")  
  
circle\_pos = [100, 300]  
elapsed\_time = 0  
  
clock = pygame.time.Clock()  
  
def easing(x):  
    return 16 \* x \* x \* x \* x \* x if x < 0.5 else 1 - math.pow(-2 \* x + 2, 5) / 2  
  
def draw\_static\_scene():  
    screen.fill(WHITE)  
  
    pygame.draw.rect(screen, BROWN, (200, 150, 20, 200))   
    pygame.draw.polygon(screen, GREEN, [(210, 130), (260, 180), (160, 180)])  
    pygame.draw.polygon(screen, GREEN, [(210, 170), (260, 220), (160, 220)])  
    pygame.draw.polygon(screen, GREEN, [(210, 210), (260, 260), (160, 260)])  
  
def draw\_dynamic\_object():  
    pygame.draw.circle(screen, BLACK, circle\_pos, CIRCLE\_RADIUS)  
  
def move\_circle():  
    global circle\_speed, elapsed\_time, circle\_pos  
      
    elapsed\_time += 1 / FPS  
      
    if elapsed\_time >= 1:  
        elapsed\_time -= 1  
      
    percentage = easing(elapsed\_time)  
      
    circle\_angle = percentage \* 360  
    radians = circle\_angle \* math.pi / 180  
      
    circle\_x = CIRCLE\_MOVE\_CENTER[0] + CIRCLE\_MOVE\_RADIUS[0] \* math.cos(radians)  
    circle\_y = CIRCLE\_MOVE\_CENTER[1] + CIRCLE\_MOVE\_RADIUS[1] \* math.sin(radians)  
  
    circle\_pos = [circle\_x, circle\_y]  
      
      
  
while True:  
    for event in pygame.event.get():  
        if event.type == pygame.QUIT:  
            pygame.quit()  
            sys.exit()  
  
    move\_circle()  
  
    draw\_static\_scene()  
    draw\_dynamic\_object()  
  
    pygame.display.flip()  
    clock.tick(FPS)