*Министерство образования Республики Беларусь*

*Учреждение образования*

*«Брестский государственный технический университет»*

*Кафедра ИИТ*

Лабораторная работа №4

По дисциплине: «Современные средства вычислительной техники»

Тема:«**Анимированные спрайты**»

Выполнил:

Студент 3-го курса

группы ИИ-23

Романюк А.П.

Проверил:

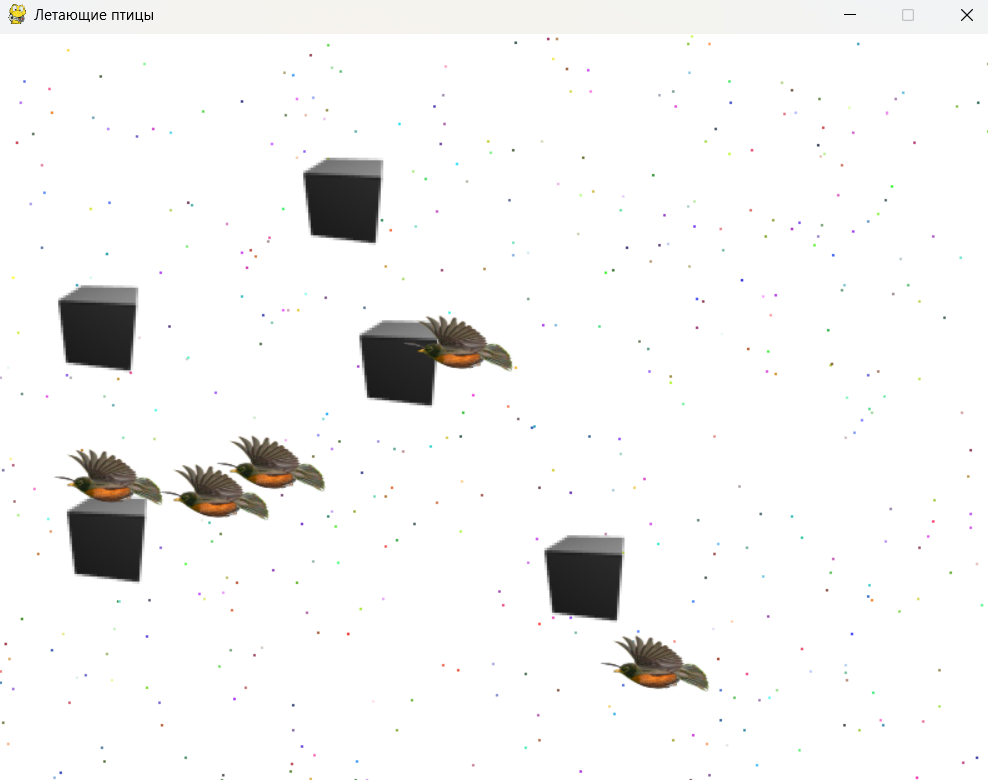
Кулеша В. И.

Брест 2025

**Цель:** освоить работу с анимированными спрайтами.

**Задание:** написать программу, которая рисует на экране анимированный спрайт; кроме анимации, спрайт должен также перемещаться по экрану.

**Ход работы**

import pygame  
import sys  
import random  
  
pygame.init()  
  
screen\_width = 800  
screen\_height = 600  
screen = pygame.display.set\_mode((screen\_width, screen\_height))  
pygame.display.set\_caption("Летающие птицы")  
  
bg\_r, bg\_g, bg\_b = 0, 0, 0  
color\_change\_speed = 1   
  
sprite\_sheet\_main = pygame.image.load('spritesheet.png')   
sprite\_sheet\_bird = pygame.image.load('robin.png')  
  
def load\_frames(sheet, rows, cols, scale=2):  
 frame\_width = sheet.get\_width() // cols  
 frame\_height = sheet.get\_height() // rows  
 frames = []  
  
 for row in range(rows):  
 for col in range(cols):  
 x = col \* frame\_width  
 y = row \* frame\_height  
 frame = sheet.subsurface(pygame.Rect(x, y, frame\_width, frame\_height))  
 scaled\_frame = pygame.transform.scale(frame, (int(frame\_width \* scale), int(frame\_height \* scale)))  
 frames.append(scaled\_frame)  
  
 return frames, int(frame\_width \* scale), int(frame\_height \* scale)  
  
frames\_main, main\_width, main\_height = load\_frames(sprite\_sheet\_main, 6, 5)  
frames\_bird, bird\_width, bird\_height = load\_frames(sprite\_sheet\_bird, 5, 5, 0.5) # Предположим 4 строки и 5 столбцов  
frames\_bird.pop()  
frames\_bird.pop()  
frames\_bird.pop()  
  
class FlyingObject:  
 def \_\_init\_\_(self, frames, width, height):  
 self.frames = frames  
 self.width = width  
 self.height = height  
 self.x = random.randint(0, max(0, screen\_width - width))  
 self.y = random.randint(0, max(0, screen\_height - height))  
 self.dx = random.choice([-3, -2, 2, 3]) # Случайная скорость по X (влево/вправо)  
 self.dy = random.choice([-2, -1, 1, 2]) # Случайная скорость по Y  
 self.current\_frame = 0  
 self.frame\_counter = 0  
 self.frame\_rate = 10 # Частота анимации  
  
 def update(self):  
 # Обновляем позицию  
 self.x += self.dx  
 self.y += self.dy  
  
 # Отскакивание от границ  
 if self.x <= 0 or self.x >= screen\_width - self.width:  
 self.dx = -self.dx # Меняем направление  
 if self.y <= 0 or self.y >= screen\_height - self.height:  
 self.dy = -self.dy # Меняем направление  
  
 # Анимация (смена кадров)  
 self.frame\_counter += 1  
 if self.frame\_counter >= self.frame\_rate:  
 self.frame\_counter = 0  
 self.current\_frame = (self.current\_frame + 1) % len(self.frames)  
  
 def draw(self, screen):  
 frame = self.frames[self.current\_frame]  
 if self.dx > 0:  
 frame = pygame.transform.flip(frame, True, False)  
  
 screen.blit(frame, (self.x, self.y))  
  
main\_objects = [FlyingObject(frames\_main, main\_width, main\_height) for \_ in range(5)] # Основной объект  
birds = [FlyingObject(frames\_bird, bird\_width, bird\_height) for \_ in range(5)] # 5 птиц  
  
num\_stars = 500  
stars = [(random.randint(0, screen\_width), random.randint(0, screen\_height),  
 random.randint(50, 255), random.randint(50, 255), random.randint(50, 255))  
 for \_ in range(num\_stars)]  
  
clock = pygame.time.Clock()  
running = True  
while running:  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 running = False  
  
 bg\_r = (bg\_r + color\_change\_speed) % 256  
 bg\_g = (bg\_g + color\_change\_speed \* 2) % 256  
 bg\_b = (bg\_b + color\_change\_speed \* 3) % 256  
  
 screen.fill((bg\_r, bg\_g, bg\_b))  
  
 for i in range(num\_stars):  
 if random.random() < 0.1: # Меняются случайные звезды  
 stars[i] = (random.randint(0, screen\_width), random.randint(0, screen\_height),  
 random.randint(50, 255), random.randint(50, 255), random.randint(50, 255))  
 pygame.draw.rect(screen, stars[i][2:], (stars[i][0], stars[i][1], 2, 2))  
  
 for cube in main\_objects:  
 cube.update()  
 cube.draw(screen)  
  
 for bird in birds:  
 bird.update()  
 bird.draw(screen)  
  
 pygame.display.flip()  
 clock.tick(30)  
  
pygame.quit()  
sys.exit()

**Вывод:** изучил базовые принципы программирования DirectDraw. Разработал программы, применяя полученные знания.