

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



ЛАБОРАТОРНА РОБОТА №4
З ДИСЦИПЛІНИ:
«Комп'ютерна графіка»

ВАРІАНТ №8

Виконав:

ст. гр. ІР-31

Касараба В.В.

Прийняла:

асистент. каф. КСА

Шпак О.І.

Львів 2021

ЗАВДАННЯ

Вивчити основні матричні перетворення на площині для побудови рухомих зображень.

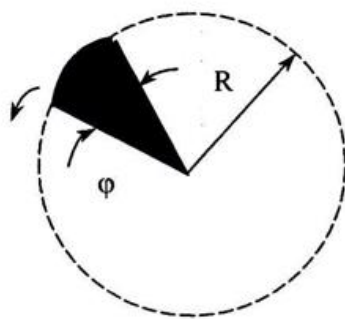
Параметри зображення задані в таблиці 1 в пікселях.

Написати програму мовою C++, що дозволяє сформувати на екрані комп'ютера рухоме зображення, використовуючи матричні операції.

Основні вимоги:

- 1) Всі завдання необхідно будувати по центру екрану.
- 2) Вивести результуючу матрицю руху для свого варіанту завдання, визначеного викладачем.

8



Рух сектора проти годинникової стрілки із зміною його кольору (сім кольорів).

Параметри: $R=80$; $\varphi=30^\circ$

Виконання роботи Python Code

```
import pygame as py
import pygame.gfxdraw

# define constants
WIDTH = 700
HEIGHT = 700
FPS = 200

# define colors
```

```

WHITE = (255, 255, 255)

GREEN = (0 , 255 , 0)
RED = (255, 0, 0)
BLUE = (0, 0, 255)
BLACK = (0, 0, 0)
VIOLET = (155, 148, 220)
PINK = (251, 96, 127)
ORANGE = (128, 0, 0)

# initialize pygame and create screen
py.init()
screen = py.display.set_mode((WIDTH , HEIGHT))
# for setting FPS
clock = py.time.Clock()

rot = 0
rot_speed = .1

def set_image_orig(color):
# define a surface (RECTANGLE)
    image_orig = py.Surface((400 , 400), py.SRCALPHA)
    # for making transparent background while rotating an image
    image_orig.set_colorkey(WHITE)
    # fill the rectangle / surface with green color
    image_orig.fill(WHITE)
    for r in range(200):
        pygame.gfxdraw.pie(image_orig,200,200,r,0,30,color)
    return image_orig

image_orig = set_image_orig(VIOLET)

# creating a copy of original image for smooth rotation
image = image_orig.copy()
image.set_colorkey(WHITE)
# define rect for placing the rectangle at the desired position
rect = image.get_rect()
x, y = WIDTH/2, HEIGHT/2
rect.center = (x, y)
# keep rotating the rectangle until running is set to False
running = True
while running:

    x, y = WIDTH/2, HEIGHT/2
    # set FPS
    clock.tick(FPS)
    # clear the screen every time before drawing new objects
    screen.fill(WHITE)

```

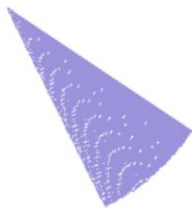
```
# check for the exit
for event in py.event.get():
    if event.type == py.QUIT:
        running = False

# defining angle of the rotation
rot = (rot + rot_speed) % 360
# rotating the original image
if 0 < rot < 50:
    color = PINK
elif 50 < rot < 100:
    color = GREEN
elif 100 < rot < 150:
    color = RED
elif 150 < rot < 200:
    color = BLUE
elif 200 < rot < 250:
    color = ORANGE
elif 250 < rot < 300:
    color = BLACK
else:
    color = VIOLET

image_orig = set_image_orig(color)
image = py.transform.rotate(image_orig, rot)
rect = image.get_rect(center = (x, y))
# drawing the rotated rectangle to the screen
screen.blit(image, rect)
# flipping the display after drawing everything
py.display.flip()

py.quit()
```

Результат виконання







Висновок:

На цій лабораторній роботі я вивчив основні матричні перетворення на площині для побудови рухомих зображень.

Для виконання роботи, я написав програму мовою Python, що дозволяє сформувати на екрані комп'ютера рухоме зображення, використовуючи матричні операції.

Код програми та результат роботи я вніс у звіт.