**Прізвище:** Йовбак

**Ім'я:** Андріанна

**Група:** КНСП-11

**Варіант:** 2

**Дата захисту:** 29.10.2020

**Кафедра:** САПР

**Дисципліна:** Методи та засоби комп’ютерного навчання

**Перевірив:** Андрущак Н.А.

**ЗВІТ**

до лабораторної роботи №2

на тему «Використання графічних модулів в Python»

**Мета роботи:** ознайомитися із можливостями мови програмування Python для графічного представлення та побудови фігур на основі модулів комп'ютерної графіки, використання графічних операторів, запис та читання з файлів.

**Відповіді на контрольні запитання:**

1. **Для чого використовується графічний модуль graphics?**

У Python графічний модуль graphics, який був розроблений Джоном Зелле, використовується для малювання та відображення примітивних графічних об’єктів на екрані.

1. **Як підключити графічний модуль graphics?** Потрібно завантажити файл і покласти в ту директорію, де ми збираємося писати код своєї програми. Далі потрібно заімпортувати потрібні компоненти бібліотеки для нашої програми. За допомогою команди import \* ми додаємо в програму всі елементи з графічного модуля graphics, які необхідні для розробки.
2. **Властивості відображення графічних об’єктів?** У Python графічний елемент, наприклад об’єкт класу Polygon, Circle чи Rectangle, має набір властивостей, які можна для нього задати. Основними та загальновідомими серед них є методи: setFill(‘color’), setOutline(‘color’), setWidth(value), які встановлюють колір, використовуваний для заповнення об'єкта, колір, для малювання контуру об'єкта та ширину об'єкту в пікселях відповідно. Крім передачі напряму назви кольору в якості аргументу, можна визначити кольори, використовуючи метод color\_rgb, який вимагає три параметри: інтенсивність червоного, зеленого і синього кольору (від 0 до 255). Для текстових об’єктів застосовуються такі функції як setFace(font), setSize(point), setStyle(style), setTextColor(color), які встановлюють такі властивості тексту як шрифт, розмір шрифту, стиль та колір відповідно.
3. **Яким чином відбувається зчитування інформації з файлу? Навести приклад.** Щоб зчитати дані з файлу у Python існує функція open(), аргументами якої є 2 параметри: назва файлу та режим, в якому ми хочемо відкрити цей файл (напр., ‘r’,’w’,’a’). Функція повертає посилання на файл, яке ми зберігаємо для подальшого використання. Метод readline () зчитує рядок з текстового файлу. Функція eval перетворює його в число. Приклад (зчитування 5 елементів з текстового файлу):

infile = open(‘elements.txt’, ‘r’)

list = []

for i in range(5):

list.append(eval(infile.readline())

1. **Яким чином відбувається зчитування та вивід координат за допомогою мишки? Навести приклад.** Коли користувач робить клік мишкою всередині вікна, розташування показника мишки в момент кліку записується для використання. Вікно має функцію під назвою getMouse, яка чекає, поки користувач не натисне у вікні, а потім повертає координати точки, де користувач натиснув. Надалі можна використовувати функції getX і getY, щоб дізнатися окремо координати Х та Y відповідно. Приклад (наповнення масиву з 5 координатами точок, введених користувачем):

points = []

for i in range(5):

points.append(win.getMouse())

print(points[i])

1. **Яка послідовність виводу графічних об’єктів?** Послідовність виводу графічних об’єктів залежить від команд заданих в програмі.
2. **Що таке масив в Python? Як відбувається його ініціалізація.** У Python масив реалізований у вигляді списку даних. Значення масиву (списку) вказуються всередині квадратних дужок, де перераховуються через кому. Елемент списку елемент можна викликати за індексом і привласнити йому нове значення. Щоб дізнатися довжину масиву можна скористаись функцією len(). (Думаю, що це питання потрапило в список контрольних питань випадково з попередньої лабораторної роботи, але відповідь все-таки сформулювала)
3. **Навести приклад запису інформації в файл.** Якщо ми хочемо записати щось в файл потрібно режим відкриття файлу в функції open() змінити з ‘r’(read) на ‘w’(write). Далі з допомогою функції write(text) відбувається запис в файл. Якщо файл не існує. То буже створений новий екземпляр з вказаним іменем. Після завершення роботи з файлом його потрбно закрити, використавши для того функцію close(). Приклад (запис у файл елементів з масиву):

file = open(‘elements.txt’, ‘w’)

list = [1,2,3,4,5]

for i in range(5):

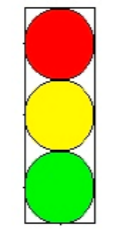
file.write(list[i])

**Індивідуальне завдання:**

*Варіант 2.*

**Завдання 1.**

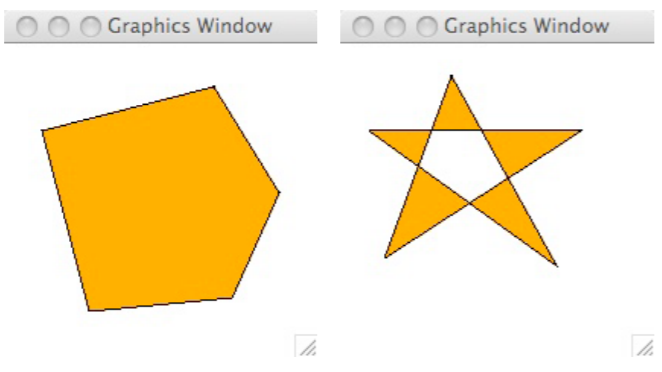
1. Намалюйте простий світлофор в графічному вікні розмірами 200 пікселів в ширину і 200 пікселів у висоту. Три вогні повинні мати діаметр 50 пікселів кожен, і світлофор повинні бути в центрі графічного вікна.



1. Використовуючи цикл, намалювати кола, які дотикаються один до одного зовнішніми частиною кола та розміщені горизонтально. Радіус найбільшого кола 60 пікселів, кожного наступного на 10 менше.

**Завдання 2.**

Напишіть програму на Python, яка попросить користувача ввести 5 точок і тоді намалює оранжевий многокуник, який з’єднує ці точки. Запустіть програму натиском на 5 різних точок у вікні. Далі запустіть програму натиснувши на вікно, і спробуйте сформувати п’ятикутну зірку. Зверніть увагу як змінюється колір, коли лінії пересікають одна одну.



**Завдання 3.**

Спростити функцію update\_board так щоб вам не потрібні були if оператори для перемикання вогників. Для цього потрібно взяти значення вогників (0 або 1), додаючи 1, і виконати модуль 2 для цього значення, щоб отримати значення перемикання. Зверніть увагу, що якщо світло має значення 1, то (1 + 1)% 2 = 0 і, якщо світло має значення 0, то (0 + 1)% 2 = 1. В обох випадках ми отримуємо значення вогника. Ви все ж будете використовувати оператор if щоб перевірити рядок і стовпець для сусідів.

**Завдання 4.**

Змініть цю програму, щоб вона мала ігрову зону 5х5, з номерами від 1 до 24 і однією порожньою клітинкою.

**Код програми:**

from graphics import \*

# TASK 1

def task\_1():

colors = ['red', 'yellow', 'green', 'blue', 'pink', 'orange']

# Part 1

def task\_1\_1():

win = GraphWin('TASK 1.1', 200, 200)

for i in range(3):

circle = Circle(Point(100, 50 \* (i + 1)), 25)

circle.setFill(colors[i])

circle.setOutline('black')

circle.setWidth(1)

circle.draw(win)

Rectangle(Point(75, 25), Point(125, 175)).draw(win)

win.getMouse()

win.close()

# Part 2

def task\_1\_2():

win = GraphWin('TASK 1.2', 420, 120)

d = 0

for i in range(6):

d += (60 - i \* 10)

circle = Circle(Point(d, 60), (60 - i \* 10))

circle.setFill(colors[i])

circle.setOutline('black')

circle.setWidth(1)

circle.draw(win)

d += (60 - i \* 10)

win.getMouse()

win.close()

task\_1\_1()

task\_1\_2()

task\_1()

# TASK 2

def task\_2():

win = GraphWin('TASK 2', 300, 300)

win.setBackground('white')

points = []

for i in range(5):

points.append(win.getMouse())

print(points[i])

polygon = Polygon(points)

polygon.setFill('orange')

polygon.draw(win)

win.getMouse()

win.close()

for i in range(2):

task\_2()

# TASK 3

def task\_3():

def initialize\_board(filename):

infile = open(filename, 'r')

board = []

for i in range(5):

board.append([])

for row in range(5):

for col in range(5):

columnvalue = eval(infile.readline())

board[row].append(columnvalue)

return board

def draw\_lines(window):

s\_x = 0

s\_y = 250

for i in range(4):

Line(Point(0, 50\*(i+1)), Point(250, 50\*(i+1))).draw(window)

if i == 3:

s\_x = 250

s\_y = 0

Line(Point(50\*(i+1), s\_x), Point(50\*(i+1), s\_y)).draw(window)

def display\_lights(window, board):

for row in range(5):

for column in range(5):

center = Point(column \* 50 + 25, row \* 50 + 25)

circ = Circle(center, 25)

if board[row][column] == 1:

circ.setFill('lightpink')

else:

circ.setFill('white')

circ.draw(window)

def update\_board(board, row, column):

board[row][column] = (board[row][column] + 1) % 2

if row > 0:

board[row - 1][column] = (board[row - 1][column] + 1) % 2

if row < 4:

board[row + 1][column] = (board[row + 1][column] + 1) % 2

if column > 0:

board[row][column - 1] = (board[row][column - 1] + 1) % 2

if column < 4:

board[row][column + 1] = (board[row][column + 1] + 1) % 2

def check\_for\_winner(board):

sum = 0

for row in range(5):

for column in range(5):

sum = sum + board[row][column]

if sum == 0:

return True

else:

return False

def clear(win):

for item in win.items[:]:

item.undraw()

win.update()

def main():

window = GraphWin('Lights Out', 250, 250)

board = initialize\_board('lights.txt')

draw\_lines(window)

display\_lights(window, board)

game\_over = False

while not game\_over:

p = window.getMouse()

column = p.getX() / 50

row = p.getY() / 50

update\_board(board, int(row), int(column))

display\_lights(window, board)

game\_over = check\_for\_winner(board)

message = Text(Point(125, 125), 'GAME OVER')

message.setSize(24)

message.setTextColor('red')

clear(window)

message.draw(window)

window.getMouse()

window.close()

main()

task\_3()

# TASK 4

def task\_4():

def initialize\_board(filename):

infile = open(filename, 'r')

board = []

for i in range(5):

board.append([])

for row in range(5):

for col in range(5):

columnvalue = eval(infile.readline())

board[row].append(columnvalue)

return board

def display\_numbers(window, board):

for row in range(5):

for col in range(5):

square = Rectangle(Point(col \* 50, row \* 50), Point((col + 1) \* 50, (row + 1) \* 50))

square.setFill('snow')

square.setOutline('maroon')

square.draw(window)

if board[row][col] != 0:

center = Point(col \* 50 + 25, row \* 50 + 25)

number = Text(center, board[row][col])

number.setSize(24)

number.setTextColor('crimson')

number.draw(window)

def update\_board(board, row, col):

if row > 0 and board[row - 1][col] == 0:

board[row - 1][col] = board[row][col]

board[row][col] = 0

return

if row < 4 and board[row + 1][col] == 0:

board[row + 1][col] = board[row][col]

board[row][col] = 0

return

if col > 0 and board[row][col - 1] == 0:

board[row][col - 1] = board[row][col]

board[row][col] = 0

return

if col < 4 and board[row][col + 1] == 0:

board[row][col + 1] = board[row][col]

board[row][col] = 0

return

def check\_for\_winner(board):

num = 1

row = 0

col = 0

while num <= 24:

if board[row][col] == num:

num = num + 1

col = col + 1

if col > 4:

col = 0

row = row + 1

else:

return False

return True

def clear(win):

for item in win.items[:]:

item.undraw()

win.update()

def main():

window = GraphWin('GAME 25', 250, 250)

board = initialize\_board('game\_25.txt')

display\_numbers(window, board)

game\_over = False

while not game\_over:

p = window.getMouse()

col = int(p.getX() / 50)

row = int(p.getY() / 50)

update\_board(board, row, col)

display\_numbers(window, board)

game\_over = check\_for\_winner(board)

message = Text(Point(125, 125), 'GAME OVER')

message.setSize(24)

message.setTextColor('red')

clear(window)

message.draw(window)

window.getMouse()

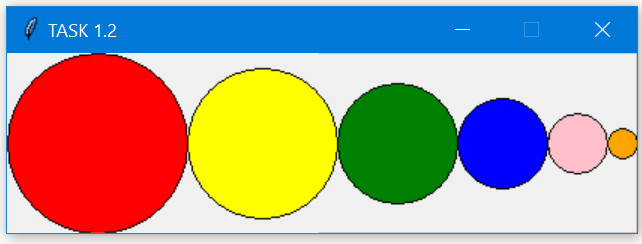
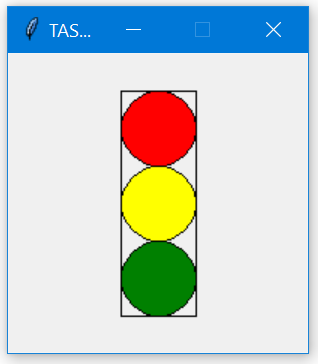
window.close()

main()

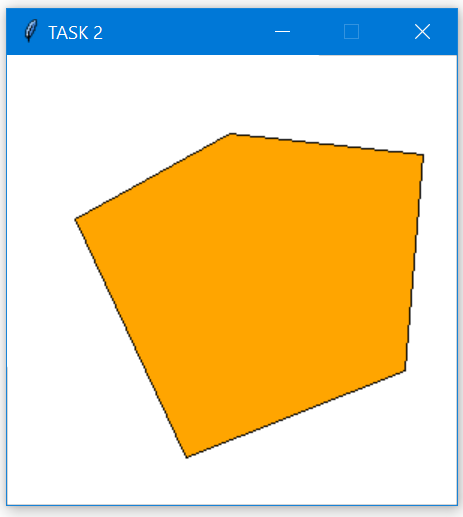
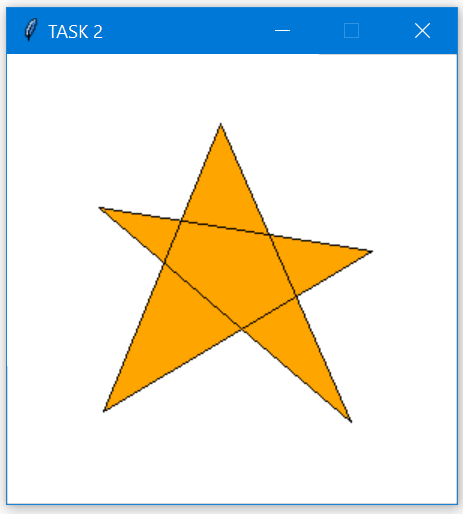
task\_4()

**Результати виконання програми:**

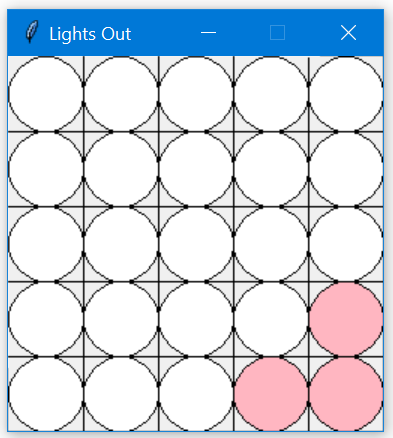
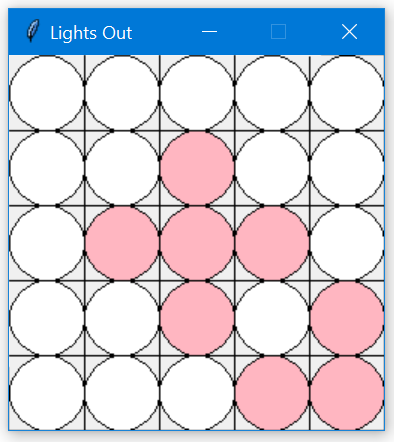
**Завдання 1**



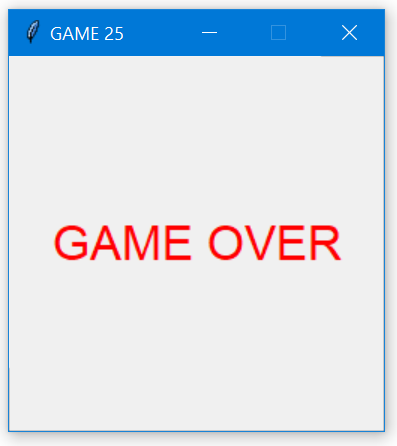
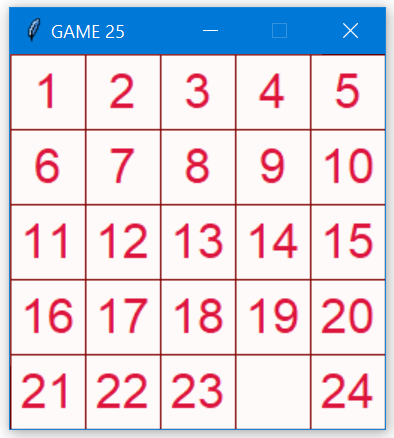
**Завдання 2**

**Завдання 3**



**Завдання 4**

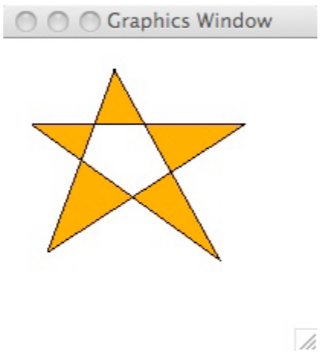
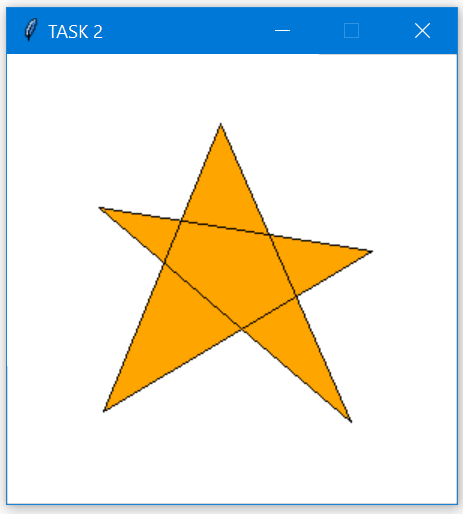


**Аналіз результатів:** В першому завданні стояла задача намалювати світлофор згідно з вказівками, які були описані в завданні. Для виконання цієї задачі було написано функцію task\_1\_1(), всередині якої була написана логіка побудови чотирикутника та відмалювання трьох вогнів світлофора, використавши цикл for. Для відображення відповідних кольорів був створений масив з назвами заливок для світлофору. Також в першому завданні стояла задача намалювати кола, які дотикаються одне до одного, використавши цикл. Для цього було створено функцію task\_1\_2(), в якій була описана логіка відмальовки кіл та знаходження їх центру та відповідного радіусу (кожен раз менше на 10). Весь код відведений для першого завдання був об’єднаний в функції task\_1().

Задачею другого завдання було надати можливість користувачу ввести 5 точок кліком миші на полотні і, на основі цих даних, побудувати п’ятикутник. Також потрібно було повторно запустити програму і користувач має знов ввести точок, але на цей раз в такому порядку, щоб сформулювати зірку. Потрібно зазначити, що в умові завдання сказано звернути увагу на заливку зірки при перетині ліній (вона порожня). Перепробувавши різні варіанти і шукавши розв’язок в мережі Інтернет, я натрапила на цікаву інформацію:

«Also, the fill may look strange on some systems like Unix (unfilled in the center) but look completely filled other systems, like Windows» [3].

Маю припущення, що через те, що виконувана програма прикладу в завданні була запущена на операційній системі MacOS заповнення зірки є порожнім. Оскільки моя ОС – Windows, то зірка виходить заповненою. Приклад:

MacOS WindowsOS

В третьому завданні стояла задача спростити функцію update\_board так, щоб if оператори для перемикання вогників були не потрібні. Для цього було запропоновано використання модулю 2 (%2), що є дуже хорошим та цікавим рішенням. Код програми можна побачити в блоці «Код програми» (task\_3()). Також в кінці гри, при перемозі користувача реалізовано вивід на полотно слів «GAME OVER».

Завдання 4 – розширення гри 15. В моєму завданні потрібно було її перетворити в гру 24. Для цього завдання була написана функція task\_4(), в якій були згруповані та, відповідно до поставленої задачі, модифіковані всі потрібні функції. Також в кінці гри, при перемозі користувача реалізовано очищення полотна та вивід на полотно слів «GAME OVER».

**Висновок:** Під час виконання лабораторної роботи я ознайомилась з мовою програмування Python зі сторони графічного представлення та побудови фігур на основі модулів комп'ютерної графіки, використання графічних операторів, запис та читання з файлів. Для малювання та відображення примітивних графічних об’єктів на екрані був використаний графічний модуль graphics, який був розроблений Джоном Зелле. Він є зручним у використанні та легким в підключені. Також ознайомилась з тим, яким способом можна створити елементарні графічні фігури(многокутник, коло, чотирикутник, текст) та відобразити їх на полотні, дізналась їх основні властивості. Ознайомилась із взаємодією з користувачем(зчитування кліку мишки) та з роботою з файлами, зчитуванням та записом даних в них.

**Перелік використаних посилань:**

1. Андрущак Н.А. Методи та засоби комп’ютерного навчання: лабораторний практикум для студентів другого (магістреського) рівня вищої освіти Інституту комп’ютерних наук та інформаційних технологій. – Львів: Видавництво Національного університету «Львівська політехніка», 2018. – 125 с.
2. Руководство по программированию на Python: <https://metanit.com/python/tutorial/>
3. Python 3 Graphics (Drawing a star) - Stack Overflow: https://stackoverflow.com/questions/26952838/python-3-graphics-drawing-a-star