

Programming Fundamentals I Lab.

11. Graphic Programming with tkinter

ชื่อ _____ รหัสนิสิต _____

ในปฏิบัติการนี้ คุณจะได้รู้จักการเขียนโปรแกรมกราฟิกเบื้องต้น

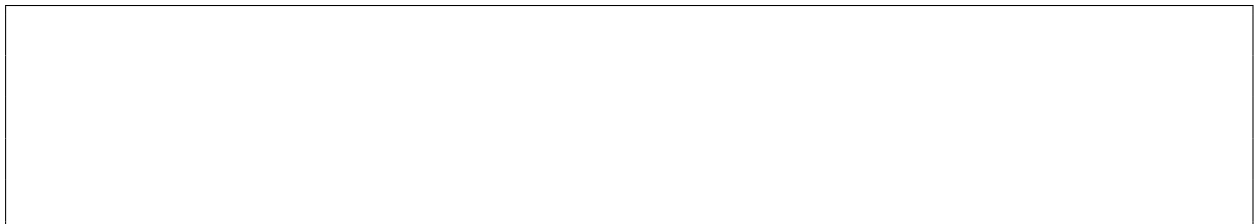
11.1 tkinter

ในการเขียนโปรแกรมกราฟิกในภาษา Python มีโมดูลหนึ่งชื่อ tkinter ที่รวบรวมคำสั่งสำคัญไว้ให้

ทดลองสร้างไฟล์ .py ใหม่ให้มีโค้ดดังนี้

```
from tkinter import *  
tk = Tk()  
btn = Button(tk, text='click me')  
btn.pack()
```

รันและบันทึกผลที่เกิดขึ้น



อธิบาย ในบรรทัดแรก เราทำการนำเข้าคำสั่งทั้งหมดที่มีอยู่ในโมดูล tkinter บรรทัดถัดมาเราสร้างตัวแปร tk ให้เป็น object ของคลาส Tk ซึ่งใช้แทนหน้าต่างโปรแกรมที่เราสามารถนำของต่าง ๆ มาใส่เพิ่มได้อย่างเช่น ปุ่ม, กล่องข้อความ

ในบรรทัดที่สาม เราสร้างตัวแปร `btn` ให้เป็น object ของคลาส `Button` ซึ่งแทนปุ่มกด โดยส่ง `tk` ไปเป็นพารามิเตอร์ตัวแรก เพื่อผูกให้ปุ่ม `btn` อยู่บนหน้าต่าง `tk` และส่งสตริง `click me` ไปใส่พารามิเตอร์ที่ชื่อ `text` เพื่อให้แสดงข้อความนั้นบนปุ่มของเรา คำสั่งสุดท้าย `btn.pack()` จะทำหน้าที่สั่งให้ปุ่ม `btn` ปรากฏขึ้นบนหน้าต่างที่ผูกไว้

ถัดมา ทดลองแก้ไขไฟล์ `.py` ให้มีโค้ดดังนี้

```
from tkinter import *
def hello():
    print('Hello there')
tk = Tk()
btn = Button(tk, text='click me', command=hello)
btn.pack()
```

ทดลองรันโปรแกรมอีกครั้ง ลองคลิกที่ปุ่ม `click me` บันทึกสิ่งที่เกิดขึ้น



คุณคิดว่าพารามิเตอร์ `command` มีไว้เพื่ออะไร



11.2 Canvas

หากเราต้องการวาดเส้นหรือภาพต่าง ๆ บนหน้าจอ เราต้องการ object ของคลาสที่ชื่อว่า `Canvas` ซึ่งแทนผืนผ้าสำหรับวาดสิ่งต่าง ๆ ลงไป

ทดลองแก้ไขไฟล์ `.py` ให้มีโค้ดดังนี้

```
from tkinter import *
tk = Tk()
canvas = Canvas(tk, width=500, height=500)
canvas.pack()
```

เมื่อทดลองรันดู จะเห็นว่าหน้าต่าง `tk` จะปรากฏด้วยขนาดกว้าง 500 pixel สูง 500 pixel แต่ยังเป็นหน้าจอเปล่า นั่นเป็นเพราะเรายังไม่ได้วาดอะไรลงไปบน `canvas` นั่นเอง

หลังจากรันโปรแกรม ทดลองพิมพ์คำสั่ง `canvas.create_line(0, 0, 500, 500)` ใน Shell บันทึกสิ่งที่เกิดขึ้นบนหน้าต่าง `tk`



ให้ทดลองแก้ไขตัวเลขทั้ง 4 ตัวในคำสั่ง `create_line` สรุปว่าตัวเลขทั้ง 4 แทนอะไร



จากการทดลอง คุณคิดว่าในหน้าต่าง `tk` จุดที่ $x=0$ และ $y=0$ อยู่มุมใด



คำสั่ง `create_line()` มีหน้าที่อะไร

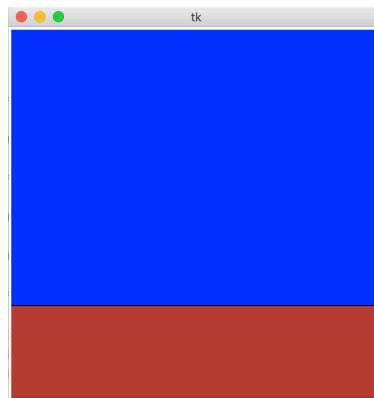


คราวนี้ให้คุณลองสั่ง `canvas.create_rectangle(20, 50, 150, 200, fill='green')` บันทึกสิ่งที่เกิดขึ้นบนหน้าต่าง `tk`



คำสั่ง `create_rectangle()` มีหน้าที่อะไร

แก้ไขไฟล์ `.py` ให้สร้างหน้าต่าง `tk` ที่มี canvas ขนาดกว้าง 400 pixel สูง 400 pixel และสร้างรูปสี่เหลี่ยมสีฟ้าให้กว้างทั่วทั้ง canvas และสูงจากขอบบนลงมา 300 pixel และสร้างรูปสี่เหลี่ยมอีกรูปให้มีสีน้ำตาลกว้างเต็ม canvas และสูง 100 pixel ชิดขอบล่าง ดังรูป



บันทึกโค้ดที่ใช้

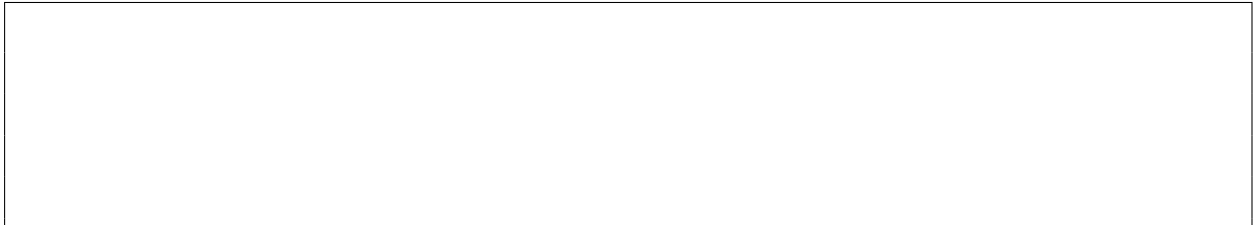
สำหรับการวาดรูปจากไฟล์รูป ในโมดูล `tkinter` นั้นรองรับการวาดรูปจากไฟล์นามสกุล `.gif` เท่านั้น ให้คุณลองนำไฟล์ชื่อ `whitesheep.gif` มาวางไว้ในโฟลเดอร์เดียวกับไฟล์ `.py` จากนั้นรันไฟล์ `.py` และพิมพ์คำสั่งต่อไปนี้ใน Shell

```
>>> sheep_image = PhotoImage(file='whitesheep.gif')
>>> canvas.create_image(100, 300, anchor=SW, image=sheep_image)
```

บันทึกผลที่เกิดขึ้นในหน้าต่าง tk



ทดลองเปลี่ยนค่า `anchor=SW` ให้เป็น NW, NE, SE ตามลำดับ บันทึกความหมายของการใช้แต่ละค่า



11.3 การสร้างภาพเคลื่อนไหว

แก้ไขไฟล์ `.py` เป็นดังนี้

```
from tkinter import *
import time
tk = Tk()
canvas = Canvas(tk, width=400, height=400)
canvas.pack()
sky_id = canvas.create_rectangle(0, 0, 400, 300, fill='blue')
floor_id = canvas.create_rectangle(0, 300, 400, 400, fill='brown')
sheep_image = PhotoImage(file='whitesheepright.gif')
sheep_id = canvas.create_image(0, 300, anchor=SW, image=sheep_image)

for i in range(80):
    canvas.move(sheep_id, 5, 0)
```

```
tk.update()
time.sleep(0.05)
```

ทดลองรันและบันทึกสิ่งที่เกิดขึ้น



ในคำสั่ง `canvas.move()` ทดลองเปลี่ยนพารามิเตอร์จาก `sheep_id` เป็น `sky_id` หรือ `floor_id` และทดลองเปลี่ยนเลข 5 กับ 0 เป็นค่าอื่น ๆ สรุปความหมายของพารามิเตอร์แต่ละตัว



หากเราต้องการสร้าง object ของแกะหลาย ๆ ตัวให้เดินอยู่ใน canvas เราสามารถสร้างเป็นคลาสสำหรับการจัดการแกะแต่ละตัวได้ ให้คุณแก้ไขไฟล์ `.py` เป็นดังนี้

```
from tkinter import *
import random
import time

class Sheep():
    def __init__(self, canvas, v=3, direction='left'):
        self.point = 2
        self.images={'center': PhotoImage(file='whitesheep.gif'),
                     'left': PhotoImage(file='whitesheepleft.gif'),
                     'right': PhotoImage(file='whitesheepright.gif')}
        self.v = v
        self.direction = direction
        self.canvas = canvas
        self.x = random.randint(0, 400-32)
```

```

        self.canvas_id = canvas.create_image(self.x, 300, anchor=SW,
                                              image=self.images[self.direction])

    def move(self):
        if self.direction == 'left':
            self.x -= self.v
            self.canvas.move(self.canvas_id, -self.v, 0)
            if self.x <= 0:
                self.direction = 'right'
                self.canvas.itemconfig(self.canvas_id,
                                       image=self.images[self.direction])
            elif self.direction == 'right':
                self.x += self.v
                self.canvas.move(self.canvas_id, self.v, 0)
                if self.x >= 400-32:
                    self.direction = 'left'
                    self.canvas.itemconfig(self.canvas_id,
                                           image=self.images[self.direction])

tk = Tk()
canvas = Canvas(tk, width=400, height=400)
canvas.pack()
canvas.create_rectangle(0, 0, 400, 300, fill='blue')
canvas.create_rectangle(0, 300, 400, 400, fill='brown')
sheeps = []
for i in range(10):
    sheeps.append(random.choice([Sheep(canvas),
                                Sheep(canvas, 0, 'center'),
                                Sheep(canvas, 1, 'right'),
                                Sheep(canvas, 2, 'left'),
                                Sheep(canvas, 1, 'left'),
                                Sheep(canvas, 2, 'right')]))

while True:
    for sheep in sheeps:
        sheep.move()
    tk.update()
    time.sleep(0.03)

```

รันและบันทึกผลที่ได้

11.4 การรับคำสั่งจากคีย์บอร์ด

ให้คุณเพิ่มคลาส Dog ต่อจากการประกาศคลาส Sheep ดังนี้

```
class Dog:
    def __init__(self, canvas):
        self.canvas = canvas
        self.images={'left':PhotoImage(file='dogleft.gif'),
                     'right':PhotoImage(file='dogright.gif')}
        self.direction='right'
        self.x = 0
        self.y = 300
        self.vx = 0
        self.vy=0
        self.canvas_id = canvas.create_image(self.x, self.y, anchor=SW,
                                              image=self.images[self.direction])

    def move(self):
        self.canvas.move(self.canvas_id, self.vx, -self.vy)
        self.y -= self.vy
        if self.y > 300:
            self.canvas.move(self.canvas_id, 0, 300 - self.y)
        if self.y < 300:
            self.vy -= 1
        else:
            self.vy = 0
            self.y = 300

        self.x += self.vx
```



```

    if self.vx > 0 and self.direction == 'left':
        self.direction = 'right'
        self.canvas.itemconfig(self.canvas_id,
                                image=self.images[self.direction])
    elif self.vx < 0 and self.direction == 'right':
        self.direction = 'left'
        self.canvas.itemconfig(self.canvas_id,
                                image=self.images[self.direction])

```

และเพิ่มคำสั่งต่อไปนี้ก่อนวนรอบ while True:

```

dog = Dog(canvas)
def move_dog(event):
    if event.keysym == 'Left':
        dog.vx = -3
    elif event.keysym == 'Right':
        dog.vx = 3
    elif event.keysym == 'space':
        dog.vy = 10

canvas.bind_all('<KeyPress-Left>', move_dog)
canvas.bind_all('<KeyPress-Right>', move_dog)
canvas.bind_all('<KeyPress-space>', move_dog)

```

และแก้ไขคำสั่งในวนรอบ while True: เป็นดังนี้

```

while True:
    for sheep in sheeps:
        sheep.move()
    dog.move()
    tk.update()
    time.sleep(0.03)

```

ทดลองรันโปรแกรม ในขณะที่โปรแกรมทำงานอยู่ ทดลองกดปุ่มลูกศรซ้าย, ขวา, และปุ่ม space-bar บันทึกผลที่เกิดขึ้น

คุณคิดว่าคำสั่ง `canvas.bind_all()` มีหน้าที่อะไร

11.5 โจทย์ปัญหา

1. จงแก้ไขเกม เมื่อผู้เล่นบังคับสุนัขไปชนกับแกะตัวใด ให้แกะตัวนั้นหายไปจากหน้าจอ
คำแนะนำ คุณสามารถลบ object ออกจาก canvas ได้ด้วยคำสั่ง `canvas.delete(object_id)`
2. จงแก้ไขเกม ให้มีกำหนดเวลาและคะแนน โดย เมื่อสุนัขจับแกะได้ตัวหนึ่ง จะได้คะแนนเท่ากับค่าในตัวแปร `point` ของแกะตัวนั้น และเมื่อเวลาสิ้นสุดหรือสามารถจับแกะได้ครบทุกตัว ให้โปรแกรมสิ้นสุดการทำงานพร้อมรายงานคะแนนรวมออกมา (ให้คุณออกแบบการรายงานคะแนนด้วยตัวเอง อาจทำงานง่าย ๆ โดยใช้คำสั่ง `print` ก็ได้ และกำหนดเวลาของเกมเองตามความเหมาะสม)
3. จงสร้างคลาสชื่อ `BlackSheep` ให้สืบทอดมาจากคลาส `Sheep` แต่ใช้รูปแกะดำในการแสดง และให้มีค่า `point` เป็น -20 แก้ไขโค้ดส่วนที่ทำการสุ่มสร้างแกะแต่ละตัว ให้สามารถสร้างแกะดำได้ด้วย
4. **พิเศษ** จงแก้ไขเกมให้ป้องกันกรณีสุนัขวิ่งออกนอกกรอบ และป้องกันการกระโดดให้สามารถกระโดดได้จากพื้นเท่านั้น