

G?.stt

HỌ TÊN: MSSV: KÝ TÊN:

TRƯỜNG: HCMUTE

MÔN: Lập trình Python

NGÀY:/...../..... (BUỔI HỌC SỐ ?)

Riêng buổi học Phòng máy ghi thêm, SỐ MÁY: PHÒNG MÁY:

=====

Buổi 14 = Bài tập 7 = Lập trình Python Game đơn giản: PyGame

1. GIỚI THIỆU CHỦ ĐỀ

NGUỒN:

<https://www.pygame.org>

THƯ VIỆN

- Thư viện chính : **PyGame**

```
python -m pip install -U pygame==2.3.0 --user
```

- Kết hợp thư viện: **Sys, Random**

LÝ THUYẾT ĐƠN GIẢN VỀ GAME

- Các game thường tổ chức vòng lặp vô tận **while True**: {lặp cho đến khi QUIT}
Đơn giản là trong vòng lặp thường có 3 thao tác chính:
 - . Lập (vẽ) màn hình (cửa sổ) trạng thái của game
 - . Nhân sự kiện (Event)
 - . Thay đổi màn hình (cửa sổ) trạng thái của game
- Surface: Khung (hình) trạng thái của Game = tương tự Layer của các đối tượng Media = là một “lớp” ảnh trong suốt. Các surface có thể được vẽ chồng lên nhau.
-> Trong Demo bên dưới: biến DISPLAYSURF là một surface đặc biệt, có kích thước bằng cửa sổ game và nằm dưới cùng.
- Sử dụng thư viện PyGame
pygame.init()
...
pygame.display.update() # hiện thị các nội dung (vẽ) kẻ trên

EX1: Màn hình (cửa sổ) Game

```
# -*- coding: utf-8 -*-
"""
Created on Thu May  3 08:45:49 2023

@author: VOXUAN
"""

# B1: Nạp thư viện pygame & các thư viện liên quan

import pygame, sys
from pygame.locals import *

# B2: khởi tạo thư viện pygame (luôn phải có)
pygame.init()

#B3: Lập cửa sổ game (400, 300). DISPLAYSURF là 1 biến kiểu surface (khung đen đen)
DISPLAYSURF = pygame.display.set_mode((400, 300))
pygame.display.set_caption('stt Ho Tên = Ex7.1: Màn hình (cửa sổ) Game')

#B4: Vòng lặp chạy liên tục Game cho đến khi nhận sự kiện QUIT (nút x : trên, phải)
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit() # thoát game
            sys.exit()    # thoát hệ thống (tức là OS)

#####MINH HỌA VẼ#####
DISPLAYSURF.fill((255, 255, 255)) # tô nền trắng RGB
#pygame.draw.rect(DISPLAYSURF, (255, 0, 0), (100, 80, 150, 50))#khung đỏ
```

EX2: Layer = Surface

```
# -*- coding: utf-8 -*-
"""
Created on Thu May  3 10:45:49 2023

@author: VOXUAN
"""

# B1: Nạp thư viện pygame & các thư viện liên quan

import pygame, sys
from pygame.locals import *

# B2: khởi tạo thư viện pygame (luôn phải có)
pygame.init()

#B3: Lập cửa sổ game (400, 300). DISPLAYSURF là 1 biến kiểu surface (khung đen đen)
DISPLAYSURF = pygame.display.set_mode((400, 300))
pygame.display.set_caption('stt Ho Tên = Ex7.2: Surface = Layer')

#B4: Vòng lặp chạy liên tục Game cho đến khi nhận sự kiện QUIT (nút x : trên, phải)
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit() # thoát game
            sys.exit()    # thoát hệ thống (tức là OS)

#####MINH HỌA SURFACE#####
DISPLAYSURF.fill((255, 255, 255)) # tô nền trắng RGB

surface2rect = pygame.Surface((150, 50))
surface2rect.fill((0, 255, 0))
pygame.draw.rect(surface2rect, (255, 0, 0), (20, 20, 50, 20))
DISPLAYSURF.blit(surface2rect, (100, 80)) # blit = place
pygame.display.update()
```

EX3: Vẽ các loại hình đơn giản

```
# -*- coding: utf-8 -*-
"""
Created on Thu May  3 11:45:49 2023

@author: VOXUAN
"""
# B1: Nạp thư viện pygame & các thư viện liên quan

import pygame, sys
from pygame.locals import *

WINDOWWIDTH = 400 # Chiều dài cửa sổ
WINDOWHEIGHT = 300 # Chiều cao cửa sổ

# Lập sẵn các màu sắc
BLACK = ( 0, 0, 0)
WHITE = (255, 255, 255)
RED = (255, 0, 0)
GREEN = ( 0, 255, 0)
BLUE = ( 0, 0, 255)

# B2: khởi tạo thư viện pygame (luôn phải có)
pygame.init()

#B3: Lập cửa sổ game (400, 300). DISPLAYSURF là 1 biến kiểu surface (khung đen đen)

DISPLAYSURF = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
pygame.display.set_caption('stt Ho Tên = Ex7.3: Vẽ hình đơn giản')
#B4: Vòng lặp chạy liên tục Game cho đến khi nhận sự kiện QUIT (nút x : trên, phải)
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit() # thoát game
            sys.exit()    # thoát hệ thống (tức là OS)

    DISPLAYSURF.fill(WHITE)
    pygame.draw.rect(DISPLAYSURF, RED, (10, 10, 100, 50))# Hình chữ nhật
    pygame.draw.rect(DISPLAYSURF, GREEN, (150, 10, 100, 50), 2)#Hình chữ nhật rỗng
    pygame.draw.circle(DISPLAYSURF, RED, (50, 100), 20) # Hình tròn
    pygame.draw.circle(DISPLAYSURF, BLUE, (200, 100), 20, 1)# Hình tròn rỗng
    pygame.draw.ellipse(DISPLAYSURF, RED, (10, 150, 100, 50))# Hình elip
    pygame.draw.ellipse(DISPLAYSURF, GREEN, (150, 150, 100, 50), 3)#Hình elip rỗng
    pygame.draw.polygon(DISPLAYSURF, RED, ((10, 220), (150, 230), (100, 290), (30, 270)))#Đa giác
    pygame.draw.polygon(DISPLAYSURF, BLUE, ((160, 220), (300, 230), (250, 290), (180, 270)), 2)#Đa giác rỗng
    pygame.draw.line(DISPLAYSURF, BLACK, (300, 50), (350, 150), 4)# Đoạn thẳng

    pygame.display.update() # hiện thị
```

MỘT SỐ HÀM VẼ HÌNH

- Một số hàm (phương thức) vẽ hình trong cửa sổ game => thực tế: game ít dùng các hàm để vẽ hình ảnh, mà thường dùng files ảnh để thêm vào trong game:

pygame.draw.rect(surface, color, rect, width) Hàm vẽ hình chữ nhật.

- **surface** Nơi sẽ vẽ = DISPLAYSURF
- **color** màu sắc = dạng danh sách [tuple] -> trong mẫu là màu đỏ
- **rect** thông số hình chữ nhật
- **width** độ dày của nét vẽ (mặc định là hình chữ nhật được tô kín)

pygame.draw.circle(surface, color, center, radius, width) Hàm vẽ hình tròn.

- **surface** nơi vẽ.
- **color** màu vẽ.
- **center** tuple (hoặc list) = tọa độ tâm hình tròn.
- **radius** bán kính hình tròn.
- **width** độ dày nét vẽ (tương tự hình chữ nhật).

pygame.draw.ellipse(surface, color, rect, width) Hàm vẽ hình elip.

- **surface** nơi vẽ
- **color** màu vẽ.
- **rect** tuple (list) gồm 4 phần tử = các thông số hình chữ nhật mà elip nội tiếp trong đó.
- **width** là độ dày nét vẽ (tương tự hình chữ nhật).

pygame.draw.polygon(surface, color, points, width) Hàm vẽ đa giác.

- **surface** nơi vẽ.
- **color** màu vẽ.
- **points** tuple (list) thể hiện các đỉnh của đa giác. Mỗi đỉnh là một tuple (hoặc list) thể hiện tọa độ.
- **width** là độ dày nét vẽ (tương tự hình chữ nhật).

pygame.draw.line(surface, color, start_pos, end_pos, width) Hàm vẽ đoạn thẳng.

- **surface** nơi vẽ.
- **color** màu vẽ.
- **start_pos** tuple (list) thể hiện tọa độ điểm đầu của đoạn thẳng.
- **end_pos** tuple (list) thể hiện tọa độ điểm cuối của đoạn thẳng.
- **width** là độ dày nét vẽ.

THAO TÁC ĐIỀU KHIỂN CỦA NGƯỜI CHƠI

- Chỉ số FPS (Frame Per Second) là tốc độ khung hình của video khi quay phim = Số của số khung hình xuất hiện trong một giây của video đó.
-> FPS càng cao, thì hình ảnh sẽ mượt hơn.
- `event.type == KEYDOWN`: để kiểm tra xem có xảy ra sự kiện `KEYDOWN` hay không.
`KEYDOWN` là sự kiện xảy ra khi có 1 phím được ấn xuống.
`event.type == KEYUP`: để kiểm tra xem có xảy ra sự kiện `KEYUP` hay không.
`KEYUP` là sự kiện xảy ra khi có 1 phím được thả ra
- `event.key == K_LEFT`: để xem phím được ấn có phải là phím **mũi tên trái** hay không
`event.key == K_RIGHT`: để xem phím được ấn có phải là phím **mũi tên phải** hay không

2. FULL CODES DEMO (Tham khảo: sv nên cá nhân hóa thông tin bài làm)

EX4: Game XE: PHẦN CĂN BẢN ===

- + Nhấn giữ phím mũi tên trái thì chiếc xe chạy sang trái
- + Nhấn giữ phím mũi tên phải thì chiếc xe chạy sang phải

```
# -*- coding: utf-8 -*-
"""
Created on Thu May 3 14:00:49 2023

@author: VOXUAN
"""
# B1: Nạp thư viện pygame & các thư viện liên quan

import pygame, sys
from pygame.locals import *

WINDOWWIDTH = 400 # Chiều dài của sổ
WINDOWHEIGHT = 300 # Chiều cao của sổ

# Lập sẵn các màu sắc
BLACK = ( 0, 0, 0)
WHITE = (255, 255, 255)
RED = (255, 0, 0)
GREEN = ( 0, 255, 0)
BLUE = ( 0, 0, 255)

# B2: khởi tạo thư viện pygame (luôn phải có)
pygame.init()

#B3: Lập cửa sổ game (400, 300). DISPLAYSURF là 1 biến kiểu surface (khung đen đen)

### Xác định FPS ###
FPS = 60 # 60 khung hình / giây
fpsClock = pygame.time.Clock()

DISPLAYSURF = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
pygame.display.set_caption('stt Ho Tên = Ex7.4: Game đơn giản = Game CHẠY XE')
```

```

class Car():
    def __init__(self):
        self.x = 100 # Vị trí của xe

        ## Tạo surface và thêm hình chiếc xe vào ##
        self.surface = pygame.image.load('img/car1.png')

    def draw(self): # Hàm dùng để vẽ xe
        DISPLAYSURF.blit(self.surface, (self.x, 100))

    def update(self, moveLeft, moveRight): # Hàm dùng để thay đổi vị trí xe
        if moveLeft == True:
            self.x -= 2
        if moveRight == True:
            self.x += 2

        if self.x + 100 > WINDOWWIDTH:
            self.x = WINDOWWIDTH - 100
        if self.x < 0:
            self.x = 0

car = Car()
moveLeft = False
moveRight = False

#B4: Vòng lặp chạy liên tục Game cho đến khi nhận sự kiện QUIT (nút x : trên, phải)
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit() # thoát game
            sys.exit()    # thoát hệ thống (tức là OS)
        if event.type == KEYDOWN:
            if event.key == K_LEFT:
                moveLeft = True
            if event.key == K_RIGHT:
                moveRight = True

        if event.type == KEYUP:
            if event.key == K_LEFT:
                moveLeft = False
            if event.key == K_RIGHT:
                moveRight = False
    DISPLAYSURF.fill(WHITE)
    car.draw()
    pygame.display.update() # hiện thị
    car.update(moveLeft, moveRight)
    pygame.display.update() # hiện thị
    fpsClock.tick(FPS) # nhảy đồng hồ với chu kỳ FPS

```

=EX5: Game ĐUA XE FULL: Racing ==

GỒM 8 PHẦN

#PHẦN 1: ĐỊNH NGHĨA CÁC THAM SỐ ##

#PHẦN 2: NỀN GAME: cuộn nền #

#PHẦN 3: XE TRONG GAME #

#PHẦN 4: XE CHƯỚNG NGẠI VẬT = XE NGƯỢC CHIỀU:obstacles #

#PHẦN 5: TÍNH ĐIỂM #

#PHẦN 6: XỬ LÝ VA CHẠM: Collision #

#PHẦN 7: CÁC THỦ TỤC CHƠI GAME #

#PHẦN 8: HÀM MAIN #

```
# -*- coding: utf-8 -*-
```

```
"""
```

```
Created on Thu May 3 15:10:40 2023
```

```
@author: VOXUAN
```

```
"""
```

```
import pygame, sys, random
```

```
from pygame.locals import *
```

```
#####
```

```
#PHẦN 1: ĐỊNH NGHĨA CÁC THAM SỐ ##
```

```
#####
```

```
###KÍCH THƯỚC KHUNG MÀN HÌNH GAME
```

```
WINDOWWIDTH = 400
```

```
WINDOWHEIGHT = 600
```

```
###KHỞI TẠO THƯ VIỆN ĐỂ DÙNG
```

```
pygame.init()
```

```
##TỐC ĐỘ KHUNG HÌNH CỦA VIDEO
```

```
FPS = 60 # Frames Per Second
```

```
fpsClock = pygame.time.Clock() #Lặp theo nhịp clock (tham số FPS)
```

```
#####
```

```
#####PHẦN 2: NỀN GAME #####
```

```
#####
```

```
#TỐC ĐỘ CUỘN NỀN
```

```
BGSPEED = 1.5 # tốc độ cuộn nền
```

```
BGIMG = pygame.image.load('img/background.png') # hình nền
```

```
# LAYER (SURFACE) NỀN
```

```
DISPLAYSURF = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
```

```
pygame.display.set_caption('stt Họ Tên = Ex7.5: Game = Game ĐUA XE')
```

```
# LỚP HÌNH NỀN = CUỘN NỀN
```

```
class Background():
```

```
    def __init__(self):
```

```
        self.x = 0
```

```
        self.y = 0
```

```
        self.speed = BGSPEED
```

```
        self.img = BGIMG
```

```
        self.width = self.img.get_width()
```

```
        self.height = self.img.get_height()
```

```

def draw(self):
    DISPLAYSURF.blit(self.img, (int(self.x), int(self.y)))
    DISPLAYSURF.blit(self.img, (int(self.x), int(self.y-self.height)))
def update(self):
    self.y += self.speed
    if self.y > self.height:
        self.y -= self.height

#####
####PHẦN 3: XE TRONG GAME #####
"""

```

- **X_MARGIN** là lề hai bên trái và phải (xe không được vượt qua đó).
- **CARWIDTH** và **CARHEIGHT** là kích thước của xe.
- **CARSPEED** là tốc độ di chuyển (tiến, lùi, trái, phải) của xe.
- **CARIMG** là ảnh chiếc xe.

```

"""
#####
#KÍCH THƯỚC XE
X_MARGIN = 80
CARWIDTH = 40
CARHEIGHT = 60
CARSPEED = 3
CARIMG = pygame.image.load('img/car.png')
#LỚP XE TRONG GAME
class Car():
    def __init__(self):
        self.width = CARWIDTH
        self.height = CARHEIGHT
        self.x = (WINDOWWIDTH-self.width)/2
        self.y = (WINDOWHEIGHT-self.height)/2
        self.speed = CARSPEED
        self.surface = pygame.Surface((self.width, self.height))
        self.surface.fill((255, 255, 255))
    def draw(self):
        DISPLAYSURF.blit(CARIMG, (int(self.x), int(self.y)))
    def update(self, moveLeft, moveRight, moveUp, moveDown):
        if moveLeft == True:
            self.x -= self.speed
        if moveRight == True:
            self.x += self.speed
        if moveUp == True:
            self.y -= self.speed
        if moveDown == True:
            self.y += self.speed

        if self.x < X_MARGIN:
            self.x = X_MARGIN
        if self.x + self.width > WINDOWWIDTH - X_MARGIN:
            self.x = WINDOWWIDTH - X_MARGIN - self.width
        if self.y < 0:
            self.y = 0
        if self.y + self.height > WINDOWHEIGHT :
            self.y = WINDOWHEIGHT - self.height

```



```
#####
#PHẦN 4: XE CHƯỚNG NGẠI VẬT = XE NGƯỢC CHIỀU:obstacles ##
"""
```

- **LANEWIDTH** là độ rộng của 1 làn xe (đường có 4 làn).
- **DISTANCE** là khoảng cách giữa các xe theo chiều dọc.
- **OBSTACLESSPEED** là tốc độ ban đầu của những chiếc xe.
- **CHANGESPEED** dùng để tăng tốc độ của những chiếc xe theo thời gian.
- **OBSTACLESIMG** là ảnh chiếc xe.

```
"""
#####
LANEWIDTH = 60
DISTANCE = 200
OBSTACLESSPEED = 2
CHANGESPEED = 0.001
OBSTACLESIMG = pygame.image.load('img/obstacles.png')
class Obstacles():
    def __init__(self):
        self.width = CARWIDTH
        self.height = CARHEIGHT
        self.distance = DISTANCE
        self.speed = OBSTACLESSPEED
        self.changeSpeed = CHANGESPEED
        self.ls = []
        for i in range(5):
            y = -CARHEIGHT-i*self.distance
            lane = random.randint(0, 3)
            self.ls.append([lane, y])
    def draw(self):
        for i in range(5):
            x = int(X_MARGIN + self.ls[i][0]*LANEWIDTH + (LANEWIDTH-self.width)/2)
            y = int(self.ls[i][1])
            DISPLAYSURF.blit(OBSTACLESIMG, (x, y))
    def update(self):
        for i in range(5):
            self.ls[i][1] += self.speed
        self.speed += self.changeSpeed
        if self.ls[0][1] > WINDOWHEIGHT:
            self.ls.pop(0)
            y = self.ls[3][1] - self.distance
            lane = random.randint(0, 3)
            self.ls.append([lane, y])
```

```
#####
#PHẦN 5: TÍNH ĐIỂM ##
#####
class Score():
    def __init__(self):
        self.score = 0
    def draw(self):
        font = pygame.font.SysFont('consolas', 30)
        scoreSurface = font.render('Score: '+str(int(self.score)), True, (0, 0, 0))
        DISPLAYSURF.blit(scoreSurface, (10, 10))
    def update(self):
        self.score += 0.02
```

```
#####
#PHẦN 6: XỬ LÝ VA CHẠM: Collision ##
#####
def rectCollision(rect1, rect2):
    if rect1[0] <= rect2[0]+rect2[2] and rect2[0] <= rect1[0]+rect1[2] and rect1[1]
        <= rect2[1]+rect2[3] and rect2[1] <= rect1[1]+rect1[3]:
        return True
    return False

def isGameOver(car, obstacles):
    carRect = [car.x, car.y, car.width, car.height]
    for i in range(5):
        x = int(X_MARGIN + obstacles.ls[i][0]*LANEWIDTH + (LANEWIDTH-obstacles.width)/2)
        y = int(obstacles.ls[i][1])
        obstaclesRect = [x, y, obstacles.width, obstacles.height]
        if rectCollision(carRect, obstaclesRect) == True:
            return True
    return False

#####
#PHẦN 7: CÁC THỦ TỤC CHƠI GAME ##
"""

    • gameStart() là phần chuẩn bị khi vừa mở game lên.
    • gamePlay() là phần chơi chính.
    • gameOver() là phần xuất hiện khi thua 1 màn chơi.

"""
#####

def gameOver(bg, car, obstacles, score):
    font = pygame.font.SysFont('consolas', 60)
    headingSurface = font.render('GAMEOVER', True, (255, 0, 0))
    headingSize = headingSurface.get_size()

    font = pygame.font.SysFont('consolas', 20)
    commentSurface = font.render('Press "space" to replay', True, (0, 0, 0))
    commentSize = commentSurface.get_size()
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            if event.type == pygame.KEYUP:
                if event.key == K_SPACE:
                    return

        bg.draw()
        car.draw()
        obstacles.draw()
        score.draw()
        DISPLAYSURF.blit(headingSurface, (int((WINDOWWIDTH - headingSize[0])/2), 100))
        DISPLAYSURF.blit(commentSurface, (int((WINDOWWIDTH - commentSize[0])/2), 400))
        pygame.display.update()
        fpsClock.tick(FPS)
```

```

def gameStart(bg):
    bg.__init__()
    font = pygame.font.SysFont('consolas', 60)
    headingSurface = font.render('RACING', True, (255, 0, 0))
    headingSize = headingSurface.get_size()

    font = pygame.font.SysFont('consolas', 20)
    commentSurface = font.render('Press "space" to play', True, (0, 0, 0))
    commentSize = commentSurface.get_size()
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            if event.type == pygame.KEYUP:
                if event.key == K_SPACE:
                    return
        bg.draw()
        DISPLAYSURF.blit(headingSurface, (int((WINDOWWIDTH - headingSize[0])/2), 100))
        DISPLAYSURF.blit(commentSurface, (int((WINDOWWIDTH - commentSize[0])/2), 400))
        pygame.display.update()
        fpsClock.tick(FPS)

```

```

def gamePlay(bg, car, obstacles, score):
    car.__init__()
    obstacles.__init__()
    bg.__init__()
    score.__init__()
    moveLeft = False
    moveRight = False
    moveUp = False
    moveDown = False
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            if event.type == KEYDOWN:
                if event.key == K_LEFT:
                    moveLeft = True
                if event.key == K_RIGHT:
                    moveRight = True
                if event.key == K_UP:
                    moveUp = True
                if event.key == K_DOWN:
                    moveDown = True
            if event.type == KEYUP:
                if event.key == K_LEFT:
                    moveLeft = False
                if event.key == K_RIGHT:
                    moveRight = False
                if event.key == K_UP:
                    moveUp = False
                if event.key == K_DOWN:
                    moveDown = False

```

```

    if isGameOver(car, obstacles):
        return
    bg.draw()
    bg.update()
    car.draw()
    car.update(moveLeft, moveRight, moveUp, moveDown)
    obstacles.draw()
    obstacles.update()
    score.draw()
    score.update()
    pygame.display.update()
    fpsClock.tick(FPS)

#####
#PHẦN 8: HÀM MAIN ##
#####

def main():
    bg = Background()
    car = Car()
    obstacles = Obstacles()
    score = Score()
    gameStart(bg)
    while True:
        gamePlay(bg, car, obstacles, score)
        gameOver(bg, car, obstacles, score)

if __name__ == '__main__':
    main()

```