



# Introducción a **PyGame**

**Patricia Q.**



# ¿Que es **PyGame**?

PyGame es un framework gratis para Python que provee módulos diseñados para crear video juegos. . Esta basado en la librería Simple DirectMedia Layer Library (SDL) que provee fácil acceso a elementos visuales y de sonido.

Funciona como interfaz de las bibliotecas SDL.

Source: [pygame.org](http://pygame.org)



# ¿Como instalo Pygame?

Pygame requiere Python; si aún no lo tiene, puede descargarlo de [python.org](https://python.org).

La mejor forma de instalar pygame es con la herramienta pip.

```
pip install pygame
```

Para ver si funciona, ejecute uno de los ejemplos incluidos:

```
python3 -m pygame.examples.aliens
```

Source: [pygame.org](https://pygame.org)



# Características

- Python, C, and Assembly
- Sistema operativo plataforma cruzada (Cross-Plataform)
- No requiere OpenGL. Pygame utiliza ya sea opengl, directx, windib, X11, linux frame buffer, y muchos otros backends ... ¡incluyendo un back-end ASCII art!
- Las CPU multinúcleo (Multi core CPU) se pueden usar fácilmente. Hacer uso de CPU multinúcleo te permite hacer más en tu juego.
- Utiliza C optimizado y código de ensamblaje para funciones principales.

# Snake Game





# **Inicializando componentes principales**



# Inicializando PyGame

```
import pygame, sys, random, time # game imports
```

```
check_errors = pygame.init()
```

```
if check_errors[1] > 0:
```

```
    print("(!) Had {0} initializing errors, exiting ...".format(
```

```
        check_errors[1]))
```

```
    sys.exit(-1)
```

```
else:
```

```
    print("(+) PyGame successfully initialized!")
```



# Componentes importantes

```
#Display  
playSurface = pygame.display.set_mode((720, 460))  
pygame.display.set_caption('Snake game!')
```

```
# FPS frames per second controller  
fpsController = pygame.time.Clock()
```





## Componentes del Snake

```
red = pygame.Color(255, 0, 0) # game over  
green = pygame.Color(0, 255, 0) # snake body  
black = pygame.Color(0, 0, 0) # score  
white = pygame.Color(255, 255, 255) # bg  
brown = pygame.Color(165, 42, 42) # food
```

```
snakePos = [100, 50]  
snakeBody = [[100, 50], [90, 50], [80, 50]]
```

```
foodPos = [random.randrange(1, 72)*10, random.randrange(1, 46)*10]  
foodSpawm = True
```

```
direction = 'RIGHT'
```

```
score = 0
```



# Game Over

```
def gameOver():  
    myFont = pygame.font.SysFont('comicsansms', 60)  
    GOText = myFont.render('Game Over', True, red)  
  
    GOrrect = GOText.get_rect()  
    GOrrect.midtop = (360, 15)  
    playSurface.blit(GOText, GOrrect)  
  
    pygame.display.flip()  
    time.sleep(3)  
    pygame.quit()  
    sys.exit()
```



## Eventos en el juego

```
def getEvent(dir):  
    for event in pygame.event.get():  
        if event.type == pygame.QUIT:  
            pygame.quit()  
            sys.exit()  
        elif event.type == pygame.KEYDOWN:  
            if event.key == pygame.K_RIGHT or event.key == ord('d'):  
                dir = 'RIGHT'  
            if event.key == pygame.K_LEFT or event.key == ord('a'):  
                dir = 'LEFT'  
            if event.key == pygame.K_UP or event.key == ord('w'):  
                dir= 'UP'  
            if event.key == pygame.K_DOWN or event.key == ord('s'):  
                dir = 'DOWN'  
            if event.key == pygame.K_ESCAPE:  
                pygame.event.post(pygame.event.Event(pygame.QUIT))  
    return dir
```

## Restricción de movimientos

```
if changeto == 'RIGHT' and not direction == 'LEFT':  
    direction = 'RIGHT'  
if changeto == 'LEFT' and not direction == 'RIGHT':  
    direction = 'LEFT'  
if changeto == 'UP' and not direction == 'DOWN':  
    direction = 'UP'  
if changeto == 'DOWN' and not direction == 'UP':  
    direction = 'DOWN'
```



## Moviendo a la serpiente

```
def moveSnake(direction):  
    if direction == 'RIGHT':  
        snakePos[0] += 10  
    if direction == 'LEFT':  
        snakePos[0] -= 10  
    if direction == 'UP':  
        snakePos[1] -= 10  
    if direction == 'DOWN':  
        snakePos[1] += 10  
  
    snakeBody.insert(0, list(snakePos)) #snake mechanism  
  
    return snakePos, snakeBody
```



## Alimentando a la serpiente

```
If snakePos[0] == foodPos[0] and snakePos[1] == foodPos[1]:
    score += 6
    foodSpawm = False
else:
    snakeBody.pop()
    if foodSpawm == False:
        foodPos = [random.randrange(1, 72)*10, random.randrange(1, 46)*10]
        foodSpawm = True

for pos in snakeBody:
    pygame.draw.rect(playSurface, green,
        pygame.Rect(pos[0], pos[1], 10, 10))

pygame.draw.rect(playSurface, brown, pygame.Rect(foodPos[0], foodPos[1], 10, 10))
```



## Checando Limites

```
def boundaries(snakePos, snakeBody):  
    if snakePos[0] > 710 or snakePos[0] < 0:  
        gameOver()  
    if snakePos[1] > 450 or snakePos[1] < 0:  
        gameOver()  
  
    for block in snakeBody[1:]: # si se golpea a si misma  
        if snakePos[0] == block[0] and snakePos[1] == block[1]:  
            gameOver()
```



# Score!

```
def showScore(choice=1):  
    sFont = pygame.font.SysFont('comicsansms', 25)  
    sText = sFont.render('Score: {0}'.format(score), True, black)  
    sRect = sText.get_rect()  
    if choice == 1:  
        sRect.midtop = (80, 10)  
    else:  
        sRect.midtop = (360, 120)  
    playSurface.blit(sText, sRect)
```





# Gracias!