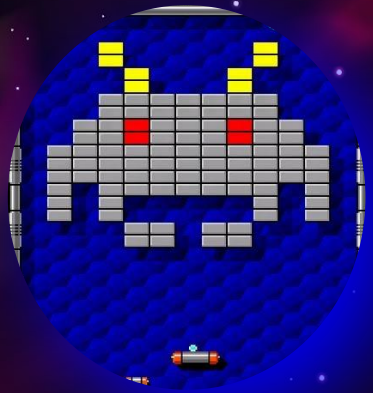
# Videojuego PYTHON

## ARKANOID



C.I.P DONAPEA 2M PRESENTACIÓN JUEGO

CHRISTOPHER ILLANES Y PATXI GOLDARAZ

# OBJETIVO

El objetivo final de este proyecto es el de crear un videojuego en PYTHON que sea parecido al mítico ARKANOID usando la biblioteca pygame.

- Crear ventana
- Añadir imágenes a ventana
- Añadir movimiento de Bola y Barra
- Mejoras posibles
- Creación de ladrillos
- Crear objetos con herencias
- Extras (Efecto nieve, música, pantalla game over)
- Sprites

Creadores de Arkanoid


# Creación del marco de la ventana

```
class Background_and_Music(py.sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.image = py.image.load("MagicBackground.png")
        self.rect = self.image.get_rect()
```

```
py.init()
VENTANAX, VENTANAY = 800,800
ventana = py.display.set_mode((VENTANAX,VENTANAY))
encabezado = py.display.set_caption("Snowball")
tiempo = py.time.Clock()
```

```
all_Sprites_Group.draw(ventana)
```

```
py.display.flip()
tiempo.tick(60)
```

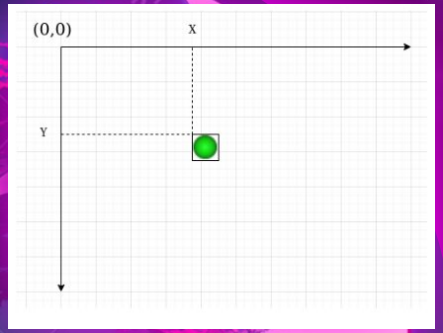
# Añadido de imágenes a la ventana

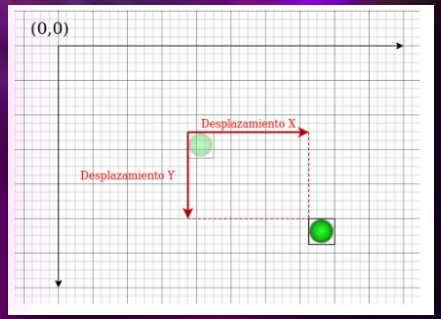
```
self.image = py.image.load("Borbuja.png")
self.image = py.transform.scale(self.image, (30,30))
self.rect = self.image.get_rect()
self.rect.center = ((VENTANAX //2, VENTANAY *0.3))
self.movimiento = [6,6]
```

```
all_Sprites_Group.draw(ventana)
```

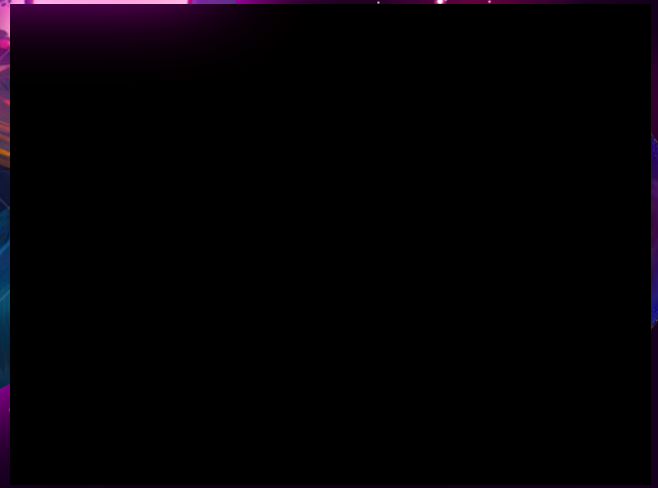
# Movimiento de los objetos

```python
class Esfera_and_GameOver(py.sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.image = py.image.load("Borbuja.png")
        self.image = py.transform.scale(self.image, (30,30)) #
        self.rect = self.image.get_rect()
        self.rect.center = ((VENTANAX //2, VENTANAY *0.3))
        self.movimiento = [6,6]
```



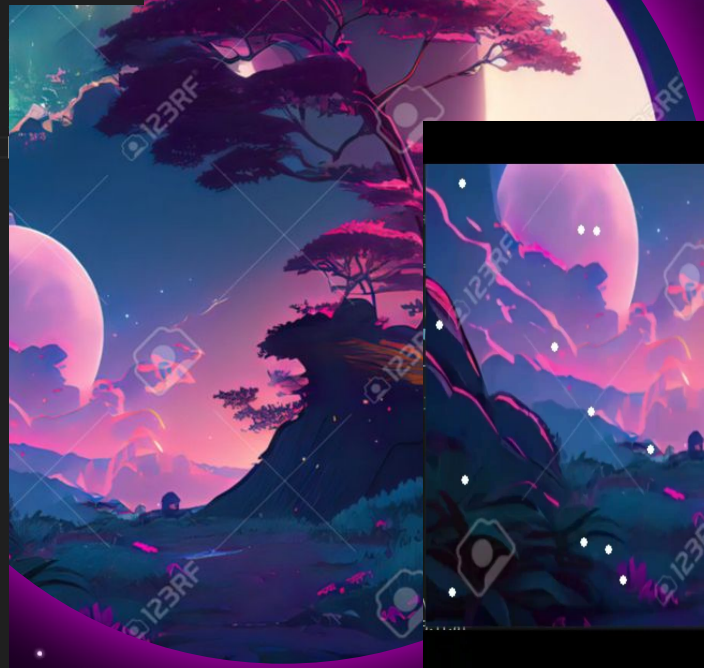

```python
def update(self):
    self.rect = self.rect.move(self.movimiento)
    if self.rect.left < 0 or self.rect.right > VENTANAX:
        self.movimiento[0] = -self.movimiento[0]

    if self.rect.top <0:
        self.movimiento[1] = - self.movimiento[1]
```

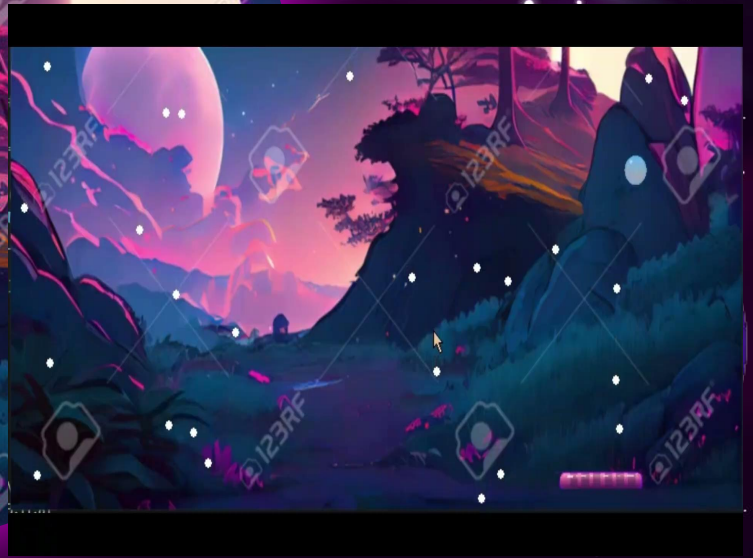


# Mejoras

```python
class BAR(py.sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.image = py.image.load("BARRR.png")
        self.image = py.transform.scale(self.image, (100,50))
        self.rect = self.image.get_rect()
        self.rect.center = ((VENTANAX//2, VENTANAY*0.97))
        self.movimiento = [4,4]
        self.timeL = None
        self.timeR = None
```

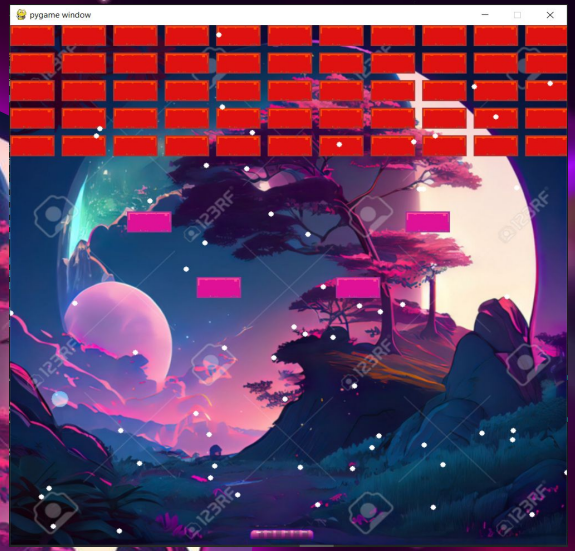
```python
def update(self):
    x = random.randint(5, 8)
    TIEMPO_BARRA = 0.5

    keys = py.key.get_pressed()
    if keys[py.K_LEFT]:
        tiempoL = py.time.get_ticks()/1000

        if self.timeL == 0:
            self.timeL = py.time.get_ticks()/1000

        if (tiempoL - self.timeL) > TIEMPO_BARRA:
            self.rect = self.rect.move(-15, 0)

        else:
            self.rect = self.rect.move(-x, 0)
    else:
        self.timeL = 0

    if keys[py.K_RIGHT]:
        tiempoR = py.time.get_ticks()/1000

        if self.timeR == 0:
            self.timeR = py.time.get_ticks()/1000

        if (tiempoR - self.timeR) > TIEMPO_BARRA:
            self.rect = self.rect.move(15,0)
        else:
            self.rect = self.rect.move(x,0)

    else:
        self.timeR = 0

    if self.rect.colliderect(esfera.rect):
        esfera.movimiento[1] = -esfera.movimiento[1]
```





# Creación de ladrillos

```python
class Ladrillo(py.sprite.Sprite):
    def __init__(self,x, y):
        super().__init__()
        self.image = py.image.load("brick.png")
        # self.image = py.transform.scale(self.image, (70,32))
        self.rect = self.image.get_rect(topleft = (x,y))
```




```python
'''Creación de los ladrillos mediante 2 bucles For.'''
Ladrillos_Group = py.sprite.Group()
FILAS = 5
COLUMNAS = 11
ANCHO_ladrillo = 64
ALTO_ladrillo = 32
ESPACIO_entre_ladrillos = 10
for fila in range(FILAS):
    for columna in range(COLUMNAS):
        x = columna * (ANCHO_ladrillo + ESPACIO_entre_ladrillos )
        y = fila * (ALTO_ladrillo + ESPACIO_entre_ladrillos )

        ladrillo = Ladrillo(x, y)
        Ladrillos_Group.add(ladrillo)
        all_Sprites_Group.add(ladrillo)
```

```python
'''Eliminación de los ladrillos al rebote'''
colisiones = py.sprite.spritecollide(esfera, Ladrillos_Group, True)
if colisiones:
    esfera.movimiento[1] = -esfera.movimiento[1]
    background.Play_Impact()
    for i in colisiones:
        SCORE += 1
        print(SCORE)
        if SCORE == 20:
            esfera.movimiento = [7,7]
            esfera.rect.move(esfera.movimiento)
            if SCORE == 40:
                esfera.movimiento = [8,8]
                esfera.rect.move(esfera.movimiento)
```

# Creación de objetos con herencia

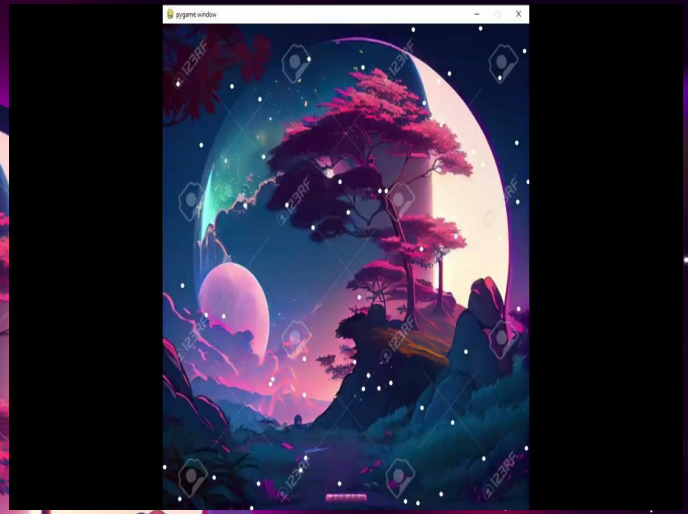
```
class Ladrillo_Irrompible(Ladrillo):
    def __init__(self,x,y,irrompibleX,irrompibleY):
        super().__init__(x,y)
        self.x = x
        self.y = y
        self.imagen = py.image.load("brick.png")
        self.image.fill((150,0,150,200), special_flags=py.BLEND_RGBA_MAX)
        self.rect = self.image.get_rect()

        self.picture = py.image.load("brick.png")
        self.picture.fill((150,0,150,200), special_flags=py.BLEND_RGBA_MAX)
        self.rectangulo1 = self.picture.get_rect()

        self.irrompibleX = irrompibleX
        self.irrompibleY = irrompibleY
```

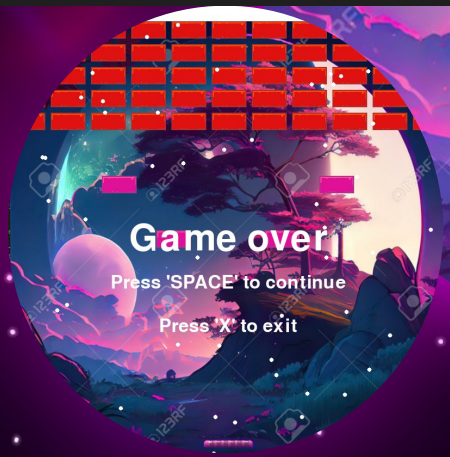
```
ladrillo_irrompible1 = Ladrillo_Irrompible(200,300, 600,300)
ladrillo_irrompible2 = Ladrillo_Irrompible(300,400, 500,400)
```

# Extras

```
''' Efecto Nieve '''
for j in lista_nieve:
    py.draw.circle(ventana,(255,255,255), (j[0],j[1]),4)
    j[1] += 1
    if j[1] > VENTANAY:
        j[1] = 0
```

```
'''Lista para guardar números aleatorios para generar el Efecto Nieve'''
lista_nieve = []
for snow in range(60):
    x = random.randint(0, VENTANAX)
    y = random.randint(0, VENTANAY)
    lista_nieve.append([x,y])
```






# Game Over and Winner

```
self.Pantalla1 = py.font.Font(None, 100)
self.Pantalla2 = py.font.Font(None, 50)
self.Pantalla3 = py.font.Font(None, 50)
self.Pantallawinner = py.font.Font(None, 150)
```

```
def GameOver(self):

    self.text1 = self.Pantalla1.render(" Game over ",True,(255,255,255))
    self.text2 = self.Pantalla2.render("Press 'SPACE' to continue", True, (255,255,255))
    self.text3 = self.Pantalla2.render("Press 'X' to exit", True, (255,255,255))
    self.texto4 = self.Pantallawinner.render("YOU WIN !0_0!", True, (255,255,255))

    if self.rect.bottom > VENTANAY:

        textorect = self.text1.get_rect()
        textorect.center = ((VENTANAX * 0.5, VENTANAY * 0.5))
        ventana.blit(self.text1,textorect)

        textorect = self.text2.get_rect()
        textorect.centerx = VENTANAX //2
        textorect.centery = VENTANAY *0.6
        ventana.blit(self.text2,textorect)


        textorect = self.text3.get_rect()
        textorect.center = ((VENTANAX//2, VENTANAY*0.7))
        ventana.blit(self.text3,textorect)

        background.Pause_music()
        background.Play_GameOver()                                    #Com

        keys = py.key.get_pressed()
        if keys[py.K_SPACE]:
            self.rect.centerx = VENTANAX //2
            self.rect.centery = VENTANAY *0.3

            background.Stop_GameOver()
            background.Unpause_Music()
```

```
def Winner(self):
    ventana.blit(self.win, self.rectan)
    winnerect = self.texto4.get_rect()
    winnerect.center = ((VENTANAX//2, VENTANAY//2))
    ventana.blit(self.texto4,winnerect)
```

```
esfera.GameOver()
if SCORE == 55:
    esfera.Winner()
    if esfera.rect.bottom >VENTANAY *0.9:
        esfera.movimiento[1] = - esfera.movimiento[1]
    for j in lista_nieve:
        py.draw.circle(ventana,(255,255,255), (j[0],j[1]),4)
        j[1] += 1
        if j[1] > VENTANAY:
            j[1] = 0
```

# Sprites

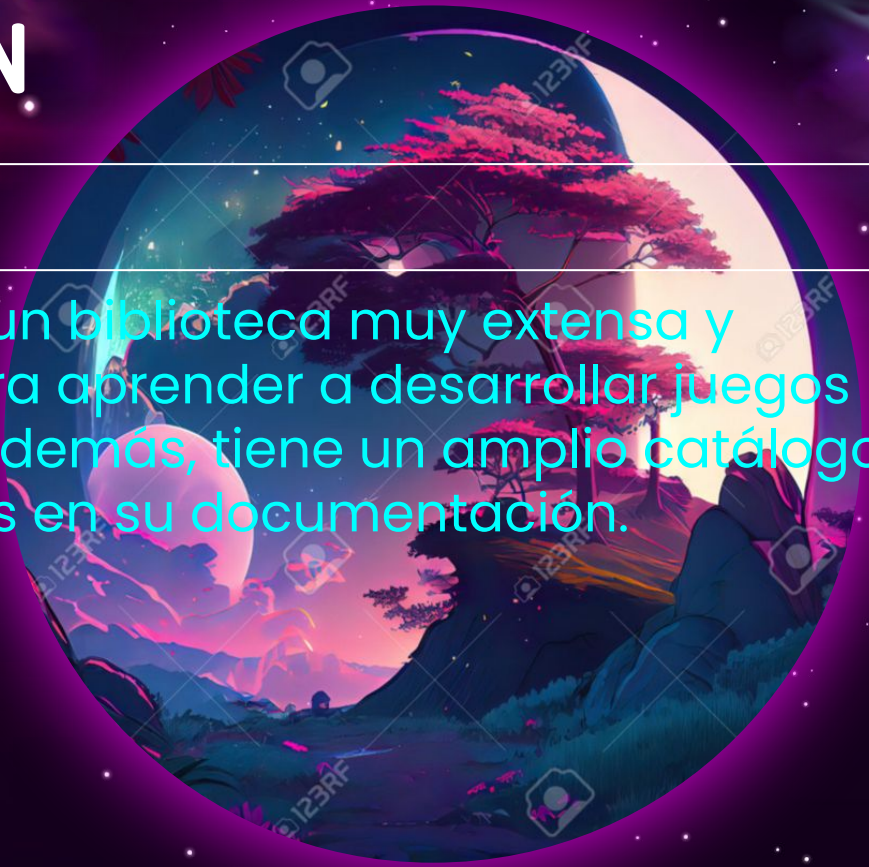
```
> class Background_and_Music(py.sprite.Sprite): ···

> class Esfera_and_GameOver(py.sprite.Sprite): ···

> class BAR(py.sprite.Sprite): ···

> class Ladrillo(py.sprite.Sprite): ···

> class Ladrillo_Irrompible(Ladrillo): ···
```

```
py.init()
VENTANAX, VENTANAY = 800,800
ventana = py.display.set_mode((VENTANAX,VENTANAY))
encabezado = py.display.set_caption("Snowball")
tiempo = py.time.Clock()

all_Sprites_Group = py.sprite.Group()

background = Background_and_Music()
esfera = Esfera_and_GameOver()
bar = BAR()

all_Sprites_Group.add(background)
all_Sprites_Group.add(esfera)
all_Sprites_Group.add(bar)


ladrillo_irrompible1 = Ladrillo_Irrompible(200,300, 600,300)
ladrillo_irrompible2 = Ladrillo_Irrompible(300,400, 500,400)
```

```
SCORE = 0
'''Bucle del juego'''
jugando = True
while jugando:
    for event in py.event.get():
        if event.type == py.QUIT:
            jugando = False

    all_Sprites_Group.draw(ventana)

    '''Eliminación de los ladrillos al rebote'''
    colisiones = py.sprite.spritecollide(esfera, Ladrillos_Group, True)
    if colisiones:
        esfera.movimiento[1] = -esfera.movimiento[1]
        background.Play_Impact()
        for i in colisiones:
            SCORE += 1
            print(SCORE)
            if SCORE == 20:
                esfera.movimiento = [7,7]
                esfera.rect.move(esfera.movimiento)
                if SCORE == 40:
                    esfera.movimiento = [8,8]
                    esfera.rect.move(esfera.movimiento)

    ladrillo_irrompible1.COLISIÓN()
    ladrillo_irrompible2.COLISIÓN()

    esfera.GameOver()
    if SCORE == 55:
        esfera.Winner()
        if esfera.rect.bottom >VENTANAY *0.9:
            esfera.movimiento[1] = - esfera.movimiento[1]
        for j in lista_nieve:
            py.draw.circle(ventana,(255,255,255), (j[0],j[1]),4)
            j[1] += 1
            if j[1] > VENTANAY:
                j[1] = 0
```

```
all_Sprites_Group.update()
```

# RESUMEN



Pygame es un biblioteca muy extensa y práctica para aprender a desarrollar juegos en 2D y 3D.Además, tiene un amplio catálogo de funciones en su documentación.