

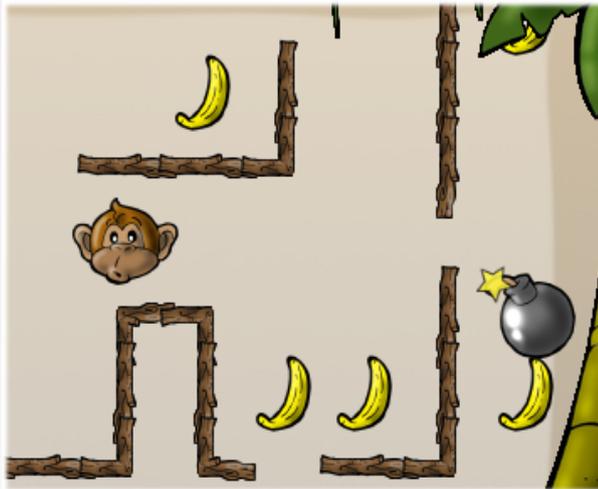
# Como desarrollar juegos con **Python y Pygame**

**Autor:** Hugo Ruscitti  
**Email:** [hugoruscitti@gmail.com](mailto:hugoruscitti@gmail.com)  
**Web:** <http://www.losersjuegos.com.ar>  
**Evento:** CaFeCONF 2007  
**Fecha:** 6 de Septiembre del 2007  
**Licencia:** Creative Commons

**Resumen:** Se mostrará como construir videojuegos en 2 dimensiones utilizando el lenguaje de programación Python y la biblioteca Pygame.

# Índice de contenidos

- ¿Por qué desarrollar videojuegos?
- Herramientas a utilizar.
- Veamos como hacer un juego paso a paso.



- Otros proyectos interesantes.
- Preguntas.

# ¿Por qué desarrollar videojuegos?

- Hacer juegos permite:
  - aprender acerca de tecnología.
  - conocer el funcionamiento de los programas.
  - encontrar una forma práctica usar matemáticas.
  - constituye una actividad muy constructiva.
- ahora entre nosotros, permite:
  - dar vida a un montón de ideas locas.
  - compartir tu experiencia de aprendizaje.
  - es todo un desafío.
  - diversión, mucha diversión...

# Herramientas a utilizar

- **python**, como lenguaje de programación.



- y **pygame** como biblioteca multimedia.



# Python, como sugerencia

- **python** es:
  - Un lenguaje de programación.
  - dinámico e interpretado.
  - multiplataforma (GNU/Linux, Windows, Mac)
  - es adecuado para casi todo proyecto.
- ¿Por qué **python**?
  - es fácil de aprender.
  - excelente documentación.
  - promueve código elegante, simple y legible.
  - puedes adaptar tu programa al cambio.

# Pygame, una biblioteca multimedia

- **pygame** se encarga de gestionar:
  - imágenes en formato PNG, BMP, PCX, TGA ...
  - sistemas de sonido, formatos MOD, OGG, MP3...
  - operaciones relacionadas con el gestor de ventana.
  - eventos de aplicación y dispositivos de entrada como mouse, teclado y joystick.
  - temporizadores.
  - Colisiones, sistema de Sprites (objetos de un juego).

# Un juego paso a paso

- Es simple, pero tiene todo lo que la mayoría de los juegos tiene.

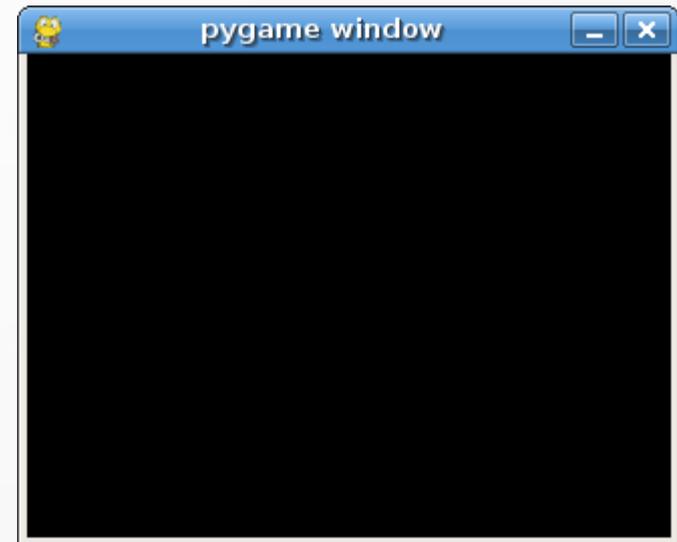


# 1 - Como crear una ventana

- Incorporar el módulo “pygame”.
- Utilizar la función “set\_mode” con el tamaño de pantalla.

```
import pygame  
pygame.display.set_mode((320, 240))
```

*código*



*resultado*



Hay ejemplo...

## 2 - Manejo de imágenes en pygame

- La función “load” permite incorporar gráficos a partir de archivos BMP, PNG, JPEG... etc.
- “load” genera un objeto “Surface” que representará a la imagen en la memoria del equipo.
- El retorno de “set\_mode” también es una superficie, pero esta representa lo que veremos en pantalla.

## 2 - Como crear objetos Surface

```
logotipo = pygame.image.load("cafeconf_2007.png")
```

*código*

```
fondo = pygame.image.load("escenario.jpg")
```

*código*

logotipo →



fondo →



## 2 - El método Blit

- Se utiliza (generalmente) para dibujar en pantalla.
- “blit” recibe la superficie a imprimir y su posición.
- La posición consiste en una coordenada (x, y).

```
screen.blit(mono, (0, 0))
```

```
screen.blit(mono, (300, 300))
```

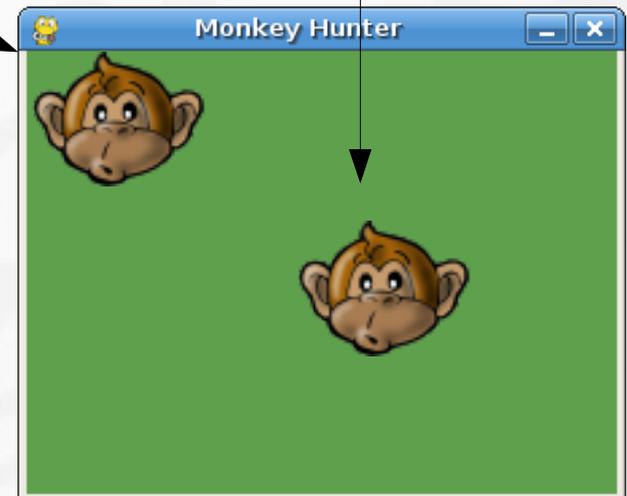
mono →



screen →

*posición (0, 0)*

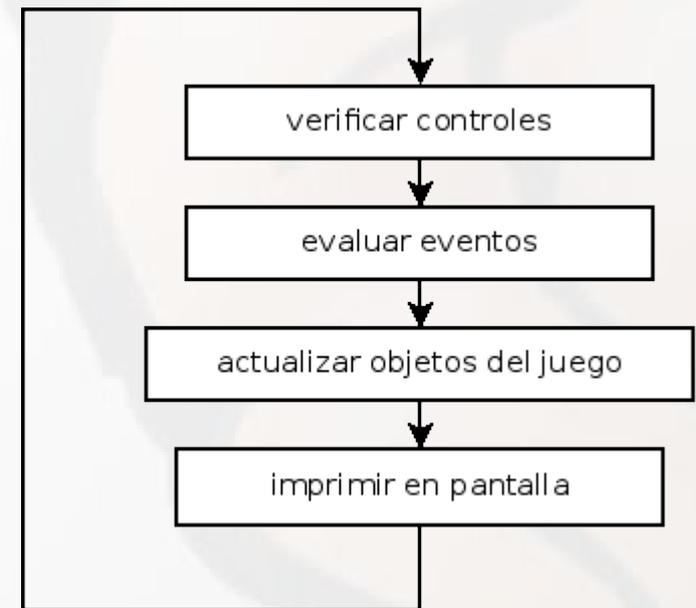
*posición (300, 300)*



# 3 - ¿ y como realizamos movimientos ?

- Los juegos generalmente utilizan un bucle de repetición (llamado “main loop”).
- Ejecuta pequeñas operaciones muy rápidamente.

```
while not salir:  
    pos_x = pos_x + 1  
  
    screen.blit(mono, (pos_x, pos_y))  
    pygame.display.flip()  
  
    # restauramos el fondo  
    screen.blit(fondo, (0, 0))
```



# 3 - ¿ y como realizamos movimientos ?

- En nuestro casos podemos cambiar poco a poco la posición del Mono en el escenario.



posicion = (100, 300)



posicion = (400, 300)



posicion = (600, 300)



Hay ejemplo...

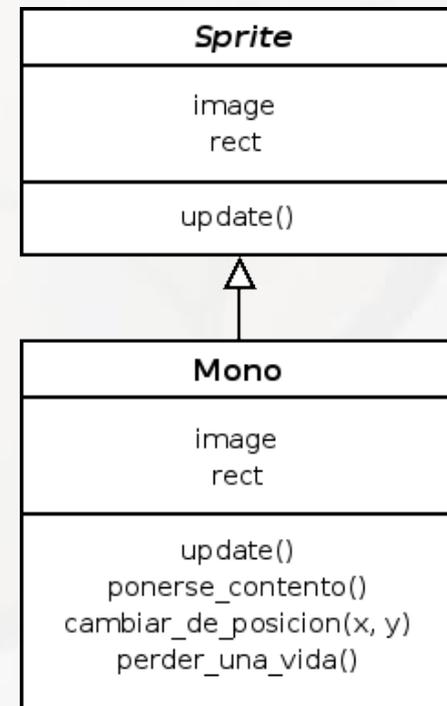
Como desarrollar juegos con **Python** y **Pygame**

## 4 - Sprite, como base para el Mono.

- Agrupa todo lo relacionado con el personaje, atributos, comportamiento.
- El método “update” contiene el comportamiento del personaje.

```
class Mono(Sprite):  
  
    def __init__(self):  
        self.image = pygame.image.load("mono.png")  
        self.rect = self.image.get_rect()  
        self.rect.move_ip(200, 300)  
  
    def update(self):  
        self.rect.x += 5
```

*código*



*diagrama*

# 4 - Detectando la pulsación de teclas

- El módulo “key” contiene una función llamada “get\_pressed”.
- “get\_pressed” nos informa el estado completo del teclado en un instante del juego.

**Ejemplo: como podemos manejar al Mono del juego**

```
def update(self):  
    teclas = pygame.key.get_pressed()  
  
    if teclas[K_LEFT]:  
        self.rect.x -= 10  
    elif teclas[K_RIGHT]:  
        self.rect.x += 10
```

*código*



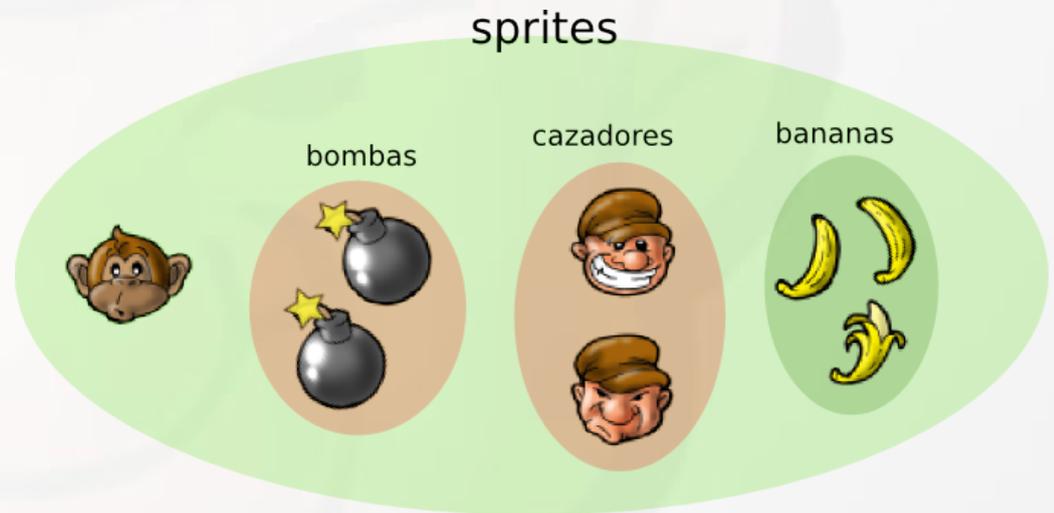
# 5 - Grupos

- La clase Group permite agrupar y clasificar sprites.
- **pygame** facilita así la administración de objetos.
- Para nuestro ejemplo podemos crear 4 grupos:

```
from pygame.sprite import Group

sprites = Group()
bananas = Group()
bombas = Group()
cazadores = Group()

sprites.add([bananas, bombas])
sprites.add([mono, cazadores])
```



# 5 - Colisiones de grupos.

- El módulo “sprite” incluye varias funciones para controlar colisiones entre objetos “Sprite”.
- La colisión se evalúa en base a los atributos “rect” de cada “Sprite”, aunque se puede modificar...

## Algunas posibilidades:

### **pygame.sprite.groupcollide (...)**

Colisiones entre miembros de dos grupos.

### **pygame.sprite.spritecollide (...)**

Detecta todas las colisiones entre un *sprite* y miembros de un grupo.

### **pygame.sprite.spritecollideany (...)**

Detecta la primer colisión entre un *sprite* y miembros de un grupo

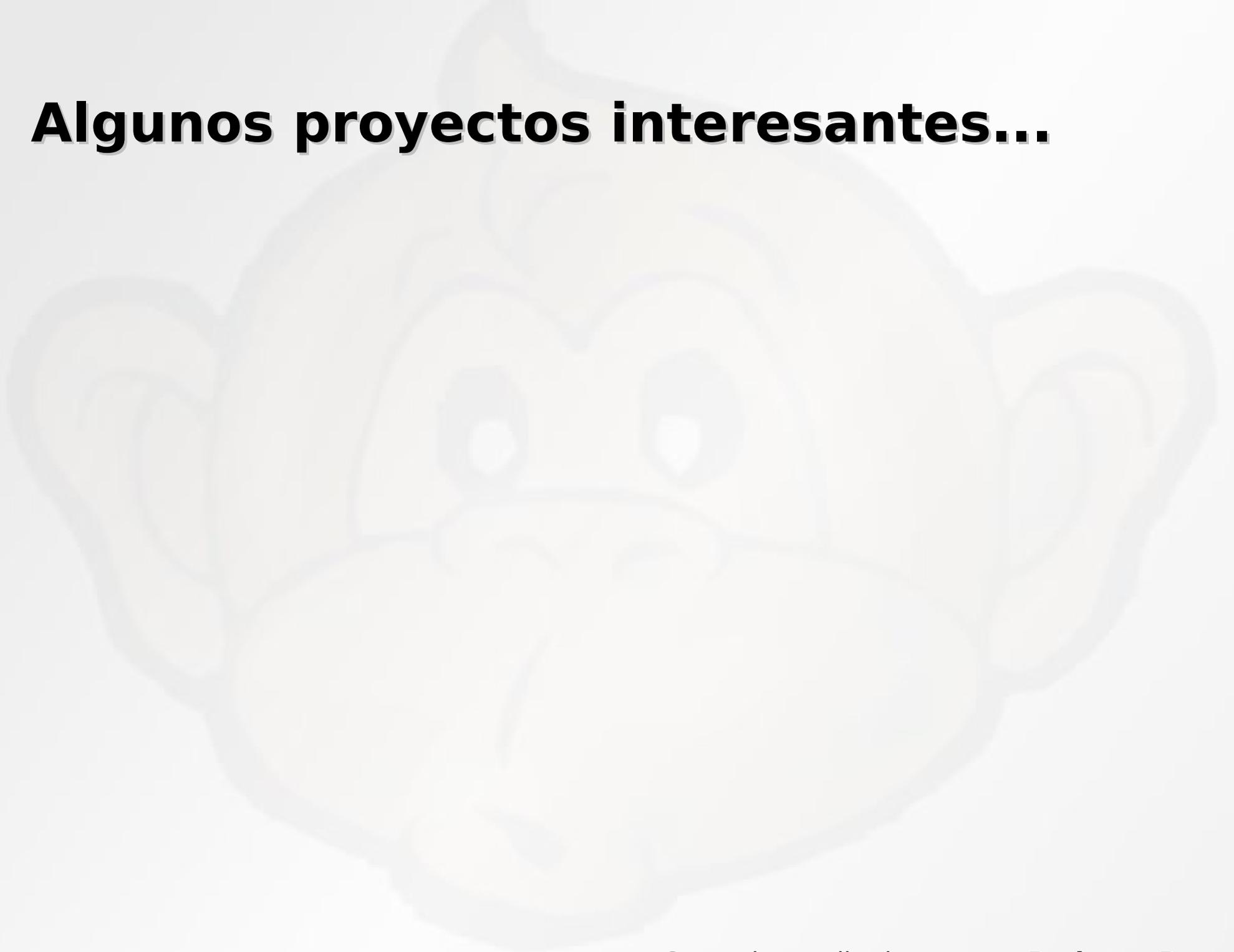


## 6 - Otras mejoras

- Una decoración alrededor del escenario.
- Un laberinto de troncos.
- Enemigos con autonomía (IA bastante trucha...).
- y movimientos en bloque...



# Algunos proyectos interesantes...



# Algunos proyectos interesantes...

## PyAr

- PyAr es el grupo de usuarios de Python en Argentina.
- Suelen participar de la competencia internacional *pyweek*.
- ... casualmente darán una charla a continuación en el **Mini Auditorio** (llamada “*pyweek un juego en 7 días*”).



# Algunos proyectos interesantes...

## LosersJuegos

- Es un proyecto que promueve el desarrollo de juegos Libres.
- Realizamos artículos, traducciones de documentación al español y juegos.
- Contamos con un sitio web, un foro y un wiki cooperativo.
- También armamos un CD ...



Losers  
Juegos



# Preguntas...



## **Programación y parloteo:**

Hugo Ruscitti - <[hugoruscitti@gmail.com](mailto:hugoruscitti@gmail.com)>



## **Gráficos e ideas locas:**

Walter Velazquez - <[wgv\\_tattoo@yahoo.com.ar](mailto:wgv_tattoo@yahoo.com.ar)>

**Web**

<http://www.losersjuegos.com.ar>