



FACULTAD DE
INGENIERIA
UNIVERSIDAD NACIONAL DE JUJUY

SECRETARIA DE EXTENSION
COORDINACION Y
PLANIFICACION

TALLER VIDEOJUEGOS en 2D

San Salvador de Jujuy

-

JUJUY

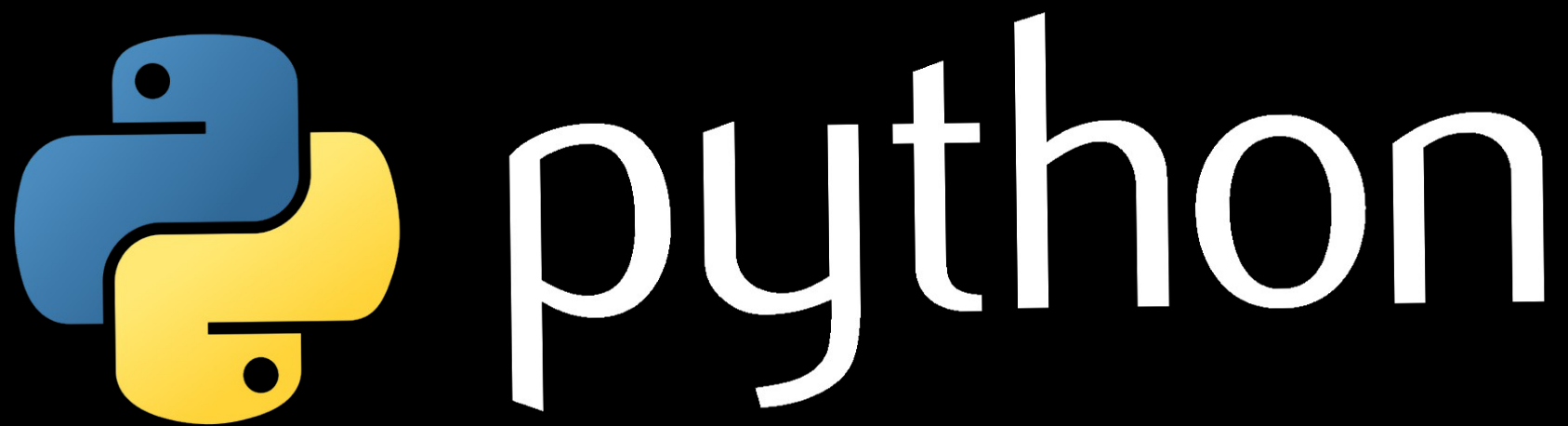
-

ARGENTINA

-

2017





El Laboratorio de Computación Científica

Presenta

Introducción a:



Aprenderemos sobre:

Lenguaje de programación Python: su historia, tipos de datos, estructuras de control, funciones, organización lógica, manejo de clases y objetos.

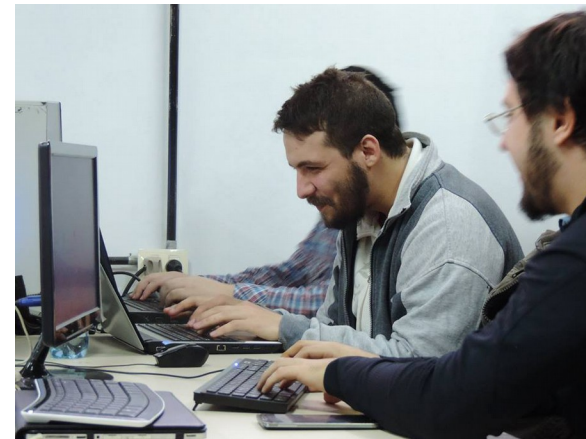
Pygame: su historia, definir superficie de trabajo, insertar un objeto en una superficie, manejo de eventos, desplazamiento automático de un objeto sobre una superficie (Timer), manejo de objetos por medio del teclado, manejo de objetos por medio del mouse, colisión entre objetos.

Mayo
20

18:00



Requisitos: ganas de aprender y conocimientos básicos en programación orientada a objetos.





¡Hoy!



SECRETARIA DE EXTENSION
COORDINACION Y
PLANIFICACION

TALLER VIDEOJUEGOS en 2D

Introducción
a Game Maker

Disertante:
Adrián Maximiliano Galán
Fac. de Cs. Exactas y
Tecnología - UNT

OBJETIVOS

Atraer estudiantes al ámbito del
desarrollo de videojuegos en 2D
Promover el uso de lenguajes
de programación especializados

PYGAME desde
cero

Disertante:
Lic. Jorge Habib Namour
Fac. de Cs. Exactas y
Tecnología - UNT

DESTINADO A

Estudiantes y Profesores de carreras
orientadas a la programación
Publico en general interesado
en el desarrollo de videojuegos



VIERNES 21
de Abril de 2017
de 15 a 19 hs.

AULA VIRTUAL
Gorriti 237
San Salvador de Jujuy



Entrada Libre y Gratuita
Cupos limitados

Más información e inscripciones a cursos.fi.unju@gmail.com



- Un poco de historia de los progenitores de nuestra “plataforma de trabajo” actual.
- Utilizando Python y Pygame.
- Introducción y paseo (:D) por algunas funcionalidades de Pygame.

Durante las prácticas, también hablaremos de cuestiones relacionadas con Python.



Guido van Rossum

1991



Sam Lantinga

1998



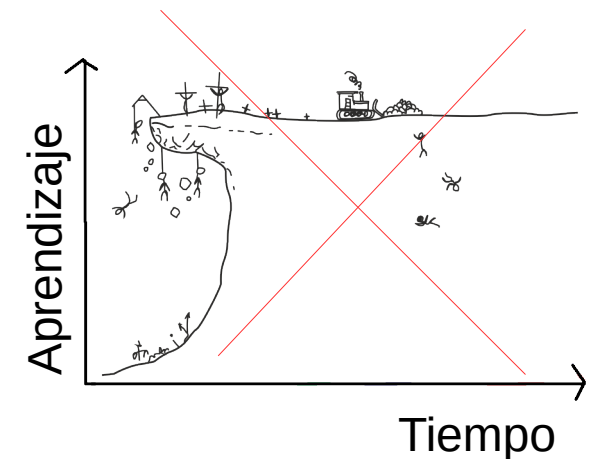
Peter Shinnars

2001



[¿Qué hay de Python?...]

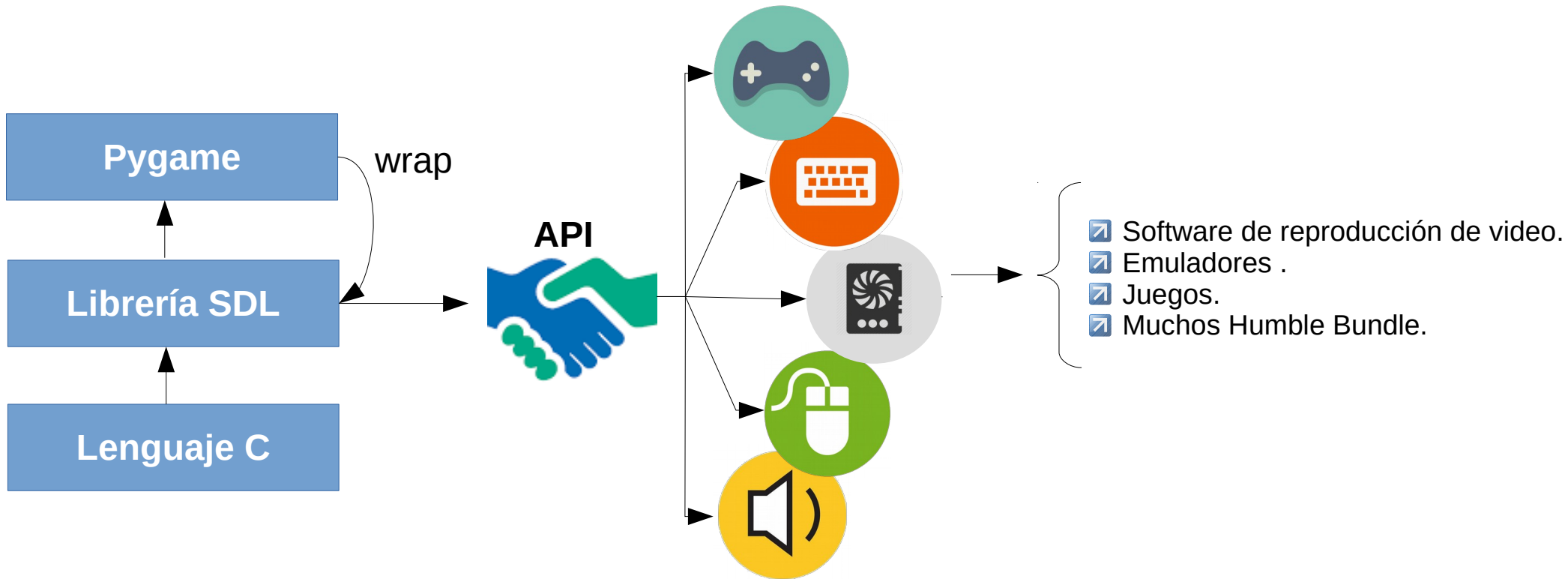
- Lenguaje de programación interpretado y compilado.
- Extensión de archivo “.py”.
- Es un lenguaje alto nivel.
- Se basa en la notación Pythonica. Los espacios en blanco son significativos.
 - La indentación denota los bloques de código. Utilizar 4 espacios (**PEP 8**).
- Multiparadigma. Paradigmas que soporta: imperativo, funcional, orientado a objetos.
- Multiplataforma.
- Tipado dinámico.
- Fuertemente tipado.
- Mantenido por la PSF (Python Software Foundation).



[¿Qué hay de SDL?...]



[¿Qué hay de SDL?...]



IMPORTANTE: SDL no es ni 2d, ni 3d, sino que puede trabajar con gráficos 3d (OpenGL) y 2d.

[¿Qué hay de Pygame?...]

- Conjunto de módulos de Python.
- No requiere OpenGL.
- Procesamiento multi núcleo.
- Utiliza código C optimizado.
- Portátil.
- Es simple y fácil de usar.
- No requiere GUI.
- Respuesta rápida a errores reportados (:foros gamers:).
- Licencia GPL: puede crear código abierto, libre, juegos, comerciales.

Página oficial: <http://pygame.org>



Módulo



[Pygame Home](#) || [Help Contents](#) || [Reference Index](#)

[BufferProxy](#) | [camera](#) | [cdrom](#) | [Color](#) | [cursors](#) | [display](#) | [draw](#) | [event](#) | [examples](#) | [font](#) | [freetype](#) | [gfxdraw](#) | [image](#) | [joystick](#) | [key](#) | [locals](#)
| [mask](#) | [math](#) | [midi](#) | [mixer](#) | [mouse](#) | [music](#) | [Overlay](#) | [PixelArray](#) | [pixelcopy](#) | [pygame](#) | [Rect](#) | [scrap](#) | [sndarray](#) | [sprite](#) | [Surface](#) |
| [surfarray](#) | [tests](#) | [time](#) | [transform](#) | [version](#)

Clase



¿Qué es un evento?

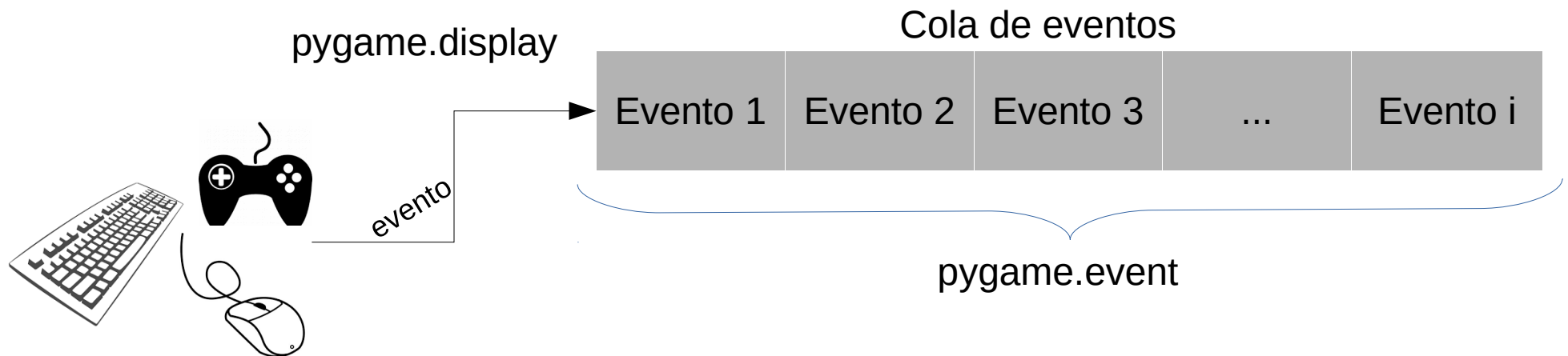
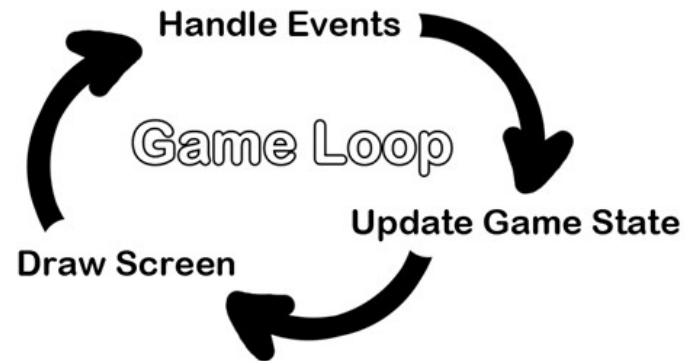
Suceso que ocurre en el sistema, definidos por el usuario o que ellos mismos provocan.

¿Cómo maneja los eventos Pygame?

Pygame captura todos los eventos y los almacena en una cola (cola de eventos).

Todos los eventos poseen un identificador.

Un usuario puede generar sus propios eventos.



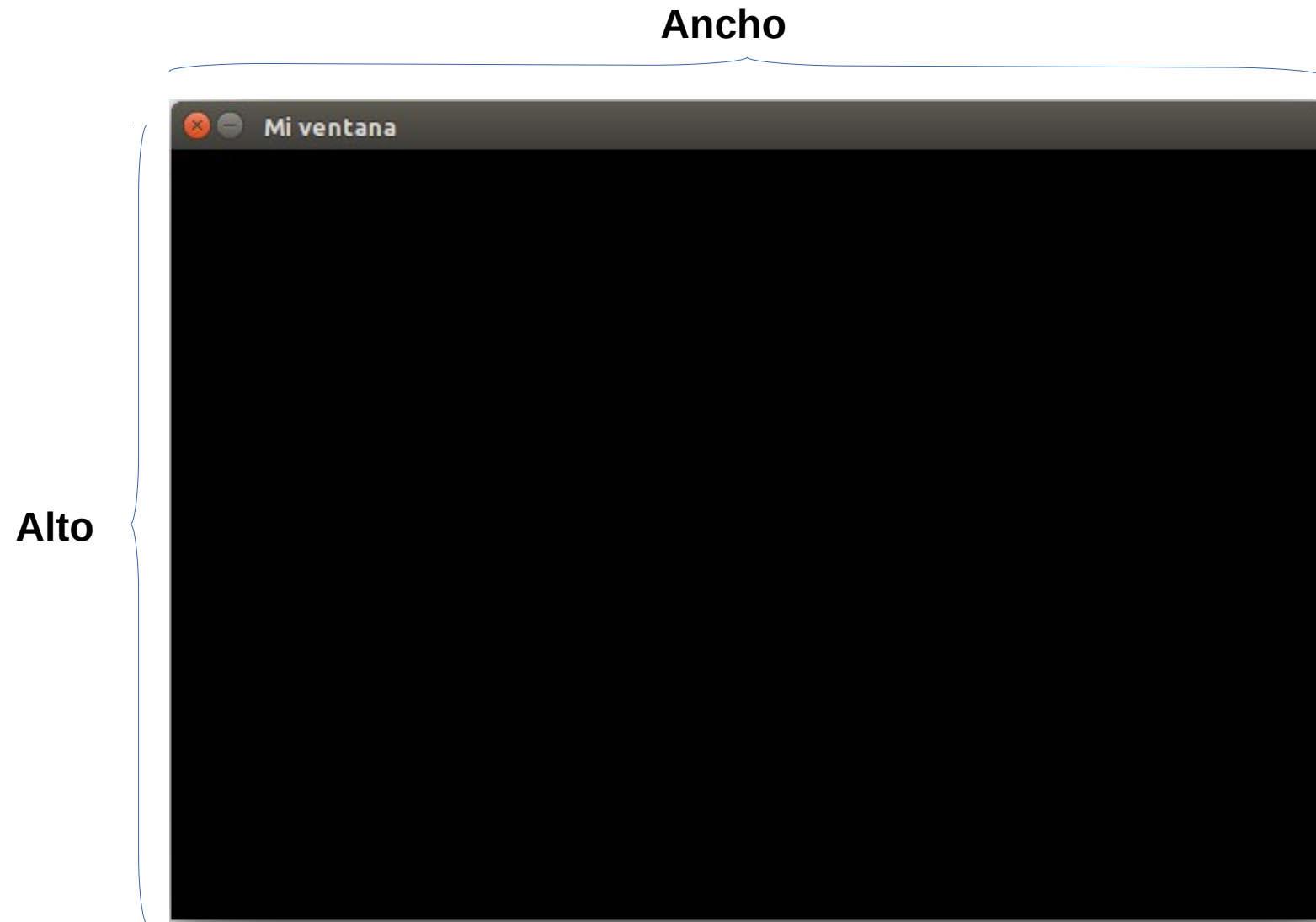
```
...  
tama = ancho, alto = [500, 500]  
s1 = pygame.display.set_mode(tama)  
while True:  
    for evento in pygame.event.get():  
        ...
```

Primeras prácticas:

- Crear una ventana (o superficie).
- Escape de una superficie.
- Insertar objeto en superficie.
- Re dimensionar objeto (imagen).
- Desplazamiento automático de un objeto.
- Desplazamiento de un objeto utilizando teclado.
- Colisión entre objetos.
- Girar objeto.

Juego a mostrar:

- ---- Snake ---->



Utilizamos el módulo “display”:

set_mode() → Surface

set_caption() → None

```
import pygame as pg
import sys
```

```
tama = ancho, alto = 600, 400
sup = pg.display.set_mode(tama)
pg.display.set_caption("Mi ventana")
```

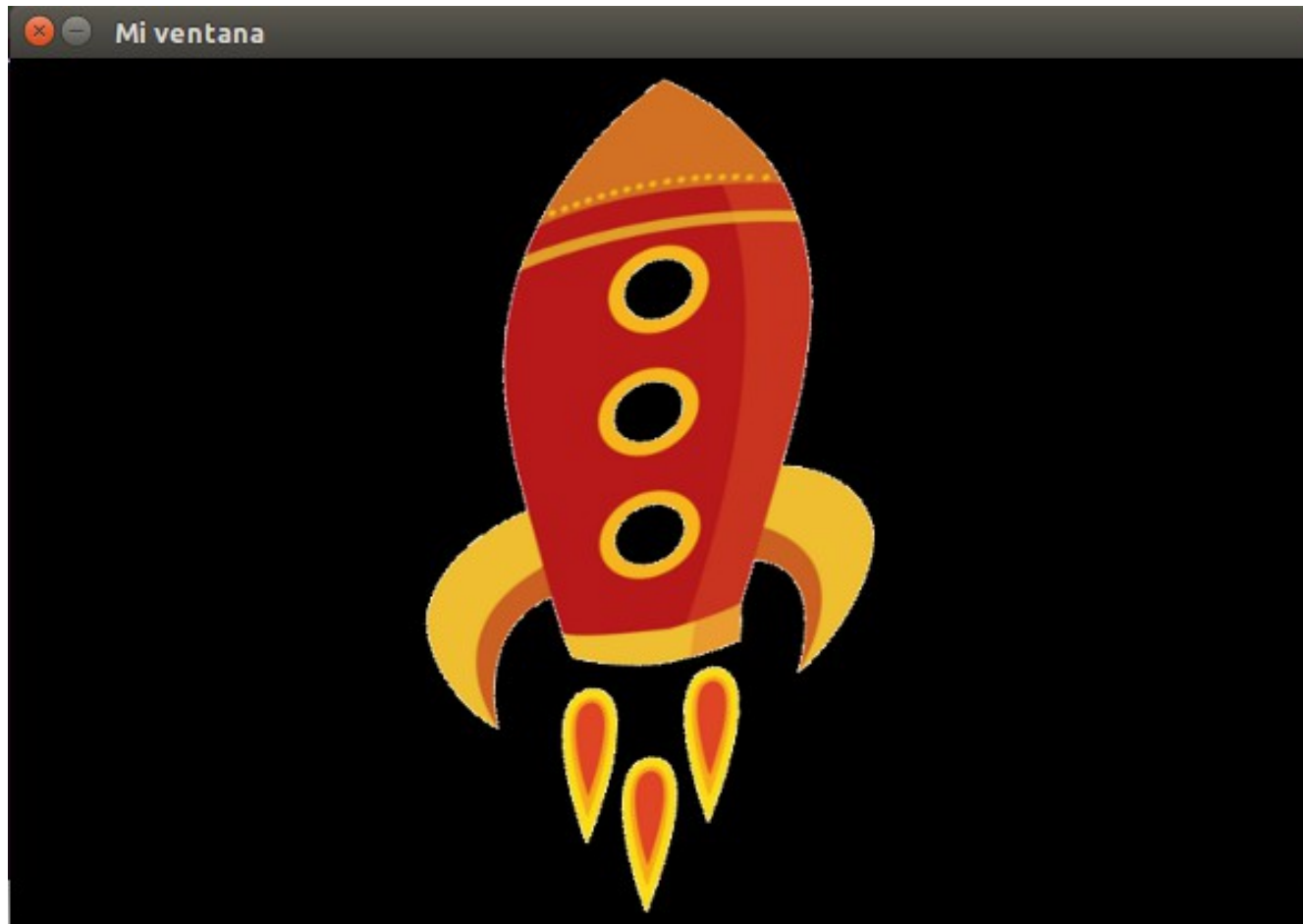
```
while True:
    pass
```

[Manejo de eventos] Accedemos a la cola de eventos, para esto necesitamos la ayuda del módulo “event”:

event.get() → Eventlist	event.EventType → Event type
-------------------------	------------------------------

▲ ID de Evento SDL

```
...  
for evento in pg.event.get():  
    if evento.type == pg.QUIT:  
        pg.quit()  
        sys.exit()  
...
```



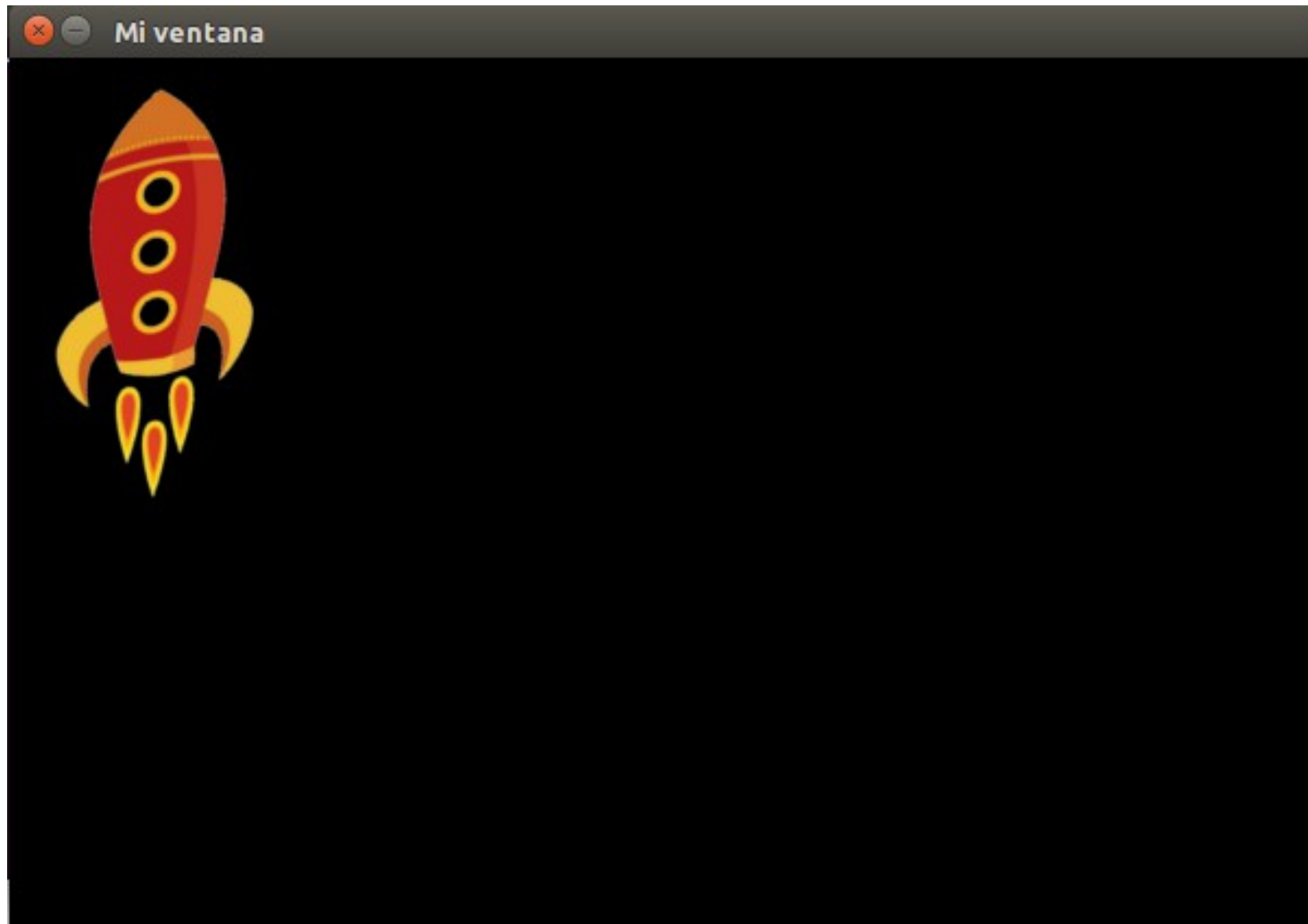
Utilizamos el módulo “image”, “display” y la clase “Surface”:

image.load() → Surface

display.update() → None

Surface.blit() → Rect

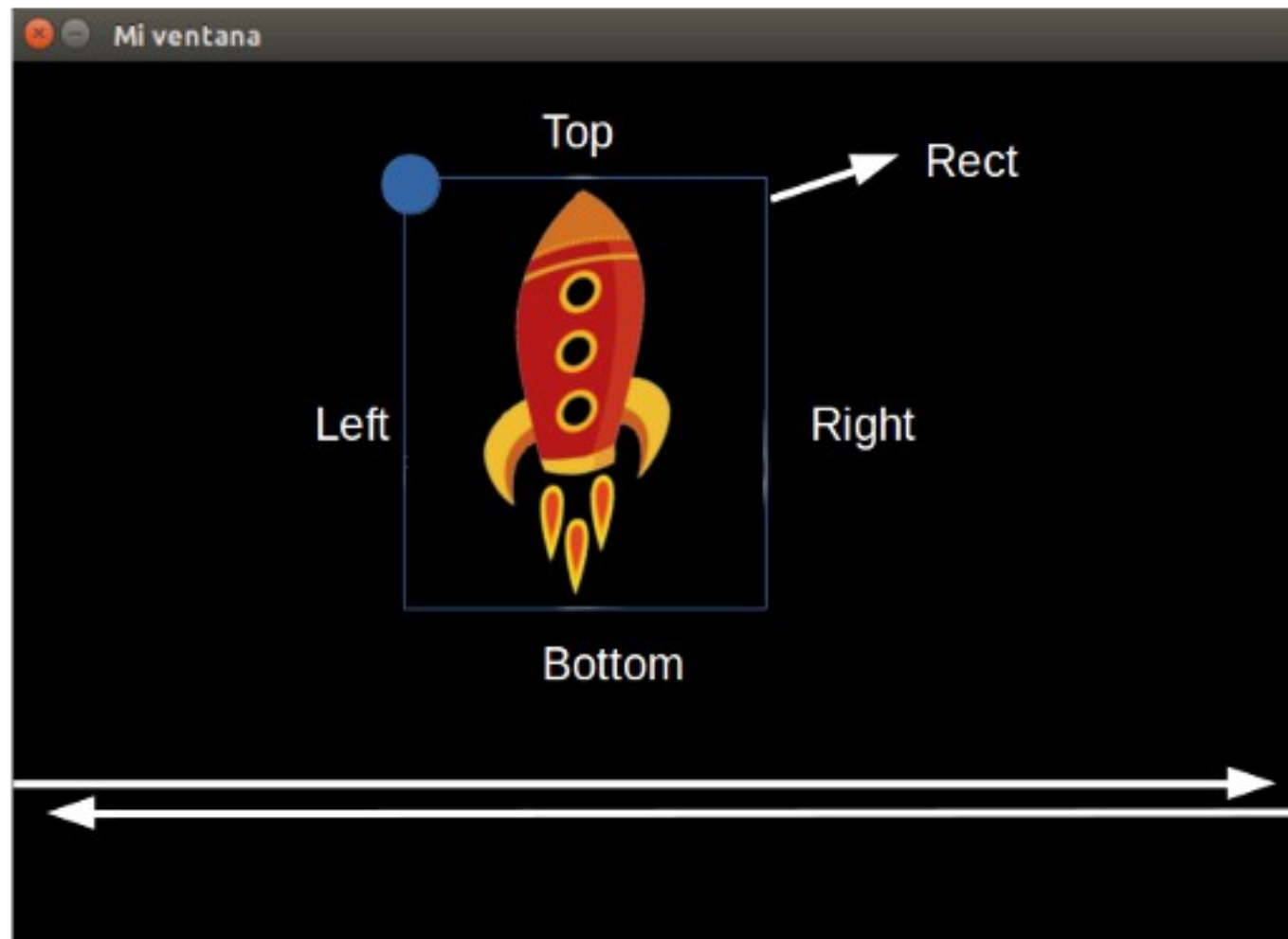
```
...  
obj = pg.image.load("nave.png")  
sup.blit(obj, (0,0))  
...  
pg.display.update()  #Al final del game loop
```



Utilizamos el módulo “transform”:

`transform.scale()` → Surface

```
...  
obj = pg.transform.scale(obj, (200,200))  
...
```



Utilizamos las clases “Surface”, “Rect” y el módulo “time”:

Surface.get_rect() → Rect	Rect.top/left/bottom/right → Int	Rect.move() → Rect
Surface.fill() → Rect	time.delay → Clock	

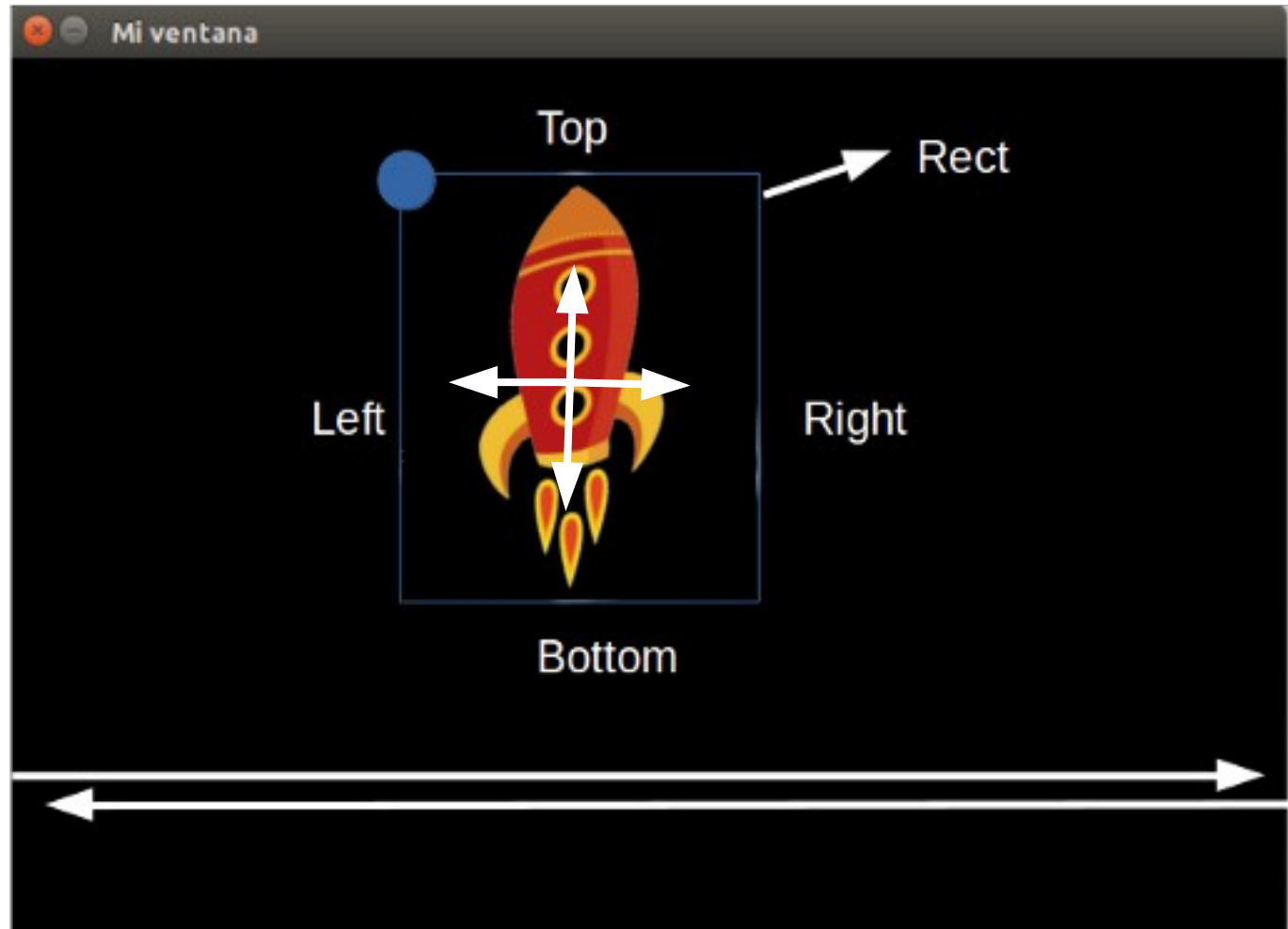
```
...
colorFondo = (0,0,0)
rectObj = obj1.get_rect()
rectObj.left = 50
rectObj.top = 50
desplazamiento = 5

...
if rectObj.left < 0 or rectObj.right > tama[0]:
    desplazamiento *= -1

rectObj = rectObj.move((desplazamiento, 0))

ventana.fill(colorFondo)
ventana.blit(obj1, rectObj)

...
```



Nuestro caso: el teclado

La cola de eventos recibe eventos `pygame.KEYDOWN` y `pygame.KEYUP` cuando los botones del teclado se presionan y se liberan. Ambos eventos tienen un atributo clave que es un identificador entero que representa cada tecla del teclado.

Utilizamos:

`event.__dict__` → dict

`pygame.KEYDOWN`

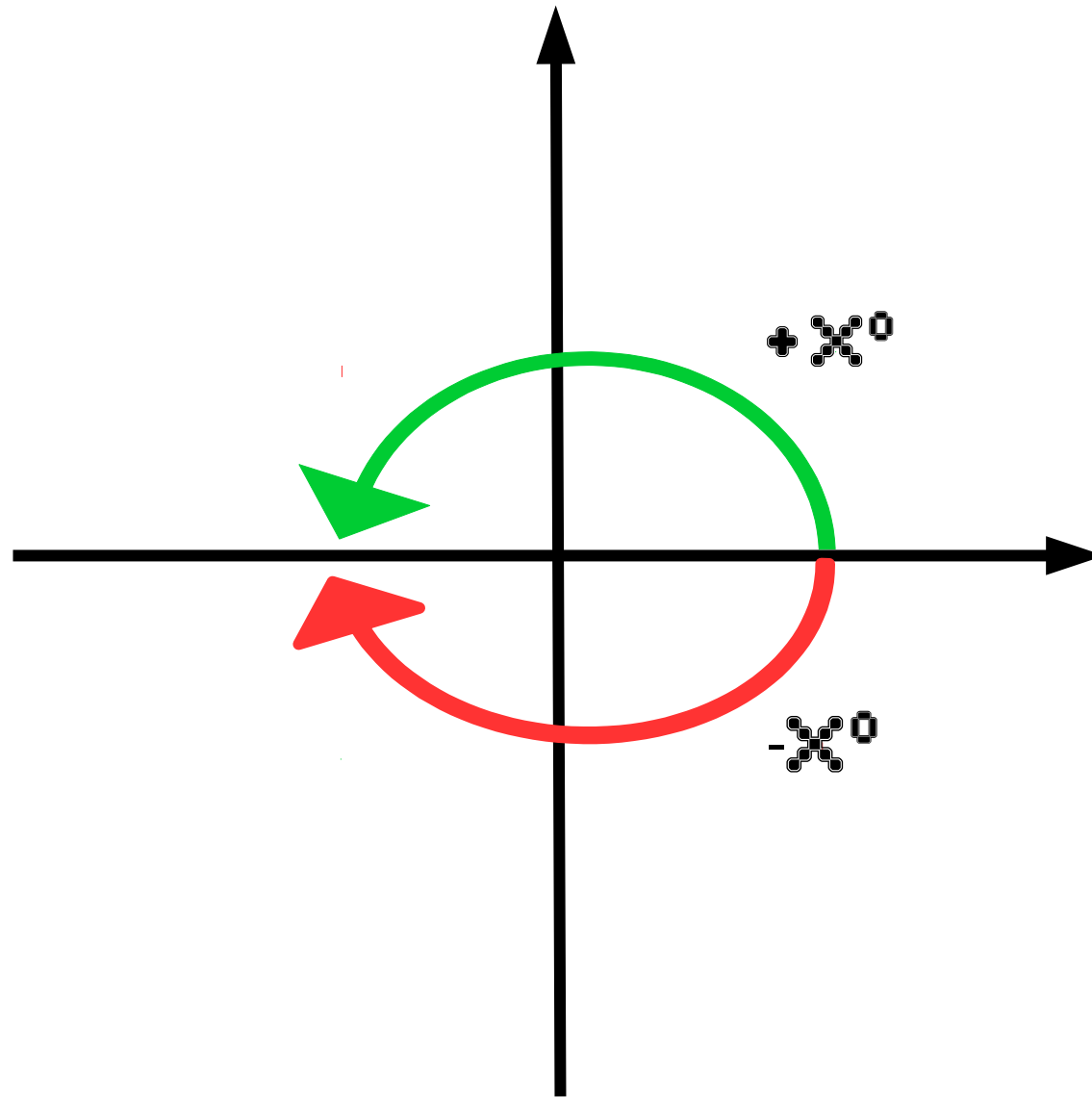
`pygame.K_RIGHT/K_LEFT/K_UP/K_DOWN` → unicode/scancode

```
...
if evento.type == pygame.KEYDOWN:
    if evento.key == pygame.K_UP:
        rectObj.top -= desplazamiento
    elif evento.key == pygame.K_RIGHT:
        rectObj.left += desplazamiento
    elif evento.key == pygame.K_DOWN:
        rectObj.top += desplazamiento
    elif evento.key == pygame.K_LEFT:
        rectObj.left -= desplazamiento
...
```

Utilizamos la clase “Rect”:

```
Rect.colliderect() → bool
```

∴ ¿Lo desarrollamos entre todos :D? ∴

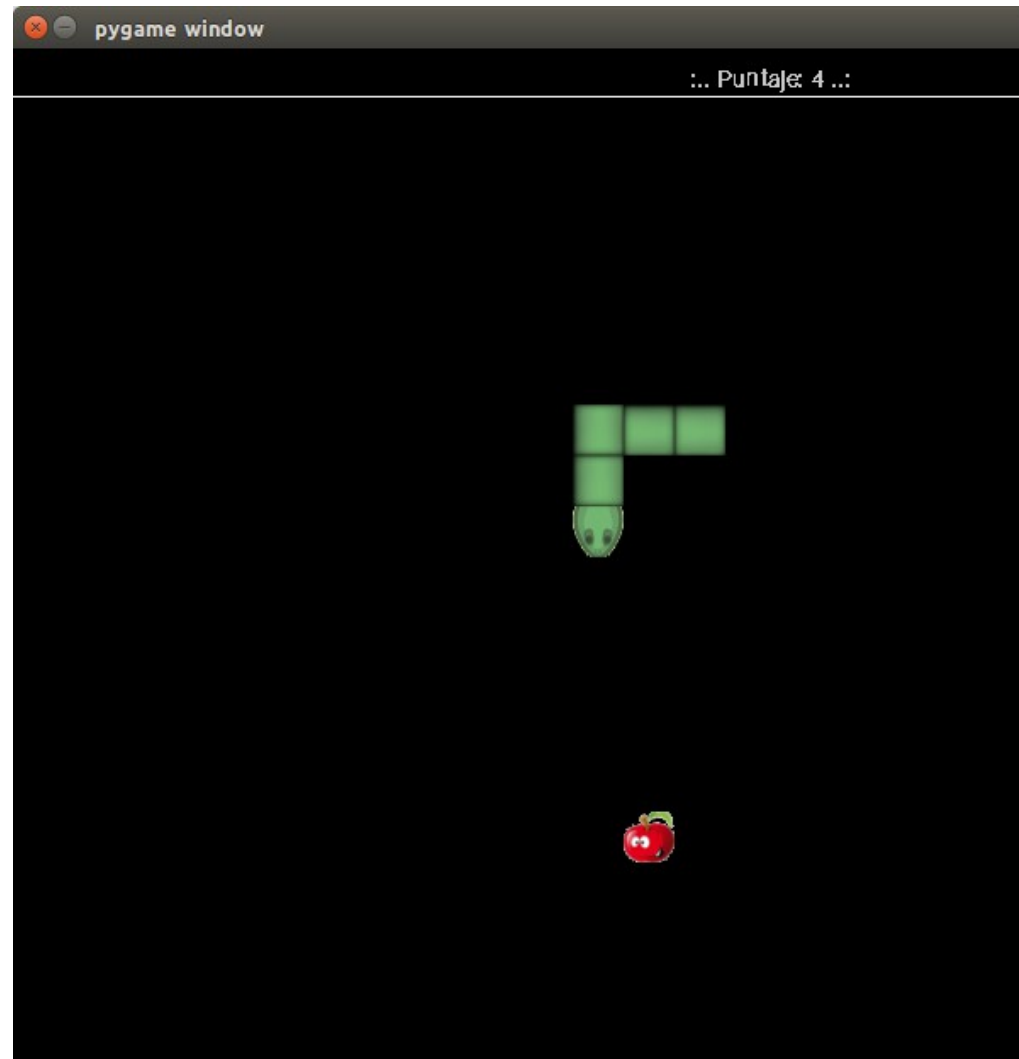


Utilizamos el módulo “transform”:

```
transform.rotate() → Surface
```

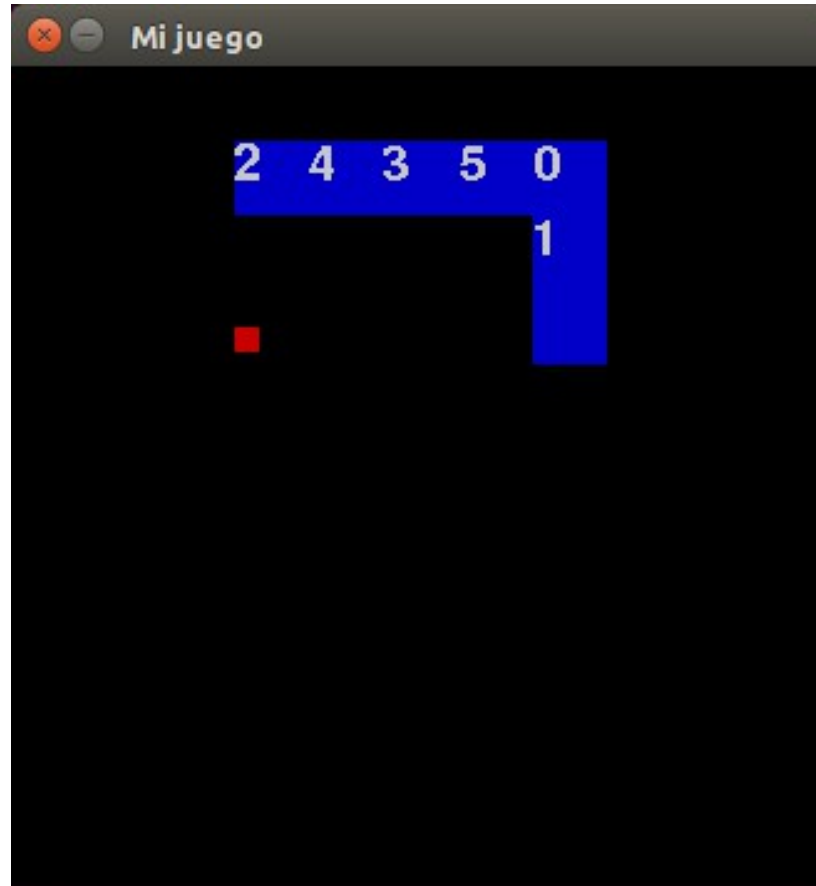
∴ ¿Lo desarrollamos entre todos :D? ∴

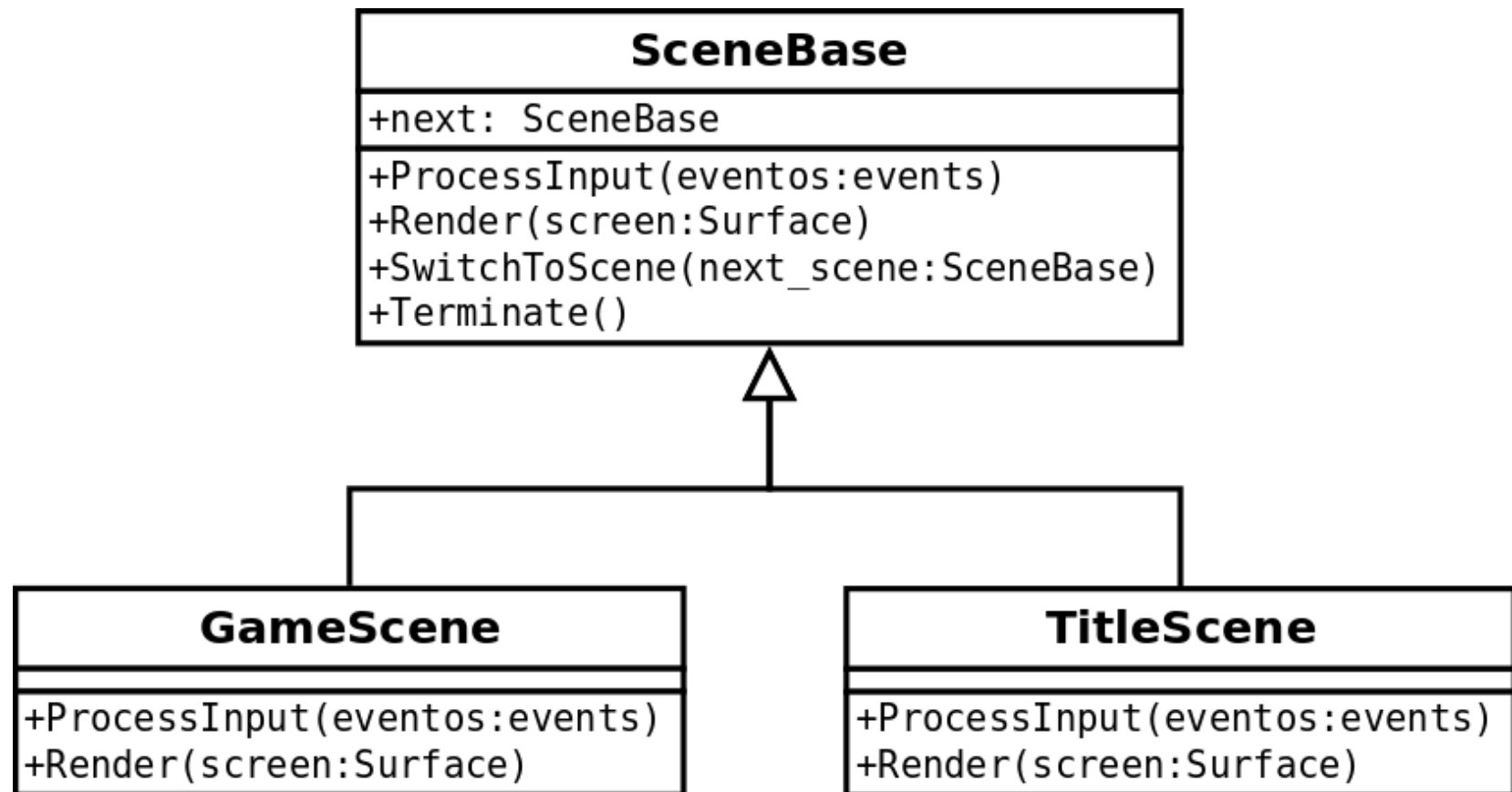
NUESTRO OBJETIVO



---- SNAKE ---->

"HISTORIA" DEL TALLER...

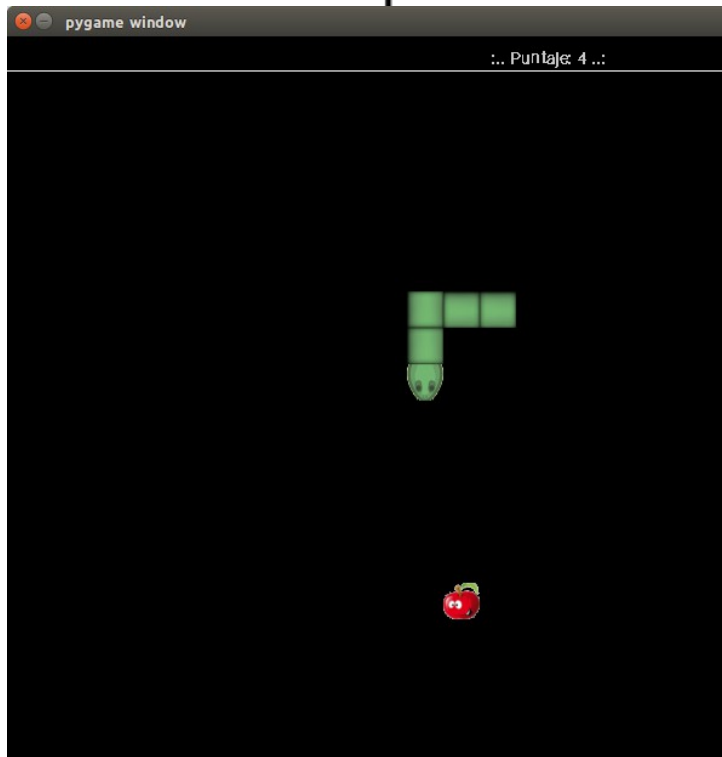
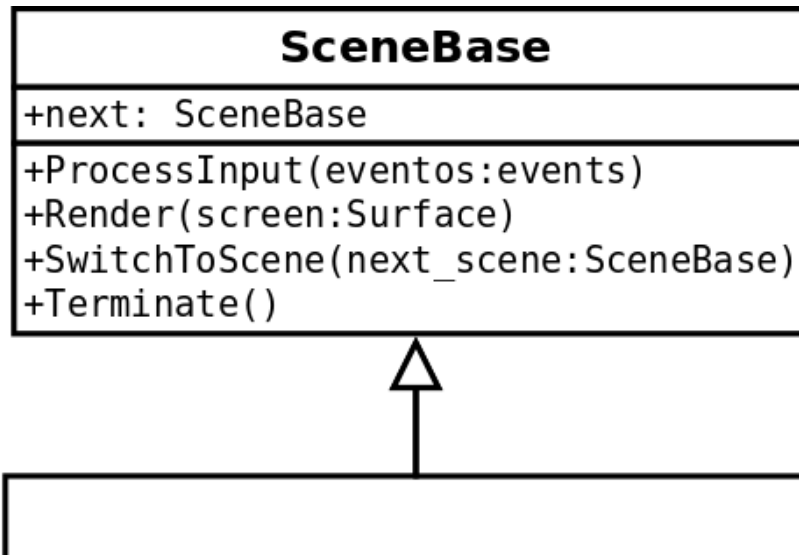




¡Y... game_run!

---- SNAKE ---->

MULTIPLES PANTALLAS

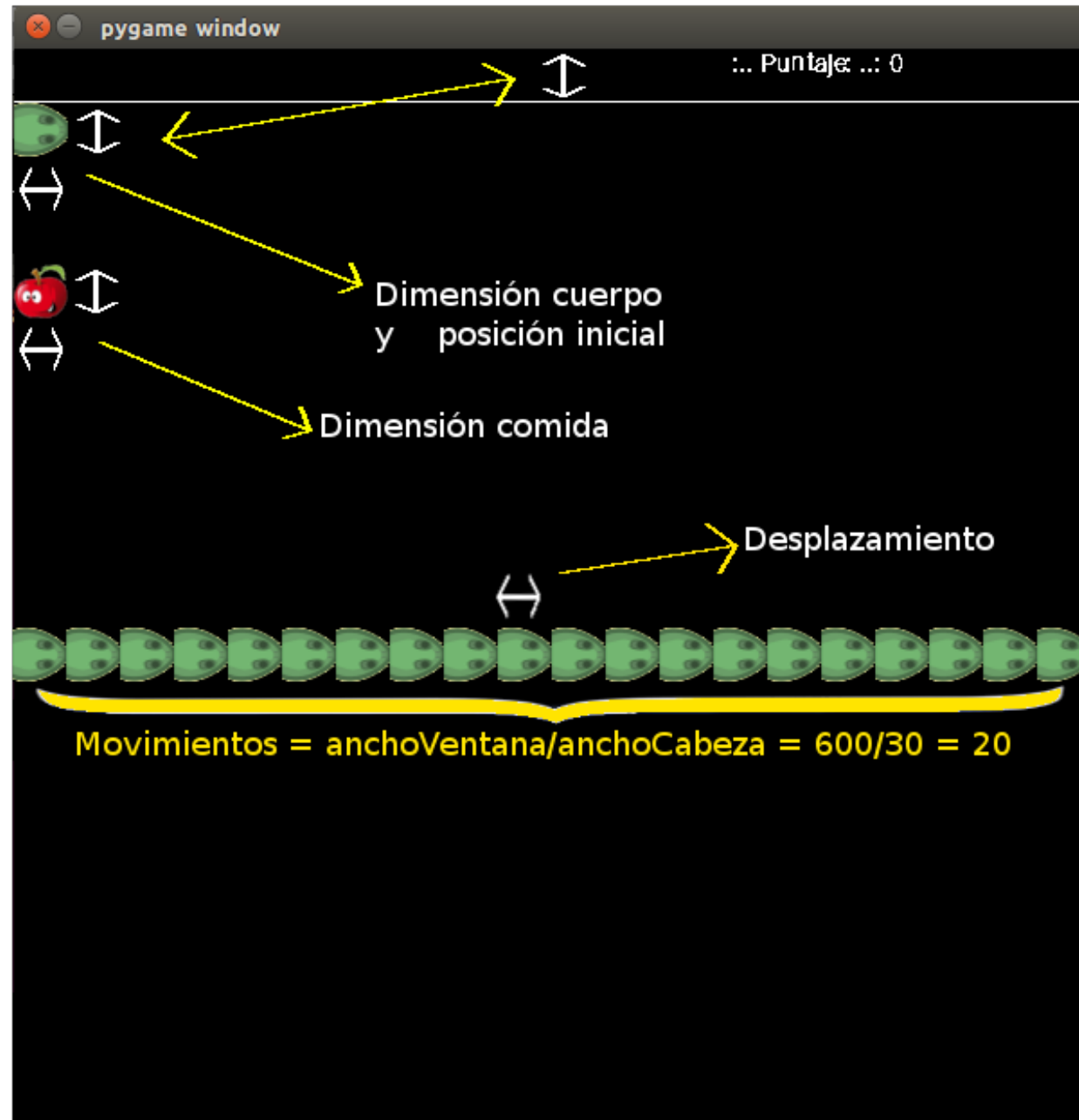


Reglas del juego:

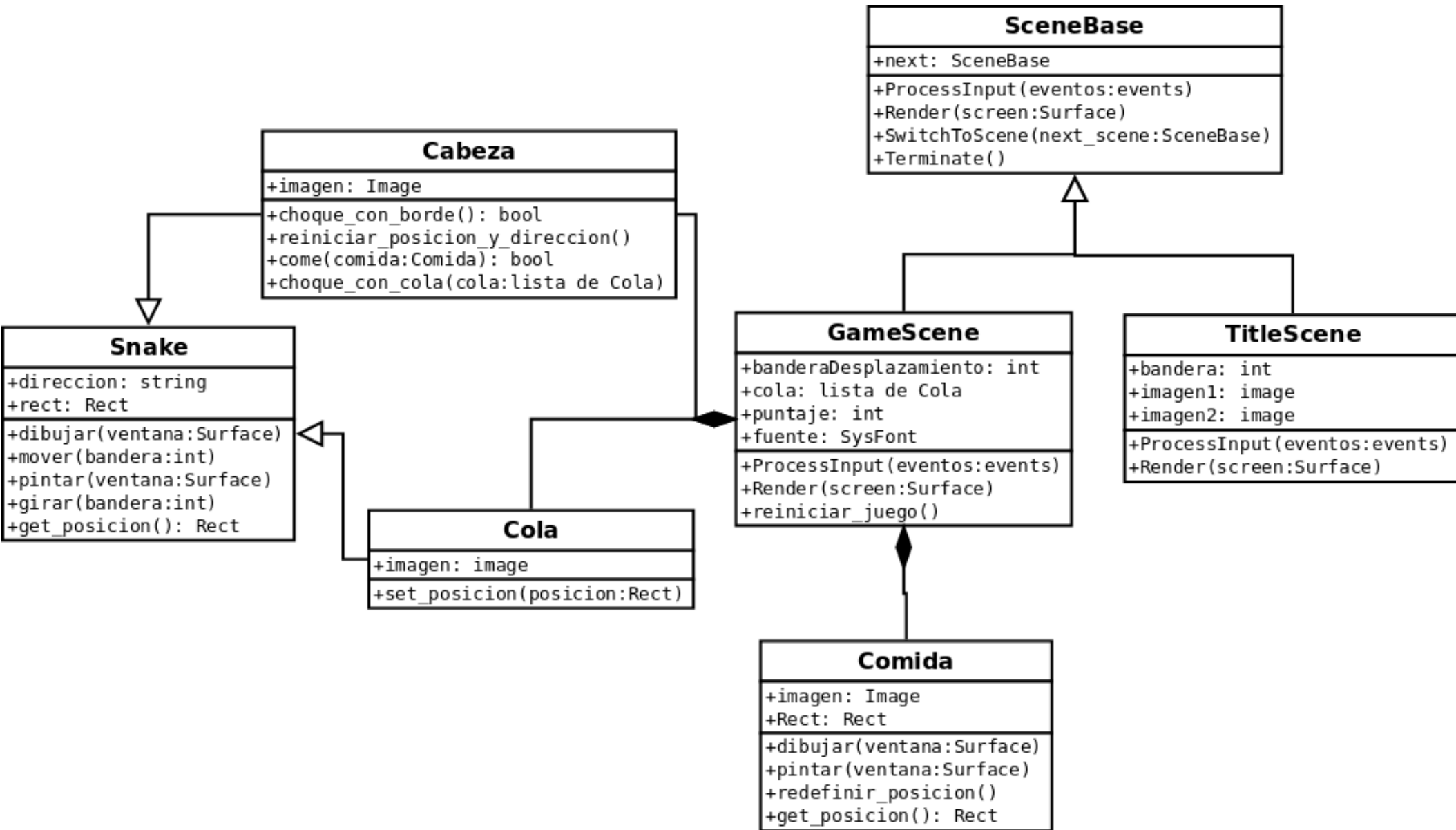
- La serpiente nunca deja de moverse.
- El jugador controla la cabeza de la serpiente con los botones: arriba, abajo, izquierda y derecha.
- Cada segmento del cuerpo de la serpiente sigue al que tiene en frente.
- Si la cabeza toca cualquiera de las paredes, o cualquier parte del cuerpo el juego termina.
- Una “pieza de comida” es colocada al azar.
- La serpiente comienza con una longitud de una unidad, y el objetivo es conducirla hasta la “pieza de comida”.
- Después de que la serpiente se come la comida, crece en longitud y un nuevo trozo de alimento se coloca al azar en el tablero.



- 1) Configuración.
- 2) Clases.
- 3) Game loop.



- Tamaño de la ventana donde se jugará.
- Colores para: fondo ventana.
- Dimensión cuerpo.
- Posición inicial del cuerpo.
- Dimensión comida.
- Desplazamiento.



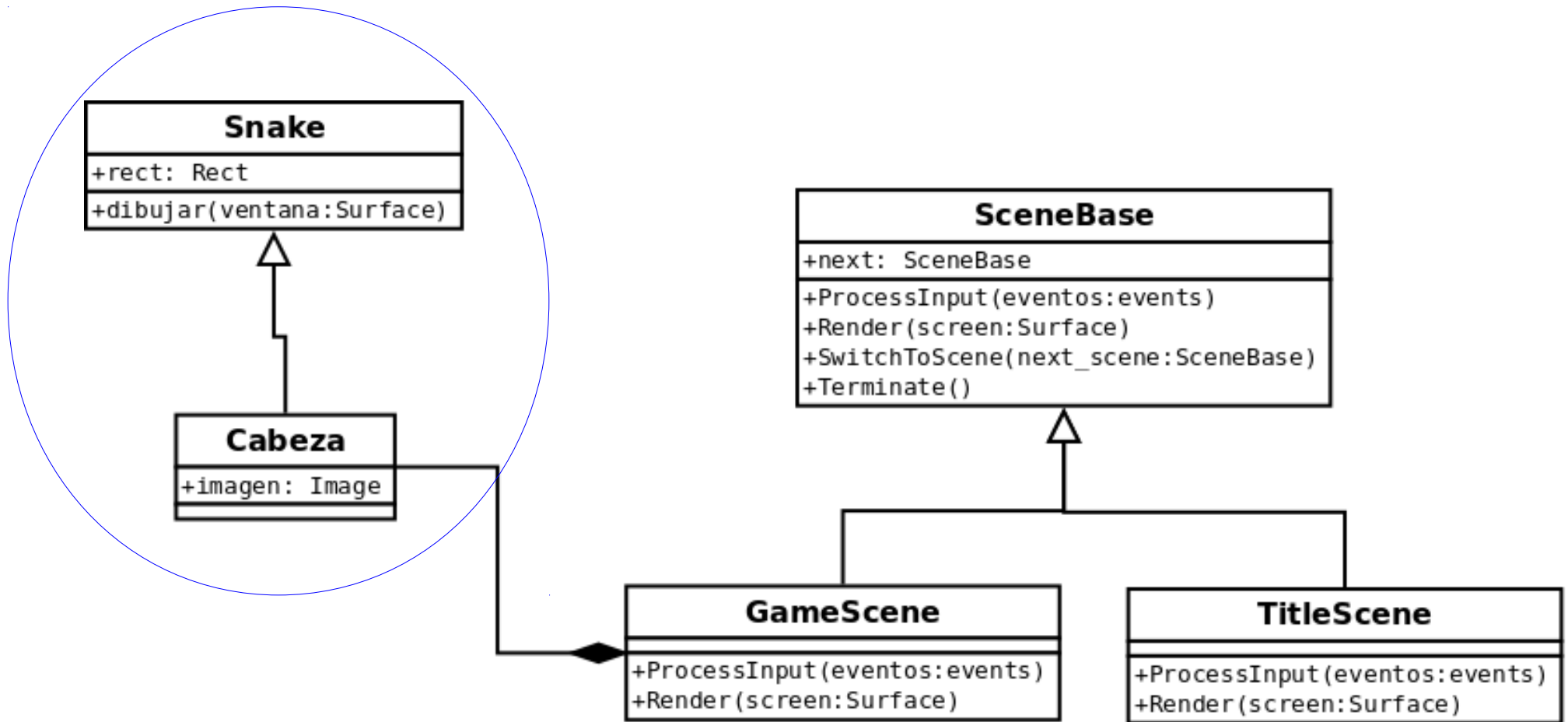
“Pasos” para el desarrollo:

- 1) Mostrar cabeza en pantalla.
- 2) Mover cabeza.
- 3) Girar cabeza.
- 4) Choque cabeza con bordes.
- 5) Mostrar comida en pantalla.
- 6) Cabeza come comida.
- 7) Alargamiento cola.
- 8) Choque cabeza con cola.
- 9) Puntaje.
- 10) Portada.
- 11) Cambio de velocidad.

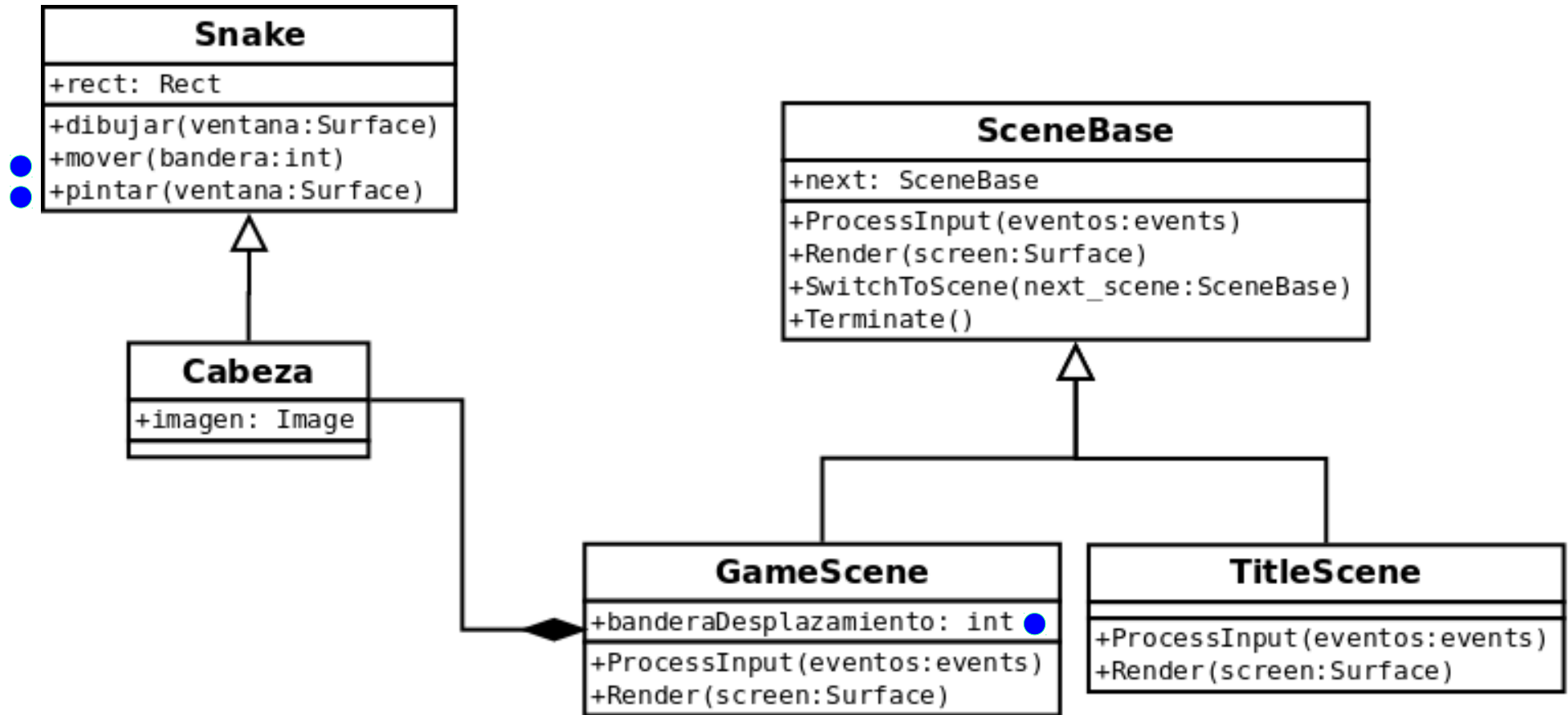


---- SNAKE ---->

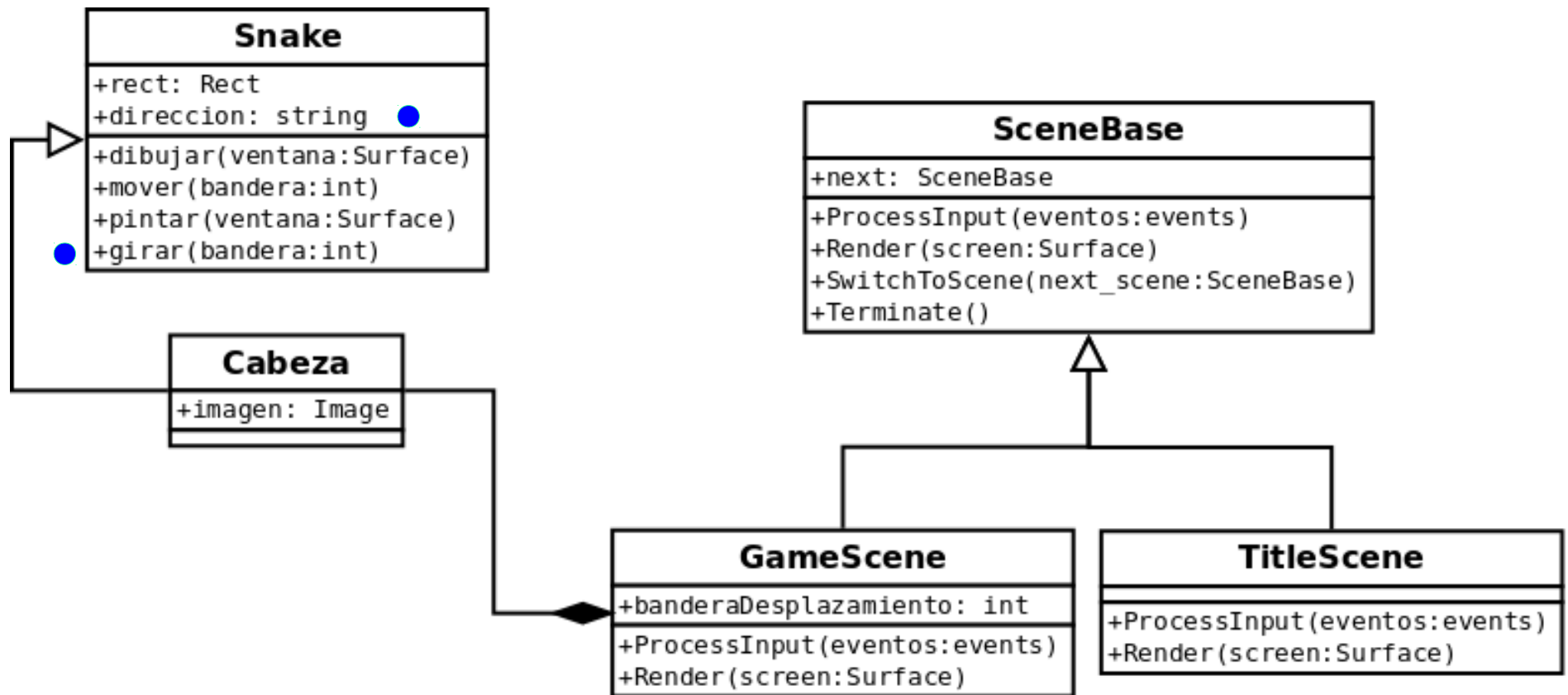
Mostrar cabeza en pantalla



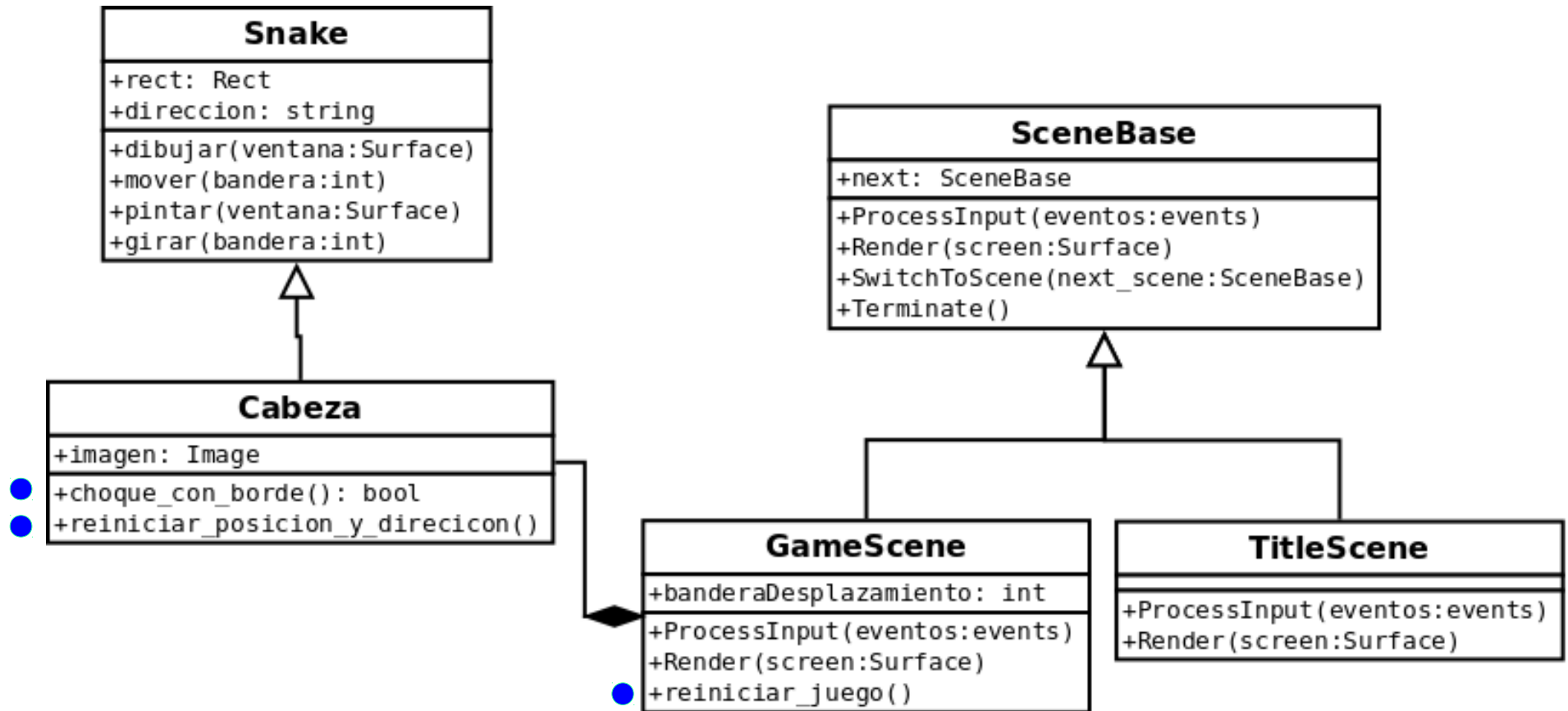
- 1) ~~Mostrar cabeza en pantalla.~~
- 2) Mover cabeza.
- 3) Girar cabeza.
- 4) Choque cabeza con bordes.
- 5) Mostrar comida en pantalla.
- 6) Cabeza come comida.
- 7) Alargamiento cola.
- 8) Choque cabeza con cola.
- 9) Puntaje.
- 10) Portada.
- 11) Cambio de velocidad.



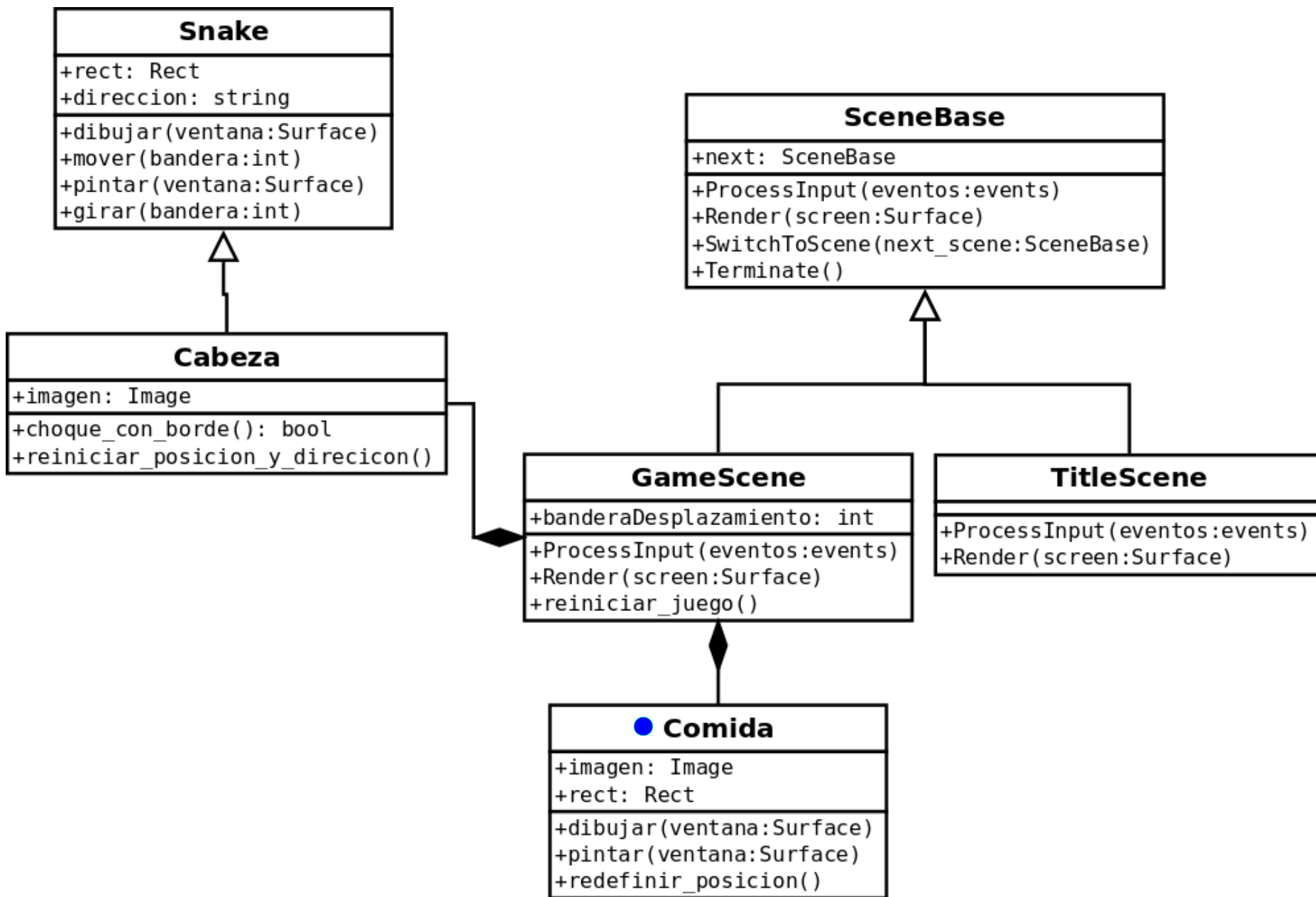
- 1) ~~Mostrar cabeza en pantalla.~~
- 2) ~~Mover cabeza.~~
- 3) Girar cabeza.
- 4) Choque cabeza con bordes.
- 5) Mostrar comida en pantalla.
- 6) Cabeza come comida.
- 7) Alargamiento cola.
- 8) Choque cabeza con cola.
- 9) Puntaje.
- 10) Portada.
- 11) Cambio de velocidad.



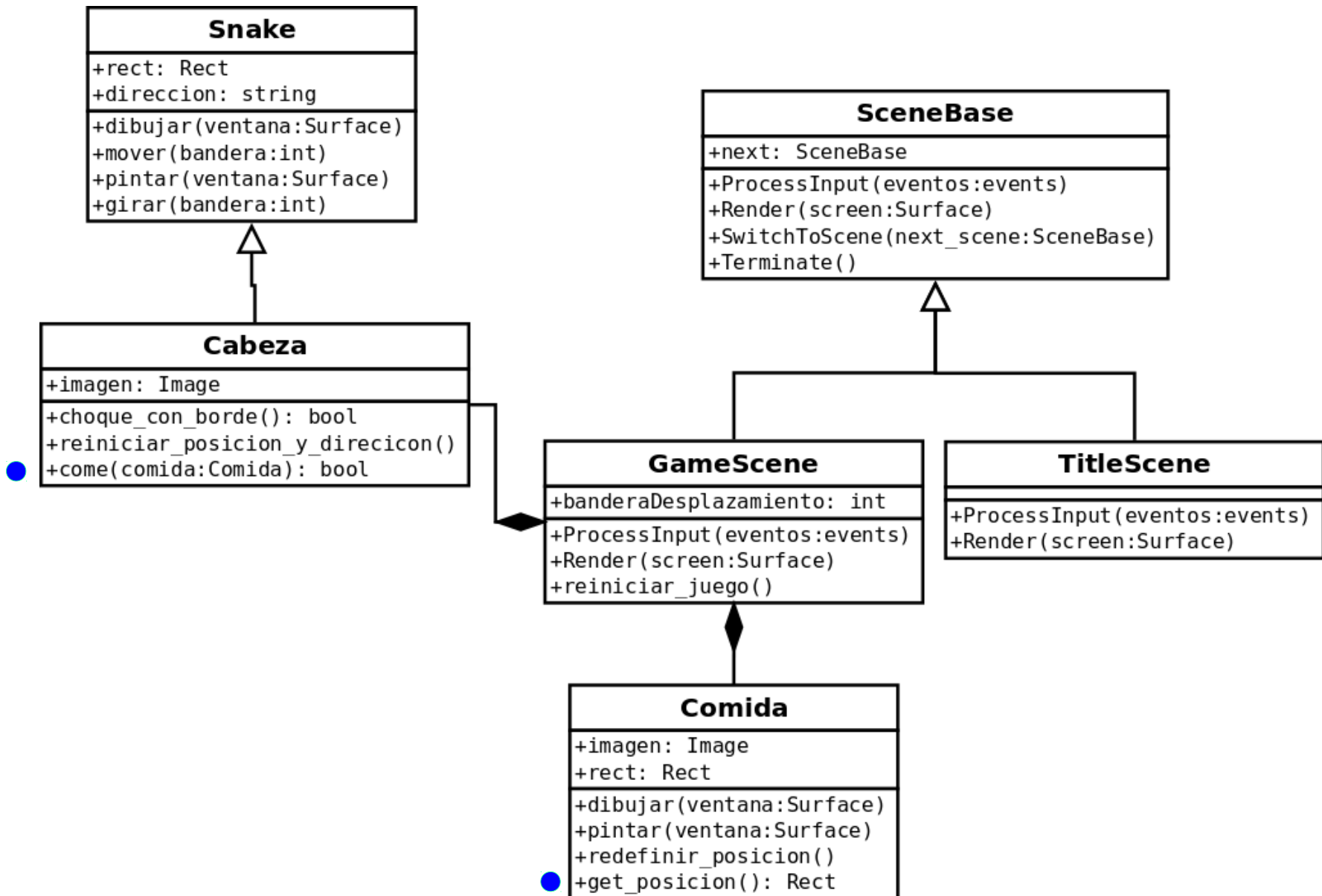
- 1) ~~Mostrar cabeza en pantalla.~~
- 2) ~~Mover cabeza.~~
- 3) ~~Girar cabeza.~~
- 4) Choque cabeza con bordes.
- 5) Mostrar comida en pantalla.
- 6) Cabeza come comida.
- 7) Alargamiento cola.
- 8) Choque cabeza con cola.
- 9) Puntaje.
- 10) Portada.
- 11) Cambio de velocidad.



- 1) ~~Mostrar cabeza en pantalla.~~
- 2) ~~Mover cabeza.~~
- 3) ~~Girar cabeza.~~
- 4) ~~Choque cabeza con bordes.~~
- 5) Mostrar comida en pantalla.
- 6) Cabeza come comida.
- 7) Alargamiento cola.
- 8) Choque cabeza con cola.
- 9) Puntaje.
- 10) Portada.
- 11) Cambio de velocidad.



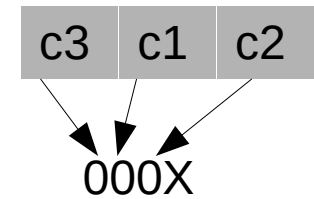
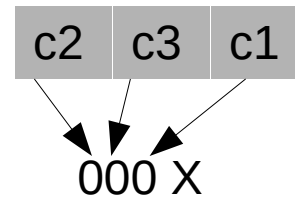
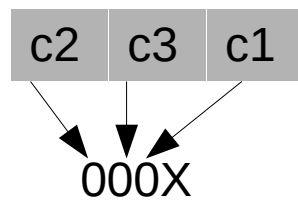
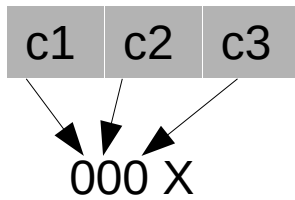
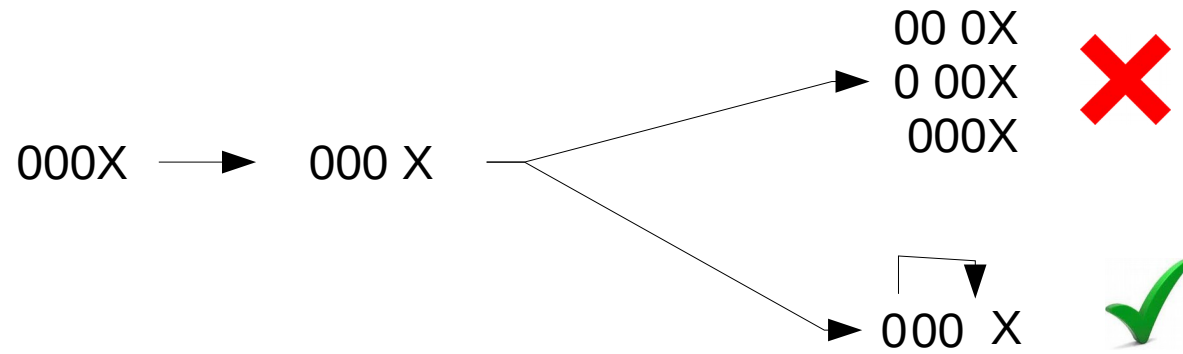
- 1) ~~Mostrar cabeza en pantalla.~~
- 2) ~~Mover cabeza.~~
- 3) ~~Girar cabeza.~~
- 4) ~~Choque cabeza con bordes.~~
- 5) ~~Mostrar comida en pantalla.~~
- 6) Cabeza come comida.
- 7) Alargamiento cola.
- 8) Choque cabeza con cola.
- 9) Puntaje.
- 10) Portada.
- 11) Cambio de velocidad.



- 1) ~~Mostrar cabeza en pantalla.~~
- 2) ~~Mover cabeza.~~
- 3) ~~Girar cabeza.~~
- 4) ~~Choque cabeza con bordes.~~
- 5) ~~Mostrar comida en pantalla.~~
- 6) ~~Cabeza come comida.~~
- 7) Alargamiento cola.
- 8) Choque cabeza con cola.
- 9) Puntaje.
- 10) Portada.
- 11) Cambio de velocidad.

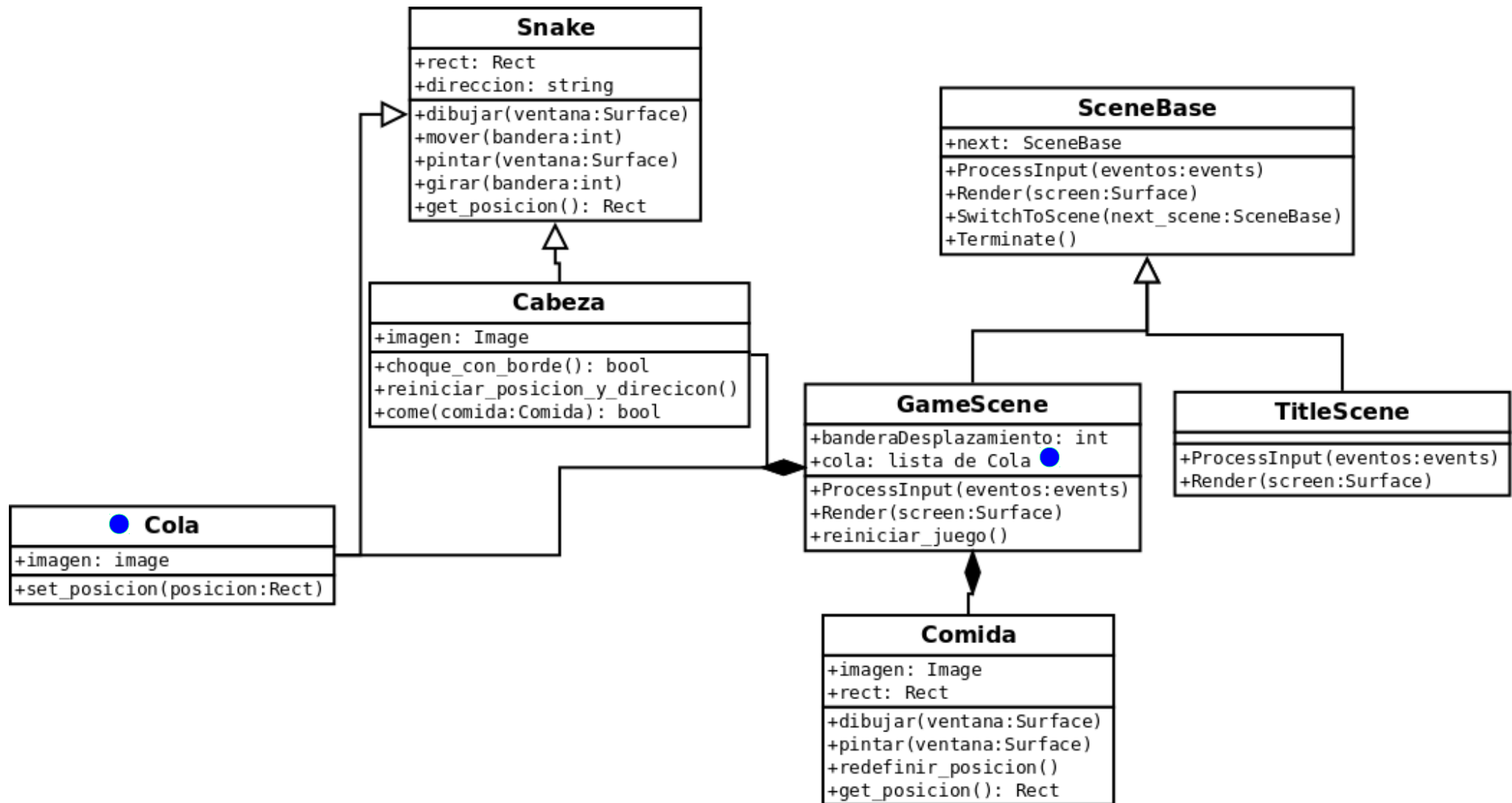
---- SNAKE ---->

OPTIMIZAR EL MOVIMIENTO DE LA
SERPIENTE



---- SNAKE ---->

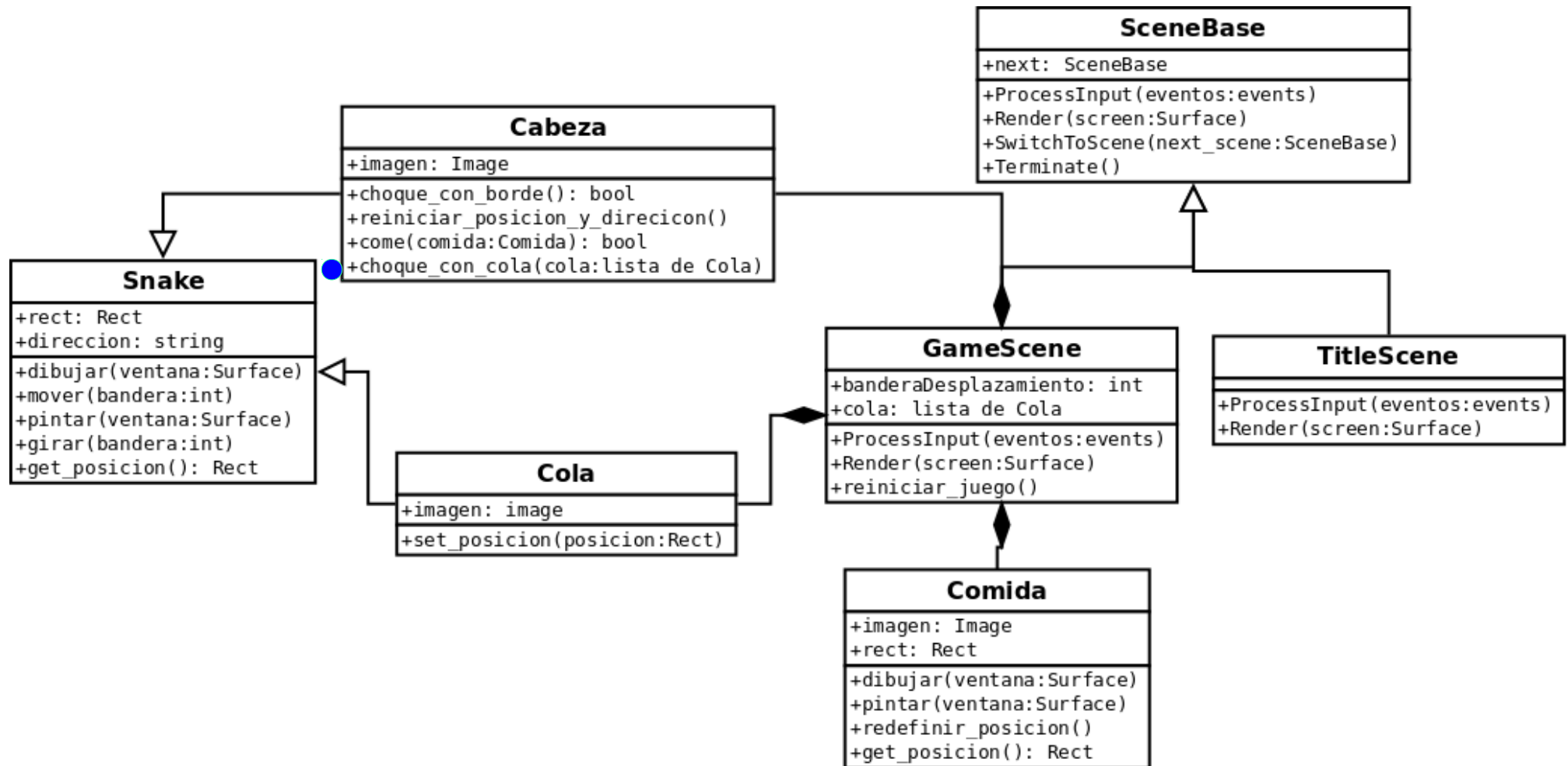
OPTIMIZAR EL MOVIMIENTO DE LA SERPIENTE



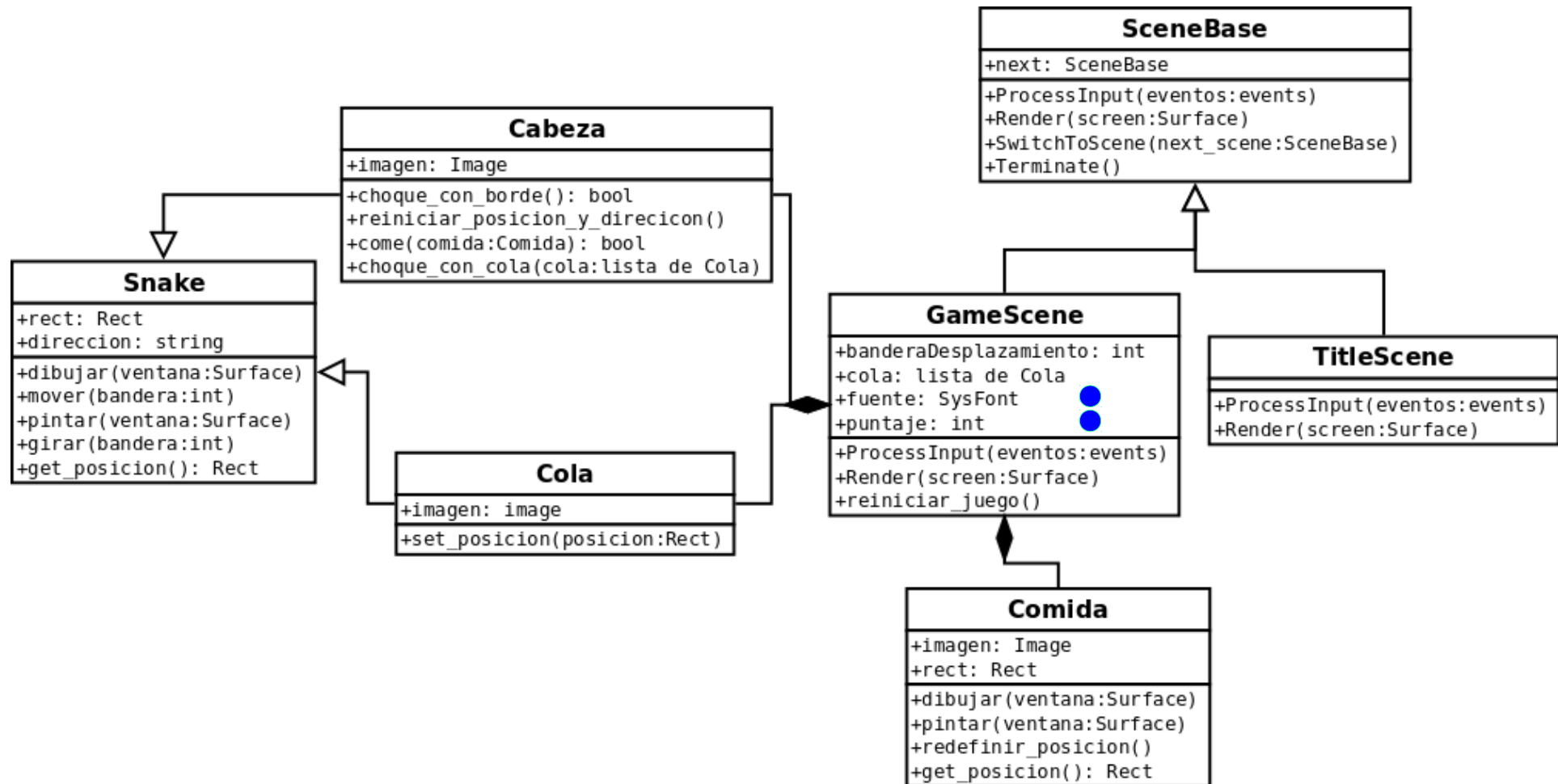
- 1) ~~Mostrar cabeza en pantalla.~~
- 2) ~~Mover cabeza.~~
- 3) ~~Girar cabeza.~~
- 4) ~~Choque cabeza con bordes.~~
- 5) ~~Mostrar comida en pantalla.~~
- 6) ~~Cabeza come comida.~~
- 7) ~~Alargamiento cola.~~
- 8) Choque cabeza con cola.
- 9) Puntaje.
- 10) Portada.
- 11) Cambio de velocidad.

---- SNAKE ---->

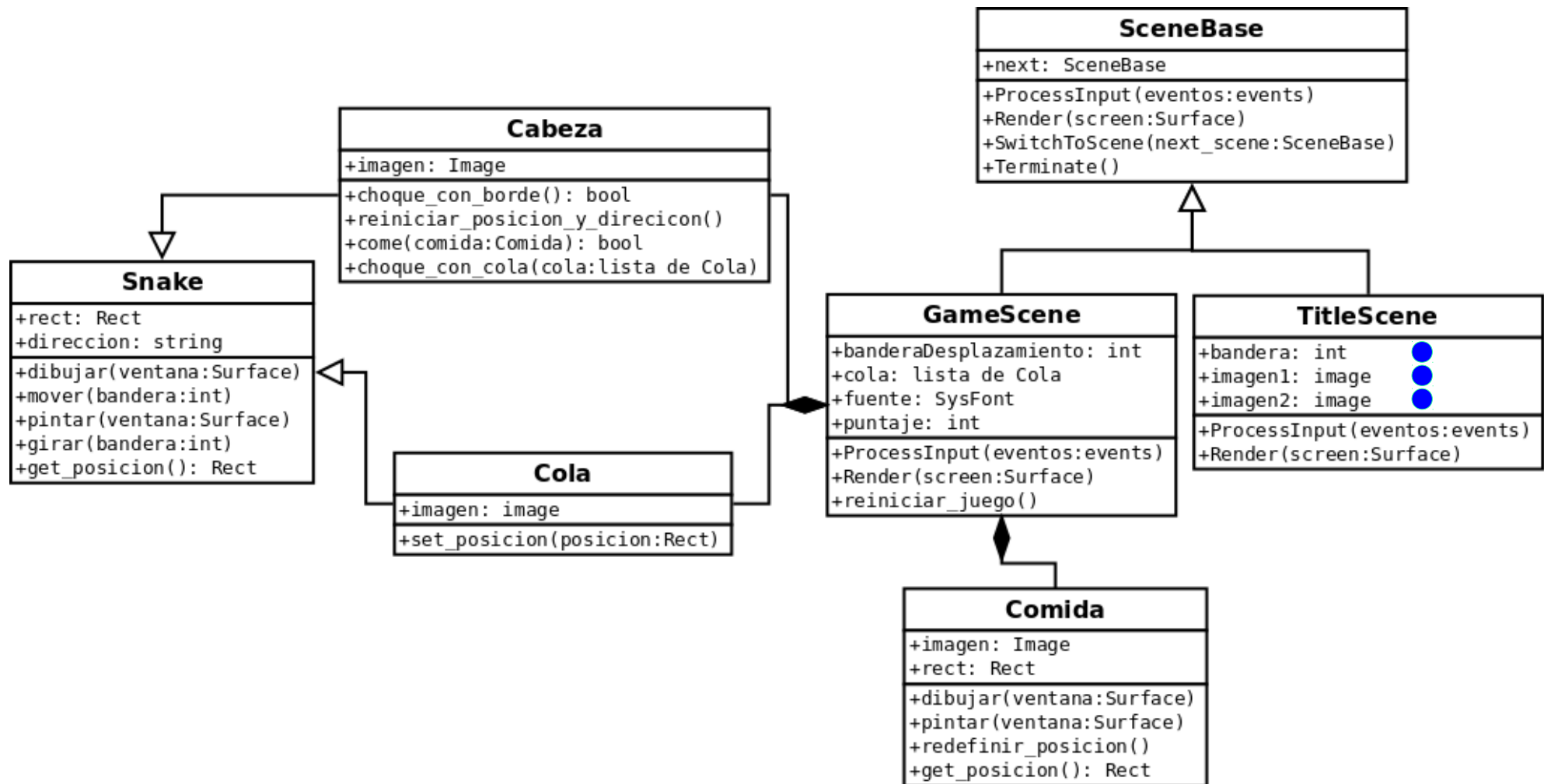
Choque cabeza con cola



- 1) ~~Mostrar cabeza en pantalla.~~
- 2) ~~Mover cabeza.~~
- 3) ~~Girar cabeza.~~
- 4) ~~Choque cabeza con bordes.~~
- 5) ~~Mostrar comida en pantalla.~~
- 6) ~~Cabeza come comida.~~
- 7) ~~Alargamiento cola.~~
- 8) ~~Choque cabeza con cola.~~
- 9) Puntaje.
- 10) Portada.
- 11) Cambio de velocidad.



- 1) ~~Mostrar cabeza en pantalla.~~
- 2) ~~Mover cabeza.~~
- 3) ~~Girar cabeza.~~
- 4) ~~Choque cabeza con bordes.~~
- 5) ~~Mostrar comida en pantalla.~~
- 6) ~~Cabeza come comida.~~
- 7) ~~Alargamiento cola.~~
- 8) ~~Choque cabeza con cola.~~
- 9) ~~Puntaje.~~
- 10) Portada.
- 11) Cambio de velocidad.



- 1) ~~Mostrar cabeza en pantalla.~~
- 2) ~~Mover cabeza.~~
- 3) ~~Girar cabeza.~~
- 4) ~~Choque cabeza con bordes.~~
- 5) ~~Mostrar comida en pantalla.~~
- 6) ~~Cabeza come comida.~~
- 7) ~~Alargamiento cola.~~
- 8) ~~Choque cabeza con cola.~~
- 9) ~~Puntaje.~~
- 10) ~~Portada.~~
- 11) Cambio de velocidad.

- 1) ~~Mostrar cabeza en pantalla.~~
- 2) ~~Mover cabeza.~~
- 3) ~~Girar cabeza.~~
- 4) ~~Choque cabeza con bordes.~~
- 5) ~~Mostrar comida en pantalla.~~
- 6) ~~Cabeza come comida.~~
- 7) ~~Alargamiento cola.~~
- 8) ~~Choque cabeza con cola.~~
- 9) ~~Puntaje.~~
- 10) ~~Portada.~~
- 11) ~~Cambio de velocidad.~~

- Lenguaje de programación: **Python v2.7.6**
- Paquete: **Pygame v1.9.2**
- Editor de texto: **Sublime v2**
- Generador de diagramas estructurados: **dia-normal v0.97.2**
- Editor de imágenes: **GIMP 2.8.10**
- Ofimática: **LibreOffice 5 - Impress**
- Resaltador de sintaxis: **www.tohtml.com**

- www.python.org
- www.pygame.org
- www.tutorialspoint.com
- <http://www.python-course.eu/>
- www.codigofacilto.com/
- python-textbok.readthedocs.org
- www.wikipedia.org
- <http://www.zlib.net/>
- www.libsdl.org

***¡MUCHAS GRACIAS
POR SU ATENCIÓN!***

Lic. Jorge Habib Namour
Universidad Nacional de Tucumán
jnamour@herrera.unt.edu.ar