

Escuela Tecnica N°35  
Ingeniero Eduardo Latzina

<b>DONKEY KONG</b>
--------------------

**Definicion Funcional**

Profesor:  
Carlos Lescano

Grupo:  
Mateo Ricci  
Ignacio Castillo  
Gustavo Delgado

# Indice

<b>Clase Objeto Movil.....</b>	<b>3</b>
<b>Clase Personaje.....</b>	<b>4</b>
<b>Clase Barril.....</b>	<b>6</b>
<b>Clase Mapa.....</b>	<b>8</b>

### **Clase ObjetoMovil**

Esta clase en si, no tiene metodos, se utilizo mas que nada como referencia y englobamiento de las clases que heredan sus atributos (Personaje y Barril)

Sus atributos son:

**RectangleShape colisionador**

**float gravity**

**float jumpSpeed**

**bool saltar= true**

**Vector2f velocity**

**int posX**

**int posY**

**int minPiso**

**bool puedeEscalar**

**int piso**

**string tipo**

**bool control=NULL**

## **Clase Personaje**

Esta clase es utilizada para el modelado de los personajes, tanto como para mario, como para la princesa.

Los atributos propios de esta clase, aparte de los que hereda de objetoMovil son:

**Texture textura**

**Sprite personaje**

**int lado**

**string tipo**

### **Personaje()**

En el constructor podemos encontrar dos condiciones, que sea de tipo mario, o que sea de tipo princesa.

Si es de tipo mario se crea el objeto en la parte inferior del mapa, con su respectiva imagen.

Y si es del tipo princesa, en la posicion superior con su imagen.

### **void mostrar(RenderWindow &v)**

Este metodo se utiliza para dibujar en la pantalla el sprite del objeto que llame al metodo, en el caso de mario, al presionar la tecla derecha o izquierda, el sprite se recorta mostrandolo mirando hacia un lado o hacia el otro.

### **void caminar()**

Este metodo realiza el movimiento en x del personaje, dependiendo del piso en que se encuentre tiene algunas restricciones, para no salirse del piso.

### **void salto()**

Se encarga de realizar el salto del personaje dependiendo de varios factores, como el piso en el que se encuentra, para que lado esta mirando.

### **void escalar(RectangleShape m)**

Se encarga de habilitar el movimiento en y del personaje, dependiendo si esta colisionando con una escalera o no.

### **void actualizar()**

Dependiendo de la posición del personaje en y, actualizará el valor de piso y del minPiso, que se utiliza como suelo en el salto.

### **bool colisionPrincesa(Personaje &target)**

Se utiliza para verificar si están colisionando dos personajes, en nuestro caso se utiliza para saber si Mario llegó a la princesa.

### **RectangleShape returnColisionador()**

Devuelve el RectangleShape colisionador del objeto que llame este método..

### **void reset()**

Se encarga de resetear los valores de la mayoría de los atributos, este método se utiliza al perder o ganar.

## Clase Barril

Se encarga de modelar los barriles que van a ir descendiendo por los pisos y escaleras del mapa. Los cuales al colisionar con mario, se pierde la partida. Hereda de ObjetoMovil sus atributos y tambien tiene propios de la clase que son:

**Texture textura**

**Sprite barril**

**RectangleShape sumadorPuntaje**

**int camino**

**Barril()**

En el constructor se genera un numero aleatorio entre 1 y 3 y se le asigna a camino, se setean las posiciones, tamaños, de su colisionador, su sprite y su sumador de puntaje.

**void escalar(RectangleShape m)**

Dependiendo de **camino** el barril se movera por los pisos y descendera al colisionar con una escalera, o tomara algun camino definido por nosotros..

**void actualizar()**

Se utiliza para establecer el piso dependieindo de la posicion en y del barril.

**void mostrar(RenderWindow &v)**

Dibuja sobre la ventana recibida como parametro (**&v**) el sprite del barril.

**void mover()**

Dependiendo del piso en el que se encuentre al llamar a este metodo, movera el barril hacia la izquierda o hacia la derecha.

**bool colision(RectangleShape target)**

Si el colisionador del barril intersecta con el target, en este caso mario, devuelve valor true

**bool puntaje(RectangleShape target)**

Si el sumadorPuntaje del barril intersecta con el target, en este caso mario, devuelve valor true

**bool salioMapa()bool salioMapa()**

Si el colisionador del barril se encuentra en el piso inferior y la posición en x es menor o igual a 68, devuelve valor verdadero, esto se utiliza para borrar el barril. De lo contrario devuelve valor falso.

## **Clase Mapa**

Esta clase se encarga de modelar los objetos del tipo mapa, que en si, son las escaleras, cuenta con el siguiente atributo:

### **RectangleShape colisionador**

#### **Mapa(int elemento)**

En su constructor, dependiendo del numero de elemento que se le pasa como parametro, se crea el objeto colocandole una posicion y tamaño determinados al colisionador.

#### **RectangleShape returnColisionador()**

Devuelve el RectangleShape colisionador cuando se llama a este metodo.