

# Modelación y Computación Gráfica para Ingenieros

## Tarea 2

PP-Block

Autor: Javier Gómez P.  
Profesor: Nancy Hitschfeld K.  
Auxiliares: Pablo Pizarro R.  
Pablo Polanco  
Ayudantes: Joaquín T. Paris  
Roberto I. Valdebenito  
Rodrigo Ramos T.

Fecha de realización: 16 de mayo de 2017  
Fecha de entrega: 15 de mayo de 2017  
Santiago, Chile

# Índice de Contenidos

<b>1. Introducción</b>	<b>1</b>
<b>2. Descripción del problema</b>	<b>2</b>
<b>3. Descripción de la solución</b>	<b>3</b>
3.1. Modelo-Vista-Controlador (MVC) . . . . .	3
3.2. Menú Principal . . . . .	3
3.3. Créditos . . . . .	5
3.4. Juego Principal . . . . .	5
<b>4. Dificultades encontradas</b>	<b>8</b>
<b>5. Conclusiones</b>	<b>9</b>

# 1. Introducción

Hoy en día, en el mundo del entretenimiento, una parte muy importante son los videojuegos. Los desarrolladores de estos se valen de herramientas computacionales para programar aplicaciones interactivas para los usuarios.

A la hora de programar un juego, el desarrollador realiza un esquema de Modelo-Vista-Controlador (MVC). Con esto, se tiene mayor claridad sobre los objetos que compondrán al videojuego, la forma en que el usuario interactúa con estos, y lo que se debe mostrar en pantalla. Estos tres elementos se relacionan entre sí, consultando y modificando valores y acciones entre ellos.

En este trabajo, se desarrollará un videojuego, clónico del famoso juego 'BB-Tan', en donde se aplicará el MVC.

## 2. Descripción del problema

Se pide desarrollar un videojuego similar a BB-Tan (fig 2.1), usando MVC.

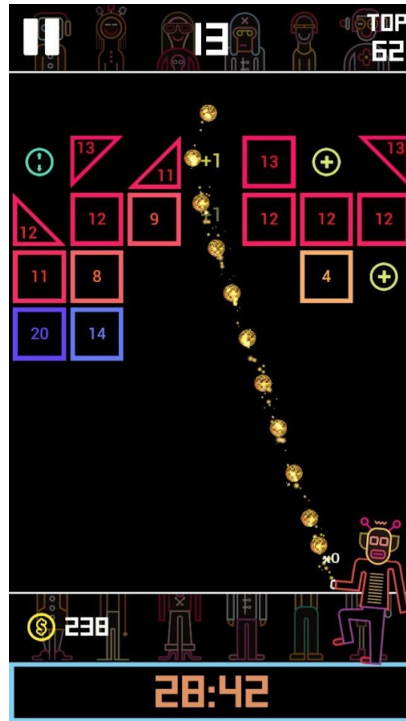


Figura 2.1: BB-Tan

Se debe implementar físicas de choques, creación y destrucción de objetos (los que en este caso serán bloques), lanzamiento de bolas, un menú principal y una guía que indique la dirección inicial de la bola.

## 3. Descripción de la solución

### 3.1. Modelo-Vista-Controlador (MVC)

Para desarrollar lo pedido, se identificaron los distintos elementos que componen el juego, y a que parte del MVC corresponden.

Modelo	Vista	Controlador
Bloques	Mostrar el menú principal	Detectar posición del mouse
Bolas	Mostrar el texto durante el juego	Detectar 'clicks' del mouse
Power Ups	Mostrar imagen en 'credits'	Detectar teclas pulsadas por el usuario
Imágenes (Jugador y credits)	Mostrar bloques dibujados	Actualizar la vista al hacer click
Texto (Puntaje, menú principal e indicadores)	Mostrar bolas dibujadas (consultar posición)	Actualizar la vista al presionar teclas
Sonidos	Reproducir sonidos	Modificar valores de Bolas (velocidad)

Se crearon las clases 'Ball', 'Block', 'PowerUp', 'Sounds' y 'Auxiliar', las que representan a cada tipo de objeto y guardan sus variables de estado.

La clase 'Auxiliar' se encarga de guardar variables de estado auxiliares para el correcto funcionamiento del juego.

Es importante indicar que la resolución a usar del juego será de  $490 \times 800$  pixeles. También es importante señalar que este juego está desarrollado en Python 2.7.12, y se utilizaron las librerías 'Pygame' (1.9.3) y 'Shapely' (1.6b4).

### 3.2. Menú Principal

La construcción del menú principal se realizó mediante bloques y texto. Según la entrada que dé el usuario (teclas que presione), se muestran distintos bloques de texto. Estos bloques de texto representan 'Play', 'Select Ball', 'Credits' y 'Quit'.

Inicialmente, se muestra la opción 'Play' como seleccionada (es decir, el texto en negro y detrás un bloque blanco). (fig 3.1)



Figura 3.1: Menú en su estado inicial

Si el usuario presiona 'Flecha Abajo', entonces el texto 'Play' se muestra en color blanco, el bloque blanco desaparece, y el texto 'Select Ball' se muestra ahora en negro, con un bloque blanco detrás. También ahora aparece un indicador ': 1' al final del texto, lo que indica que bola está seleccionada (fig 3.2). Al presionar 'Flecha Izquierda' o 'Flecha Derecha' se cambia este valor, y con esto, la bola seleccionada. (La bola cambiará de color y forma, efecto que se apreciará durante el juego)



Figura 3.2: Menú con 'Select Ball' seleccionado

Para el resto de las opciones del menú, el procedimiento es análogo.

### 3.3. Créditos

Se implementó una opción ‘Credits’ en el menú principal, la cual al seleccionarla, muestra una imagen desplazándose verticalmente. Si el usuario presiona la tecla ‘Escape’, vuelve al menú principal.



Figura 3.3: Pantalla de créditos

### 3.4. Juego Principal

En la pantalla del juego principal, se muestran en la parte superior, el contador de puntos, un texto que indica cómo salir del juego, y una línea horizontal que delimita el área de juego.

A continuación, se generan 7 bloques aleatorios (50 % de probabilidad de ser un bloque de lado 70px, 25 % de probabilidad de ser un PowerUp y 25 % de probabilidad de ser vacío).

El juego inicia cuando el usuario hace ‘click’ con el mouse y se muestra una guía entre el puntero del mouse y la bola a lanzar. La bola comienza con su movimiento, y al colisionar con un bloque o con el borde del área de juego, esta rebota y cambia su dirección. Si toca un bloque, a este se le resta un punto a su resistencia. Si la resistencia del bloque llega a 0, el bloque desaparece. Lo anterior está implementado dentro de la clase ‘Ball’, con un método *move()* que inicia movimiento de la bola.

Si la bola toca el borde inferior del área de juego, se da por terminada la jugada, se aumenta el puntaje del jugador en 1 punto, se bajan 70px todos los bloques y se generan 7 nuevos bloques de la misma manera que la anterior. Se cambia la posición del punto donde serán lanzadas las bolas al punto donde la primera bola tocó el borde inferior. Junto con esto, se mueve la imagen del jugador justo abajo de este punto (bajo el borde inferior del área de juego). (fig 3.4)

Luego se chequea si la última fila está vacía. En caso de no estarlo, se da por terminado el juego y se muestra en la pantalla ‘Game Over’.

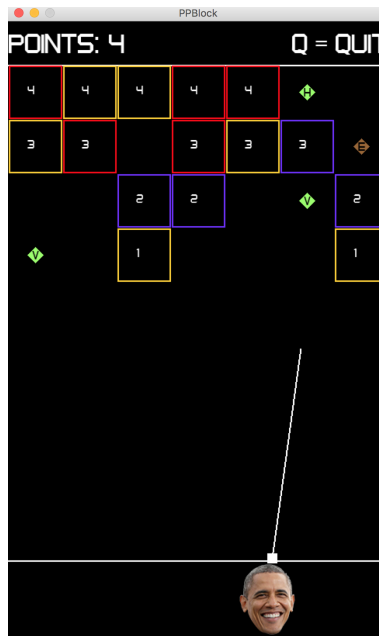


Figura 3.4: Juego Principal

Anteriormente se indicó que dependiendo del valor de ‘Select Ball’ en el menú principal, la bola cambia de forma y de color. En la figura 3.4 se puede ver la bola como un cuadrado de color blanco. Esto sucede cuando el valor de ‘Select Ball’ en el menú principal es 1.

En la figura 3.5 se observa una bola triangular de color rosado. Esto sucede cuando ‘Select Ball:’ es 2.

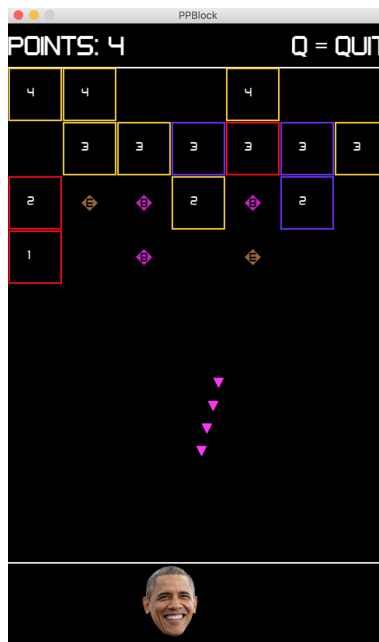


Figura 3.5: Juego Principal con ‘Select Ball: 2’



En la figura 3.6 se observa una bola triangular de color azul. La forma de esta es también triangular, pero en otro sentido ('de cabeza'). Esto sucede cuando 'Select Ball:' es 3.

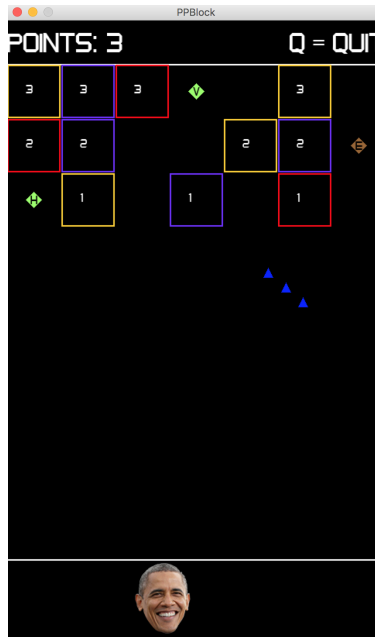


Figura 3.6: Juego Principal con 'Select Ball: 3'

## 4. Dificultades encontradas

Principalmente, el aspecto que más trabajo requirió, y también más líneas de código, fue el movimiento de las bolas, el que incluía las colisiones con los distintos tipos de elementos (bloques, PowerUps, bordes) y que estaba condicionado a las variables generales del juego (clase Auxiliar).

La dificultad de programar esto se debió a la cantidad de condiciones a considerar, debido a los múltiples escenarios que se pueden presentar. Fue necesario tener en cuenta si la jugada había comenzado, si aún el jugador podía jugar (no había perdido), si las bolas habían tocado el borde inferior, etc.

También, debido a la geometría del juego, existió una dificultad al detectar las colisiones, ya que los bloques se encuentran muy cerca uno de otro y es difícil determinar con total claridad con qué bloque colisiona la bola (y con cual borde del bloque).

La generación de las figuras no fue tan complicada gracias al módulo ‘centered figure’, proporcionado por Pablo Pizarro R.

La implementación del resto de los elementos no generó mayor dificultad.

## 5. Conclusiones

Al realizar este trabajo, se logró comprender el funcionamiento de la librería ‘Pygame’ y ‘Shapely’, además de comprender los distintos elementos del MVC y cómo se relacionan entre si.

También se comprendió la forma en que Pygame lee las interacciones de los usuarios y las relaciona con una acción de los objetos existentes.

Se logró dimensionar el trabajo requerido para desarrollar un juego, y los aspectos a considerar durante su desarrollo (la cantidad de múltiples escenarios que podrían presentarse).