



**Universidad  
Rey Juan Carlos**

# Ping Pang Pung

**Universidad Rey Juan Carlos Móstoles**

**Grado de Diseño y Desarrollo de Videojuegos**

*14/01/2019*

## Práctica Obligatoria

**Programación Avanzada**

**Grupo**

Ana María Alcaide Recio  
Carlos Marqués Gonzalez  
Sergio Sánchez-Uran López

## Contenido

MANUAL DEL USUARIO .....	3
DESCRIPCIÓN DEL JUEGO .....	4
DISEÑO REALIZADO Y CRITERIOS.....	4
UML .....	6
MIEMBROS DEL GRUPO .....	7
ANEXO INFORMATIVO.....	7
REQUISITOS CUMPLIMENTADOS .....	7

## MANUAL DEL USUARIO

Al ejecutar el código el procedimiento es sencillo, los únicos que tendremos que hacer es comenzar a jugar.

Los botones necesarios para esto son:

**A:** Para movernos hacia la izquierda

**D:** Para movernos hacia la derecha

**Space:** Para disparar

**H:** Para iniciar el juego.

**E:** Para reiniciar el juego manteniendo las mejores puntuaciones guardadas.

**B:** Para reiniciar el juego, reestableciendo a 0 la mejor puntuación.

**ESC:** Para salir del juego en cualquier momento.

**O:** Rotar en el eje de las Y hacia la izquierda la pantalla

**P:** Rotar en el eje de las Y hacia la derecha la pantalla

**X:** Rota en el eje de las Z hacia abajo

**Z:** Rota en el eje de las Z hacia arriba

**W:** Desplazamiento hacia la izquierda

**Q:** Desplazamiento hacia la derecha

La dinámica del juego es la siguiente:

El juego del que partimos es el clásico Pang pero con visión en 3 dimensiones.

La primera partida comienza con 3 bolas rojas y un minion en el centro que será nuestro personaje principal para la partida.

Tendremos que disparar con el botón espacio, lo que permitirá al personaje destruir las bolas que se dividirán en dos más pequeñas y a su vez si impactamos otra vez se dividirán en otras dos más pequeñas y al impactar una tercera vez desaparecerán.

En caso de que las bolas colisionen con el jugador la pantalla quedará congelada y aparecerá un menú contextual dónde el usuario podrá ver su mejor puntuación, su puntuación actual, una instrucción que dirá “si desea seguir jugando pulse E” en la que se guardará la mejor puntuación para la siguiente partida, y otro menú contextual en el que dirá “presione b para restaurar la mejor puntuación”, donde la mejor puntuación se restablecerá a 0. Presionando cualquiera de las dos teclas (B o E) comenzará una nueva

partida. En caso de no querer otra, lo único que deberá hacer éste será darle a escape (ESC).

En el caso de que supere la primera pantalla, **hemos implementado una segunda pantalla** que cargará al terminar la primera.

Cuando una bola impacte contra nosotros la situación es la misma que en la primera partida.

En caso de ganar la saltará un mensaje diciendo “ENHORABUENA”.

## DESCRIPCIÓN DEL JUEGO

El juego realizado es una versión tridimensional del clásico Pang con dos pantallas en el que en la primera pantalla aparecerán 3 bolas y deberemos esquivarlas, de tal manera que, al dispararlas, éstas se subdividen en otras dos bolas y así sucesivamente hasta que el tamaño de la bola sea tan pequeño que con un disparo podamos hacerla desaparecer.

El objetivo del juego es evitar que las pelotas toquen al personaje e intentar explotarlas todas para poder avanzar en el juego. Tratando de superarte a ti mismo y tus puntuaciones anteriores.

## DISEÑO REALIZADO Y CRITERIOS

Para la realización del juego nos hemos basado en un juego llamado Pang, en el que gracias al motor base implementado en clase, podíamos añadir una serie de modificaciones para poder llevar a cabo el juego con los requisitos pedidos.

Para su implementación hemos realizado lo siguiente:

Hemos creado la clase Particula, que es la que tiene el rebote con el tiro parabólico para las pelotas y hereda de sólido, porque tiene posición velocidad y color y así obtiene sus métodos, además de contener los métodos propios de partícula con sus consecuentes características.

Hemos usado la clase modelo y la clase esfera para el disparo del modelo (minion).

Dentro de glMotorJuegosBase hemos declarado un array de punteros `vector<Particula*> solidos;` Aquí metemos las bolas de tipo partículas para poder hacer la colisión y comprobar si existen las colisiones sólo entre esfera y la pelota.

En cargarPartida creo las bolas de tipo partícula y las meto en el array de punteros, y también en escena para que las muestre por pantalla al renderizar.

Todo lo metemos en escena, donde añadimos las pelotas, los planos, etc. Ponemos el vector y así saber cuáles son las pelotas que tenemos que calcular sin tener que recorrer el array de la escena entera.

Para renderizar lo metemos todo dentro del `e.add` que sería la escena y cuando reproduce el render hace la escena y las colisiones, solo de las pelotas.

Cuando pasamos de una pantalla a otra vaciamos el array, metemos otras seis pelotas y las posiciones que tengan y se vuelve a empezar.

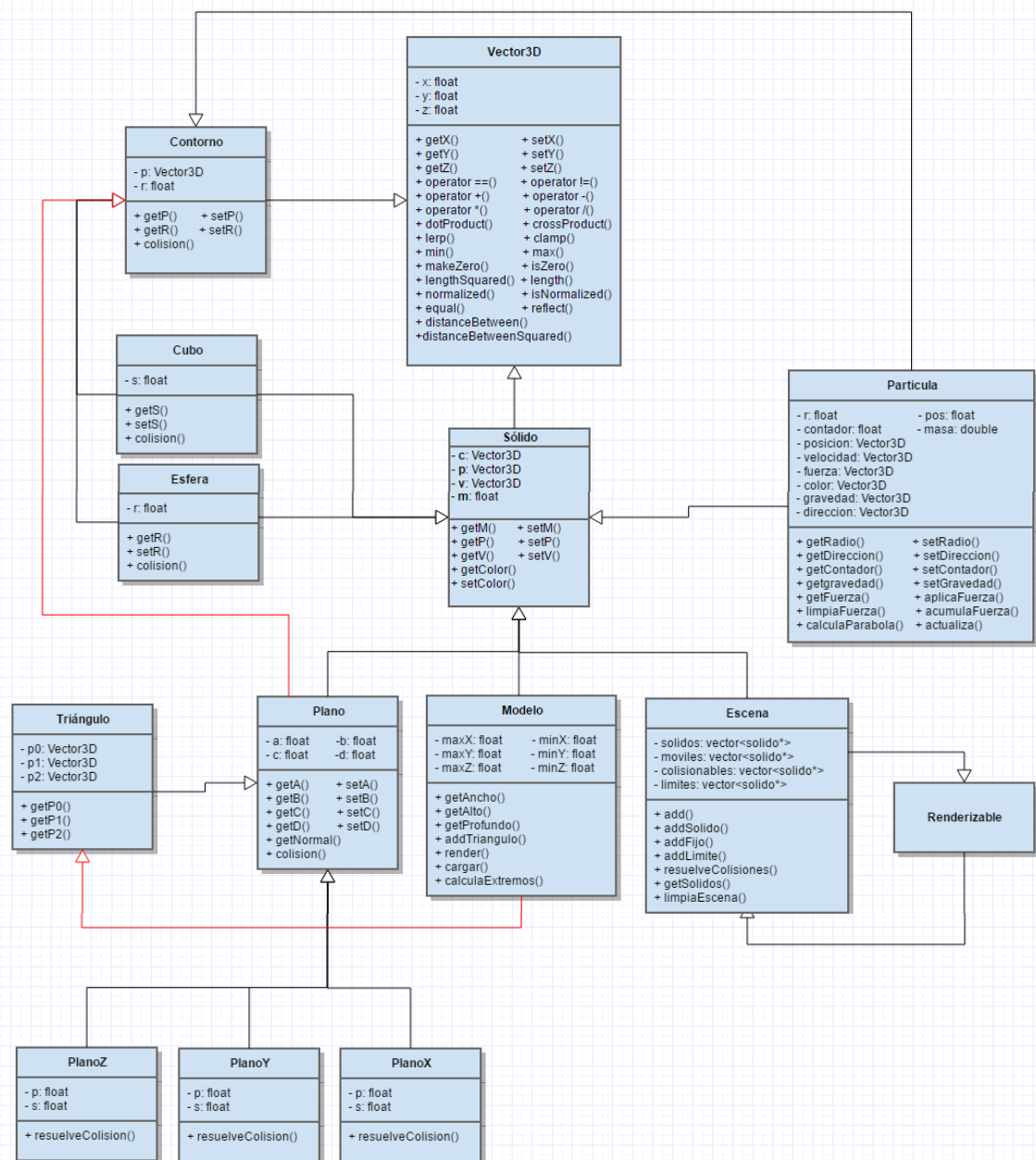
También se creó una función `rand` para crear números aleatorios y generar las pelotas aleatorias y sus posiciones.

Partida, es un contador que cuando destruyes una bola, y cuando llega a cierto número de bolas se sabe que ha acabado y carga el siguiente nivel.

En un fichero almacenamos la mejor puntuación y la puntuación que el jugador va obteniendo en cada partida. Así si en algún momento se deseara ampliar la práctica, con requisitos como, mostrar un ranking por pantalla con más jugadores, no costaría nada, solo valdría con leer el fichero.

En resumen, esto sería una breve descripción de los criterios seguidos para el desarrollo de la práctica del juego Ping Pang Pung.

# UML



## MIEMBROS DEL GRUPO

Nombre y Apellidos	Número de expediente
Ana María Alcaide Recio	9
Carlos Marqués Gonzalez	100
Sergio Sánchez-Uran López	178

## ANEXO INFORMATIVO

Dentro de la carpeta podrá ver dos subcarpetas una con contenido para abrir en Visual Studio y otra con contenido para Linux.

En la carpeta para Practica\_Obligatorio\_PA\_VisualStudio deberá abrir el archivo con nombre OpenGLBase del tipo Microsoft Visual Studio. Y ejecutarlo en modo Release.

## REQUISITOS CUMPLIMENTADOS

Uso del motor gráfico	✓
Juego completo con nivel	✓
Manual de usuario y documento	✓
Documento con las decisiones del juego	✓
Menú inicial	✓
Mejores puntuaciones	✓
Guarda ranking y restaurarlo	✓
Modificar la orientación de la vista	✓
Comentar el código	✓
Añadir alguna mejora (más niveles)	✓
Escritura en fichero de puntuación	✓