

Memoria Videojuego

Álvaro Barrios Gil

Adrián Fañanás Baeza

David Carmona Casado

Alberto Milián Barquín

03/05/2017



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza



Contenido

1. Introducción	4
1.1. Descripción del juego	4
2. Equipo de desarrollo	5
3. Ficha técnica	5
4. Historia	6
5. Objetivos del juego.....	6
6. Ventana Inicial de Juego.....	7
7. Clasificación PEGI	9
8. Power-Ups	10
9. HUD	11
10. Enemigos: IA.....	12
11. Tiempo y puntuaciones	15
12. Controles	16
13. Audio y música	16
14. Requisitos y Resoluciones	16
15. Diagrama de estados del juego	17
16. Jugabilidad del 3D	18
17. Pantalla de juego 2D y 3D	20
18. Cambios respecto a la primera versión.....	22
19. Cronograma de actividades y reparto de tareas.....	24
20. Bibliografía	26
ANEXO I: Capturas del Juego.....	27





1. Introducción

1.1. Descripción del juego

Bomberman [1] es una franquicia de videojuegos estratégico-laberínticos originalmente desarrollada por Hudson Soft. El juego original fue lanzado en 1983 (figura 1.1.1), y desde entonces se han lanzado nuevos juegos. Este personaje tiene más de 60 juegos como protagonista, antagonista o secundario como en Bomberman 64 y Wario Blast, y se encuentra en prácticamente todas las consolas de todo el mundo. Tiene un gran éxito en Japón, aunque éste no es tan grande en América.

El juego consiste en poner bombas para ir rompiendo bloques hasta encontrar la salida que está debajo de uno de ellos y matar a los diferentes enemigos que están en el laberinto antes de que se acabe el tiempo. Debajo de los bloques pueden aparecer diferentes power-ups que aumentan el número de bombas, su potencia, velocidad de movimiento, etc....

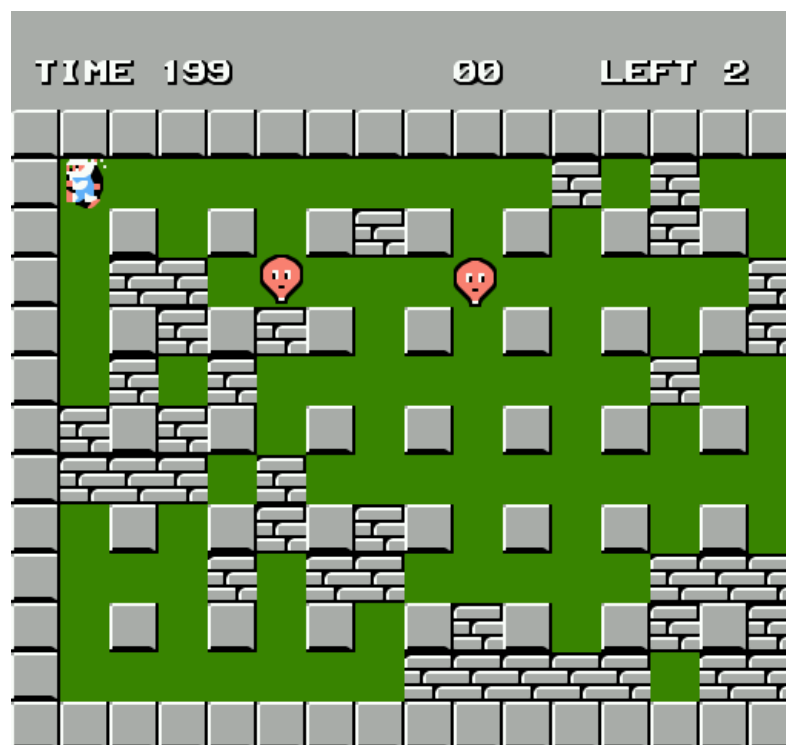


Figura 1.1.1. Bomberman NES (1983).

2. Equipo de desarrollo

El equipo encargado en el desarrollo del videojuego está compuesto por cuatro miembros. Cada miembro del equipo tendrá un rol determinado dentro del equipo:

- **Programación:** Álvaro Barrios, Adrián Fañanás, David Carmona, Alberto Milián.
- **Arte:** Álvaro Barrios.
- **Diseño:** Adrián Fañanás.
- **Inteligencia Artificial:** David Carmona.
- **Calidad:** Alberto Milián.

3. Ficha técnica

- **Desarrollo:** Grupo 5.
- **Producción:** Grupo 5.
- **Género:** Árcade, Acción.
- **Número de jugadores:** 1 Jugador.
- **Plataforma:** PC.
- **Formato:** Digital.
- **Textos:** Español.
- **Voces:** No.
- **Online:** No.
- **Clasificación:** PEGI 3.



4. Historia

Bomberman es un robot que vivía felizmente con su mujer en una casita a las afueras de la ciudad, Bomberman trabajaba en la producción de bombas en una fábrica de la ciudad. Un día mientras trabajaba, BomberGirl (su mujer) fue secuestrada por el malvado bomberBlack.

Bomberman enfadado por la situación decide irle a buscar al castillo donde su mujer está retenida y rescatarla. El juego contará las aventuras que Bomberman vivirá dentro del castillo para poder rescatar a su mujer, vencer a BomberBlack y volver a vivir felizmente con su mujer.

¿Conseguirá vencer a BomberBlack? ¿Rescatará sana y salva a BomberGirl?

5. Objetivos del juego

El juego consta de un modo arcade en el cual un jugador deberá pasar los tres niveles disponibles sin perder las dos vidas que se le proporcionan. En cada nivel el jugador deberá manejar al personaje de bomberman a través de un laberinto formado por bloques y pasillos. El jugador podrá poner una única bomba en cualquier casilla libre de los pasillos y, después de 3 segundos, esta explotará en un radio de hasta 7 casillas dejando libre al jugador de poner otra. El número de bombas a colocar es uno, excepto cuando el Bomberman coja power-ups que aumenten el número de bombas.

Esta explosión puede acabar con enemigos, ciertos bloques o con el propio jugador (eliminando una vida de su contador).

El nivel se visualizará por completo al jugador, por lo que no hay ningún tipo de scroll de la pantalla.

Para pasar de nivel, el jugador deberá poner bombas en el laberinto para acabar con todos los monstruos del nivel y, al destruir ciertos bloques, encontrar la salida. Si la salida se encuentra antes de acabar con todos los enemigos, no será utilizable y el jugador deberá acabar con ellos para activarla y pasar de nivel.



6. Ventana Inicial de Juego

La ventana principal del juego (figura 6.0.1) consta de las siguientes opciones:

- **New game:** Empezar una nueva partida
- **Scores:** Sirve para ver las puntuaciones finales
- **Options:** Permite configurar las opciones del juego
- **Exit:** Cerrar y salir del juego.



Figura 6.0.1. Menú principal.

En el menú de configuración de opciones (figura 6.0.2) se puede cambiar la resolución del juego, activar o desactivar sonido y efectos de sonido y cambiar los gráficos del juego de 2D a 3D.

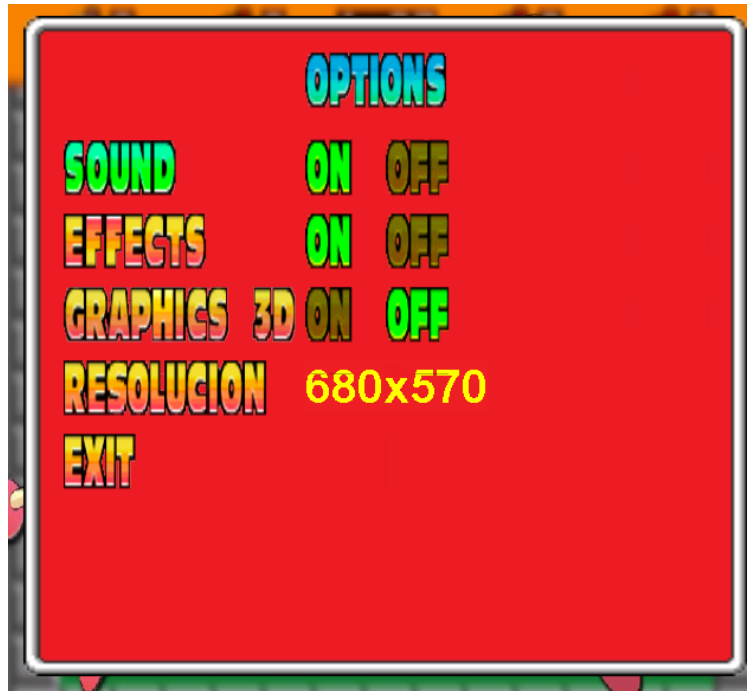


Figura 6.0.2. Ventana de opciones.

La ventana de puntuaciones(Scores) (figura 6.0.3) permite ver las 5 mejores puntuaciones logradas en el juego, para salir de esta ventana hay que pulsar la barra espaciadora.



Figura 6.0.3. Ventana de puntuaciones.

7. Clasificación PEGI

Las etiquetas PEGI [2] se colocan en el anverso y el reverso de los estuches e indican uno de los siguientes niveles de edad: 3, 7, 12, 16 y 18. Indican de manera fiable la idoneidad del contenido del juego en términos de protección de los menores. La clasificación por edades no tiene en cuenta el nivel de dificultad ni las habilidades necesarias para jugar.

El contenido de este bomberman se corresponde con la etiqueta PEGI 3 (figura 7.0.1).

El contenido de los juegos con esta clasificación se considera apto para todos los grupos de edades. Se acepta cierto grado de violencia dentro de un contexto cómico (por lo general, formas de violencia típicas de dibujos animados como Bugs Bunny o Tom y Jerry). El niño no debería poder relacionar los personajes de la pantalla con personajes de la vida real, los personajes del juego deben formar parte exclusivamente del ámbito de la fantasía. El juego no debe contener sonidos ni imágenes que puedan asustar o amedrentar a los niños pequeños. No debe oírse lenguaje soez.



Figura 7.0.1. PEGI 3.

8. Power-Ups

En los diferentes laberintos, al romper los bloques pueden aparecer diferentes power-ups que se enuncian a continuación:



Speed-Up: Incrementa la velocidad del Bomberman.



Speed-Down: Decrementa la velocidad del Bomberman.



Salto: Permite saltar 1 muro, este power-up desaparece con su uso.



Tiempo: Incrementa el tiempo en 10 segundos.



1-UP: Aumenta el contador de vidas de Bomberman en uno, en el caso de que se tengan 3 vidas no tendrá ningún efecto.



Bomb-Up: Incrementa el número de bombas que puede poner el jugador, hasta un máximo de 5.



Fire-Down: Decrementa en 1 el rango de las bombas hasta un mínimo de 1



Fire-Up: Incrementa en 1 el rango de las bombas hasta un máximo de 7.



Full-Fire: Aumenta a 7 el rango de las bombas.

Si el jugador coge dos power-ups repetidos, se mantendrá el efecto del último power-up encontrado en el caso de Speed-Up, Speed-Down y Salto. Los power-ups se mantendrán en pantalla durante 10 segundos, si se acaba el tiempo, desaparecerán. Entre niveles no se conservan los power-ups.

9. HUD

En la parte superior de la pantalla de juego se visualiza un apartado (figura 9.0.1) en el que se aprecia claramente las vidas restantes, la puntuación, los power-Ups obtenidos y el tiempo restante para pasar el nivel.

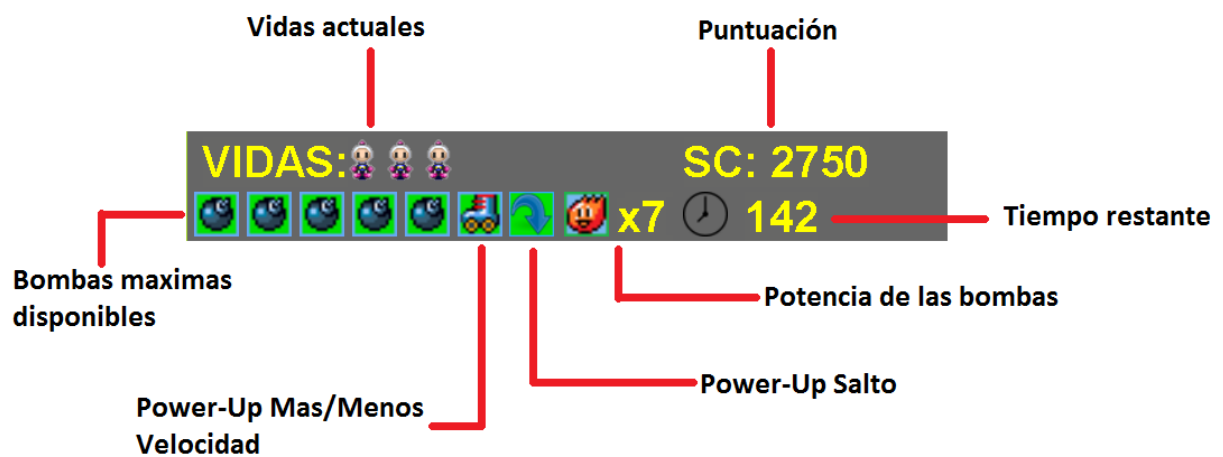


Figura 9.0.1. HUD.



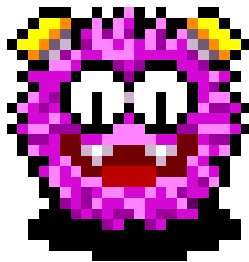
10. Enemigos: IA

Los enemigos varían dependiendo del nivel en el que se encuentra el jugador. Hay tres tipos de inteligencia, el primer nivel de inteligencia se moverá de forma “pseudoaleatoria”, el segundo, tendrá la capacidad de esquivar bombas, y el tercer nivel se moverá hacia la posición del Bomberman con la intención de matarlo.

De esta manera cada uno de los enemigos dispondrá de uno de los niveles de inteligencia en función de la dificultad que se quiera dar al jugador. Los tipos de IA y los enemigos se detallarán a continuación:



- **Poro Verde:** Enemigo con movimiento “pseudoaleatorio”



- **Poro Rosa:** Enemigo que esquiva bombas.



- **Poro Azul:** Enemigo que va hacia BomberMan.

El enemigo con movimiento “pseudoaleatorio” va a tener una mínima memoria para saber cuál ha sido su último movimiento, por tanto, de donde ha venido, de tal

forma que no vuelva atrás y evitar movimientos excesivamente simples. Para ello, guardo en una variable de tipo entero, cuál ha sido mi último movimiento y a partir de ahí calculo la probabilidad de moverme en las otras tres direcciones posibles, favoreciendo que se mueva hacia sitios inexplorados, para eso en cada dirección inexplorada genero un numero aleatorio y lo multiplico por 0.7. Sin embargo, voy a penalizar volver atrás y esa dirección la multiplico por 0.1. Como se muestra en la figura:

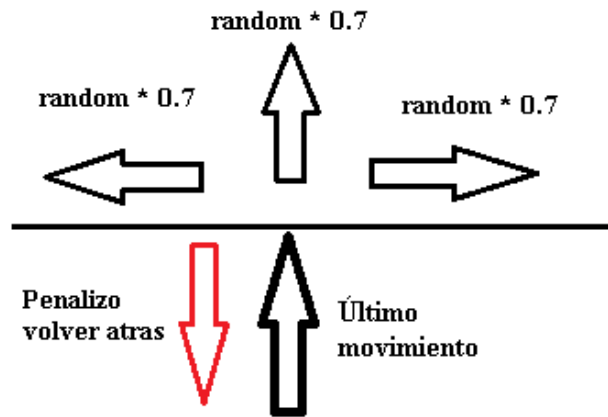


Figura 10.0.1. Algoritmo enemigo pseudoaleatorio

El enemigo que esquiva bombas se implementó mediante una red neuronal con Neuroph [3], concretamente un perceptrón multicapa. La red consta de 5 neuronas de entrada, 20 neuronas en la capa intermedia y 4 en la salida.

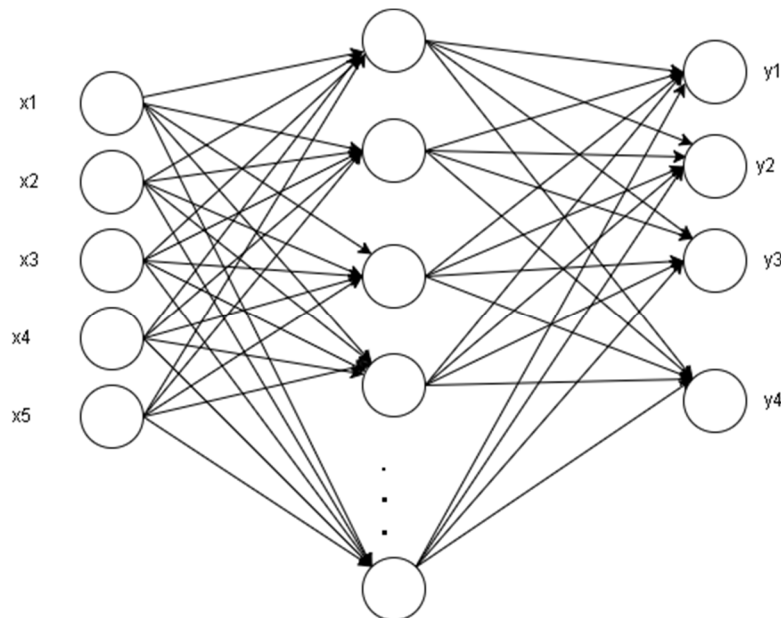


Figura 10.0.2. Red neuronal del enemigo que esquiva bombas

Las 5 neuronas de entrada corresponden a codificar el espacio del enemigo en un radio de dos casillas:

00000	00001	00010	00011	00100
00101	00110	00111	01000	01001
01010	01011	Enemigo	01100	01101
01110	01111	10000	10001	10010
10011	10100	10101	10110	10111

Como podemos observar son 24 diferentes configuraciones posibles, se necesitan 5 bits para representar todos los casos, esos 5 bits representan las neuronas de entrada.

Cada casilla, en un radio de proximidad 2 respecto al enemigo, tiene una codificación como se ha mostrado anteriormente. Cuando el Bomberman ponga una bomba, si ésta se encuentra en el espectro de proximidad del enemigo (proximidad 2), se calculará los inputs de la red en función de la posición de la bomba respecto al enemigo.

Las cuatro neuronas de salida representan en qué dirección puedo moverme (con 1) o en cuál no (con 0), y_1 codifica "arriba", y_2 codifica "abajo", y_3 codifica "izquierda" e y_4 codifica "derecha". Por ejemplo, si el Bomberman pone una bomba justo en la casilla de encima donde está el enemigo:

00000	00001	00010	00011	00100
00101	00110	00111	01000	01001
01010	01011	Enemigo	01100	01101
01110	01111	10000	10001	10010
10011	10100	10101	10110	10111

Se activará esa input, mediante unos cálculos que determinan la posición relativa de la bomba respecto al enemigo. Las partes sombreadas de naranja indican el radio de acción de la bomba (en el caso por defecto, que la bomba sea de radio 1). Con esa input, las outputs deseadas son que se mueva a la derecha o a la izquierda por tanto $y_1 = 0$, $y_2 = 0$, $y_3 = 1$, $y_4 = 1$.

Como las bombas pueden tener hasta radio 7, en las configuraciones introducidas a la red neuronal se decidió siempre el movimiento "seguro", es decir, suponiendo que la bomba sea de radio máximo. Esto influye en los outputs de la red, se restringen más los movimientos posibles pero aumentas la probabilidad de vida del enemigo.

Este enemigo, tiene un radio de proximidad respecto al Bomberman, si traspasa la frontera se comporta como se ha mencionado anteriormente, pero si está lejos, se comporta igual que el enemigo “pseudoaleatorio”.

El enemigo que se mueve hacia el Bomberman se basa en el mismo algoritmo explicado en la presentación anterior de la versión 2D. Es un algoritmo basado en reglas, pero tiene en cuenta dónde hay bloques que pueden obstruirle para elegir una ruta mejor. Al igual que la IA que esquiva bombas, tiene un radio de proximidad, si se acerca al Bomberman va hacia él, si está fuera del radio se comporta como el enemigo “pseudoaleatorio”.

11. Tiempo y puntuaciones

El jugador dispondrá de un tiempo limitado para terminar cada nivel. Este será de 4 minutos y, si no ha conseguido acabarlo en ese tiempo, el jugador morirá, y perderá una vida.



Figura 11.0.1. HUD: Tiempo restante para pasar el nivel.

El jugador dispondrá también de un contador de puntos, estos puntos se pueden conseguir a lo largo del juego de diferentes maneras, ya sea:

- Acabando con los enemigos: 300 puntos por enemigo.
- Acabando los niveles en el menor tiempo posible: cada segundo restante será un punto extra en la puntuación final.
- Destruyendo cajas: 50 Puntos por caja.



Figura 11.0.2. HUD: Contador de puntuación.

12. Controles

Los controles del juego son los siguientes (figura 12.0.1):

- **Flechas de dirección:** Controlar el bomberman y moverse por el menú.
- **Enter:** Seleccionar una opción del menú/saltar escenas de la historia
- **Tecla U:** Saltar si el bomberman dispone del power-up correspondiente.
- **Barra espaciadora:** Poner bombas.

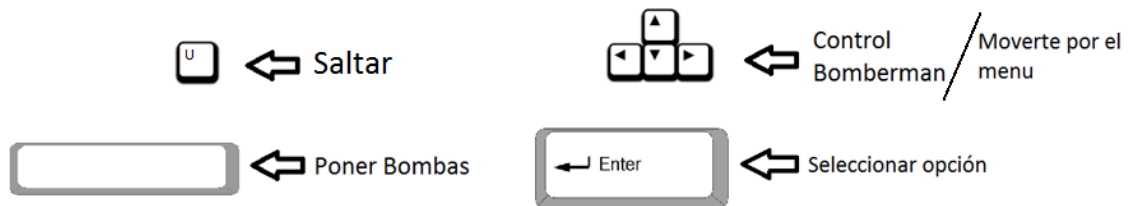


Figura 12.0.1. Controles del juego.

13. Audio y música

Tanto música como efectos de sonido estarán en formato wav. Los efectos de sonido (explosión de bomba, nivel finalizado, moverse por el menú...) se tomarán del Bomberman original (NES).

14. Requisitos y Resoluciones

Los requisitos para poder ejecutar correctamente el juego son los siguientes:

- Versión de java jre y jdk 1.8.0_121 o superior.
- La tarjeta gráfica debe soportar OpenGL 3.0 o superior para el 3D.
- La tarjeta gráfica debe soportar OpenGL 2.9.1 o superior para el 2D.

Durante el juego se puede cambiar la resolución de este, las resoluciones elegidas son las siguientes:

- **Ventana Pequeña:** 425x365
- **Ventana Media:** 680x570
- **Ventana Grande:** 850x690

15. Diagrama de estados del juego

El diagrama de estados que seguirá el juego es el siguiente (figura 14.0.1), consta de 1 estado inicial llamado "INICIO" y 2 estados finales llamados "WIN" y "GAME_OVER", para simplificar el diagrama (figura 14.0.2) se van a juntar los estados de "FASEN" y "FASEN_3D", se entrara en un estado u otro en función de si esta activada la opción de los gráficos 3D o no.

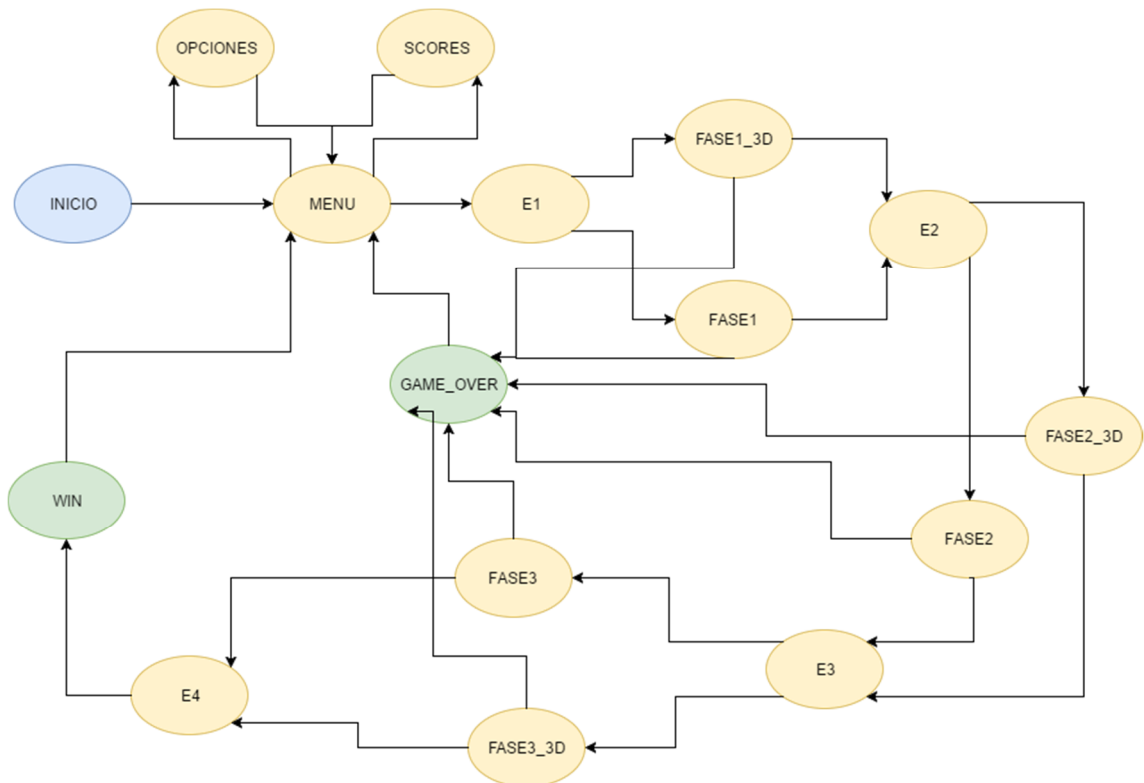


Figura 15.0.1. Diagrama de estados.



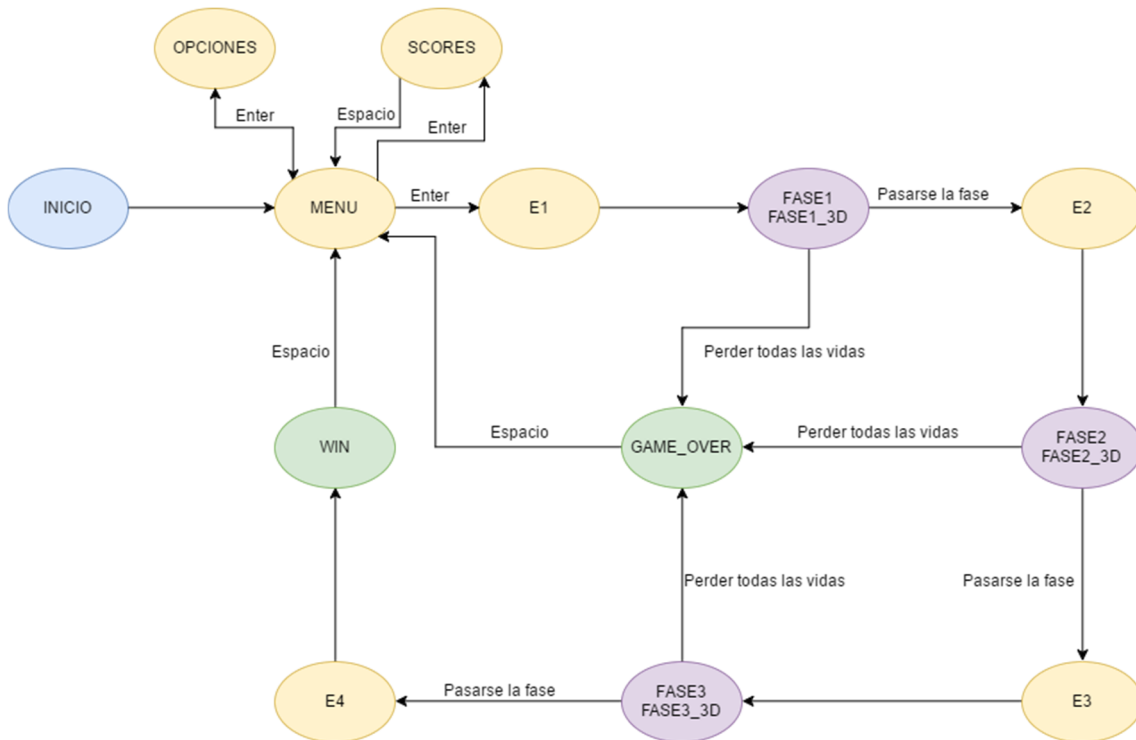


Figura 15.0.2. Diagrama de estados simplificado.

Las transiciones entre los estados son automáticas a no ser que se indique lo contrario en el diagrama de la figura 15.0.2.

16. Jugabilidad del 3D

Para el paso del 2D al 3D se utilizó la librería LWJGL. En las configuraciones del juego, existe una opción que permite visualizar el juego en 2D o 3D, activando o desactivando esa opción. En primer lugar se crearon los cubos que iban a representar los bloques del nivel. Cada cubo consta de 24 vértices (cuatro vértices por cada cuadrado * 6 caras), cada cubo tiene la textura correspondiente dependiendo si es bloque destructible que tiene un color marrón o si es irrompible que tiene color gris. La cámara en 3D se encarga de seguir al jugador desde la parte de atrás, ofreciendo una visión completa del Bomberman. Pulsando la tecla "1", la cámara tiene un pitch = 20 lo que proporciona una visión bastante horizontal sobre el jugador. Pulsando la tecla "2", la cámara se posiciona desde arriba del jugador con un pitch = 90, para mayor visibilidad del laberinto y pulsando la tecla "3", es una vista isométrica sin llegar a ser fija. Manteniendo click izquierdo del ratón y arrastrando hacia arriba o abajo, la cámara se mueve sobre el eje Y ofreciendo cualquier libertad de ángulo sobre el jugador.

En principio, se iba a representar el Bomberman como un cilindro ya que no se tenía el tiempo necesario para realizar una funcionalidad completa del juego en su versión 3D. Más adelante, se observó que daba un aspecto muy pobre al juego y se tomó la decisión de crear un modelo 3D. Para el Bomberman, se ha descargado un modelo 3D en formato .obj y la textura correspondiente, ofreciendo visualmente mejores resultados. El jugador en esta versión tiene la posibilidad de saltar, se tuvo que gestionar el salto con la gravedad que proporcionaba el suelo al descender mediante diferentes variables enteras permitiendo que el movimiento fuera lo más fluido y natural posible. Se tuvieron inicialmente algunos problemas ya que si mantenías la tecla de saltar el personaje “volaba”, que se corrigieron añadiendo algunas comprobaciones e introduciendo una variable booleana que indicara si el personaje se encuentra en el aire o no.

Finalmente se añadieron varios NPCs que se comportan como los enemigos con movimiento “pseudoaleatorio” explicados en el apartado de IA. Aunque en esta versión la funcionalidad es reducida, ya que no te matan aunque pasen por encima de ti.

Además, para mejorar la visualización del juego y eliminar los efectos como “dientes de sierra” se ha añadido un filtro de antialiasing.

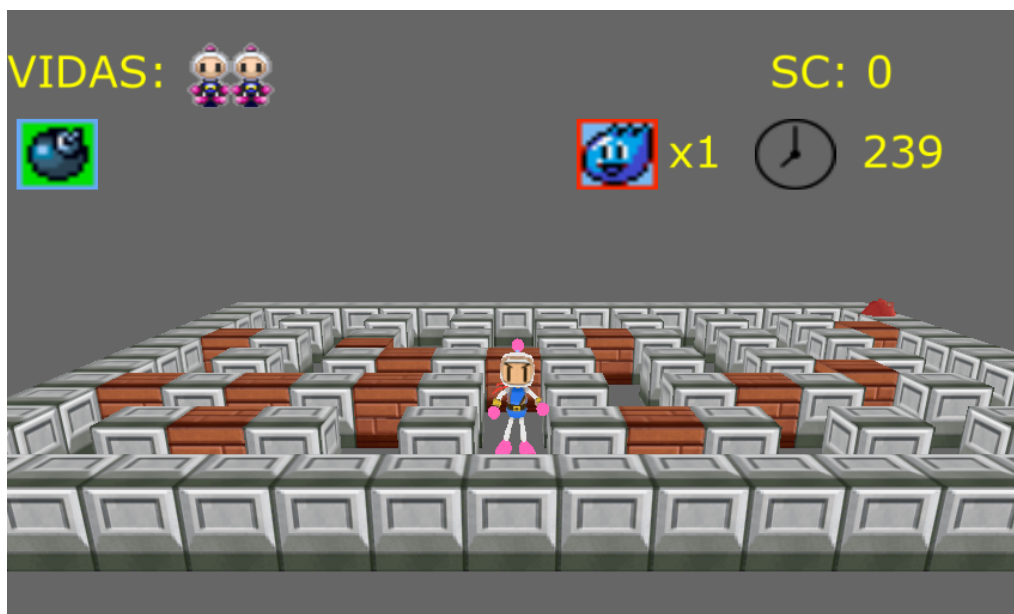


Figura 16.0.1. Imagen del juego en 3D.

17. Pantalla de juego 2D y 3D

En las siguientes imágenes se mostrará una imagen final de la pantalla de juego de uno de los niveles en 2D en la que podremos observar todos los detalles mencionados anteriormente: enemigos, power-Ups, hud...



Figura 17.0.1. Imagen del juego 2D, explosión de una bomba rango 7.

Para el juego en 3D se ha optado por introducir 3 vistas de la cámara principal a la cual se le podrá hacer un scroll de la cámara en horizontal, las vistas proporcionadas son las siguientes:



Figura 17.0.2. Cámara vertical.



Figura 17.0.3. Cámara horizontal.



Figura 17.0.4. Cámara isométrica.

18. Cambios respecto a la primera versión

Desde la primera versión presentada a principios de abril el juego ha sufrido muchos cambios que enumeraremos a continuación:

- Cambio del sistema del juego de posiciones a matriz.
- Cambio en la gestión de colisiones.
- Se ha implementado el ranking de puntuaciones.
- Ahora se puede cambiar la resolución del juego.
- Se ha implementado el menú de opciones (Sonido, Efectos, Resolución, Gráficos 2D/3D).
- Se han añadido más tipos de power-ups (fire-up, fire-down, full-fire, tiempo extra, salto).
- Mejora en animaciones.
- HUD nuevo y mejorado.
- Mejora de la jugabilidad.

- Ajuste de la velocidad del bomberman.
- Mejora de gráficos.
- Cambio de la historia del juego.
- Corrección de diversos errores.
- Añadido diferentes rangos de explosiones.
- Cambios en sprites de enemigos (ahora se diferencian por tipos).
- Implementación de la red neuronal para el enemigo que esquiva bombas.
- Mejoras en la IA aleatoria.
- Mejoras en la IA que persigue al jugador.
- Cambios de los mapas de las fases 2 y 3.
- Implementación de la versión 3D.
- Implementación de varios tipos de cámara para la versión 3D.
- Ahora el bomberman puede tener hasta 3 vidas.
- Cambio en el sistema de puntuaciones.
- Las explosiones de las bombas destruyen power-ups.
- Las explosiones de las bombas hacen explotar otras bombas.
- Cambio en el reescalado de las imágenes para adaptarse a las diferentes resoluciones.
- Reescalado de las letras del menú.
- Añadidas físicas para la versión 3D.
- Añadido filtro antialiasing en la versión 3D.
- Modificada la máquina de estados del juego.



19. Cronograma de actividades y reparto de tareas

ACTIVIDADES	MESES											
	Abril				Mayo				Junio			
	1	2	3	4	1	2	3	4	1	2	3	4
Animación sprites	█	█	█									
Diseño de los niveles			█									
Desarrollo del menú			█									
Implementación colisiones		█										
Diseño de la IA		█	█									
Implementación de los power-ups			█	█								
Redacción del GDC					█							
Aprendizaje OpenGL (LWJGL)	█	█	█	█	█							
Efectos de sonido				█								
Sistema de puntuación y tiempo			█									
Diseño de la historia e introducción		█										
Corrección de errores	█	█	█	█	█							
Implementación versión 3D				█	█							
Cambios en la base del juego	█	█										
Cambios en jugabilidad		█	█	█								
Cambios de resolución		█										
Implementación cámaras 3D		█										
Físicas 3D				█								
Cambio del HUD			█									
Implementación ranking puntuaciones			█									
Timers y Delays para animaciones/Menú		█	█									



Cada actividad ha sido desarrollada por uno o varios miembros del equipo:

Animación de sprites: Álvaro Barrios y Alberto Milián.

Diseño de los niveles: Adrian Fañanás, Álvaro Barrios, Alberto Milián.

Desarrollo del menú: Adrian Fañanás.

Implementación de las colisiones: Alberto Milián.

Diseño de la IA: David Carmona.

Implementación de los power-ups: Álvaro Barrios, Alberto Milián y Adrian Fañanás.

Redacción del GDC: Alberto Milián, Adrian Fañanás y David Carmona.

Aprendizaje de OpenGL: David Carmona, Alberto Milián, Álvaro Barrios y Adrian Fañanás.

Efectos de sonido: Álvaro Barrios y David Carmona.

Sistema de puntuación y tiempo: David Carmona y Adrian Fañanás.

Diseño de historia e introducción: Adrian Fañanás.

Corrección de errores: Alberto Milián, David Carmona, Álvaro Barrios y Adrian Fañanás.

Implementación versión 3D: David Carmona y Álvaro Barrios.

Cambios en la base del juego: Alberto Milián.

Cambios en la jugabilidad: Alberto Milián.

Implementación cámaras 3D: David Carmona.

Físicas 3D: David Carmona y Álvaro Barrios.

Cambios del HUD: Adrian Fañanás.

Implementación ranking puntuaciones: Adrian Fañanás.

Timers y Delays para animaciones/Menús: Alberto Milián.

20. Bibliografía

- [1] [https://es.wikipedia.org/wiki/Bomberman_\(videojuego\)](https://es.wikipedia.org/wiki/Bomberman_(videojuego))
- [2] <http://www.pegi.info/es/index/id/96/>
- [3] http://neuroph.sourceforge.net/sample_projects.html



ANEXO I: Capturas del Juego

