



Universidad Carlos III de Madrid

Escuela Politécnica Superior

Ingeniería en Informática

Diseño e implementación de un “Bomberman” mediante XNA 3.1 aplicando inteligencia artificial

Autor: Juan Ramón Martínez Navarro

Tutor: Juan Peralta Donate



ÍNDICE

- Introducción
- Objetivos
- Bomberman
- XNA
- Requisitos y funcionalidades
- Diseño y clases
- Conclusiones
- Líneas futuras
- Demo



INTRODUCCIÓN

- Los videojuegos son un mercado emergente
 - Desarrollo tecnológico
 - Beneficios
 - Crecimiento de mercado
 - Evolución
- El modelo de negocio evoluciona
 - Evitando grandes crisis
 - Nuevo modelo
 - ◊ Juegos arcade, casuales, indie, bajo demanda
 - ◊ Plataformas de juego online
 - ◊ Plataforma de entretenimiento: cine, música
- Nuevas posibilidades para desarrolladores amateur



OBJETIVOS

- Principales
 - Clon de "Bomberman"
 - Visual Studio .NET y XNA 3.1
 - Aplicar inteligencia artificial
 - Inclusión de un generador de niveles
- Secundarios y personales
 - Desarrollo de un videojuego completo
 - Aprendizaje
 - Primer paso en el desarrollos futuros



BOMBERMAN I

- 1983 primera versión llamada “[Eric and the Floaters](#)”
 - Publicado por Hudson
 - Muy sencillo en diseño y jugabilidad
- 1985 primera versión con el [diseño](#) actual
 - Rediseño del personaje
 - Inclusión de nuevos enemigos
 - Inclusión de Power Ups
 - Mejora de jugabilidad
 - Modo Battle
- Versiones hasta hoy día
 - Bomberman Online, Bomberman DS, Bomberman Live



BOMBERMAN II

- Aspectos básicos del [juego](#)
 - Laberinto cerrado de dimensiones 11x13 casillas
 - Formado por bloques rompibles e irrompibles
 - Los bloques irrompibles tienen posiciones fijas
 - Los bloques rompibles tiene posiciones aleatorias
 - Bombas
 - Power Ups
 - Enemigos
- Mecánica del juego
 - Tiempo limitado
 - Eliminar enemigos
 - Buscar salida



XNA

- Alternativas a XNA
 - Potentes y avanzados
 - Limitaciones
 - Sólo código VS Sólo interfaz
 - Dependencias
- XNA
 - Sobre .NET Framework 2.0
 - Entorno y SDK's gratuitos
 - Programación
 - Múltiples funciones
 - Plataformas finales
 - Plataforma de distribución



REQUISITOS Y FUNCIONALIDADES

- Mantener mecánica del juego original
- Aplicar IA
- Generar niveles
- Incluir Power Ups
- Configuración del juego
 - Dificultad
 - Controles
 - Música y efectos de sonido
- Registrar la puntuación máxima



DISEÑO Y CLASES I

- Diseño y clases: cuatro elementos principales
 - Pantallas
 - ◊ ScreenManager, InputState y GameScreen
 - Jugador
 - ◊ Sprite, Peon y Jugador
 - Enemigos
 - ◊ Sprite, Peon, Enemigo y Jefe
 - Niveles
 - ◊ Nivel, Casilla, Objeto y Sprite



DISEÑO Y CLASES II

- Pantallas
 - ScreenManager
 - Varios tipos de pantalla según la funcionalidad
 - ◊ MenuScreen
 - ◊ LoadingScreen
 - ◊ BackgroundScreen
 - ◊ GameplayScreen



DISEÑO Y CLASES III

- Jugador
 - Respuesta a los controles
 - ◊ Teclado, mando XBOX 360 y Zune
 - Animación de los desplazamientos
 - ◊ Sprites 2D
 - Control y detección de colisiones
 - ◊ Básica y avanzada



DISEÑO Y CLASES IV

- Detección de colisiones de jugador
 - Aplicación de [métodos](#) lógico y físico
 - ◊ Lógico
 - ◊ Físico
 - ◊ !!Erroresjj
 - Depuración del [método](#)
 - ◊ Detección por píxel
 - ◊ No refleja la mecánica de los “Bomberman” originales
 - Nueva implementación
 - ◊ Colisión [básica](#)
 - ◊ Colisión [avanzada](#)



DISEÑO Y CLASES V

- Enemigo
 - Inteligencia Artificial
 - ◇ [Diagramas](#) de estados
 - ◇ Tres tipos de [enemigos](#) básicos y un enemigo final
 - Estructura similar a la de jugador
 - ◇ Animación
 - ◇ Colisiones
 - Comportamiento
 - ◇ Estados y temporizadores de control



DISEÑO Y CLASES VI

- Niveles
 - Agrupados en mapas
 - ◊ Mismas características visuales
 - Generados de forma automática
 - ◊ A partir de matrices de códigos
 - Posición de bloques rompibles
 - Posición de bloques irrompibles
 - Posición jugador
 - Posición aleatoria enemigos
 - Generación salida



DISEÑO Y CLASES VII

- Elementos adicionales
 - [Power Ups](#)
 - ◇ Bomb Up
 - ◇ Remote Bomb
 - ◇ Fire Up
 - ◇ Heart
 - Cámara
 - ◇ Camera2D
 - Nombre de usuario
 - ◇ Teclado virtual



CONCLUSIONES

- Acierto en la selección de XNA como base
 - Curva de aprendizaje
 - Obtención de resultados
 - Comunidad de usuarios
- Clon de "Bomberman" fiel al original
- Inclusión de IA
- Generación de niveles
- Objetivos personales cumplidos



LÍNEAS FUTURAS

- Inclusión del modo de juego Battle
- Inclusión de modo de juego online
- Permitir guardar la partida
- Desbloquear mapas y niveles con passwords
- Inclusión de un editor gráfico de niveles



DEMO



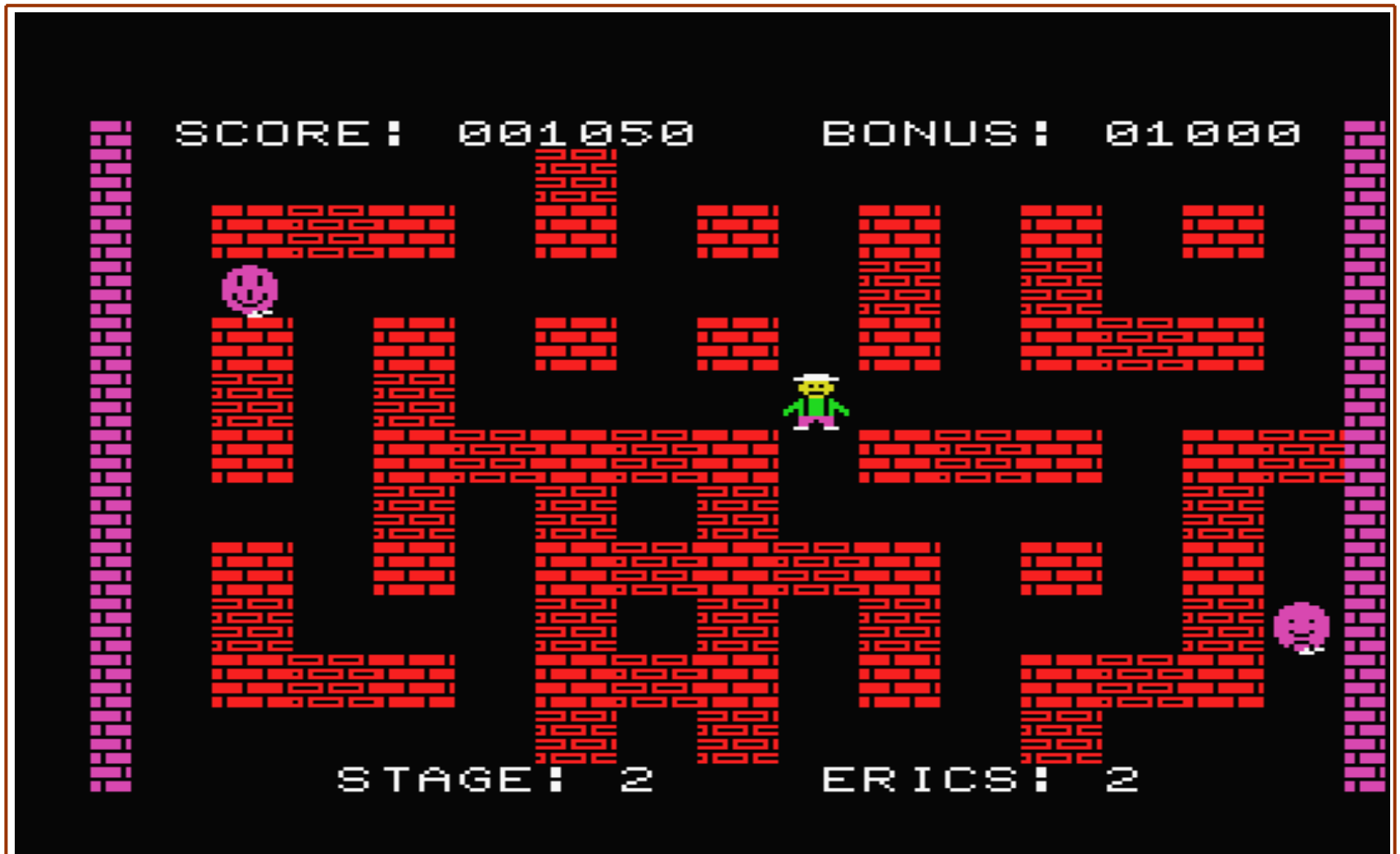


PREGUNTAS





BOMBERMAN I





BOMBERMAN I



Diseño e implementación de un "Bomberman" mediante XNA 3.1 aplicando IA



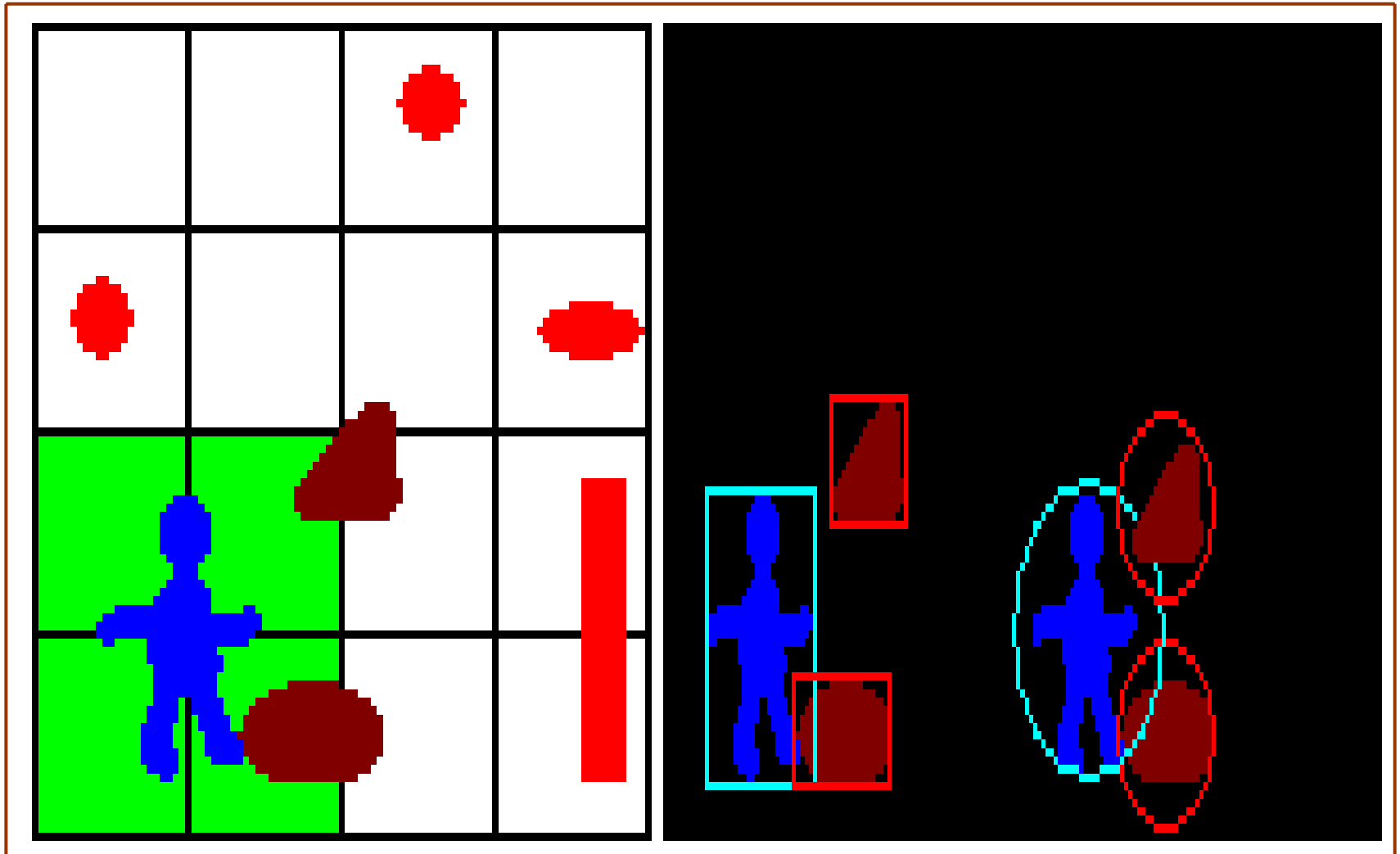
BOMBERMAN II



Diseño e implementación de un "Bomberman" mediante XNA 3.1 aplicando IA

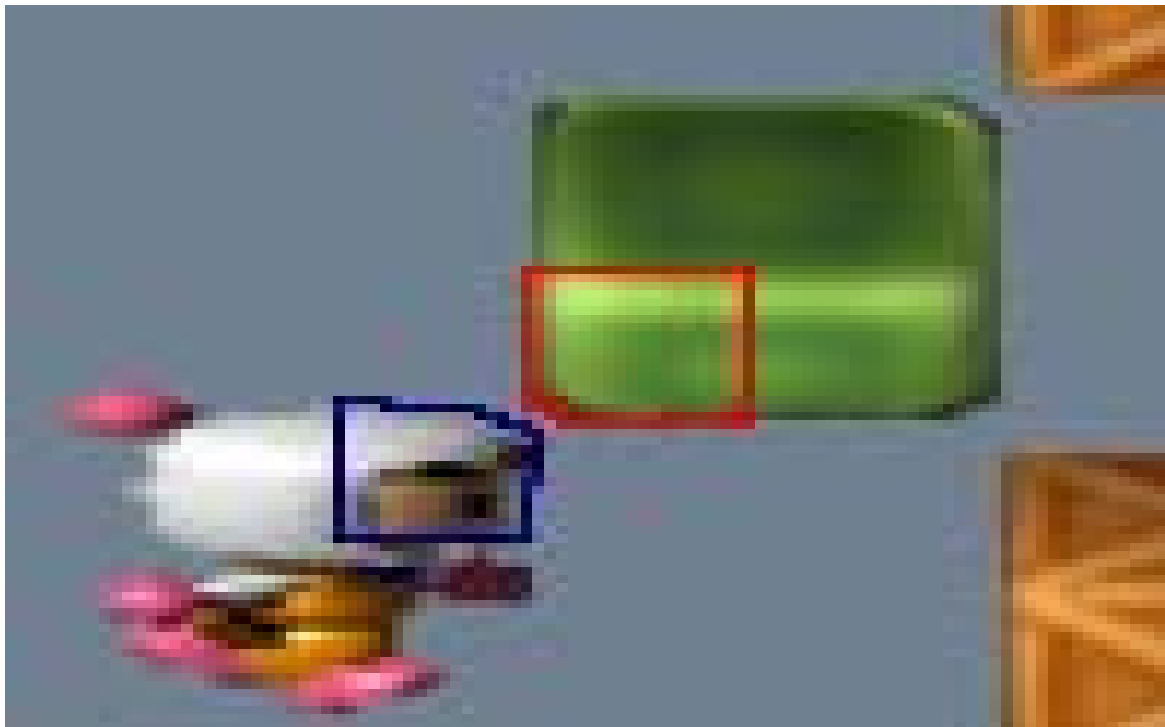


DISEÑO Y CLASES IV



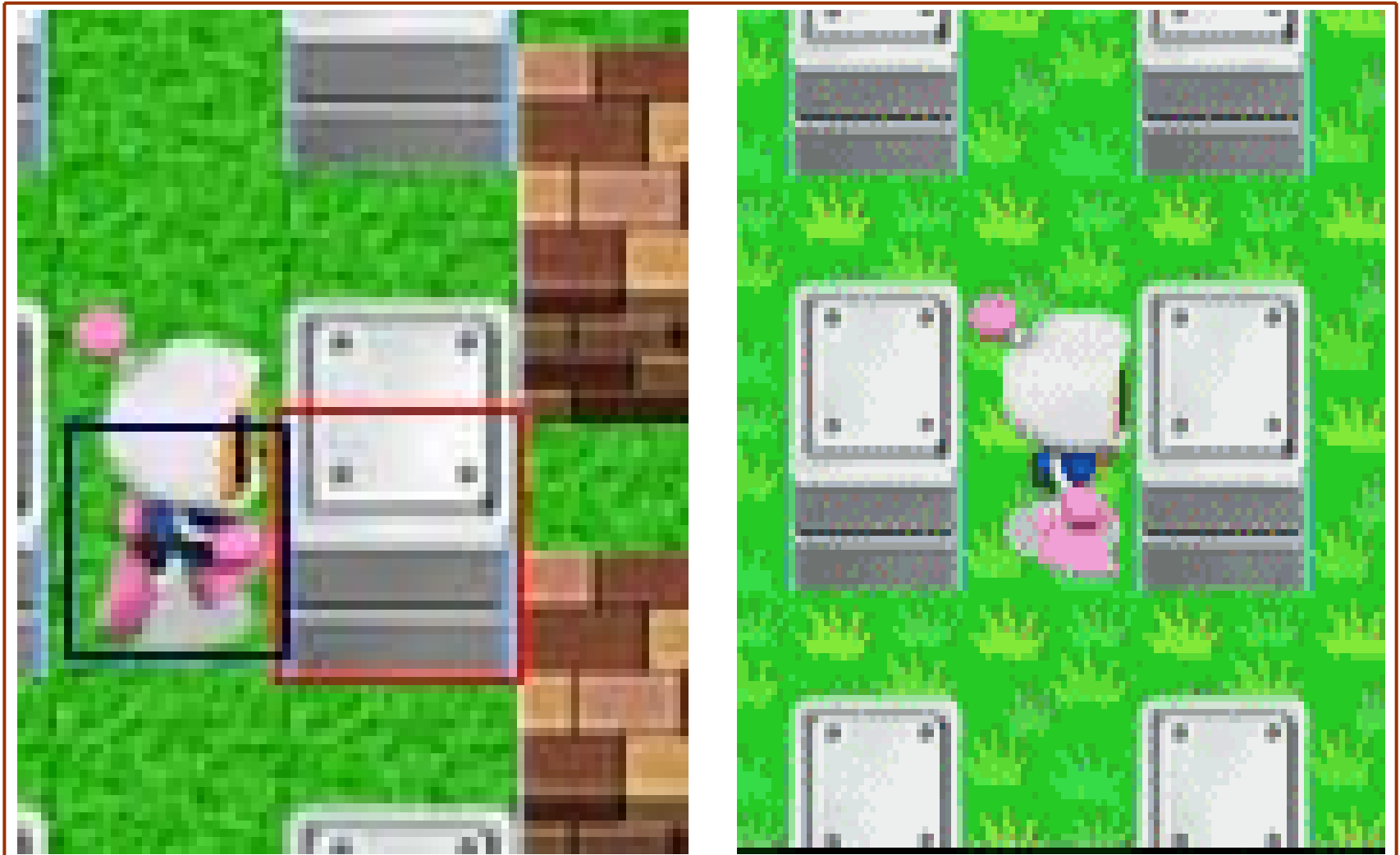


DISEÑO Y CLASES IV





DISEÑO Y CLASES IV



Diseño e implementación de un "Bomberman" mediante XNA 3.1 aplicando IA



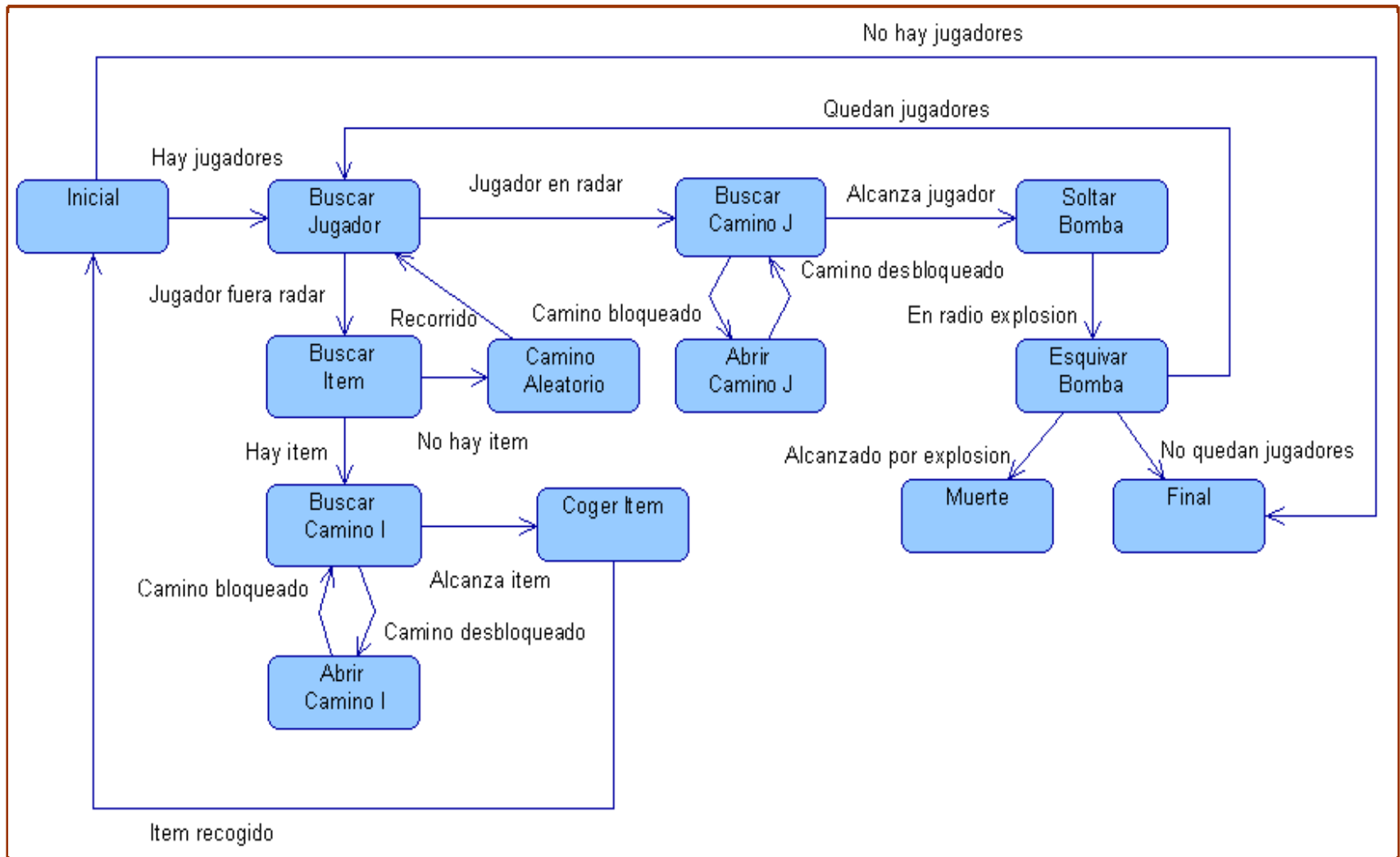
DISEÑO Y CLASES IV

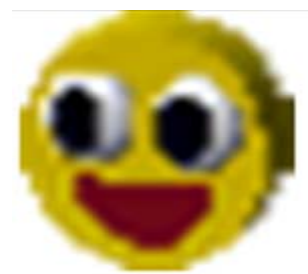


Diseño e implementación de un "Bomberman" mediante XNA 3.1 aplicando IA

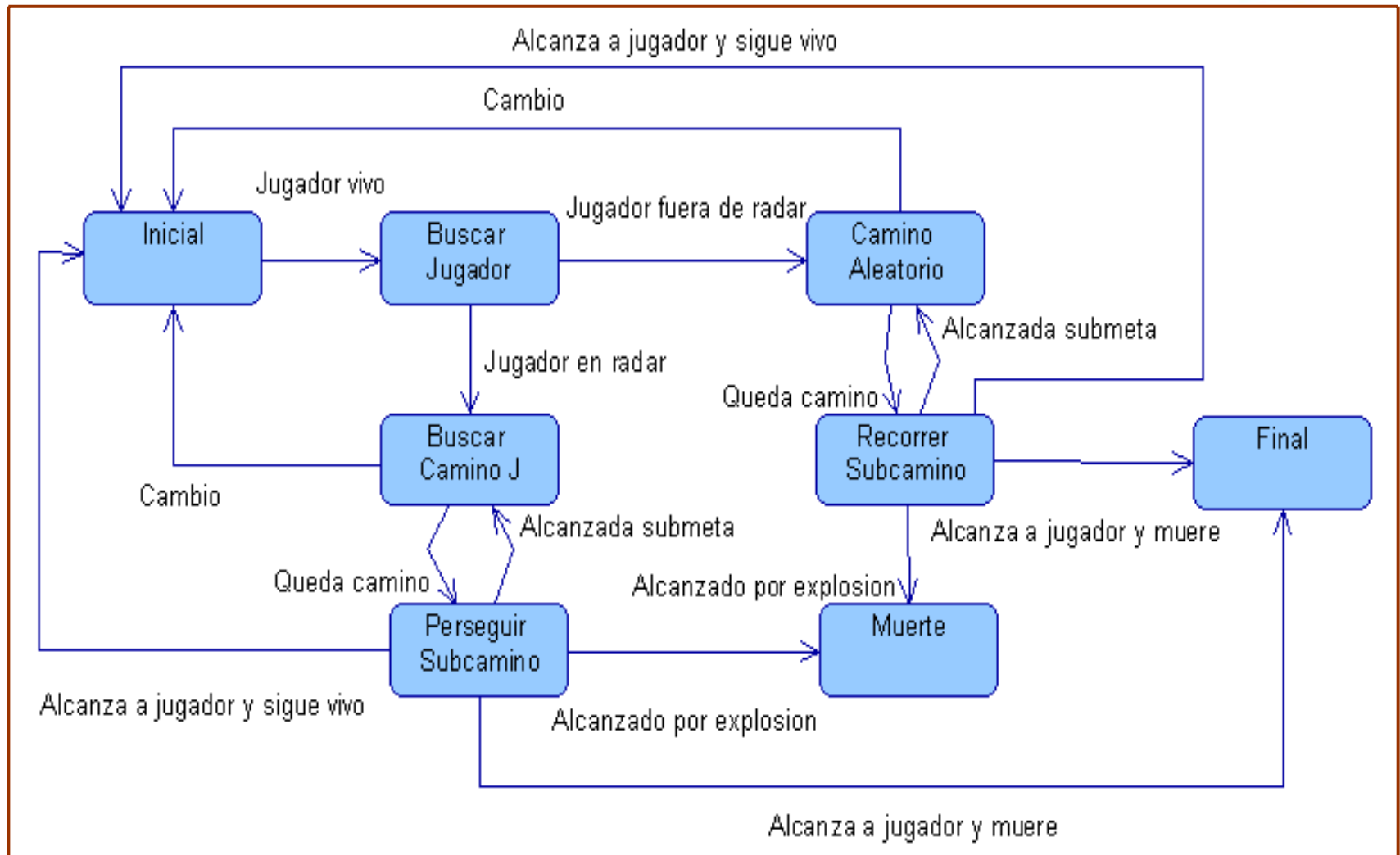


DISEÑO Y CLASES V



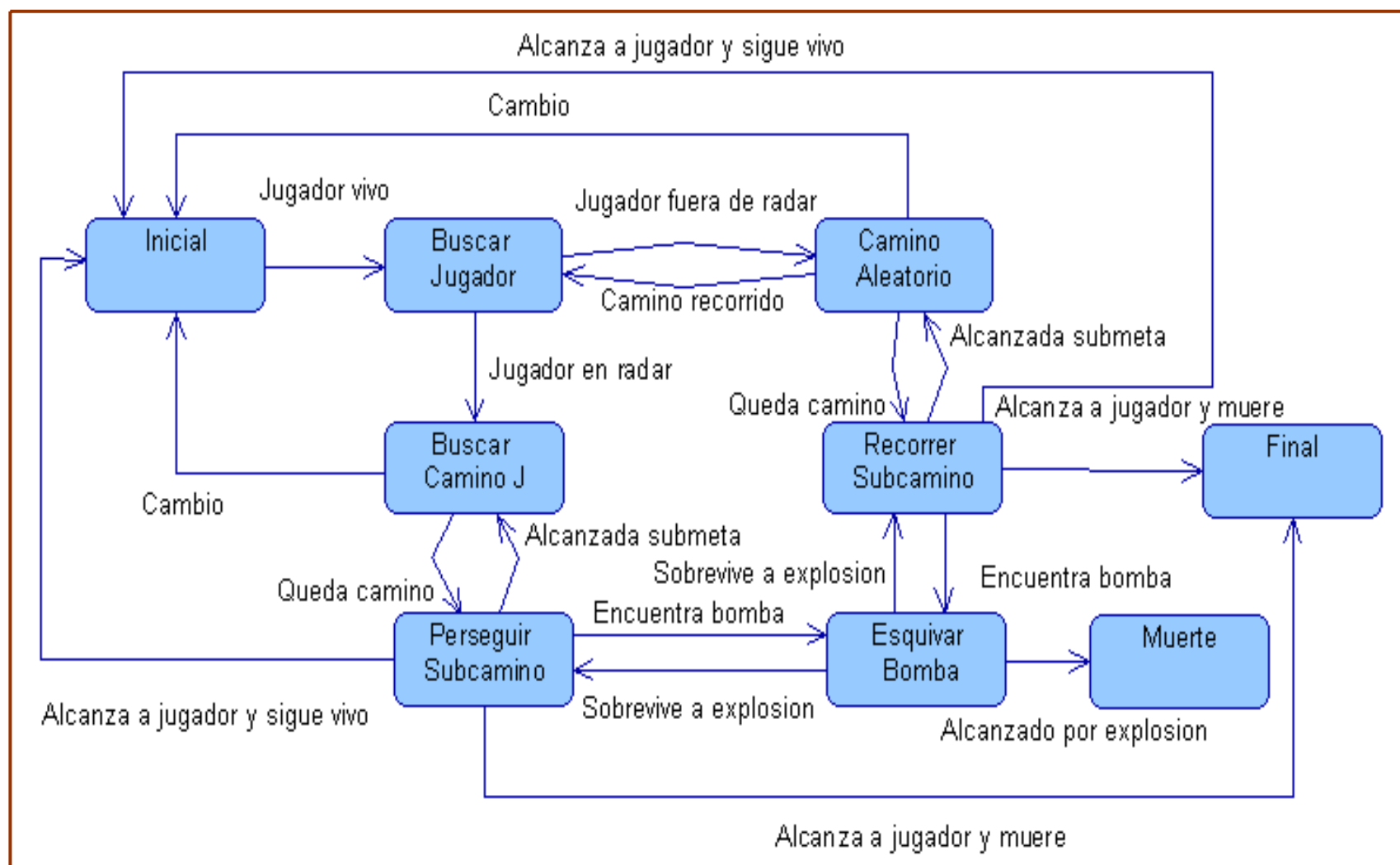


DISEÑO Y CLASES V

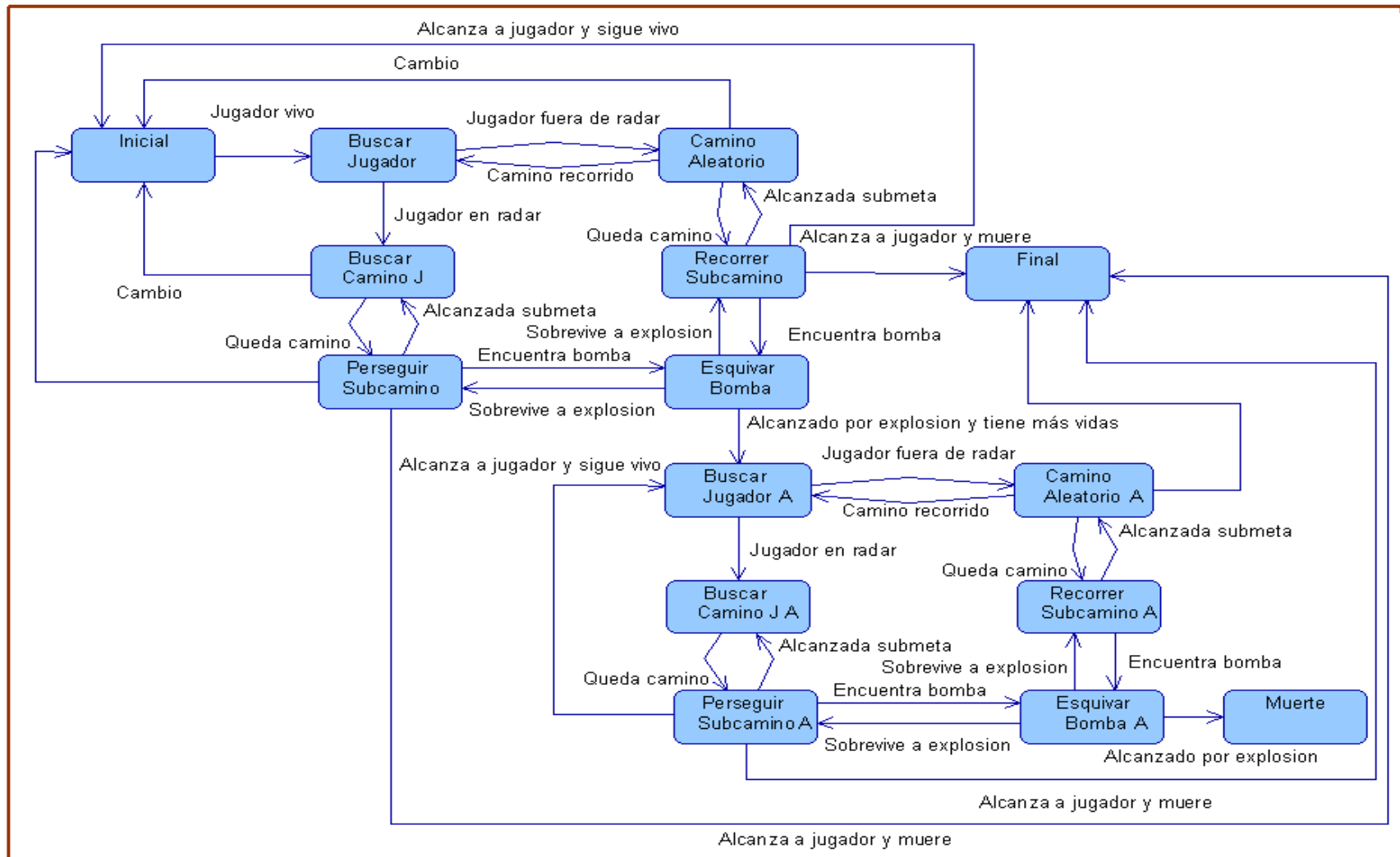




DISEÑO Y CLASES V







DISEÑO Y CLASES V





DISEÑO Y CLASES V

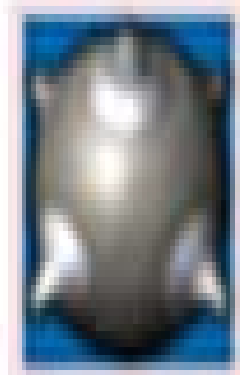
Enemigo				
Nombre	Enemigo_B	Enemigo_C	Enemigo_S	Jefe
Valor	100	250	500	1000
Velocidad	0.5	1	0.5/1	0.5/1.5
Vidas	1	1	2	4
Dificultad	Baja	Media	Alta	Muy Alta



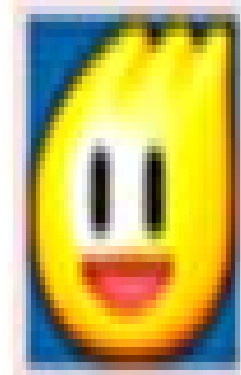
DISEÑO Y CLASES VII



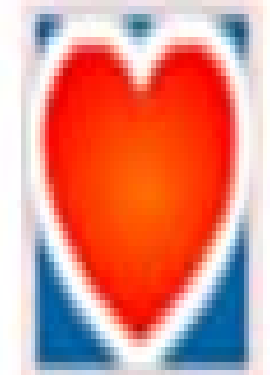
Bomb Up



Remote Bomb



Fire Up



Heart