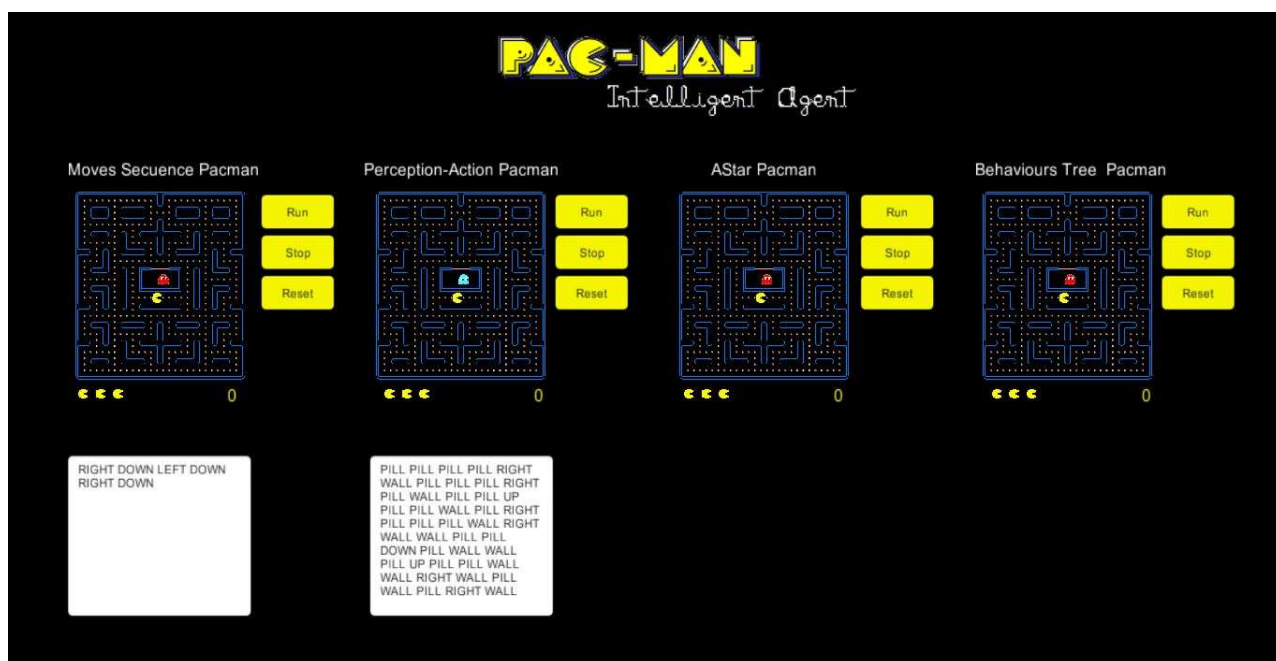


PACMAN Intelligent Agent

Este proyecto ha sido desarrollado para la asignatura de **Sistemas Inteligentes** del itinerario de computación del **Grado en Ingeniería Informática** de la **ULL**.

“

NOTA: El README.md será el informe a entregar.



MIEMBROS DEL GRUPO

Paz Méndez, Germán (alu0100503647@ull.edu.es)

Hernández Pérez, Víctor (alu0100697032@ull.edu.es)

INTRODUCCIÓN

En este documento se recogen todos los aspectos descriptivos del desarrollo e implementación del proyecto final propuesto para la asignatura Sistemas inteligentes.

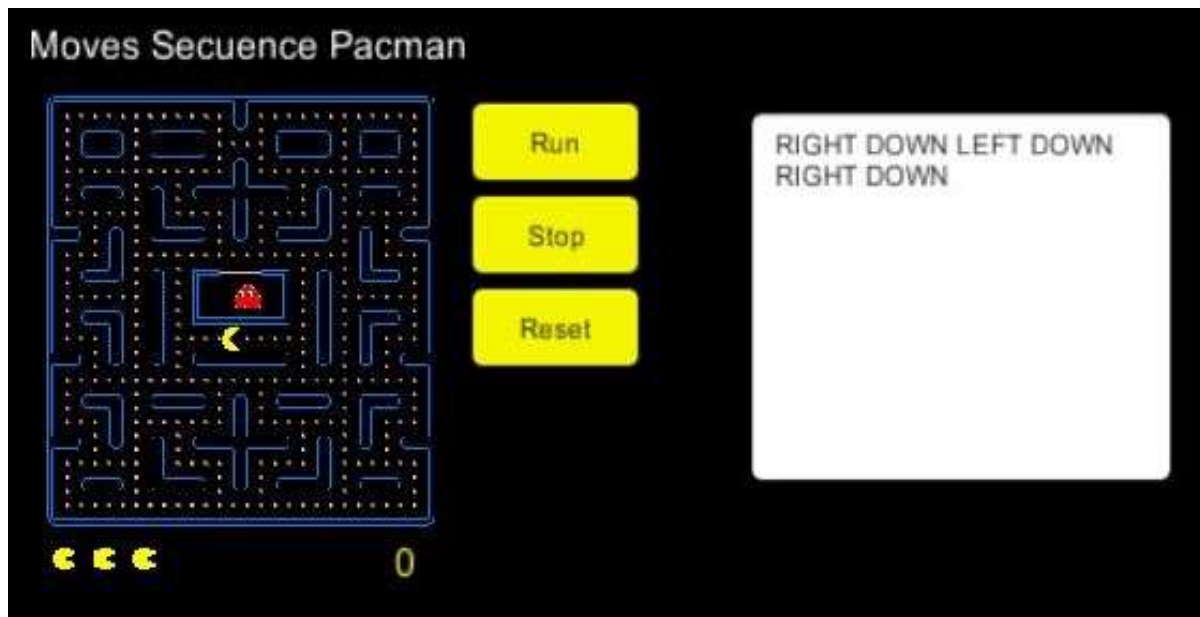
DESCRIPCIÓN

El proyecto ha consistido en la elaboración de varios **agentes inteligentes** para el juego de PacMan utilizando diferentes arquitecturas inteligentes.

“

NOTA: *PacMan* es un conocido juego en tiempo real que ofrece una interesante plataforma para la investigación y el desarrollo de la IA.

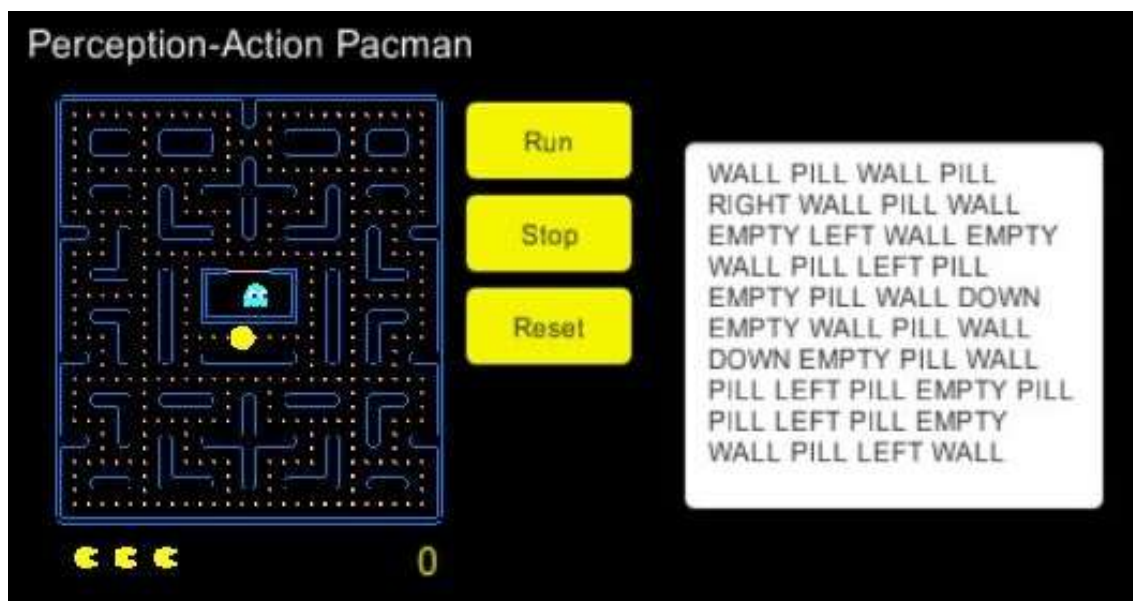
PACMAN basado en una Secuencia de movimientos:



Definimos un agente inteligente `PacMan` que intentará resolver el laberinto de una manera pre-programada antes de que se inicie el juego. Para ello le indicaremos una secuencia de movimientos que deberá seguir para completar el nivel.

Establecemos la posibilidad de que el agente obtenga **"una programación bien adquirida"** para conseguir un **"objetivo perseguido"**, por ello intentamos contemplar este paradigma a través de la posibilidad de establecer unos movimientos iniciales con los que sabemos que el pacman alcanzará la victoria.

PACMAN basado en una tabla de Percepción-Acción:



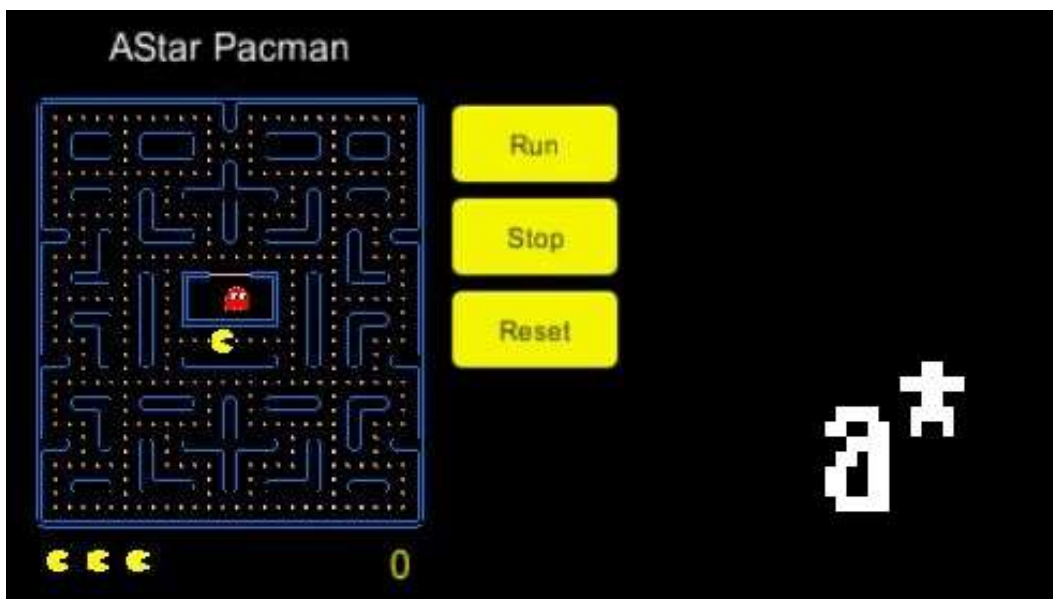
Definimos un agente inteligente `PacMan` de reflejo simple que intentará superar el nivel actuando según las percepciones de objetos o fantasmas que encuentre a su alrededor. Para ello utilizaremos una tabla de Percepción-Acción como la siguiente:

UP	RIGHT	DOWN	LEFT	PERCEPTION / ACTION
WALL	WALL	WALL	WALL	STOP
WALL	WALL	WALL	PILL	LEFT
WALL	WALL	PILL	WALL	DOWN
WALL	PILL	WALL	WALL	RIGHT
PILL	WALL	WALL	WALL	UP
PILL	PILL	WALL	WALL	DOWN
PILL	PILL	PILL	WALL	UP
PILL	PILL	PILL	PILL	RIGHT
WALL	PILL	PILL	PILL	UP
WALL	WALL	PILL	PILL	UP
WALL	WALL	PILL	PILL	UP

“

NOTA: Un inconveniente típico de este sistema es que la tabla puede ser enorme y difícil de construir si establecemos muchas percepciones posibles, por ello hemos añadido a WALL la percepción de los fantasmas y a PILL los espacios vacíos reduciendo así el tamaño de la misma.

PACMAN basado en el algoritmo A*:



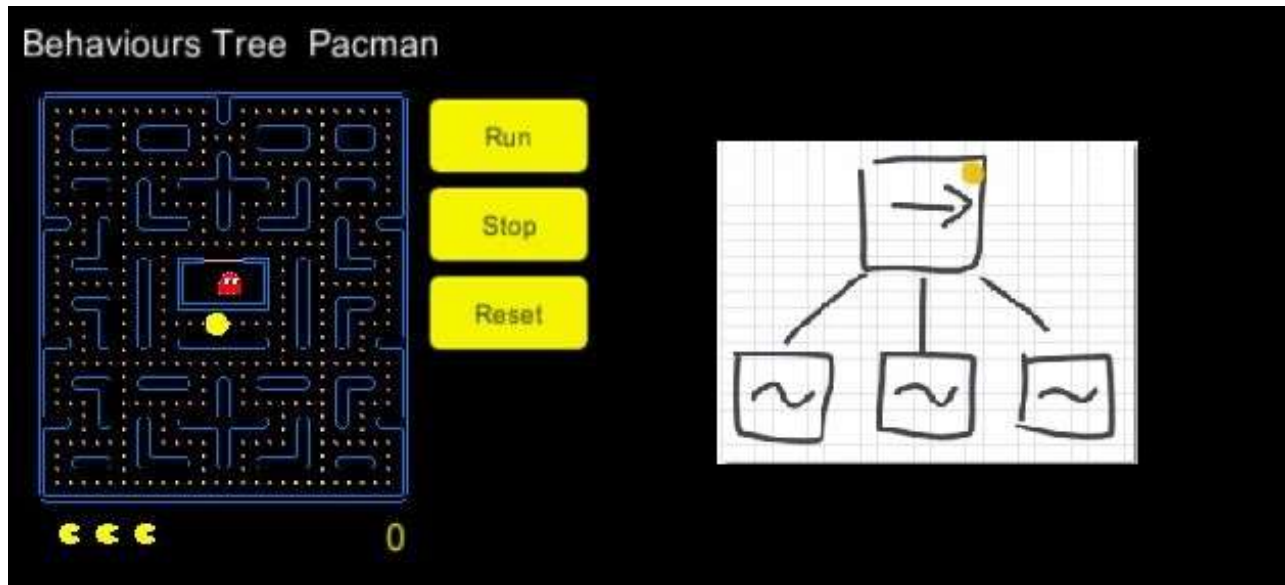
Definimos un agente inteligente `PacMan` que utilizará el algoritmo de A*, siendo este el más conocido en juegos para encontrar la ruta optima.

“

NOTA: EL primer juego en utilizar el A* fue el Pacman, aunque en este caso eran los fantasmillas los que lo implementaban.

Por lo tanto el único objetivo del agente será el de perseguir las píldoras. Para ello hará uso del A*. Además si el fantasma se cruza en el camino del pacman este cambiará la trayectoria para no perder alguna de sus tres vidas.

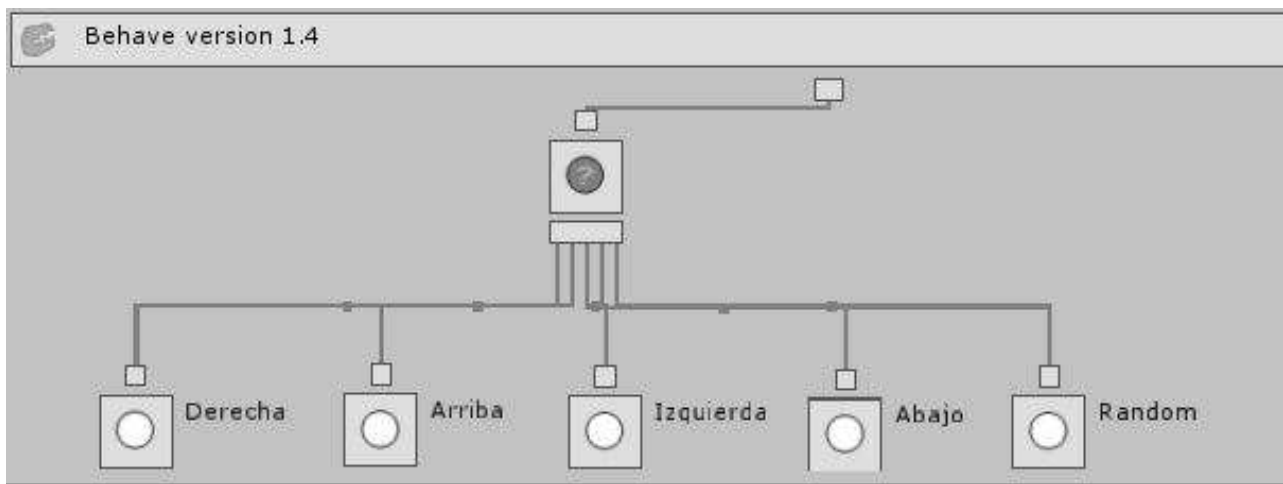
PACMAN basado en un arbol de comportamiento:



Definimos un agente inteligente `PacMan` que simula un comportamiento simple mediante un árbol de comportamiento.

Esta es una de las técnicas que se ha venido utilizando en los últimos años para poder modelar Inteligencia artificial en videojuegos. Cada día ha ido ganando cada vez más adeptos a lo largo de los años.

Como todo lo que es representado por un árbol la visualización de lo que se pretende hacer es bastante sencillo y esto nos permite ganar en tiempo de desarrollo. En nuestro agente hemos añadido un selector y cinco acciones:



Con estas cuatro arquitecturas pretendemos obtener la máxima puntuación en el juego del Pacman.

“

NOTA: En 1999 el jugador **Billy Mitchell** realizó una **partida perfecta de Pac-Man**, entendiéndose como tal una partida en la que el jugador completo los 255 niveles con la puntuación máxima sin ser capturado ni una sola vez. La puntuación máxima es de 3.333.360 puntos.

Hemos establecido la posibilidad de modificar los diferentes archivos de configuración de varios agentes inteligentes Pacman. Con esto podremos determinar la mejor arquitectura para realizar una partida perfecta.

PROYECTOS SIMILARES

OnLine

- MS. Pacman ([URL](#))

Vídeos

- Pacman agent using AI ([URL](#))
- Playing Pacman using AIXI Approximation ([URL](#))
- Pac-Man played by an AI agent (CS188.1x) ([URL](#))

RECURSOS

A lo largo del proyecto hemos utilizado los siguientes **recursos** para la creación del agente inteligente Pacman:

Wikipedia

- Agente inteligente ([URL](#))
- Pacman ([URL](#))
- A* ([URL](#))
- Árbol de comportamiento ([URL](#))
- Tablas de Persepción-Acción ([URL](#))

Lenguajes, plataformas y librerías

- C# ([URL](#))
- UNITY ([URL](#))
- BEHAVE ([URL](#))

Material utilizado

- Sprites



TECNOLOGÍAS DE IA

En el proyecto implementamos **varias arquitecturas** de agente de IA:

- Pre-programación o programación adquirida inicialmente
- Tablas de Persepción-Acción
- A*
- Árbol de comportamiento

DESARROLLO

Lista de tareas y cronograma

Etapas proyecto	Asignación
Propuesta	Nombre
1 Propuesta inicial	Germán Paz Méndez & Víctor Hernández Pérez
Búsqueda de recursos	Nombre
Entrevistas grupales	Víctor Hernández Pérez
2 Búsqueda de información	Germán Paz Méndez
3 Búsqueda de contenido open source	Víctor Hernández Pérez
4 Búsqueda de diferentes arquitecturas de agentes inteligentes	Germán Paz Méndez
Definición formal	Nombre
5 Primera organización de análisis	Germán Paz Méndez & Víctor Hernández Pérez
Informe actualizado	Germán Paz Méndez & Víctor Hernández Pérez
6 Definición del problema	Germán Paz Méndez & Víctor Hernández Pérez
Conceptualizando	Nombre
7 Diseño de la aplicación	Germán Paz Méndez & Víctor Hernández Pérez
8 Definición de las arquitecturas	Germán Paz Méndez & Víctor Hernández Pérez
Diseño	Nombre
9 Juego	Germán Paz Méndez
10 Componentes Juego	Víctor Hernández Pérez
11 Diseño final	Germán Paz Méndez & Víctor Hernández Pérez
Implementación	Nombre
12 Juego	Víctor Hernández Pérez
13 Lógica del juego	Germán Paz Méndez & Víctor Hernández Pérez
14 Lógica del agente	John Daniels
Ensayos y entrega	Nombre
15 Testeando código	Germán Paz Méndez & Víctor Hernández Pérez
16 Testeando juego	Víctor Hernández Pérez
17 Entrega	Germán Paz Méndez & Víctor Hernández Pérez
Evaluación	Nombre
18 Primera convocatoria	Germán Paz Méndez & Víctor Hernández Pérez

The Gantt chart displays the project schedule from January 2015 to January 2016. The timeline is marked by months: January 2015, December 2015, and Enero 2016. Tasks are represented by colored bars indicating their duration:

- Task 1 (Propuesta inicial):** January 1-2, 2015.
- Task 2 (Búsqueda de recursos):** January 3-10, 2015.
- Task 3 (Entrevistas grupales):** January 11-18, 2015.
- Task 4 (Búsqueda de información):** January 19-26, 2015.
- Task 5 (Búsqueda de contenido open source):** January 27-February 3, 2015.
- Task 6 (Búsqueda de diferentes arquitecturas de agentes inteligentes):** February 4-11, 2015.
- Task 7 (Primera organización de análisis):** February 12-19, 2015.
- Task 8 (Informe actualizado):** February 20-27, 2015.
- Task 9 (Definición del problema):** February 28-March 6, 2015.
- Task 10 (Diseño de la aplicación):** March 7-14, 2015.
- Task 11 (Definición de las arquitecturas):** March 15-22, 2015.
- Task 12 (Juego):** March 23-30, 2015.
- Task 13 (Componentes Juego):** March 31-April 7, 2015.
- Task 14 (Diseño final):** April 8-15, 2015.
- Task 15 (Juego):** April 16-23, 2015.
- Task 16 (Lógica del juego):** April 24-May 1, 2015.
- Task 17 (Lógica del agente):** May 2-9, 2015.
- Task 18 (Primera convocatoria):** May 10-17, 2015.

Longer bars represent sub-phases or ongoing activities:

- Entrevista grupales:** January 11-18, 2015.
- Informe actualizado:** February 20-27, 2015.
- Conceptualizando:** March 7-14, 2015.
- Diseño:** March 23-30, 2015.
- Implementación:** April 16-23, 2015.
- Ensayos y entrega:** May 10-17, 2015.

ESTADO ACTUAL

Fase Final. Entregado!

PROBLEMAS ENCONTRADOS

A lo largo del proyecto nos hemos topado con **diferentes problemas**, entre los cuales destacamos:

- Dificultades con la idea general planteada en el comienzo.
- Implementación desde cero del juego Pacman.
- Aprendizaje de librerías externas.
- Tiempo limitado.

POSIBLES MEJORAS

Hemos detectado **varias limitaciones** que corregiremos en un futuro implementándolas:

- Implementación de píldoras de poder.
- Implementación de varios niveles.
- Implementación de fantasmas inteligentes.
- Mejorar los algoritmos para que Pacman perciba mejor a los fantasmas.
- Implementar un árbol de comportamiento más complejo.
- Añadir los sonidos típicos del juego.

FUNCIONAMIENTO

Tenemos un archivo ejecutable (**test1.EXE**) para Windows con el cual el usuario lo podrá ejecutar.

A través de este ejecutaremos el programa y dispondremos para cada arquitectura del agente implementada los botones **START**, **STOP** y **RESET**:

- **START**: ejecuta la simulación de la arquitectura planteada.
- **STOP**: para la simulación de la arquitectura planteada.
- **RESET**: reinicia la simulación de la arquitectura planteada.

Para las arquitecturas "Moves Sequence Pacman" y "Perseption-Action Pacman" **disponemos de una entrada de datos para su configuración de movimientos** según arquitectura.

Con esto disponemos de las herramientas necesarias para ver las diferentes implementaciones de los agentes programados.

CONCLUSIONES

En líneas generales hemos cumplido el objetivo y hemos llegado a crear un pacman inteligente mediante las diferentes arquitecturas expuestas en este informe.

