

## Proyecto Programado #2 – Valor 20%

### Objetivos

- Utilizar grafos para desarrollar la inteligencia artificial de los fantasmas en el juego
- Mejorar la eficiencia del juego mediante el uso de técnicas de hashing
- Implementar y resolver colisiones en la interfaz gráfica del juego
- Aplicar técnicas que mejoren la eficiencia y el tiempo de respuesta de un programa

### Instrucciones

El proyecto consistirá en desarrollar una versión del juego clásico de Arcade, llamado **Pac Man**. Más información del juego en el siguiente enlace:

<https://es.wikipedia.org/wiki/Pac-Man>

Nota: Si escriben “**pacman**” en Google, les aparecerá un “**doodle**” divertido.

Deberán incluir los elementos básicos del juego (pacdots, frutas, los fantasmas, atajos en el laberinto, las píldoras de poder, etc.).

Para la implementación de la inteligencia artificial de los fantasmas, pueden guiarse con los siguientes lineamientos:

#### 1. Comprender los comportamientos de los Fantasmas

- **Blinky (Rojo)**: Persigue directamente a Pacman
- **Pinky (Rosa)**: Intenta predecir la dirección de Pacman y cortarle el paso
- **Inky (Cian)**: Utiliza tanto la posición de Pacman como la de Blinky para calcular su objetivo, lo que lo hace más impredecible
- **Clyde (Naranja)**: Se comporta de manera errática, a veces persiguiendo a Pacman y otras veces alejándose

#### 2. Modos de comportamiento

- **Modo de Persecución (Chase)**: Los fantasmas persiguen activamente a Pacman
- **Modo de Dispersión (Scatter)**: Cada fantasma se dirige a una esquina específica del laberinto
- **Modo Asustado (Frightened)**: Cuando Pacman come una píldora de poder, los fantasmas se vuelven vulnerables y cambian de dirección aleatoriamente

### 3. Algoritmos de Búsqueda

- Utilicen algoritmos de búsqueda en grafos como BFS (Breadth-First Search) o DFS (Depth-First Search) para que los fantasmas encuentren el camino más corto hacia Pacman
- Implementen un sistema de pesos en los caminos para que los fantasmas puedan tomar decisiones más inteligentes sobre qué ruta seguir

### 4. Predicción de Movimiento

- Para fantasmas como Pinky, implementen una lógica que intente predecir la dirección futura de Pacman basándose en su velocidad y dirección actuales

### 5. Optimización y Eficiencia

- Asegúrense de que las decisiones de los fantasmas se tomen en tiempo real sin afectar el rendimiento del juego. Esto puede implicar optimizar los algoritmos de búsqueda y minimizar el uso de recursos.

La aplicación de hashing en la implementación de Pacman puede ser muy beneficiosa para mejorar la eficiencia y el rendimiento del juego. Algunos ejemplos donde lo pueden aplicar:

#### 1. Gestión de Elementos del Juego

- Utilicen una tabla hash para almacenar y gestionar las posiciones de los puntos, frutas y otros elementos del juego. Esto permite acceder rápidamente a estos elementos sin necesidad de recorrer toda la lista de objetos

#### 2. Detección de Colisiones

- Una tabla hash para detectar colisiones entre Pacman, los fantasmas y otros objetos del juego. Al usar hashing, pueden reducir el tiempo de búsqueda y comparación, haciendo que la detección de colisiones sea más rápida y eficiente

#### 3. Optimización de Movimientos

- Usen hashing para almacenar y recuperar rápidamente las rutas óptimas calculadas para los fantasmas. Esto puede ayudar a que los fantasmas tomen decisiones más rápidas y eficientes sobre sus movimientos

#### 4. Almacenamiento de Estados del Juego

- Una tabla hash para almacenar diferentes estados del juego, como las posiciones de Pacman y los fantasmas en diferentes momentos. Esto puede ser útil para implementar funciones de deshacer o para analizar el rendimiento del juego

Además, deben poder guardar el juego en cualquier momento y recuperarlo cuando se inicie de nuevo la aplicación.

Recursos adicionales:

<https://tech.gobetech.com/31539/cual-es-el-algoritmo-utilizado-por-los-monstruos-perseguidores-en-pac-man.html>

<https://www.toolify.ai/es/ai-news-es/descubre-el-sorprendente-comportamiento-de-los-fantasmas-en-pacman-1126891>

<https://rd.udb.edu.sv/bitstreams/663722f7-0734-4bff-ad10-fbbc98814444/download>

<https://www.jovenesenlaciencia.ugto.mx/index.php/jovenesenlaciencia/article/download/3944/3427/12936>

### Características del Programa

- El programa debe ser desarrollado en Python 3.10 o superior
- Asegúrense de aplicar conceptos de programación orientada a objetos
- Para la interfaz gráfica, pueden utilizar la librería que ustedes deseen
- Pueden apoyarse con el uso de herramientas de AI tales como Copilot o ChatGPT. Nada más deben comentar en el código las funciones o métodos que hayan sido creadas con esta tecnología

### Observaciones

- La tarea debe ser realizada en grupos de cuatro personas como máximo
- Se debe entregar el proyecto con el código fuente y librerías necesarias para su ejecución, así como cualquier otro archivo que se requiera
- La fecha límite para la entrega del proyecto es el día Domingo 10 de noviembre de 2024
- En caso de copia (dos proyectos o más con mucha similitud que lo demuestren) o plagio (códigos descargados de Internet, libros o cualquier otro material), la nota del proyecto es automáticamente 0
- Debe indicar todos los recursos consultados para la elaboración del proyecto en la documentación dentro del código fuente (enlaces, libros, etc.)

### Tabla de evaluación

Rúbrica	Valor
Funcionalidad y ejecución del programa	60%
Elementos utilizados / Optimización	25%
Buen uso de los recursos AI	10%
Documentación del código	5%
	100%