****

**Grupo no. 3**

Sustentantes:

**Deybby Rosario 2024-0504**

**Sarah Peña 2024-0506**

**Carlos Roa 2024-0510**

**Héctor Reyes 2024-0526**

Materia:

**Juegos Inteligentes**

Docente:

**Carlos Bienvenido Ogando Montas**

Sección:

**2025-C-3-1908-3242-TIA-603**

Fecha:

**08-10-2025**

Tema:

**FlowFree - Búsqueda desinformada e informada**

**Introducción**

El presente proyecto corresponde a la implementación del juego **Flow Free**, adaptado al entorno de consola en lenguaje **Python**, con el propósito de aplicar y comparar **agentes de búsqueda clásicos** en Inteligencia Artificial.

El juego *Flow Free* consiste en conectar pares de puntos de igual color en una cuadrícula, formando “tuberías” que cubran todo el tablero sin que se crucen entre sí. La complejidad del problema aumenta conforme crece el tamaño de la cuadrícula y el número de colores a conectar.

El desarrollo del sistema incluye tanto el **motor del juego**, jugable por humanos mediante controles de teclado, como tres agentes automatizados que resuelven los tableros empleando diferentes estrategias de búsqueda:

* **DFS (Depth-First Search)** o búsqueda en profundidad.
* **BFS (Breadth-First Search)** o búsqueda en amplitud.
* **A\*** o búsqueda informada con heurísticas múltiples.

Cada algoritmo fue diseñado bajo una estructura modular, extendiendo una clase base común para la recolección de métricas de rendimiento.

El sistema genera reportes automáticos en archivos de salida que contienen las métricas principales de cada ejecución: número de nodos expandidos, costo de la ruta, profundidad alcanzada, tiempo total y uso máximo de memoria RAM.

Si bien no todos los tableros pudieron resolverse completamente, los resultados obtenidos permiten **evaluar el comportamiento y eficiencia de cada técnica de búsqueda** frente a distintos tamaños de tablero y niveles de dificultad, observando cómo el tiempo y la memoria crecen según la complejidad del problema.

**Objetivos**

**Objetivo general**

Desarrollar un sistema de resolución automática para el juego *Flow Free* mediante la implementación y comparación de distintos algoritmos de búsqueda en Inteligencia Artificial, analizando su desempeño en diferentes niveles de dificultad.

**Objetivos específicos**

1. Implementar una versión jugable por consola del juego *Flow Free*, con lectura dinámica de tableros desde archivos .txt.

2. Diseñar e integrar tres agentes de búsqueda (DFS, BFS y A\*) que puedan resolver los niveles de manera autónoma.

3. Incorporar un sistema de medición de métricas de rendimiento para evaluar el comportamiento de los algoritmos.

4. Ejecutar un **benchmark experimental** sobre diferentes tableros (fácil, medio y difícil), registrando resultados de tiempo, memoria, profundidad y nodos expandidos.

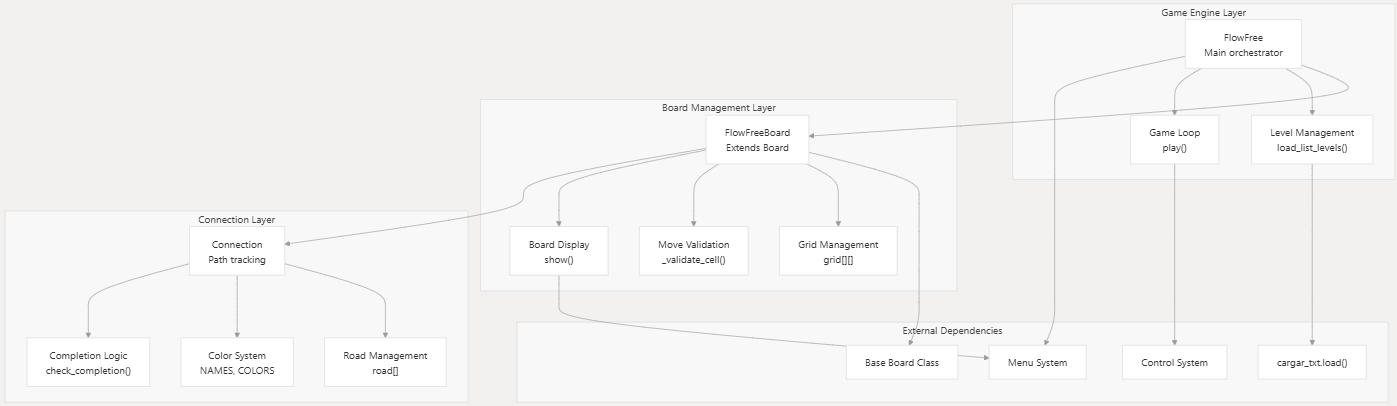
5. Analizar los resultados obtenidos para identificar ventajas, limitaciones y posibles mejoras en las estrategias implementadas.

**Metodología**

**Game Engine**

El motor sigue una **arquitectura en capas**, con separación clara entre:

* **Orquestación del juego** (gestión de menús, niveles y flujo de juego).
* **Gestión del tablero** (estado del tablero, validación de movimientos, visualización).
* **Control de conexiones** (creación, extensión y ruptura de caminos de color).



|  |  |  |  |
| --- | --- | --- | --- |
| Componente | Responsabilidad principal | Métodos clave | Líneas |
| FlowFree | Orquestación del juego, gestión de niveles, bucle de juego | app(), play(), select\_level() | 212-428 |
| FlowFreeBoard | Gestión del estado del tablero, visualización, validación de movimientos | show(), \_validate\_cell(), percentage\_filled() | 89-211 |
| Connection | Seguimiento y validación de caminos individuales | add\_to\_road(), check\_completion(), break\_road() | 10-85 |

**Gestión del Estado del Juego**

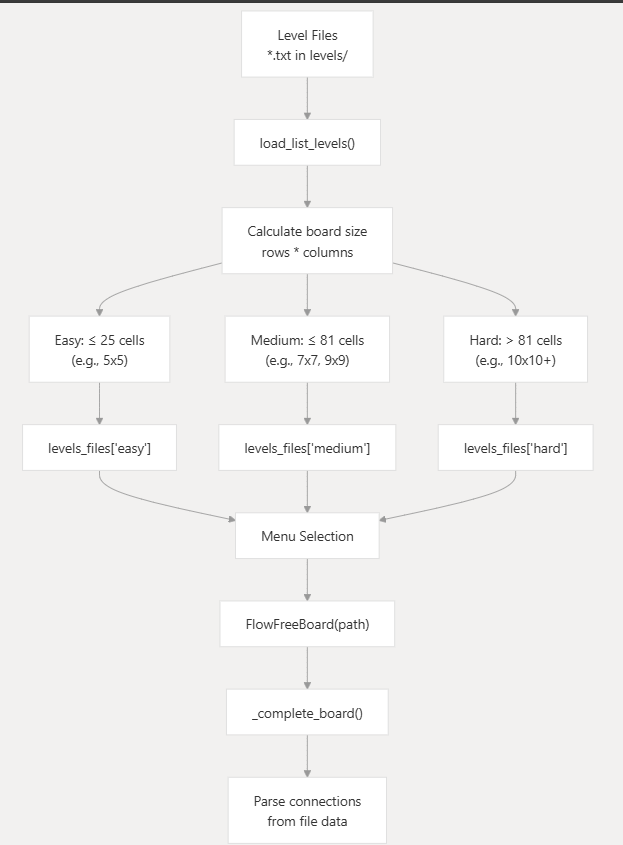
El motor mantiene varias variables clave para coordinar el flujo del juego:

|  |  |
| --- | --- |
| Variable | Función |
| self.board | Instancia activa de FlowFreeBoard. |
| self.last\_color\_position | Rastrea la conexión que se está dibujando actualmente. |
| self.last\_move | Permite validar movimientos y realizar deshacer. |
| self.levels\_files y self.levels\_files\_names | Gestionan los niveles disponibles y sus nombres para la UI. |

**Secuencia del Flujo de Juego**

El flujo típico de ejecución del juego va desde el inicio hasta la finalización de un nivel:

1. **Carga del menú principal** (app()).
2. **Selección de nivel** y carga de datos.
3. **Inicialización del tablero** (FlowFreeBoard).
4. **Bucle de juego** gestionado por play().
5. **Validación de movimientos y actualización del estado** según acciones del jugador.
6. **Finalización de nivel** y retorno al menú principal.



**Carga de Niveles e Inicialización del Tablero**

El motor incluye un sistema de gestión de niveles:

1. **Descubrimiento de archivos**: Escanea el directorio levels/ buscando archivos .txt.
2. **Extracción de metadatos**: Obtiene dimensiones del tablero y número de colores desde el nombre del archivo.
3. **Clasificación por dificultad**: Según la cantidad total de celdas del tablero.
4. **Generación de menú**: Crea nombres de niveles formateados para la interfaz.
5. **Inicialización del tablero**: Carga el nivel seleccionado en una instancia de FlowFreeBoard.

**Validación de Movimientos y Lógica de Juego**

El motor implementa un sistema de validación que maneja distintos escenarios:

|  |  |  |  |
| --- | --- | --- | --- |
| Escenario | Condición | Acción | Referencia de código |
| Celda inválida | !\_validate\_cell(x, y) | Ignorar movimiento | 345-346 |
| Mismo lugar | move == last\_move | Deseleccionar / deshacer | 349-358 |
| Nuevo inicio | !last\_color\_position | Inicia un nuevo camino | 369-383 |
| Conexión completada | Conecta con círculo del mismo color | Completa el camino | 387-399 |
| Intersección de camino | Movimiento hacia celda existente | Rompe el camino en ese punto | 405-409 |
| Extensión de camino | Celda adyacente válida | Extiende el camino actual | 418-421 |

**Resultados y Análisis**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Dificultad | Algoritmo | Movimientos (cost\_of\_path) | Nodos expandidos | Profundidad | Máx. Prof. | Tiempo (s) | Memoria (MB) |
| Fácil (5x5) | A\* | 21.0 | 190.18 | 21.0 | 9.93 | 0.0140 | 0.0631 |
|  | BFS | 21.0 | 87.33 | 21.0 | 9.33 | 0.0103 | 0.0082 |
|  | DFS | 21.0 | 122.25 | 21.0 | 12.25 | 0.0075 | 0.0078 |
| Medio (7x7) | A\* | 46.04 | 2300.71 | 44.2 | 19.0 | 0.0784 | 0.0014 |
|  | BFS | 44.0 | 254.00 | 44.0 | 22.0 | 0.0150 | 0.0000 |
|  | DFS | 43.75 | 425,962.25 | 43.75 | 27.0 | 5.3618 | 0.0000 |