

Informe de Práctica: Implementación del 3-en-Raya en python

Adrián Martínez Pérez
Guillem Arnau Vallejos

Enero 2025

Índice

1. Introducción	2
2. Descripción del Juego	2
3. Diseño e Implementación	2
4. Pruebas y Validación	2
5. Dificultades y Aprendizajes	3
6. Reparto de Responsabilidades	3
7. Conclusión	3

1. Introducción

La presente práctica tiene como objetivo la implementación de un "tablero abstracto" que permita jugar al clásico juego del 3-en-rama, así como a algunas de sus variantes. El desarrollo de este proyecto busca fomentar el trabajo en equipo, la comprensión de conceptos de programación funcional y el diseño de estructuras de datos reutilizables.

El informe detalla el proceso de diseño, implementación, pruebas y validación del sistema, así como las decisiones tomadas durante su desarrollo. Finalmente, se incluye una reflexión sobre las dificultades enfrentadas y los aprendizajes adquiridos.

2. Descripción del Juego

El 3-en-rama es un juego de estrategia en el que dos jugadores colocan sus piezas en un tablero de 3×3 casillas. El objetivo en la variante estándar es alinear tres piezas consecutivas en horizontal, vertical o diagonal.

En esta práctica, se consideran las siguientes variantes del juego:

- **Juego estándar:** El primer jugador en alinear tres piezas gana.
- **Variante "misery":** El primer jugador que alinea tres piezas pierde.

El tablero se ha diseñado para ser flexible, permitiendo ajustar su tamaño y el número de piezas por jugador.

3. Diseño e Implementación

El programa desarrollado consta de dos componentes principales:

- **El tablero abstracto:** Implementado en el archivo `abs.board.py`, contiene las estructuras de datos y funciones necesarias para gestionar el estado del juego.
- **Drivers:** Programas que permiten la interacción con los jugadores. Se han utilizado tanto interfaces basadas en texto como en gráficos (usando Pygame).

El diseño del tablero abstracto se basa en principios de programación funcional, empleando clausuras y variables `nonlocal` para encapsular el estado del juego. Además, se ha asegurado que el código sea modular y fácil de extender.

4. Pruebas y Validación

Para garantizar el correcto funcionamiento del sistema, se realizaron las siguientes pruebas:

- Simulaciones de partidas completas en ambas variantes del juego.
- Validación cruzada con los drivers proporcionados y modificados.

Los resultados obtenidos coinciden con las expectativas y confirman que el sistema implementado cumple con los requisitos de la práctica.

5. Dificultades y Aprendizajes

Durante el desarrollo, enfrentamos las siguientes dificultades:

- Comprensión y aplicación de clausuras en Python.
- Diseño modular que permita cambios en las reglas del juego sin afectar el núcleo del sistema.

Como resultado, hemos adquirido habilidades avanzadas en programación funcional y diseño de software colaborativo.

6. Reparto de Responsabilidades

El trabajo se distribuyó de la siguiente manera:

- Adrián Martínez Pérez: Implementación del tablero abstracto y pruebas con drivers.
- Guillem Arnau Vallejos: Diseño de variantes del juego y desarrollo de interfaces gráficas.

7. Conclusión

El desarrollo de esta práctica ha sido una experiencia enriquecedora que nos ha permitido consolidar conceptos clave de programación y trabajo en equipo. El sistema implementado cumple con los objetivos planteados y ofrece una base sólida para futuras mejoras.