

Fundamentos de Inteligencia Artificial

- Reto - Agente Connect-4 -

1 Objetivo

El objetivo de este proyecto es diseñar, implementar y evaluar un agente inteligente capaz de jugar Connect-4 de manera autónoma, aplicando los conceptos y técnicas vistos en clase. Se busca que los estudiantes demuestren comprensión teórica, capacidad de implementación práctica y pensamiento crítico al analizar los resultados obtenidos. Además, se valorará la creatividad en las decisiones de diseño y la calidad de la validación experimental.

2 Entregables

Los entregables son los siguientes:

1. Cada grupo deberá entregar (un integrante del grupo lo manda por mensaje privado en Teams) un enlace al repositorio en GitHub que contenga
 - a) el código completo, los datos necesarios para la ejecución y una breve guía de uso.
 - b) un Python notebook especial con nombre ‘entrega.ipynb’ en el que se realiza todo el estudio y donde se generan todas las gráficas relevantes para la validación y optimización del agente (según criterios 2 & 3 en la rúbrica) y que posiblemente se muestran durante de la presentación.
2. Cada *integrante* del grupo deberá presentar un documento individual en formato PDF (máximo dos páginas) en MS Teams que resuma los aspectos más relevantes de su contribución personal al proyecto, destacando las partes del trabajo en las que participó, los desafíos enfrentados y los principales logros alcanzados.

Como aporte debe anexar el enlace que contiene el commit cocncreto de esas contribuciones.

Nos interesan solo los aportes *highlight* (máximo 2). Si se presentan más que 2 no se valora ninguna.

También debe contener su reflexión de la solución con propuestas de mejora.

3. una presentación de 10 - 12 minutos (al menos 3 minutos por integrante) demostrando . Puede ser sostenido en español o inglés.

3 Rúbrica de Evaluación

El puntaje de cada integrante depende en algunos aspectos de la solución grupal pero generalmente también de la evidencia (en github y por PDF entregable) y sustentación de sus propias contribuciones. Criterios marcados con I y G en la siguiente rúbrica son influidos por rendimiento del individuo, del grupo, o ambos.

Criterio	20% (Mínimo)	50% (Básico)	80% (Bueno)	100% (Excelente)
1. Implementación correcta y original de los métodos (G + I)	El agente es sintácticamente funcional y puede participar en el torneo, tal vez con errores lógicos o conceptuales; se observan partes de código tomadas de fuentes externas sin adaptación.	El agente funciona de forma general usando métodos no-triviales del curso; se basa en gran medida en código proporcionado o ajeno, con poca aportación propia.	Implementación mayormente propia y técnicamente correcta; sin errores graves; comprensión clara de los conceptos vistos en clase.	Implementación completamente correcta y 100% desarrollada por el equipo (sin uso de IA generativa o código copiado); trabajo original, bien comentado y totalmente comprensible.
2. Validación del agente (G)	Solo existen pruebas anecdóticas o evaluaciones subjetivas del funcionamiento (“parece jugar”); sin datos cuantitativos.	Se presenta, para cada jugador (rojo y amarillo) una curva de aprendizaje visualizada que demuestra que el agente aprende o cambia su rendimiento; resultados documentados de forma básica.	Se realizaron varios experimentos o curvas de aprendizaje para dar una idea más profunda de la estabilidad del rendimiento; resultados comparados de manera visual o estadística (por ejemplo, mostrando la media y desviación estándar en la curva).	Existen visualizaciones adicionales (otras que curvas de aprendizaje como avg reward en tiempo) que demuestren un buen funcionamiento de (i) o componentes internos y/o (ii) comportamiento del agente. (Sorpréndanos!)
3. Optimización del agente (G + I)	Se modifican algunos parámetros sin una justificación clara.	Se comparan al menos dos configuraciones o métodos; la elección final está razonablemente justificada.	Optimización sistemática: se usan los mismos elementos como en la validación del agente pero para al menos dos implementaciones/versiones <i>diferentes</i> de la cual la mejor se usó en la versión final	Identificación de una métrica propia (otra que rendimiento final) que usa mejorar el rendimiento formal y incentiva un cambio puntual en la implementación y que se realizó y comprobó (independientemente del resultado).
4. Reflexión y análisis (limitaciones y mejoras) (I)	Los resultados solo se describen; sin análisis ni identificación de debilidades.	Se mencionan limitaciones acertadas y se explican sus raíces, pero las propuestas de mejora son vagas o poco realistas.	Se identifican claramente los cuellos de botella y limitaciones; se proponen mejoras coherentes y técnicamente razonables; se entiende la relación causa–efecto.	Ánalisis profundo y estratégico: limitaciones bien definidas, causas explicadas analíticamente y mejoras concretas con alto impacto potencial (por ejemplo, cambios de arquitectura o representación de datos); priorización y argumentación claras.
5. Presentación y comunicación (G + I)	Presentación básica pero poco estructurada o difícil de seguir.	Presentación comprensible, pero poco visual o con participación desigual del grupo.	Estructura clara, visualizaciones ilustrativas y explicación comprensible de las ideas principales.	Presentación profesional y bien coordinada; comunicación fluida, visualmente atractiva y convincente incluso para un público no especializado.

Además, después de una semana de trabajo se exige una versión inicial del agente que ya es capaz de siempre ganar un jugador aleatorio. Sin una evidencia de un agente así (con demostración empírica) el grupo sale del proyecto.

4 Cronograma

- 12.11.: Lanzamiento del proyecto, apertura de gradescope para propósitos de testeo
- 16.11 - 19.11.: Evidencia de progreso para permanecer en el reto
- 23.11.: Entrega del agente final
- 24.11.: Ejecución del torneo
- 25.11./26.11.: Presentación/Sustentación y evaluación de las soluciones

Cada grupo tiene que consistir de 2 o 3 integrantes. No se permiten “grupos” de una persona ni con más de 3 estudiantes.