DCIE | Divulgación Científica de Investigación y Emprendimiento
ISSN: 3028-9157 (En línea)
ISSN: 3028-9157
http://www.dcie.org.pe/index.php/dcie/index

Gamificación en IA

Juegos como motores del aprendizaje activo

C. Astengo Noguez, L. Martinez Elizalde
Tecnológico de Monterrey (MÉXICO)

Resumen

Este artículo documenta un rediseño del curso *Agentes Inteligentes*, impartido a estudiantes de Ciencia de Datos y Matemáticas de 5 semanas en Febrero-Marzo de 2025. La innovación central consistió en convertir los juegos en el eje estructural del curso: desde dinámicas de cartas y plataformas digitales como *Kahoot*, *Quizlet* y *Blooket*, hasta la creación de testbeds interactivos en Unity. Los estudiantes no solo participaron en estas actividades, sino que también diseñaron sus propios juegos para reforzar conceptos clave de IA. Esta experiencia evidenció que la gamificación, cuando es participativa y significativa, puede fomentar pensamiento algorítmico, comprensión profunda y habilidades de diseño pedagógico.

Introducción

La enseñanza tradicional de la inteligencia artificial suele apoyarse en enfoques abstractos y teóricos. En contraste, el rediseño del curso *Agentes Inteligentes* se propuso convertir el aula en un entorno lúdico, donde cada actividad se transformara en una experiencia de juego significativa. Lejos de usar los juegos como simples complementos, estos fueron el vehículo principal para construir conocimiento.

Se partió de problemas clásicos de IA, como el *Mars Rover*, pero recontextualizados en un entorno gamificado. Cada etapa del curso incorporó una o más dinámicas de juego: desde resolver desafíos inspirados en juegos de mesa y videojuegos, hasta competencias interactivas con herramientas digitales. Más aún, los estudiantes diseñaron actividades propias donde se integraron heurísticas, búsquedas, arquitecturas de agentes y restricciones.

Juegos aplicados en clase

1. Juegos de cartas

Se incorporó el juego de cartas comercial *Potato Pirates* como herramienta didáctica para enseñar fundamentos de programación y pensamiento algorítmico. En este juego, cada estudiante asume el rol de un capitán pirata con una flota de barcos y una tripulación de papas. Las cartas representan instrucciones de programación: las cartas de acción ejecutan ataques, mientras que las cartas de control introducen estructuras como bucles (*for*, *while*) y condicionales (*if-else*). Los jugadores deben programar secuencias de ataque colocando hasta tres cartas por barco, que se activan en turnos posteriores. Esta mecánica

ISSN: 3028-9157
http://www.dcie.org.pe/index.php/dcie/index

simula la construcción de funciones y la lógica de control de flujo en programación. Además, las cartas sorpresa permiten interrupciones en cualquier momento, emulando eventos asincrónicos y manejo de excepciones. La dinámica del juego promueve la comprensión de conceptos como modularidad, reutilización de código y depuración, al tiempo que fomenta habilidades de pensamiento lógico y resolución de problemas en un entorno lúdico y colaborativo

2. Juegos clásicos como entornos de resolución de problemas en IA

Una de las estrategias más efectivas del curso fue el uso de problemas clásicos convertidos en juegos para ilustrar cómo los agentes inteligentes abordan situaciones con espacio de búsqueda definido, objetivos claros y restricciones operativas. A continuación se detallan cuatro de estos desafíos:

1. Gato (Tic-Tac-Toe)

Planteamiento del problema:
Juego por turnos entre dos jugadores (X y O) en un tablero 3x3. El objetivo es alinear tres símbolos iguales (horizontal, vertical o diagonal).

Enfoque de IA:
El juego fue abordado desde la perspectiva de un **agente basado en objetivos** que selecciona movimientos para maximizar la probabilidad de ganar o empatar.

Espacio de búsqueda:
Cada nodo representa una configuración del tablero. El espacio completo tiene hasta 9! ≈ 362,880 configuraciones posibles, aunque muchas son inválidas o simétricas.

Este juego permitió analizarlo no solo como una dinámica lúdica, sino como un problema clásico de inteligencia artificial formulado como búsqueda en árboles. A pesar de que el espacio de búsqueda es grande pero finito, se aprovechó la estructura del juego para introducir técnicas de reducción del espacio explorado, como la identificación de jugadas simétricas mediante rotaciones y reflexiones del tablero. Esta simplificación permitió modelar estrategias eficientes de resolución y, en particular, fue útil para introducir el algoritmo *minimax*, mostrando cómo un agente racional puede evaluar y seleccionar la mejor jugada posible considerando las respuestas óptimas del oponente.

DCIE | Divulgación Científica de Investigación y Emprendimiento
ISSN: 3028-9157 (En línea)
ISSN: 3028-9157
http://www.dcie.org.pe/index.php/dcie/index

Pseudocódigo (Minimax simplificado):

```
def minimax(tablero, jugador):
    if es_estado_terminal(tablero):
        return utilidad(tablero)

    if jugador == "X":
        mejor_valor = float("-inf")
        for movimiento in movimientos_posibles(tablero):
            valor = minimax(resultado(tablero, movimiento, "O"))
            mejor_valor = max(mejor_valor, valor)
        return mejor_valor
    else:
        mejor_valor = float("inf")
        for movimiento in movimientos_posibles(tablero):
            valor = minimax(resultado(tablero, movimiento, "X"))
            mejor_valor = min(mejor_valor, valor)
        return mejor_valor
```

2. Misioneros y Caníbales

Planteamiento del problema:
Tres misioneros y tres caníbales deben cruzar un río utilizando una barca con capacidad para dos personas. En ningún momento, en ninguna orilla, los caníbales pueden ser mayoría.

Enfoque de IA:
Se usó un **agente de búsqueda ciega** (búsqueda en anchura o profundidad), adecuado para espacios con restricciones lógicas estrictas.

Espacio de búsqueda:
Cada estado se representa como una tupla:
(M_izq, C_izq, B_izq) → número de misioneros, caníbales y barca en la orilla izquierda.
El espacio completo es pequeño (~32 estados válidos), pero con muchas transiciones inválidas.

Pseudocódigo (búsqueda por anchura):

ISSN: 3028-9157
http://www.dcie.org.pe/index.php/dcie/index

```python
def busqueda_en_anchura(estado_inicial):
    frontera = cola([estado_inicial])
    visitados = set()

    while not frontera.esta_vacia():
        estado = frontera.desencolar()

        if es_estado_meta(estado):
            return reconstruir_camino(estado)

        for sucesor in generar_sucesores(estado):
            if sucesor not in visitados:
                visitados.add(sucesor)
                frontera.encolar(sucesor)
```

3. El caballo en el tablero de ajedrez (Knight's Tour)

Planteamiento del problema:
Mover un caballo en un tablero de ajedrez (8x8) de modo que pase exactamente una vez por cada casilla.

Enfoque de IA:
Este problema fue abordado con un **agente de búsqueda heurística**, usando una variante de búsqueda con *backtracking* y la heurística de Warnsdorff (priorizar casillas con menos movimientos posibles).

Espacio de búsqueda:
64! posibilidades, pero enormemente reducido al aplicar restricciones y heurísticas.

Pseudocódigo (búsqueda con backtracking + heurística):

```python
def recorrido_caballo(x, y, paso, tablero):
    if paso == 64:
        return True

    for movimiento in ordenar_movimientos_posibles(x, y, tablero):
        nx, ny = movimiento

        if es_valido(nx, ny, tablero):
            tablero[nx][ny] = paso

            if recorrido_caballo(nx, ny, paso + 1, tablero):
                return True

            tablero[nx][ny] = -1

    return False
```

DCIE | Divulgación Científica de Investigación y Emprendimiento
ISSN: 3028-9157 (En línea)
ISSN: 3028-9157
http://www.dcie.org.pe/index.php/dcie/index

4. Las ocho reinas

Planteamiento del problema:
Colocar 8 reinas en un tablero de ajedrez de 8x8 sin que ninguna se ataque (no compartir fila, columna ni diagonal).

Enfoque de IA:
El problema se resolvió con un **agente que aplica satisfacción de restricciones** (CSP), usando backtracking con forward-checking.

Espacio de búsqueda:
Cada reina debe ocupar una fila distinta; así, el espacio se reduce a permutaciones de columnas (8! = 40320), pero muchas combinaciones violan las restricciones.

Pseudocódigo (CSP con forward-checking):

```
def resolver_8_reinas(fila, tablero):
    if fila == 8:
        return True

    for columna in range(8):
        if no_hay_conflicto(tablero, fila, columna):
            tablero[fila] = columna

            if resolver_8_reinas(fila + 1, tablero):
                return True

            tablero[fila] = -1

    return False
```

Estos juegos clásicos permitieron a los estudiantes entender los conceptos fundamentales de agentes inteligentes no solo desde la teoría, sino desde la acción y la experimentación. Cada juego proporcionó un entorno controlado donde se discutió qué tipo de agente era adecuado, cómo se definía el espacio de estados y qué algoritmos podían resolverlos de forma óptima o razonable. Esta combinación entre lo lúdico y lo técnico sirvió como puente entre el razonamiento humano y el computacional.

3- Plataformas digitales como espacios de juego estructurado

Una innovación clave del curso fue retar a los alumnos a diseñar sus propias dinámicas gamificadas para reforzar los conocimientos básicos y habilidades de Inteligencia Artificial. Estas actividades utilizaron plataformas comerciales como:

DCIE | Divulgación Científica de Investigación y Emprendimiento
ISSN: 3028-9157 (En línea)
ISSN: 3028-9157
http://www.dcie.org.pe/index.php/dcie/index

Kahoot!

Se utilizó para activar conocimientos previos, realizar quizzes rápidos y fomentar la participación sincrónica. Su dinámica competitiva, inmediata y visual hizo que incluso los repasos de teoría fueran momentos de alta energía y motivación.

Sitio: https://kahoot.com

Quizlet

Con esta herramienta se crearon tarjetas de estudio para conceptos técnicos como *tipos de agentes*, *heurísticas*, *ciclo percepción-acción* y *tipos de entorno*. Las modalidades de prueba, emparejamiento y competencia permitieron revisar contenidos de forma autónoma o grupal.

Sitio: https://quizlet.com

Blooket

Fue la plataforma preferida para repasar temas más complejos gracias a su capacidad para transformar un simple cuestionario en una experiencia envolvente (por ejemplo, modo *Battle Royale*, *Gold Quest*, o *Factory*). Se usó para reforzar búsquedas, restricciones, grafos, árboles y tendencias actuales en IA.

Sitio: https://www.blooket.com

Este enfoque promovió empatía pedagógica y una apropiación crítica del conocimiento, los estudiantes no solo comprendieron los conceptos, sino que reflexionaron sobre cómo enseñarlos de forma significativa.

Simulaciones interactivas Gamificadas

Paralelamente, los estudiantes desarrollaron un testbed en Unity con agentes en Python que se comunicaban mediante una API REST. Aunque técnica, esta parte se estructuró también como un juego: cada escenario planteaba diferentes retos desde un entorno plano hasta laberintos complejos con obstáculos.

Los equipos de estudiantes programaron la lógica de distintos tipos de agentes inteligentes con el propósito de comparar sus desempeños en diversos entornos o escenarios simulados. Cada equipo diseñó experimentos específicos *agente–ambiente*, ejecutándolos múltiples veces para generar datos observables y así identificar cuál arquitectura de agente resultaba más eficiente según el tipo de entorno. Como requisito, todos los equipos debían implementar al menos dos agentes:

ISSN: 3028-9157
http://www.dcie.org.pe/index.php/dcie/index

a) un agente reactivo simple y
b) un agente deliberativo con búsqueda informada mediante el algoritmo A*.

Adicionalmente, se ofreció la opción de incluir un agente con capacidad de aprendizaje. En clase se revisó el algoritmo *Q-learning*, y varios equipos optaron por explorar este enfoque, mientras que otros implementaron redes neuronales como base de sus agentes. Esta dinámica promovió la experimentación comparativa y el análisis empírico, conectando teoría, implementación y toma de decisiones basadas en evidencia.

Resultados

El desempeño académico del grupo, conformado por 28 estudiantes, fue notable. La calificación final promedio del curso fue de 96.4, con una mediana de 97.5 y un rango que osciló entre 82 y 100, lo que refleja una consistencia sobresaliente en el aprendizaje. En el proyecto final sobre agentes inteligentes, la media fue de 31 puntos sobre un total de 34.5, con resultados que variaron entre 22.2 y 34.5. Cabe destacar que 15 estudiantes decidieron participar en el trabajo opcional, una actividad voluntaria de diseño y evaluación de agentes de aprendizaje, y todos obtuvieron una calificación perfecta de 100/100. Este nivel de participación voluntaria y desempeño excelente evidencia un alto grado de motivación intrínseca y dominio técnico en los estudiantes más comprometidos.

Los datos sugieren que el enfoque gamificado del curso no solo facilitó el aprendizaje, sino que lo potenció significativamente: los estudiantes no solo comprendieron los conceptos, sino que los aplicaron en contextos complejos, diseñaron sus propias soluciones y demostraron autonomía y entusiasmo en su proceso de formación.

Reflexión: jugar para aprender, aprender para diseñar

En este curso, la gamificación no fue un simple complemento metodológico, sino el núcleo estructural de la experiencia de aprendizaje. Al integrar juegos analógicos, simulaciones, plataformas digitales y diseño de actividades por parte de los propios estudiantes, se transformó la dinámica del aula en un laboratorio de creación, prueba e iteración.

Desde la perspectiva del modelo SAMR de Puentedura (), se alcanzó el nivel más alto: redefinición. Las tareas tradicionales fueron reemplazadas por desafíos diseñados por los propios alumnos, donde el conocimiento no se memorizaba, sino que se construía colectivamente a través del juego, la competencia saludable y la colaboración activa.

Este enfoque generó un clima de aula seguro y estimulante, donde el error se valoró como parte natural del proceso creativo y no como un indicador de fracaso. En este entorno, la

ISSN: 3028-9157
http://www.dcie.org.pe/index.php/dcie/index

motivación extrínseca (puntajes, reconocimiento, juego) se entrelazó con una motivación intrínseca profunda: el deseo de resolver, de crear, de comprender, de superar obstáculos no impuestos sino autoimpuestos.

El resultado fue una experiencia educativa donde muchos estudiantes superaron incluso las expectativas del curso, desarrollando soluciones complejas, explorando técnicas avanzadas como el *Q-learning* o redes neuronales, y demostrando que un entorno gamificado bien diseñado puede despertar un nivel de compromiso y desempeño académico excepcional.

Conclusiones

La experiencia mostró que la gamificación, cuando es participativa y deliberada, puede potenciar el aprendizaje en cursos técnicos como IA. Lejos de ser un recurso superficial, los juegos se convirtieron en estructuras profundas de comprensión, evaluación y creación.

Diseñar juegos para otros implicó que los estudiantes reorganizaran su conocimiento, identificaran lo esencial y comunicaran ideas complejas de manera accesible. En este sentido, la gamificación fue un catalizador de habilidades transversales: comunicación, empatía, creatividad, pensamiento crítico y trabajo colaborativo.