

Proyecto 2 – Vivir o Morir

Expectimax y Minimax

Estudiante: Jhonatan Alejandro Solano Mendoza

Asignatura: Introducción a la Inteligencia Artificial

Profesor: Arles Ernesto Rodríguez Portela

13 de noviembre de 2025

Resumen

Este proyecto presenta el desarrollo del juego “Vivir o Morir”, una simulación de supervivencia entre un sobreviviente y una horda creciente de zombies, implementada en Python. El objetivo es analizar el desempeño de los algoritmos Expectimax y Minimax en un entorno dinámico y adversarial. El sobreviviente utiliza Expectimax para maximizar su expectativa de supervivencia ante movimientos inciertos, mientras que los zombies aplican una estrategia Minimax local para acorralarlo. Se modelaron funciones evaluadoras basadas en la distancia Manhattan y se definió una meta de supervivencia de diez turnos. Los resultados muestran que Expectimax permite al sobreviviente mantener ventaja con pocos enemigos, pero pierde eficacia a medida que la cantidad de zombies crece, demostrando los límites de decisión racional frente a la complejidad exponencial del entorno.

Índice

1. Introducción	3
2. Planteamiento del Problema	3
3. Objetivos del Proyecto	3
3.1. Justificación	4
3.2. Desarrollo Teórico y Metodológico	4

3.2.1. Estructura del Proyecto	4
3.2.2. Teoría de Juegos y Toma de Decisiones	5
3.3. Comparación de estrategias empleadas	5
3.3.1. Función Evaluadora del Sobreviviente	6
3.3.2. Función Objetivo y de Pérdida	6
3.3.3. Dinámica del Juego	7
3.4. Análisis de Resultados	7
3.5. Conclusiones	8
4. Repositorio	8
5. Referencias	8
6. Anexos	9
6.1. Instrucciones para ejecutar de forma local	9

1. Introducción

El presente proyecto tiene como propósito el desarrollo de un videojuego de simulación denominado “**Vivir o Morir**”, diseñado e implementado en el lenguaje de programación Python. El juego recrea una situación de supervivencia entre un único sobreviviente y una horda de zombies que crece exponencialmente en cada ronda. El sobreviviente utiliza un algoritmo de decisión inteligente basado en *Expectimax*, mientras que los zombies emplean un modelo *Minimax* simplificado de persecución local.

El objetivo principal consiste en que el sobreviviente logre sobrevivir durante diez turnos consecutivos, tomando decisiones óptimas frente a la amenaza, mientras los zombies intentan capturarlo. Este proyecto integra conceptos de inteligencia artificial, funciones evaluadoras, teoría de juegos y optimización bajo incertidumbre, lo que permite comprender cómo los agentes racionales pueden aprender a sobrevivir en entornos dinámicos y hostiles.

2. Planteamiento del Problema

La problemática a resolver consiste en crear un entorno controlado donde un agente inteligente (el sobreviviente) deba tomar decisiones óptimas ante un entorno adverso que evoluciona de forma impredecible. En este caso, los zombies representan agentes hostiles que actúan de manera competitiva, y el sobreviviente debe anticipar sus movimientos para maximizar su probabilidad de supervivencia.

La dificultad radica en que el número de zombies aumenta exponencialmente (2, 4, 8, 16, 32...), haciendo que la complejidad del entorno crezca rápidamente. Esto plantea un desafío propio de los problemas de inteligencia artificial con múltiples agentes, donde el espacio de búsqueda y las decisiones óptimas se vuelven costosas de calcular.

3. Objetivos del Proyecto

Objetivo General:

- Desarrollar una simulación en Python que represente el comportamiento de un sobreviviente controlado mediante el algoritmo *Expectimax* frente a una horda de

zombies controlados mediante *Minimax*, analizando su desempeño en términos de supervivencia y toma de decisiones óptimas.

Objetivos Específicos:

1. Implementar una estructura modular en Python que integre la configuración del entorno, las estrategias de los agentes, la simulación y la visualización.
2. Diseñar e implementar funciones evaluadoras, de pérdida y heurísticas que determinen el valor de cada estado del juego.
3. Analizar matemáticamente las funciones utilizadas por cada agente para la toma de decisiones.
4. Ejecutar experimentos con diferentes cantidades de zombies para evaluar el rendimiento del sobreviviente.
5. Presentar un análisis de resultados, conclusiones y reflexiones sobre el comportamiento emergente de los agentes.

3.1. Justificación

La inteligencia artificial moderna no se limita a la predicción o clasificación, sino que busca modelar comportamientos adaptativos en entornos dinámicos. Este proyecto demuestra cómo, a través de estructuras como *Expectimax* y *Minimax*, un agente puede aprender a sobrevivir en condiciones de incertidumbre y hostilidad. Además, fomenta la comprensión de conceptos como racionalidad, evaluación de estados, estrategias adversariales y búsqueda heurística, aplicados a un contexto lúdico pero académico.

3.2. Desarrollo Teórico y Metodológico

3.2.1. Estructura del Proyecto

El proyecto está compuesto por cinco módulos principales:

1. **config.py**: Define los parámetros del entorno (dimensiones, direcciones, condiciones de victoria).
2. **strategies.py**: Contiene la lógica de decisión del sobreviviente y los zombies.

3. **simulation.py**: Controla el flujo del juego, turnos y condiciones de victoria o derrota.
4. **visualization.py**: Muestra gráficamente los movimientos de los agentes en tiempo real.
5. **main.py**: Coordina las rondas y resume los resultados finales.

3.2.2. Teoría de Juegos y Toma de Decisiones

El modelo implementado se fundamenta en la teoría de juegos de suma cero, donde un jugador intenta maximizar su ganancia mientras el otro intenta minimizarla. En este caso:

- El sobreviviente busca **maximizar su esperanza de supervivencia**.
- Los zombies buscan **minimizar la distancia** al sobreviviente, intentando capturarlo.

Expectimax El algoritmo *Expectimax* se utiliza para modelar el comportamiento del sobreviviente. Este asume que los movimientos de los zombies son parcialmente aleatorios y toma decisiones maximizando el valor esperado de la función de utilidad:

$$V(s) = \begin{cases} \text{Eval}(s), & \text{si } s \text{ es terminal} \\ \max_{a \in A(s)} \mathbb{E}_{s'}[V(s')], & \text{si es turno del sobreviviente} \end{cases}$$

donde $A(s)$ son las acciones posibles y $\mathbb{E}_{s'}[V(s')]$ representa el valor esperado de los futuros estados posibles.

Minimax Local Los zombies se modelan mediante una estrategia adversarial local basada en *Minimax*. Aunque no calculan todas las posibles trayectorias, se mueven siempre hacia la posición que minimiza su distancia Manhattan al sobreviviente:

$$D(a, b) = |x_a - x_b| + |y_a - y_b|$$

3.3. Comparación de estrategias empleadas

En el desarrollo del juego se aplicaron dos enfoques de toma de decisiones opuestos: El algoritmo Expectimax, utilizado por el sobreviviente, y el algoritmo Minimax local,

empleado por los zombies.

La siguiente tabla resume las principales diferencias conceptuales entre ambos agentes:

Cuadro 1: Comparación entre las estrategias de decisión de los agentes.

Agente	Estrategia	Tipo de decisión	Racionalidad
Sobreviviente	Expectimax	Basada en expectativas (mundo estocástico)	No asume que el enemigo es perfecto
Zombies	Minimax local	Basada en minimización (mundo determinista)	Persigue y rodea al sobreviviente

3.3.1. Función Evaluadora del Sobreviviente

La función de evaluación se define como:

$$f(s) = 2,5 \cdot \bar{d} + p$$

donde \bar{d} es la distancia promedio a los zombies y p es una penalización basada en la distancia mínima:

$$p = \begin{cases} -100, & \text{si un zombie está en la misma celda} \\ -40, & \text{si está a distancia 1} \\ -10, & \text{si está a distancia 2} \\ 0, & \text{si está más lejos} \end{cases}$$

Esta función evalúa la calidad de una posición: mientras mayor sea la distancia promedio y menor la cercanía extrema, mejor será el estado para el sobreviviente.

3.3.2. Función Objetivo y de Pérdida

La **función objetivo** del sobreviviente es maximizar su esperanza de vida durante 10 turnos:

$$\max \mathbb{E}[f(s_t)], \quad t \in [1, 10]$$

Mientras que la **función de pérdida** se define como:

$$L(s) = -f(s)$$

cuando el sobreviviente es capturado o se encuentra en un estado de peligro extremo.

3.3.3. Dinámica del Juego

El juego se desarrolla en un tablero de 15x10. El sobreviviente inicia en la esquina inferior izquierda y los zombies aparecen en posiciones aleatorias. Cada turno se realiza la siguiente secuencia:

1. El sobreviviente evalúa todas sus posibles acciones (moverse o quedarse).
2. Aplica *Expectimax* con profundidad limitada para estimar la mejor acción.
3. Los zombies se mueven localmente para reducir su distancia al sobreviviente.
4. Si algún zombie alcanza al sobreviviente, la simulación termina.

El sobreviviente gana si logra sobrevivir los 10 turnos sin ser alcanzado.

3.4. Análisis de Resultados

Durante las pruebas se realizaron rondas con diferentes cantidades de zombies: 2, 4, 8, 16 y 32. En cada nivel se jugaron dos rondas, y se observaron los siguientes patrones:

- Con 2 y 4 zombies, el sobreviviente logró sobrevivir consistentemente, mostrando desplazamientos estratégicos hacia zonas más abiertas.
- Con 8 zombies, el comportamiento fue más variable; en ocasiones fue rodeado debido a su profundidad de búsqueda limitada.
- A partir de 16 zombies, las victorias se redujeron drásticamente, ya que la expansión de la horda reduce significativamente el espacio de maniobra.

La visualización permitió observar cómo el sobreviviente prioriza movimientos que maximizan su distancia promedio frente a los zombies, evidenciando un comportamiento inteligente emergente a partir de simples reglas heurísticas.

3.5. Conclusiones

Este proyecto permitió comprender de forma práctica los principios de la inteligencia artificial aplicada a la toma de decisiones bajo incertidumbre y entornos adversariales. Entre las conclusiones principales destaco:

1. El algoritmo *Expectimax* permitió al sobreviviente mostrar un comportamiento racional y adaptable.
2. La heurística implementada fue efectiva para entornos con pocos zombies, pero su rendimiento decrece con la complejidad del escenario.
3. La simulación demuestra que la profundidad de búsqueda y la aleatoriedad influyen significativamente en los resultados.
4. El enfoque modular permitió desarrollar un entorno fácilmente escalable para futuros experimentos o visualizaciones más avanzadas.

4. Repositorio

<https://github.com/JhonatanSolano/ProyectoZombie>

5. Referencias

- Russell, S. y Norvig, P. (2010). *Artificial Intelligence: A Modern Approach* (3rd ed.). Prentice Hall.
- Sutton, R. S. y Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.
- Millington, I. y Funge, J. (2016). *Artificial Intelligence for Games* (3rd ed.). CRC Press.
- Python Software Foundation. (2024). *Python Language Reference, version 3.12*. Recuperado de: <https://docs.python.org/3/>
- Matplotlib Developers. (2024). *Matplotlib Documentation*. Recuperado de: <https://matplotlib.org/stable/contents.html>

- NumPy Developers. (2024). *NumPy Reference Guide*. Recuperado de: <https://numpy.org/doc/stable/>
- OpenAI. (2023). *Expectimax and Minimax Decision-Making Algorithms*. Recuperado de: <https://openai.com/research>
- Apuntes del curso de Introducción a la Inteligencia Artificial. Universidad Nacional Abierta y a Distancia (UNAD), 2025.

6. Anexos

6.1. Instrucciones para ejecutar de forma local

1. Instalar dependencias:

```
pip install -r requirements.txt
```

2. Ejecutar:

```
python main.py
```