Escuela de Ingeniería en Computación

Curso: IC-3002 Análisis de Algoritmos

Profesor: Ing. Joss Pecou Johnson

TEC | Tecnológico de Costa Rica

Proyecto #2: Blackjack

Descripción General:

Este proyecto consiste en el desarrollo de un juego de **Blackjack** en el que los jugadores podrán enfrentarse entre sí o contra una inteligencia artificial (IA). La IA utilizará un enfoque basado en algoritmos probabilísticos para tomar decisiones sobre las jugadas (hit o stand), además de mejorar sus estrategias mediante aprendizaje reforzado (Q-learning). El juego permitirá agregar múltiples jugadores o jugar únicamente contra la IA.

El proyecto tiene como objetivo no solo simular el juego de **Blackjack**, sino también explorar cómo un agente IA puede aprender a tomar decisiones óptimas en un entorno estocástico, utilizando probabilidades y aprendizaje automatizado. Además, se integrarán funciones avanzadas, como el conteo de cartas, para que la IA tome decisiones más informadas y el programa debe ser desarrollado utilizando **Python** y **Flask** para la interfaz gráfica.

Requisitos del Juego:

1. **Reglas Básicas del Juego:**
    - El juego debe seguir las reglas estándar de Blackjack: el objetivo es que los jugadores (humanos o IA) se acerquen lo máximo posible a 21 puntos sin exceder ese número.
    - Cada jugador puede tomar las decisiones de **hit** (pedir una carta) o **stand** (quedarse con su mano actual).
    - La banca sigue reglas predeterminadas para sus decisiones, como pedir cartas hasta alcanzar al menos 17 puntos.
    - Los jugadores pueden ganar, perder o empatar según las reglas del juego.
2. **Implementación del Algoritmo Probabilístico (Monte Carlo):**
    - La IA debe calcular la probabilidad de que pedir una carta adicional le permita ganar o, al menos, no superar los 21 puntos.
    - Las probabilidades se basarán en la baraja restante y las cartas visibles en el juego.
    - El cálculo probabilístico debe influir en las decisiones de la IA en cada ronda.

3. **Implementación del Algoritmo de Inteligencia Artificial (Q-Learning):**
    - La IA debe entrenarse utilizando aprendizaje reforzado para mejorar sus decisiones a lo largo de múltiples rondas del juego.
    - Se deben definir estados (combinaciones de cartas visibles y puntaje actual), acciones posibles (hit o stand), y recompensas asociadas a ganar, perder o empatar.
    - A lo largo del entrenamiento, la IA debe optimizar sus jugadas para maximizar su tasa de victorias.
    - Se debe implementar un sistema que permita visualizar el progreso de la IA durante el entrenamiento. (por ejemplo, porcentaje de victorias a lo largo de las múltiples rondas, decisiones tomadas con acierto y decisiones erróneas, jugadas realizadas por conteo, probabilidad calculada o al azar, decisiones tomadas durante la ronda anterior y número de partidas jugadas hasta el momento)
4. **Conteo de Cartas para la IA:**
    - La IA debe ser capaz de utilizar una estrategia de conteo de cartas que le permita realizar un seguimiento de las cartas que han sido repartidas.
    - Esta información debe ser integrada en su cálculo probabilístico, permitiendo a la IA mejorar sus predicciones sobre las cartas restantes en la baraja.
    - El sistema de conteo debe tener impacto en las decisiones de la IA y su tasa de éxito en el juego.
5. **Jugadores:**
    - En el juego debe haber al menos dos instancias de la IA participando en la partida, y el jugador humano puede retirarse antes de cada partida.
6. **Interfaz de Usuario:**
    - Se debe proporcionar una interfaz gráfica sencilla que permita a los jugadores humanos interactuar con el juego.
    - La interfaz debe mostrar las cartas del jugador, las cartas visibles de la banca, y las opciones de juego (hit o stand).
    - Al finalizar cada partida, los jugadores pueden acceder a un resumen con estadísticas clave: porcentaje de manos ganadas por la IA, porcentaje de decisiones probabilísticas acertadas tanto por la IA como para los jugadores, evolución de la estrategia de la IA a lo largo de las rondas.
    - Debe mostrarse el porcentaje de la probabilidad para que el usuario pueda ganar si solicita una carta o no.

7. **Opcionales:**
    - Se puede implementar un sistema de **niveles de dificultad** para la IA. A niveles más bajos, la IA podría tomar decisiones más aleatorias o no utilizar el conteo de cartas. En los niveles más altos, podría optimizar al máximo sus jugadas usando tanto el algoritmo probabilístico como el aprendizaje reforzado con conteo de cartas.
    - Implementar la posibilidad de desafiar a los jugadores mediante fichas, esto permite tener un monto inicial utilizando fichas para cada jugador incluyendo la IA, con la opción de seleccionar cuanto se quiere apostar, doblar apuestas o all in. Esto añadiría más complejidad y opciones estratégicas tanto para los jugadores humanos como para la IA.
8. **Documentación Externa:**
    - La documentación externa debe incluir las instrucciones de ejecución y uso del programa.
    - Debe de incluir una descripción sobre las estrategias de la IA, incluyendo cómo el algoritmo probabilístico y el conteo de cartas afectan sus decisiones

Entrega:

La entrega del proyecto debe incluir:

- Código fuente completo.
- Documentación interna y externa.
- Programa funcional

Formato de Entrega:

El proyecto debe de entregarse por medio de la documentación externa con el nombre "Nombre1_Nombre2-Proyecto2", que contenga las instrucciones para ejecutar el proyecto y el enlace del repositorio de Github para descargar y ejecutarlo.

Utilizar la estrategia de ramificación para el repositorio y manejo de cambios dentro del programa. Evitar utilizar un solo commit en la rama ***main***.

Fecha de Entrega:

El proyecto debe entregarse el día 19 de noviembre del 2024 con hora límite de las 11:55 p.m., cualquier proyecto entregado después de la hora establecida será penalizada con 2^n por ciento, donde n representa n la cantidad de días de retraso por la entrega.

Aspectos para evaluar:

Rubro	Puntos
Jugabilidad	**40**
Algoritmo probabilístico	**15**
Implementación de IA	**15**
Estadísticas	**5**
Probabilidad de ganar	**5**
Implementación de Conteo	**10**
Documentación interna	**5**
Documentación externa	**5**
Dificultad(opcional)	**10**
Apuestas(opcional)	**10**
Total	**120**

Liberías recomendadas:

1. **numpy** y **random** para la simulación del juego y el cálculo de probabilidades.
2. **gym** o una implementación manual para la creación de entornos de aprendizaje reforzado.
3. **matplotlib** para la visualización de los resultados y el progreso del modelo.