

Centro Académico de Limón

Escuela de Ingeniería en Computación

Análisis de Algoritmos
IC - 3002

PY02 - Blackjack

Profesor: Joss Pecou Johnson

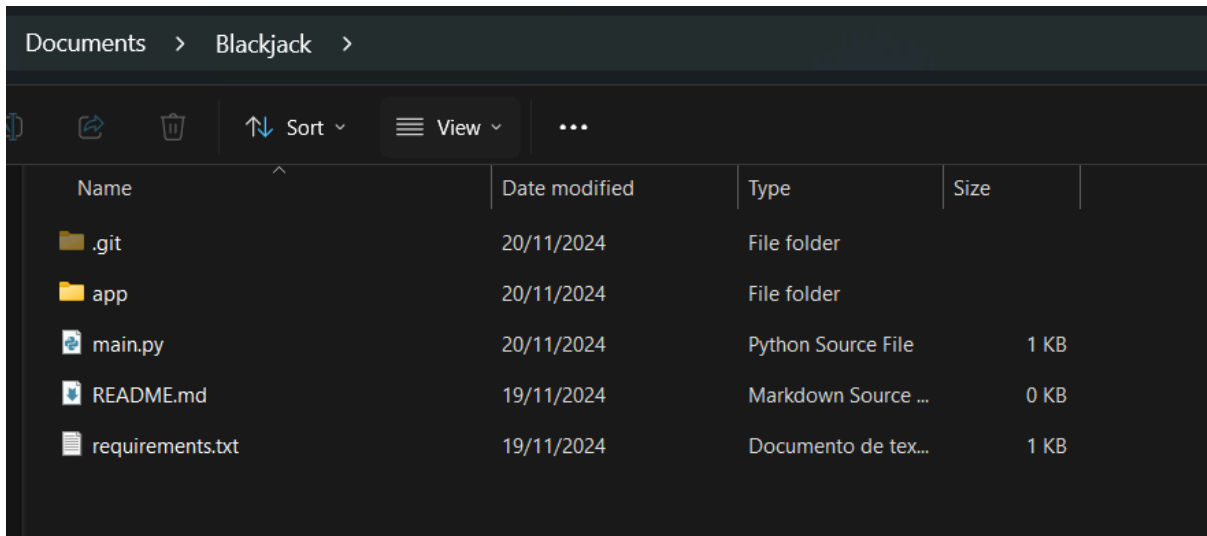
Elaborado por:

Dylan Jordany Rodríguez
Yosimar Antonio Montenegro

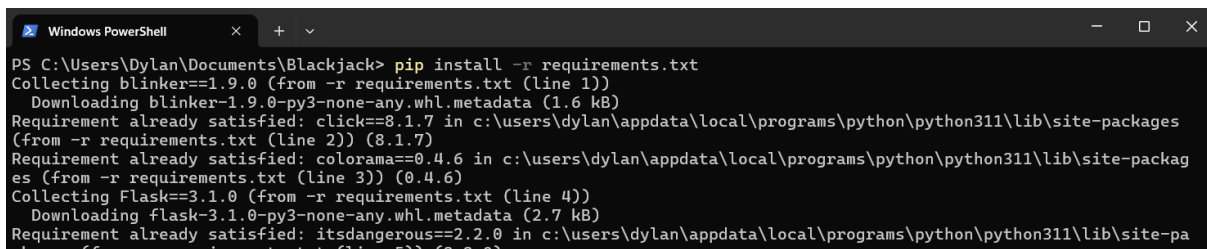
GR60

II Semestre - 2024

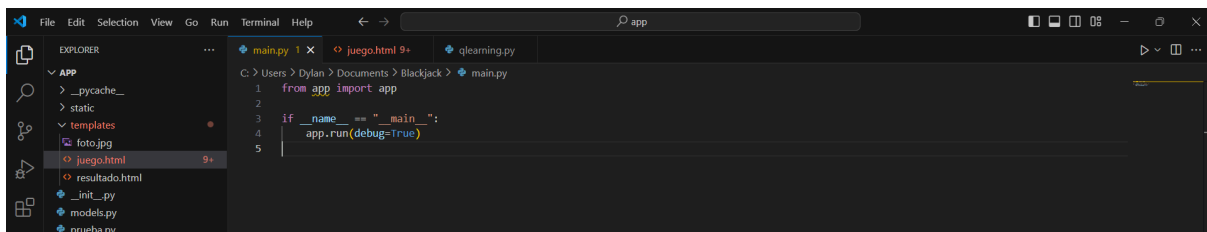
Instrucciones De Ejecución y Uso Del Programa



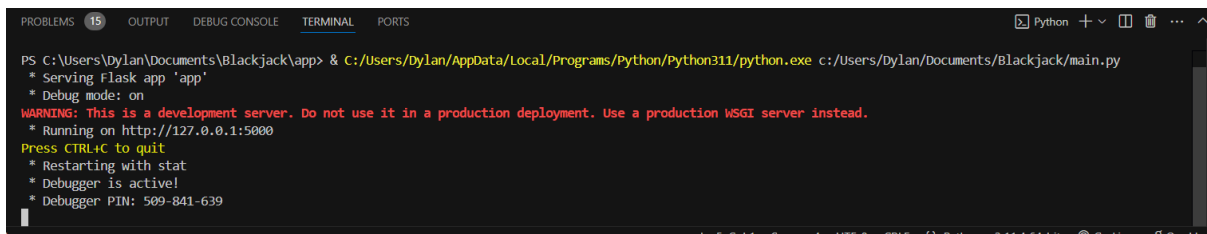
Estando en la raíz del proyecto se debe de ejecutar el siguiente comando: `pip install -r requirements.txt`



Posterior a esto se debe de ejecutar el main.py



Se debe de acceder a la dirección que se indica en la línea Running on `http://127.0.0.1:5000`



En la ventana del juego se muestran los botones para realizar los movimientos de pedir carta o plantarse. Se le muestra al jugador la probabilidad de pasarse de 21 si pide otra carta



Al pasar las rondas los demás jugadores van jugando, en este caso el jugador perdió y se muestra el botón para volver a jugar otra partida

Implementación de la IA

La IA para este juego se diseñó usando Q Learning, este mismo permite que la IA tome decisiones óptimas a lo largo del juego aprendiendo de experiencias y entrenando antes de realizar una jugada. Este modelo como tal usa una tabla Q que tiene estados del juego como un puntaje o cartas visibles en el dealer, dándole posibles acciones como pedir carta o plantarse. Durante la fase de entrenamiento la IA explora diferentes formas mediante simulaciones previas, actualizando los valores de la tabla Q según una recompensa que se le haya dado. Cada victoria, empate y derrota influye en las acciones tomadas.

Esta IA evalúa su estado actual y consulta con la tabla Q para tomar decisiones más inteligentes. Al iniciar la IA toma decisiones imprecisas debido a su tasa de exploración, pero una vez ha entrenado suficiente empieza a tomar acciones con sentido mientras su tasa de reducción afecta la exploración. Se hace uso de una fórmula del Q Learning.

Implementación del algoritmo

El algoritmo implementa una simulación de Monte Carlo para calcular la probabilidad de no superar 21 puntos al pedir una carta adicional en Blackjack. El método `probabilidadDeNoSuperar21` realiza 1000 simulaciones (ciclos) donde en cada una: crea una copia de la baraja y del jugador actual, baraja las cartas aleatoriamente, reparte una carta y verifica si la puntuación resultante es menor o igual a 21. Cuenta cuántas veces no se pasa de 21 (`noPasarse`) y al final calcula el porcentaje de éxito dividiendo este contador entre el número total de simulaciones y multiplicando por 100. Este enfoque de Monte Carlo permite estimar de manera práctica el riesgo de "pasarse" al pedir una carta adicional, lo que ayuda al agente de Q-Learning a tomar decisiones más informadas sobre si pedir o plantarse en una situación dada.