

Prueba 3 : Parte II (*Take-Home*)

Curso de Pensamiento Computacional

"If I have seen further it is by standing on the shoulders of
giants" (*Sir Isaac Newton*)

Darío Creado F. / dario.creado@colegiolagirouette.cl
Curso de Pensamiento Computacional
Colegio La Girouette

June, 2022

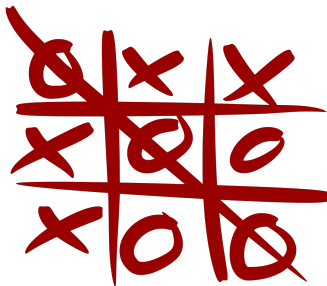
Problema Extra *Tic Tac Toe*



Problema Extra *Tic Tac Toe*

Descripción

TicTacToe, Gato o Tres en línea, es un conocido juego de estrategia en la que se busca completar tres elementos similares en línea, ya sea horizontal, vertical o diagonal. Para ello se trabaja con cruces y círculos, sobre un tablero de 3×3 .



Problema Extra *Tic Tac Toe*

Este es un juego cuya cantidad de posibles combinaciones es conocida; además existe una estrategia estudiada que puede ser aplicada tanto por un jugador como por un computador que desee siempre ganar o empatar.

Actividad - Requisitos

Para lograr la realización de este juego, se deben cumplir con las siguientes acciones.

Task 1: Jugabilidad inicial (50 %) El juego en su forma inicial, en que juegan 2 jugadores humanos, deberá satisfacer las siguientes especificaciones:

- Contar con una interfaz sencilla, que contenga las instrucciones de cómo jugar, comenzando siempre con cruces.
- Permitir el ingreso de los nombres de los jugadores.
- Poder jugar una ronda de 3 juegos, indicando al final el nombre del jugador ganador o si hubo un empate.
- Se debe ir alternando qué jugador comienza en cada ronda.
- En cada ronda, se debe indicar el jugador que gana.

Problema Extra *Tic Tac Toe*

Actividad - Requisitos

- Chequeo de errores. El programa debe ser robusto y alertar acerca de los siguientes posibles errores.
 - a) No permitir que los jugadores ingresen el mismo nombre.
 - b) No permitir sobrecribir una posición ya ocupada.
 - c) No permitir el ingreso de una posición no existente, pensando en un número o letra que no corresponda a una posición válida.
 - d) En esencia, con un programa robusto se refiere a que ante entradas que no corresponden, el programa pueda seguir funcionando sin cerrarse abruptamente.
 - e) Se debe agregar una opción de finalizar el juego sin haber terminado todas las rondas. (Rendirse).

Actividad - Requisitos

Task 2: Incorporar computador (30 %) En esta actividad, se debe incorporar un computador que sea capaz de jugar el juego de forma aleatoria (nivel fácil) y con un leve nivel de inteligencia (nivel medio). Este debe considerar las siguientes características:

Nivel Fácil

- Mismas condiciones de Task 1.
- El computador debe hacer solo movimientos válidos, es decir, utilizar una celda disponible y dentro del rango aceptado.
- Se debe ir alternando quién comienza cada ronda.
- Los movimientos deben ser aleatorios por parte del computador.

Actividad - Requisitos

Nivel Medio

- Si el computador tiene un movimiento en el que puede ganar, debe ganar.
- Si el jugador está por ganar, y es turno del computador, el computador debe bloquear su acción. En otras situaciones, se comportará de forma aleatoria.
- Se sugiere que esta parte se construya sobre el código del **Task 1**. Para esto, considere la creación de una pantalla inicial que permita seleccionar entre los distintos modos de jugar.

Actividad - Requisitos

Task 3: Incorporar registro de última partida. (20 %) Esta actividad corresponde a registrar los movimientos de la última partida realizada. Para esto, al finalizar cada ronda, deberá desplegar la opción:

¿Desea mostrar la última ronda?

- En la que al responder afirmativamente, deberá replicar los movimientos realizados, independiente si el modo de juego es persona vs. persona o persona vs. computador.
- Es su deber idear una forma de almacenar esta información a medida que el juego avanza y de eliminarla al comenzar cada ronda, una idea podría ser guardar la información en un archivo externo de cualquier extensión.

Problema Extra *Tic Tac Toe*

Comentarios Finales - Instrucciones

La implementación deberá contar con un soporte gráfico utilizando alguna librería de este tipo PyGame, Turtle, Arcade, etc.

Se pide para el desarrollo de los modos Fácil / Medio el uso de librerías especializadas (random,etc.) y una investigación (no muy extensa) de algoritmos de inteligencia para la resolución de este juego (relacionar con lo visto en clases sobre ML y DL en lo posible)

Comentario : Un ejemplo de *output* del juego en su versión inicial y más sencilla (sólo interacción por medio de consola) se adjunta en un archivo .txt