DOCUMENTACION JUEGO CEPHALOPOD ILEANA SANABRIA

Describiré cada estructura en detalle y justificaré su elección:

Matriz de Números (matriznum):

Descripción: matriznum es una matriz bidimensional de enteros con un tamaño de 5x5. Cada elemento de la matriz almacena un número entero que representa la cantidad de fichas en esa posición del tablero.

Justificación: Esta matriz se utiliza para mantener un registro preciso de la cantidad de fichas en cada celda del tablero. Es una elección apropiada porque proporciona una representación numérica de la disposición de las fichas en el juego y facilita las comprobaciones lógicas, como determinar si un jugador puede comer fichas adyacentes.

Matriz de Letras (matrizlet):

Descripción: matrizlet es una matriz bidimensional de caracteres con un tamaño de 5x5. Cada elemento de la matriz almacena un carácter que representa el color de las fichas en esa posición del tablero.

Justificación: Esta matriz se utiliza para representar visualmente el estado del juego. Los caracteres (letras) en esta matriz representan los colores de las fichas en el tablero. Es esencial para mostrar claramente el tablero a los jugadores y ayuda a identificar las fichas de cada jugador.

Entero (eleccion):

Descripción: eleccion es una variable entera que almacena la elección del jugador sobre si desea jugar contra la máquina o con un amigo. Puede tomar el valor 1 o 2.

Justificación: Esta variable permite al jugador especificar su preferencia de modo de juego. Su elección afecta la lógica del juego, ya que determina si se jugará en modo un jugador o dos jugadores. Esta elección personaliza la experiencia del juego.

Entero (eleccion2):

Descripción: eleccion2 es una variable entera que almacena la elección del jugador sobre el color de las fichas del Jugador 1 (rojo o verde). Puede tomar el valor 1 o 2.

Justificación: El color de las fichas del Jugador 1 es una preferencia del jugador y afecta la representación visual del juego. La elección del color personaliza la experiencia y permite que los jugadores se identifiquen fácilmente en el tablero.

Enteros (fila y columna):

Descripción: fila y columna son variables enteras que almacenan las coordenadas (fila y columna) en el tablero que el jugador selecciona para colocar una ficha.

Justificación: Estas variables son esenciales para que el jugador especifique dónde desea colocar su ficha en el tablero. Permiten al jugador interactuar con el juego de manera efectiva al indicar la ubicación deseada.

Arreglos (matcomer, coordenadas, eleccion):

Descripción: Estos arreglos se utilizan para almacenar temporalmente datos relacionados con la lógica de comer fichas adyacentes. matcomer registra la elección de comer o no, coordenadas almacena las coordenadas de las fichas adyacentes y eleccion almacena elecciones relacionadas con comer fichas.

Justificación: Estas estructuras temporales son esenciales para llevar a cabo comprobaciones y actualizaciones relacionadas con comer fichas adyacentes. Ayudan a gestionar la lógica del juego y las decisiones del jugador de manera eficiente.

Caracteres (letra, jugador1, jugador2):

Descripción: letra, jugador1, y jugador2 son variables de tipo caracter que representan letras y colores de los jugadores.

Justificación: Estas variables se utilizan para mostrar visualmente las letras (colores) de las fichas de los jugadores y para identificar al Jugador 1 y al Jugador 2. Facilitan la representación gráfica del estado del juego y ayudan a identificar a qué jugador pertenece cada ficha.

Enteros (contX, contO):

Descripción: contX y contO son variables enteras que almacenan el recuento de fichas con un valor de 6, lo que indica una posible victoria, para cada jugador.

Justificación: Estas variables son esenciales para determinar al ganador del juego. Basándose en el recuento de fichas con un valor específico, permiten decidir el resultado del juego de manera precisa.

Entero (turno):

Descripción: turno es una variable entera que lleva un registro del turno actual del jugador (1 o 2).

Justificación: El turno es fundamental para alternar entre los jugadores y permitir que jueguen por turnos. Determina cuál de los jugadores realiza una acción en un momento dado.

Arreglo de Caracteres (auxiliar):

Descripción: auxiliar es un arreglo de caracteres que se utiliza para almacenar temporalmente la elección de fichas de un jugador.

Justificación: Esta estructura de datos temporal es útil para realizar comprobaciones y actualizaciones relacionadas con la elección de fichas de un jugador, como cuando se selecciona el color de las fichas al inicio del juego.

Cada una de estas estructuras de datos se ha elegido cuidadosamente según su función en el juego y su relevancia para el seguimiento del estado del juego, la interacción con el jugador y la lógica del juego. Estas estructuras son cruciales para el funcionamiento correcto del juego y contribuyen a la experiencia del jugador.

FUNCIONES DEL JUEGO:

int obtenerPosicionAleatoria(int \*fila, int \*columna): Esta función se encarga de obtener una posición aleatoria en la matriz (tablero) para el jugador o la máquina. Se genera una fila y una columna aleatorias, asegurándose de que la casilla esté vacía antes de asignar las coordenadas a los punteros fila y columna. Retorna 1 en caso de éxito.

void comerDadosAleatorio(int matcomer[2][4], char matrizlet[5][5], int matriznum[5][5], int fila, int columna): Esta función permite que el jugador o la máquina "coman" los dados adyacentes de forma aleatoria. Se verifican los números adyacentes y se toma una decisión aleatoria sobre cuáles comer. En función de la elección, se actualiza el tablero. Esta función no se proporciona en el código, pero su propósito es gestionar el juego cuando se come el dado adyacente.

void printfas(): Esta función imprime una especie de título o diseño del juego en la consola. Es principalmente estética y no influye en la lógica del juego.

int obtenerPosicion(int \*fila, int \*columna): Permite al jugador humano elegir una posición en el tablero ingresando una letra y un número que representan la fila y la columna, respectivamente. Verifica que la entrada sea válida y que la casilla esté vacía. Retorna 1 en caso de éxito.

int reconocimiento(int matriznum[5][5], int fila, int columna): Esta función verifica si una posición en el tablero está ocupada o no. Si la casilla está vacía, devuelve 1; de lo contrario, devuelve 0.

void comerDados(int matcomer[2][4], char matrizlet[5][5], int matriznum[5][5], int fila, int columna): Permite al jugador comer los dados adyacentes en función de ciertas reglas. Verifica cuáles dados son adyacentes y si su suma es menor o igual a 6, permitiendo al jugador elegir si desea comerlos. Actualiza el tablero en consecuencia.

int juego(int eleccion, int \*eleccion2, int matriznum[5][5], char matrizlet[5][5]): Esta función es la función principal del juego. Controla el flujo del juego, alternando entre los turnos del jugador 1 y el jugador 2 (o la máquina). Permite a los jugadores o la máquina realizar movimientos, come los dados adyacentes y verifica el resultado del juego (ganador o empate).

inicio(): Esta función muestra una presentación visual en la consola, imprime las reglas del juego y espera a que el usuario presione Enter para continuar.

eleccion1(int \*e): Esta función solicita al usuario que elija entre jugar con la máquina o con un amigo. El valor seleccionado se almacena en la variable e.

eleccioncolor(int \*e): Esta función solicita al jugador 1 que elija el color con el que desea jugar (rojo o verde). El valor seleccionado se almacena en la variable e.

inicializarMatriz(int matriznum[5][5], char matrizlet[5][5]): Inicializa las matrices matriznum y matrizlet con valores iniciales (0 y espacios en blanco).

imprimirjuego(int matriznum[5][5], char matrizlet[5][5], int \*eleccion2): Esta función imprime el estado actual del juego en la consola, mostrando el tablero y las fichas según las matrices matriznum y matrizlet.

trofeo(): Muestra un trofeo en la consola. Se usa para indicar que un jugador ha ganado.

ganadorX(): Muestra un mensaje y gráficos de victoria en la consola para el jugador X.

ganadorO(): Muestra un mensaje y gráficos de victoria en la consola para el jugador O.

empate(): Muestra un mensaje y gráficos de empate en la consola.

inic(): Imprime un mensaje visual en la consola.

Este juego es una implementación de un juego de estrategia en el que los jugadores llenan casillas y tratan de maximizar su puntaje. El juego incluye una variedad de funciones para manejar la lógica del juego, la entrada de los jugadores y la visualización de la interfaz del juego.