Análisis – Juego de Totito – Alejandro Sagui – C

1. Requisitos que va a tener mi sistema
   1. Hacer un tablero en la consola cuadricula 3 x 3 que este vacía al inicio
   2. Debe dejar que dos personas jueguen por turnos, colocando su X u O en la casilla
   3. Ver que no se pueda poner algo en una casilla ocupada
   4. Permitir que mi sistema determine el turno del jugador 1 y el jugador 2
   5. Asignarle la variable X a un jugador y la variable O a otro
   6. Acabar la partida cuando un jugador logre alinear tres variables consecutivas en fila, columna o diagonal o cuando el tablero este lleno en un empate
   7. Preguntar si se quiere volver a jugar para reiniciar el tablero y si me dicen que no, finalizar el programa
   8. Hacer un reporte al terminar el juego con el numero de partidas ganadas por cada jugador y decir que jugador ha ganado más
   9. Usar arraylist para que los datos se guarden
   10. Que mi programa interactue con el usuario solo en la clase main
2. Clases identificadas y su propósito
   1. Clase juego
      1. Acá se hará el juego del totito
      2. Este va a controlar los turnos, reiniciará el tablero y hará el informe
   2. Clase tablero
      1. Hará la cuadrícula 3 x3
      2. Este va a controlar las casillas, las posiciones de las variables y va a ver si hubo victoria o empate
   3. Clase jugador
      1. Representa a los dos jugadores de mi partida
      2. Tendrá el nombre del jugador, su variable (X u O) y las partidas que ha ganado
   4. Clase main
      1. Acá se manejara toda la interacción con el usuario
      2. Es el punto en el que empiece el programa
3. Atributos y métodos de mis clases
   1. Clase juego
      1. Atributos
         1. Jugador1: jugador = primer jugador
         2. jugador2: jugador = segundo jugador
         3. tablero: tablero = mi representación del tablero
         4. turnoActual: jugador = me dira de quien es el turno
         5. partidasJugadas : int = va a decir cuantas partidas se han jugado
      2. Métodos
         1. iniciarPartida() = empieza la partida, alterna el turno
         2. cambiarTurno() = pasa elturno al otro jugador
         3. veriticarinPartida() = mira si hubo victoria o empate
         4. generarReporte() = este devolverá el resumen de la sesión
         5. jugarMovimiento(int columna, int fila) = va a dejar poner la marca del jugador en la casilla que este vacía
   2. Clase main
      1. Métodos
         1. Main(String[] args) = acá empezara el punto de partida, pide los nombres, empieza el juego y muestra resultados e inormes
   3. Clase tablero
      1. Atributos
         1. red : String[3][3] = matriz del tablero
      2. Métodos
         1. inicializar() = crea mi tablero vacio
         2. estaLleno() = booleano que dará true si las casillas están ocupadas
         3. colocarMarca(int fila, int columna, String marca) = Coloca la marca de un jugador en la posición que diga
         4. mostrarTablero() = muestra una representación del tablero que se este jugando
   4. Clase jugador
      1. Atributos
         1. Nombre : string = nombre del player
         2. Marca : string = X u O
         3. partidasGandas: int = numero de partidas ganadas
      2. Métodos
         1. IncrementaVictoria() = suma una victoria al registro del jugador
         2. getNombre() = nombre del jugador
         3. getMarca() = Devuelve la marca
         4. getVictorias() = da las victorias

Diseño

Diagrama

El contenido generado por IA puede ser incorrecto.