

Laboratorio

Análisis

i. Requisitos funcionales del sistema

1. El sistema debe permitir que dos jugadores compitan por turnos.
2. El tablero debe estar compuesto por fichas ocultas con símbolos duplicados colocados aleatoriamente.
3. Cada símbolo debe aparecer exactamente dos veces.
4. En cada turno, un jugador selecciona dos casillas para revelar.
5. Si las fichas coinciden, el jugador gana un punto y conserva el turno.
6. Si no coinciden, las fichas se ocultan nuevamente y el turno pasa al otro jugador.
7. El juego finaliza cuando todas las fichas han sido emparejadas.
8. Al finalizar el juego, se debe mostrar la cantidad de pares obtenidos por cada jugador y se indica el ganador.
9. Se debe ofrecer la opción de iniciar una nueva partida o terminar la sesión.
10. Se debe validar que no se seleccione la misma casilla dos veces en un turno.
11. Se debe impedir seleccionar fichas ya emparejadas.
12. El sistema debe ser flexible para tableros de distintos tamaños.
13. La interacción con el usuario se debe realizar exclusivamente desde la clase principal, es decir, clase Main.
14. Se prohíbe el uso de System.out.println y Scanner fuera de la clase Main.
15. Se penaliza el uso de while(true) con break.

ii. Clases propósito

1. Ficha: Esta representa una ficha del tablero con su símbolo y estado.
2. Tablero: Este contiene la matriz de fichas y gestiona su inicialización y visualización.
3. Jugador: Almacena el nombre del jugador y la cantidad de pares encontrados.
4. JuegoMemoria: Este controla la lógica del juego, turnos, validaciones.
5. Main: Esta es la clase donde se ejecuta donde interactúa con el usuario y coordina el juego.

iii. Atributos y propósito

Clase Ficha

1. String símbolo: Representa el emoji o símbolo de la ficha.
2. Boolean emparejada: Indica si la ficha ya fue emparejada.
3. Boolean visible: Indica si la ficha está actualmente revelada.

Estos atributos permiten controlar el estado de cada ficha y su visibilidad en el tablero.

Clase Tablero

1. Ficha [][] matriz: Matriz bidimensional que representa el tablero.
2. Int filas: Cantidad de filas de tablero.
3. Int columnas: Cantidad de columnas del tablero.

Se necesitan estos atributos para una estructura dinámica para representar el tablero y permitir flexibilidad en su tamaño.

Clase Jugador

1. String nombre: Identificador del juego.
2. Int paresEncontrados: Contador de pares emparejados por el jugador.

Estas atribuciones permiten llevar el registro individual de cada jugador y determinar al ganador.

Clase JuegoMemoria

1. Tablero tablero: Instancia del tablero actual.
2. Jugador jugador1: primer jugador.
3. Jugador jugador2: Segundo jugador.
4. Jugador turnoActual: Referencia al jugador que tiene el turno.
5. Int totalPares: Total de pares posibles en el tablero.

Esta clase centraliza el control del juego, gestiona turnos y verifica condiciones de victoria.

Clase Main

Esta no contiene atributos, pero gestiona la entrada y salida del usuario y la ejecución del juego. Cumple con la restricción de interacción exclusiva con el usuario y evita penalizaciones.

iv. Métodos y propósito

Clase Ficha

1. mostrar(): Devuelve el símbolo si está visible o un marcador oculto.
2. revelar(): Cambia el estado de visibilidad a verdadero.
3. ocultar(): Cambia el estado de visibilidad a falso.
4. emparejar(): Marca la ficha como emparejada.

Estos permiten controlar el estado de cada ficha durante el juego.

Clase Tablero

1. inicializar(): Asigna símbolos aleatorios duplicados a las fichas.
2. mostrarTablero(): Imprime el estado actual del tablero.
3. getFicha(int fila, int columna): Devuelve la ficha en una posición específica.

4. todasEmparejadas(): Verifica si todas las fichas han sido emparejadas.

Estas facilitan la gestión del tablero y su visualización.

Clase Jugador

1. incrementarPares(): Aumenta el contador de pares encontrados.
2. getNombre(): Devuelve el nombre del jugador.
3. getPares(): Devuelve la cantidad de parejas encontradas.

Estos permiten acceder y modificar la información del jugador.

Clase JuegoMemoria

1. jugarTurno(int f1, int c1, int f2, int c2): Ejecuta la lógica de un turno.
2. validarSeleccion(int f, int c): Verifica si una casilla es válida.
3. cambiarTurno(): Alterna el turno entre jugadores.
4. determinarGanador(): Compara los pares encontrados y determina el ganador.
5. reiniciar(): Reinicia el juego con un nuevo tablero.

Estas encapsulan la lógica del juego y aseguran su correcto funcionamiento.

Clase Main

1. main(String[] args): Punto de entrada del programa.
2. mostrarMenu(): Presenta opciones al usuario.
3. leerEntrada(): Captura las coordenadas seleccionadas.
4. mostrarResultadosFinal(): Muestra el resultado de la sesión.

Estos permiten la interacción con el usuario y el control de las demás clases.