

Documentación

Enlace de la documentación:

https://docs.google.com/document/d/1ci7yetqH-W8fGF8o4ikjlp_6Ky8UK-os5dmiXLxBfOM/edit?usp=sharing.

Tipo

```
CuatroEnLinea
```

Descripción

Cuatro en línea es un juego para 2 jugadores en el cual los jugadores van soltando fichas en el tablero de forma alterna con respecto a los turnos.

El objetivo del juego es conseguir colocar 4 fichas consecutivas ya sea de forma horizontal, vertical o diagonal.

Constructores

```
CuatroEnLinea( int filas, int columnas,  
                String jugadorRojo,  
                String jugadorAmarillo)
```

- **post:** Partida iniciada con el tablero “VACIO” en todas sus posiciones.
Comienza siendo el turno de “jugadorRojo”

Métodos

```
void inicializarTablero()
```

- **post:** Devuelve la matriz con casilleros vacíos en 'i' e 'j' mientras son menores a 'filas' y 'columnas'.

```
int contarFilas()
```

- **post:** devuelve la cantidad máxima de fichas que se pueden alinear en el tablero.

```
int contarColumnas()
```

- **post:** devuelve la cantidad máxima de fichas que se pueden alinear en el tablero.

```
Casillero obtenerCasillero(int fila, int columna)
```

- **pre :** fila está en el intervalo [1, contarFilas()], columnas está en el intervalo [1, contarColumnas()].
- **post:** indica qué ocupa el casillero en la posición dada por fila y columna.

```
void soltarFichaEnColumna(int columna)
```

- **pre :** el juego no terminó, columna está en el intervalo [1, contarColumnas()] y aún queda un Casillero.VACIO en la columna indicada.
- **post:** deja caer una ficha en la columna indicada.

```
int ultimaFilaVacía(int columna)
```

- **pre:** el juego no terminó, fila está en el intervalo [ultimaFilaVacía][columna - 1]
- **post:** devuelve como resultado 'int ultimaFilaVacía'.

```
void cambioDeTurno()
```

- **post:** devuelve el turno rojo en primer lugar, luego al ser 'true' lo convierte a 'false' y es turno del 'amarillo'. Al ser turno 'amarillo' 'true', convierte a turno 'rojo'.

```
boolean termino()
```

- **post:** indica si el juego terminó porque uno de los jugadores ganó o no existen casilleros vacíos.

```
boolean hayGanador()
```

- **post:** indica si el juego terminó y tiene un ganador.

```
String obtenerGanador()
```

- **pre :** el juego terminó.
- **post:** devuelve el nombre del jugador que ganó el juego.

Tipo

```
AnalizarFicha
```

Descripción

AnalizarFicha se encarga de verificar si la ficha soltada es parte de una secuencia de cuatro en línea ya sea de forma horizontal, vertical o diagonal.

Constructores

```
CuatroEnLinea( Casillero[][] tablero,  
               int filas, int columnas )
```

- **post:** Guarda en los atributos la posición de la ficha que será analizada.

Métodos

```
boolean calcularCuatroEnLinea()
```

- **post:** Devuelve si la ficha analizada pertenece a una secuencia de 4 en línea ya sea de forma vertical, horizontal o diagonal.

```
int analizarAumentando( int filas, int columnas)
```

- **pre:** Ingresar la cantidad de filas y columnas que se deben ir agregando para formar la secuencia deseada.
- **post:** Devuelve la cantidad de fichas contiguas según la secuencia indicada.

```
boolean seguirAvanzando()
```

- **post:** Devuelve si existe alguna ficha contigua .

```
boolean seguirAvanzando()
```

- **post:** Devuelve si la ficha contigua es igual a la ficha anterior o si ya existen 4 fichas consecutivas iguales.

```
boolean siguienteCasillero()
```

- **post:** Devuelve el siguiente casillero;