

# Manual del Usuario

## Introducción

Este juego se basa en el clásico Tatetí en la que los jugadores colocan fichas en un tablero y pueden usar cartas con diferentes efectos estratégicos para cambiar el curso del juego.

## Objetivo del Juego

El objetivo del juego es lograr formar una línea de fichas del mismo color en el tablero en cualquiera de sus ejes: X, Y o Z.

## Componentes del Juego

1. Tablero 3D: El tablero está compuesto por casilleros ubicados en una estructura tridimensional. Los jugadores colocan sus fichas en estos casilleros.
2. Fichas: Cada jugador tiene fichas que puede colocar en los casilleros del tablero. Las fichas tienen un color y una posición en el tablero.
3. Cartas: Durante el juego, los jugadores pueden jugar cartas que tienen efectos especiales, como bloquear fichas, anular casilleros o cambiar el color de las fichas.
4. Dados: Los dados se lanzan para determinar el resultado de ciertas acciones del juego.

## Reglas Básicas

1. Turnos: Los jugadores se alternan para colocar sus fichas en el tablero o jugar cartas.
2. Efectos de las Cartas: Cada carta tiene un efecto específico que puede influir en el juego. Por ejemplo, puede hacer que un jugador pierda su turno o que un casillero sea anulado.
3. Bloqueo de Fichas: Un jugador puede bloquear una ficha para evitar que se mueva durante un turno.
4. Colocación de Fichas: Los jugadores deben colocar sus fichas en un casillero vacío del tablero. Si un casillero está bloqueado, no se puede colocar una ficha en él.

## Funcionalidad de las Cartas

Las cartas en el juego tienen diferentes acciones que afectan al jugador o al tablero.

Algunas de estas acciones son:

- Perder el turno: Un jugador pierde su turno si juega esta carta.
- Volver atrás jugada: Un jugador puede deshacer su última jugada.
- Bloquear ficha: Bloquea una ficha de otro jugador para evitar que se mueva.
- Anular casillero: Anula un casillero, impidiendo que se pueda colocar una ficha en él.
- Cambiar color ficha: Cambia el color de una ficha en una posición específica del tablero.
- Intercambiar fichas: Permite intercambiar las posiciones de dos fichas en el tablero.

## Cómo Jugar

1. Lanza Dado: El jugador lanza un dado para determinar qué acción puede realizar. Dependiendo del valor obtenido, se le dará la opción de jugar una carta o colocar una ficha en el tablero.
2. Colocar una Ficha: El jugador selecciona un casillero vacío en el tablero y coloca su ficha allí.
3. Jugar Carta: El jugador puede elegir una carta de su mano y utilizarla para realizar una acción especial en el juego.

## Fin del Juego

El juego termina cuando uno de los jugadores consigue alinear tres fichas del mismo color en el tablero en cualquiera de los ejes (X, Y o Z). El jugador que logre esto es el ganador.

---

## Manual del Programador

### Introducción

Este sistema está compuesto por varias clases que modelan un juego tridimensional donde los jugadores colocan fichas en un tablero y pueden usar cartas para influir en el juego. La arquitectura está diseñada para ser extensible y modular, siguiendo buenas prácticas de programación como la encapsulación y la validación de entradas.

### Estructura de Clases

1. Clase Carta: Representa una carta en el juego. Cada carta tiene una acción asociada y puede realizar diferentes efectos cuando se juega. La clase maneja la lógica de los efectos de las cartas, como perder el turno, bloquear fichas, cambiar el color de las fichas, etc.
  - Atributos:
    - accion: La acción que realiza la carta (por ejemplo, "Perder el turno", "Bloquear ficha").
    - id: Un identificador único para cada carta.
  - Métodos:
    - jugarCarta: Ejecuta la acción de la carta sobre el jugador o el tablero.
2. Clase Casillero: Representa un casillero en el tablero 3D. Cada casillero tiene una posición en el espacio tridimensional y puede contener una ficha.
  - Atributos:
    - x, y, z: Coordenadas en el espacio 3D.
    - ficha: La ficha que ocupa el casillero.
  - Métodos:
    - colocarFicha: Coloca una ficha en el casillero.
    - toString: Representa el casillero como una cadena de texto.
3. Clase Dado: Representa un dado utilizado para lanzar en el juego y determinar el resultado de ciertas acciones.
  - Atributos:
    - cantidadDeCaras: El número de caras del dado (por defecto, 2).
  - Métodos:

- lanzarDado: Lanza el dado y devuelve un número aleatorio entre 1 y la cantidad de caras.
4. Clase Ficha: Representa una ficha en el tablero. Cada ficha tiene un color, una posición en el tablero y un estado de bloqueo.
    - Atributos:
      - color: El color de la ficha.
      - x, y, z: Las coordenadas de la ficha en el tablero.
      - bloqueada: Un indicador de si la ficha está bloqueada.
    - Métodos:
      - mover: Mueve la ficha a una nueva posición si no está bloqueada.
      - bloquear: Bloquea la ficha para evitar que se mueva.
      - desbloquear: Desbloquea la ficha para que pueda moverse.
  5. Clase Tablero3D: Representa el tablero 3D donde se colocan las fichas. Esta clase maneja la validación de las posiciones de las fichas y la ejecución de las jugadas.
    - Atributos:
      - casilleros: Una matriz de casilleros que forman el tablero.
    - Métodos:
      - validarPosicion: Valida si una posición (x, y, z) en el tablero es válida.
      - bloquearFicha: Bloquea una ficha en una posición específica.
      - anularCasillero: Anula un casillero para que no se pueda usar.
      - cambiarColorFicha: Cambia el color de una ficha en el tablero.

## Buenas Prácticas de Programación

1. Encapsulamiento: Las clases utilizan métodos getters y setters para acceder a los atributos privados, asegurando que los datos sean manipulados de forma controlada.
2. Validación de Entradas: Antes de realizar cualquier acción, las clases validan que los objetos involucrados no sean null y que las posiciones sean válidas.
3. Modularización: El código está dividido en clases pequeñas y manejables, lo que permite la reutilización del código y facilita su mantenimiento.
4. Excepciones: Se manejan las excepciones adecuadamente, proporcionando mensajes de error claros en caso de que se ingrese una acción no válida o se intente realizar una operación no permitida.