

# Determinants

## Tarea Programada - Álgebra Lineal para Computación MA - 0292

### Universidad de Costa Rica

Juan Pablo Vindas Ureña - C38554

Jhonny Leung Quesada - C34259

Profesor: Jorhan Chaverri Hernández

#### Lógica:

Todo se desarrolla en el método “update” de la clase “Game”, apoyándose de la clase “Deck”, y la clase “Matrix”.

Update:

#### 1. Inicialización

- Se inicializan matrices y arreglos para llevar un registro de los determinantes recientes de cada jugador.
- Se muestra una matriz inicial (main\_matrix) y su determinante.
- Se le presenta al usuario la opción de generar una nueva matriz con el método (init\_matrix), repitiendo este proceso hasta que el usuario lo desee.

#### 2. Determinante principal

- Se guarda el determinante de la matriz inicial como referencia (main\_determinant).

#### 3. Turnos de jugadores

- En un ciclo de 5 iteraciones (5 turnos, por cada jugador):
  - **Jugadores modifican matrices:** Cada jugador modifica una entrada de su matriz con las cartas que se le brindan (5 cartas).
  - **Validaciones de determinantes:** Si el determinante es igual al original (main\_determinant), el jugador debe volver a elegir una carta y cambiar una entrada hasta que cambie el determinante. Si un determinante es repetido respecto a turnos anteriores, el jugador pierde su turno. Si el determinante es 21, y ningún otro jugador obtiene un determinante de 21, el jugador gana automáticamente y el juego termina.

#### 4. Puntajes:

- En el caso de que el determinante sea positivo se calcula el puntaje del jugador basado en el valor absoluto de la diferencia entre el determinante actual y el determinante original. De lo contrario (determinante negativo) al puntaje total anterior se le suma directamente el determinante de turno.

#### 5. Guardado de determinantes:

- Se actualizan los arreglos de “p1\_dets”, “p2\_dets” y “p3\_dets” con los últimos determinantes para la mecánica de bloqueo. Además, se guardan para determinar ganadores o perdedores en caso de un empate cuando acabe el juego.

#### 6. Estado del juego: Los jugadores al final de cada turno pueden:

- Jugar el turno cambiando una entrada de la matriz.
- Conseguir una nueva baraja (únicamente tres veces por juego)
- Finalizar el juego.

#### 7. Determinación del ganador o perdedor

- Cuando no hay más turnos restantes, se determina el ganador utilizando el método “get\_ranking” presente en la clase GUI, el ganador o perdedor es determinado a partir de los siguientes requisitos:

d) *Ganador* será el jugador quien tenga más puntos (positivos) al final de la última ronda. Ahora, el primer jugador que alcance, en alguno de los turnos, el valor 21 en el determinante gana inmediatamente. Por otro lado, si todos los jugadores llegan al final con puntaje negativo, entonces nadie ganará, y se llamará *perdedor* quien tenga menos puntos (más puntos negativos). El programa debe indicar cuál jugador ganó/perdió y su puntaje respectivo. En caso de un empate: el *ganador* será quien haya tenido el primer *determinante de turno* mayor; o el *perdedor* será quien haya tenido el último *determinante de turno* menor.

### **Parte gráfica e interactiva:**

En esta parte, se aprovechan clases de manejo general como “GUI”, que contiene objetos y elementos que manejan el apartado visual, además de métodos de dibujado presentes en clases “Deck” y “Matrix”

#### **1. Librería:**

- Se utilizó una librería presente en el paquete de Java llamada **“JFrame”** y **“java.awt”**, la cuál ofrece diversas herramientas para crear ventanas, dibujar imágenes, permitir transparencia, manejo de el mouse y soporte para dibujar strings con fuentes personalizadas.

#### **2. Interacción:**

- Utilizando las plantillas brindadas por java.awt, se creó una clase para el manejo del mouse, permitiendo almacenar las coordenadas del mouse, y los botones que se estén presionando. A partir de estas funcionalidades se implementó una clase “Button” para permitir interacciones como:
  - Seleccionar entradas de la matriz
  - Seleccionar cartas para cambiar las entradas
  - Botones como el generador de nuevos mazos durante el juego y nuevas matrices antes de iniciar el juego
  - El botón para terminar el juego
  - Manejo de menus

Esto con el fin de permitir una experiencia más interactiva y cómoda para el usuario.

#### **3. Gráficos:**

- Las imágenes (sprites) presentes en el juego son originales, creadas utilizando software gratuito por los creadores del juego, utilizando las herramientas brindadas por la librería, con funciones como Graphics2D para dibujarlas en pantalla
- La fuente utilizada para el texto presente en el proyecto es **“Upheaval”**
- Clases como “Background”, “Element”, “MovingShape” y entre otros manejan la parte visual, montando los fondos y el movimiento de los elementos en pantalla para crear un atractivo al juego y atención mayor de el usuario.