Escuela Politécnica Superior

Asignatura: Técnicas de programación avanzada

Práctica Final TPA

Documentación

NOMBRE Y APELLIDOS DE AMBOS ESTUDIANTES:
Jeronimo Boza García

UNIVERSIDAD NEBRIJA

Tabla de contenido

Descripción y reglas de la aplicación (Juego del UNO)

El juego de cartas Uno es un juego de mesa popular y entretenido que se juega con una baraja especial.

Descripción del Juego

Uno es un juego de cartas en el que el objetivo principal es ser el primero en deshacerse de todas las cartas de tu mano. El juego se juega con una baraja especial de 108 cartas, que incluyen cartas de colores y cartas de acción especiales.

La Baraja

La baraja de Uno consta de:

- Cartas de colores: Cuatro colores (rojo, verde, azul, amarillo), con números del 0 al 9.
- Cartas de acción:
  - Salta: El siguiente jugador pierde su turno.
  - Reversa: Invierte el sentido del juego.
  - Toma Dos: El siguiente jugador debe tomar dos cartas y pierde su turno.
  - Comodín: Permite cambiar el color actual.
  - Comodín Toma Cuatro: Permite cambiar el color actual y el siguiente jugador debe tomar cuatro cartas.

  **(El cambio de color se realizará por defecto de manera aleatoria)

Reglas Básicas

Preparación del Juego

- Se barajan las cartas y cada jugador recibe 7 cartas.
- Se coloca el resto de la baraja boca abajo para formar el mazo de robo.
- Se voltea la primera carta del mazo de robo para iniciar la pila de descarte.

Si es una carta de acción, se aplica la acción inmediatamente.

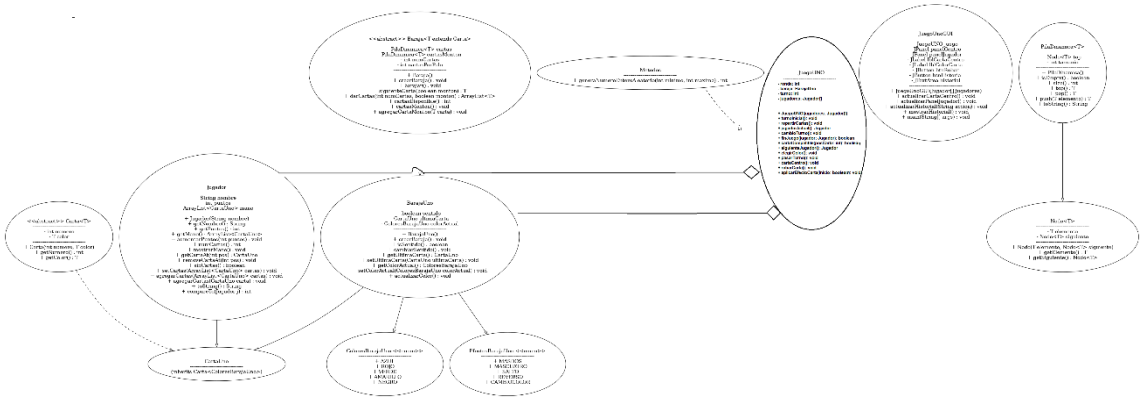
Desarrollo del Juego

- Los jugadores se turnan en sentido horario (a menos que una carta Reversa cambie la dirección).
- En su turno, un jugador debe jugar una carta de su mano que coincida en color o número con la carta superior de la pila de descarte, o jugar una carta de acción.
- Si no puede jugar ninguna carta, debe robar una del mazo. Si puede jugar esta carta, puede hacerlo inmediatamente. Si no, su turno termina.

Finalización del Juego

- El primer jugador en quedarse sin cartas gana la partida.

UML de la aplicación



Enlace a la imagen

CartaUno

- Representa una carta en el juego con propiedades como color, número, y efecto.
- Métodos para obtener propiedades y verificar compatibilidad.
- Representa la baraja de cartas.
- Métodos para dar cartas, sacar la siguiente carta, barajar, cambiar el sentido del juego, y actualizar el color.

- **Atributos**:
  - numero: Número de la carta.
  - color: Color de la carta.
  - efecto: Efecto especial de la carta (si tiene).
- **Métodos**:
  - isEspecial(): Verifica si la carta es especial.
  - compatible(CartaUno otraCarta): Verifica si la carta es compatible con otra carta.
  - getColor(): Devuelve el color de la carta.
  - getNumero(): Devuelve el número de la carta.
  - getEfecto(): Devuelve el efecto especial de la carta.

-

Nodo<T>

- Nodo genérico usado en PilaDinamica.
- Contiene un elemento y un puntero al siguiente nodo.

PilaDinamica<T>

- Implementación de una pila genérica dinámica.
- Métodos para verificar si está vacía, obtener el tamaño, acceder al elemento en la parte superior, eliminarlo (pop), añadir un nuevo elemento (push), y convertir la pila a una cadena.

- **Atributos**:
  - top: Nodo en la parte superior de la pila.
  - tamanio: Tamaño de la pila.

- **Métodos**:
  - isEmpty(): Verifica si la pila está vacía.
  - size(): Devuelve el tamaño de la pila.
  - top(): Devuelve el elemento en la parte superior de la pila.
  - pop(): Remueve y devuelve el elemento en la parte superior de la pila.
  - push(T elemento): Agrega un elemento a la pila.
  - toString(): Devuelve una representación en cadena de la pila.

Jugador

- Representa a un jugador en el juego.
- Métodos para gestionar el nombre, puntos, mano de cartas, y funcionalidades relacionadas con el juego.
- **Atributos:**
    - nombre: Nombre del jugador.
    - mano: Lista de cartas del jugador (ArrayList<CartaUno).

- **Métodos:**
    - Métodos de acceso y modificación (getNombre, getMano, setCartas, agregarCartas).
    - Métodos para manipular la mano (numCartas, getCartaAt, removeCartaAt, sinCartas).
    - Implementación del método compareTo para comparar jugadores por puntos.

JuegoUNO

- Maneja la lógica del juego UNO.
- Métodos para inicializar el juego, gestionar turnos, verificar el estado del juego, calcular puntos, cambiar el turno, y aplicar efectos de las cartas especiales.

- **Atributos**:
    - baraja: La baraja de cartas del juego.
    - turno: Índice del jugador cuyo turno es el actual.
    - jugadores: Array de jugadores participando en el juego.

- **Métodos**:
    - turnoInicial(): Selecciona aleatoriamente el jugador inicial.
    - repartirCartas(): Reparte las cartas iniciales a los jugadores.
    - mostrarTurnoActual(): Muestra el jugador que tiene el turno actual.
    - mostrarCartasJugadorActual(): Muestra las cartas del jugador actual.
    - cambioTurno(): Cambia el turno al siguiente jugador.
    - ganadorPartida(): Devuelve el jugador que ha ganado la partida actual.
    - finJuego(): Indica si el juego ha terminado.
    - cartaCompatible(int posCarta): Verifica si una carta seleccionada es compatible con la carta en el centro.
    - siguienteJugador(): Devuelve el siguiente jugador según el sentido del juego.
    - elegirColor(): Selecciona aleatoriamente un color cuando se juega una carta especial que cambia el color.
    - pasarTurno(): Cambia el turno sin realizar acciones adicionales.
    - cartaCentro(): Muestra la carta en el centro del juego.

- robarCarta(): Permite al jugador actual robar una carta de la baraja.
- aplicarEfectoCarta(boolean inicio): Aplica el efecto de una carta especial.

JuegoUnoGUI

- Interfaz gráfica para el juego UNO.
- Contiene paneles para mostrar la carta central y la mano del jugador, y botones para robar cartas, pasar el turno, y mostrar el historial de acciones.

- **Atributos:**
  - Gestiona componentes gráficos como paneles, etiquetas, botones, y un área de texto para el historial.
- **Métodos:**
  - JuegoUnoGUI(Jugador[] jugadores): Constructor que inicializa la GUI y configura eventos.
  - actualizarCartaCentro(): Actualiza la carta central mostrada en la interfaz.
  - actualizarPanelJugador(): Actualiza la visualización de las cartas del jugador actual.
  - actualizarHistorial(String accion): Añade acciones al historial.
  - mostrarHistorial(): Muestra el historial en un cuadro de diálogo.
  - main(String[] args): Método principal para ejecutar la aplicación GUI.

BarajaUno

- **Atributos**:
  - cartas: Lista de cartas en la baraja.
  - sentido: Sentido del juego (normal o inverso).
  - colorActual: Color actual en juego.

- **Métodos**:
  - darCartas(int numCartas, boolean inicial): Devuelve una lista de cartas para ser repartidas a un jugador.
  - getUltimaCarta(): Devuelve la última carta jugada.
  - agregarCartaMonton(CartaUno carta): Agrega una carta al montón.
  - cambiarSentido(): Cambia el sentido del juego.
  - actualizarColor(): Actualiza el color actual en juego.
  - barajar(): Baraja las cartas del montón para seguir jugando.