Universidad San Carlos de Guatemala
Centro Universitario de Occidente
Ingeniería en Ciencias y Sistemas
Laboratorio de Estructura de Datos

UNO

Objetivo General

El objetivo de este proyecto es desarrollar una versión avanzada del juego de cartas UNO mediante la consola de C++ (con opción a interfaz gráfica). El sistema debe ser altamente parametrizable, permitiendo activar o desactivar reglas especiales y variantes del juego como "UNO Flip".

Objetivos Específicos

- Implementar estructuras de datos desde cero.
- Asegurar que todas las estructuras se actualicen correctamente.
- Ofrecer una interfaz intuitiva

Descripción

Se necesita hacer un prototipo eficiente de lo que es un juego de UNO bastante avanzado, dónde hay varias opciones las cuales se pueden jugar, la cantidad de jugadores es de 2 mínimo, y sin límite superior (lista enlazada circular), dónde el objetivo del juego es el primero que baje todas las cartas, se pide analizar y usar las estructuras más adecuadas para cada uno de los procesos del juego.

Requerimientos Técnicos Obligatorios

- **Lenguaje: C++**
- **Gestión de Memoria:** Uso obligatorio de punteros y estructuras dinámicas.
- **Paradigma:** Programación Orientada a Objetos (POO).
- **Interfaz:** Consola de texto (CLI) con menús claros. Se permite el uso de librerías gráficas (SFML, SDL, Qt) de forma opcional.

Lógica del Mazo y Jugadores

Dinámica de Escalamiento

- **Jugadores:** Mínimo 2, máximo definido por la memoria del sistema.
- **Mazos Dinámicos:** El mazo base tiene 108 cartas. Se debe añadir un mazo extra completo por cada 6 jugadores adicionales:
    - 1-6 jugadores: 1 mazo (108 cartas).
    - 7-12 jugadores: 2 mazos (216 cartas).
    - n jugadores: ((n_jugadores - 1) / 6 +1) mazos.

Estructura de Datos del Mazo

El mazo principal debe implementarse estrictamente como una **Pila (Stack)** propia

- **Barajado:** Implementar un algoritmo de aleatoriedad antes de apilar las cartas.

Configuración de Partida (Flags de Reglas)

Antes de iniciar, el sistema debe permitir activar/desactivar las siguientes opciones:

1. **Acumulación (Stacking):**
   - Permite responder a un +2 con otro +2 (acumula +4 al siguiente y así sucesivamente).
   - Permite responder a un +4 con otro +4.
   - Restricción: No se puede poner un +2 sobre un +4.
2. **El Reto del +4:**
   - Si un jugador lanza un +4, el afectado puede "Retar".
   - Si el lanzador tenía otra carta del color actual en su mano o un número actual: el lanzador roba 4.
   - Si el lanzador no tenía otra opción: el que retó roba 6 (los 4 originales + 2 de penalización).
3. **Modo de Robo:**
   - Opción A: Robar sólo 1 carta y pasar.
   - Opción B: Robar hasta poder jugar (sin límite).
4. **Grito de "¡UNO!":**
   - El jugador debe escribir "UNO" antes de terminar su turno si le queda 1 carta.
   - Si no lo hace y alguien lo reporta antes del siguiente turno: roba 2.
   - Si alguien reporta erróneamente: el reportador roba 2.
5. **Ganar con negra:**
   - El jugador puede ganar aún si su última carta es negra.
   - Si no está habilitada está opción el jugador no puede bajar de último una carta que sea negra, si un jugador su última carta es negra, no podrá jugar, debe robar una carta hasta que pueda jugar según el modo de robo.

Expansión: UNO Flip (Lado Oscuro)

Si se activa esta opción, se añaden 8 cartas de flip por cada deck (2 de cada color) que activan el cambio de lado.

Característica	Lado Claro (Normal)	Lado Oscuro (Dark)
Cartas de Robo	+1 (en vez de +2) / +2 multicolor	+3 / +6 multicolor

Salto	Salta al siguiente	**Salto a todos** (vuelve a jugar el lanzador)
Reverse	Cambiar dirección	Cambiar dirección
Comodín Color	Normal	**Color Eterno:** El siguiente roba hasta hallar el color
Victoria	Normal	Normal
Colores	rojo, amarillo, azul, verde	rosa, turquesa, naranja, violeta.
Cartas Números	Normal	Normal

Cuando se active estas cartas, se debe hacer aleatorias las partes de adelante y atrás, por ejemplo si en una partida delante de una carta verde número 2, hay un reverse rosa, no necesariamente la siguiente partida exista esta carta (la combinación), eso sí, deben haber la misma cantidad de 1, 2, 3… y de accionables con sus colores en cada partida de cada lado.

Cartas Personalizadas

Cada grupo de alumnos debe inventar e implementar **2 tipos de cartas nuevas** que no sean simples variaciones numéricas. Deben ser creativas.

Entregables

1. **Código Fuente:** Organizado en archivos `.h` y `.cpp`.
2. **Diagrama de Clases (UML):** Explicando la jerarquía de las cartas.
3. **TAD**s: Con toda su estructuras
4. **Manual de Usuario:** Explicando cómo activar las reglas y cómo jugar.
5. **Análisis de Complejidad:** Breve explicación del tiempo de ejecución de barajar y buscar cartas en la mano.
6. Makefile funcional
7. Ejecutable compilado

8. **Archivo README** con instrucciones de compilación

Importante

- Lenguaje a utilizar C/C++.
- Es obligatoria la programación orientada a objetos.
- El programa debe ejecutarse en consola o en aplicación de computadora.
- Las malas prácticas de desarrollo serán penalizadas.
- Las copias obtendrán nota de cero y se notificará a coordinación.
- Se requiere la creación obligatoria de un archivo Makefile, para obtener el compilado del programa.
- El uso de inteligencia artificial no consciente será penalizado hasta un 90%.

Entrega

La fecha de entrega es el día 24-01-2026 a las 23:59 cómo fecha límite. Los componentes se entregarán en un repositorio de Git a través Classroom

Notas Importantes

- Usar la terminal (se puede usar interfaz gráfica si desea)
- No habrá prórroga en la entrega
- Compilar sin warnings ni errores
- Documentar el código apropiadamente
- No se aceptan copias, si hay copias entre los alumnos se notificará a dirección y se pondrá 0 al proyecto
- La revisión se hará de forma oral, si se identifica desconocimiento del código realizado de parte del alumno que incurra en posible copia, compra de código o uso de IA será penalizado según sea la gravedad del caso.
- La revisión se hará en la máquina del revisor en primera instancia, si no funciona se dará la oportunidad de revisar en su máquina sobre el 60% de la nota.