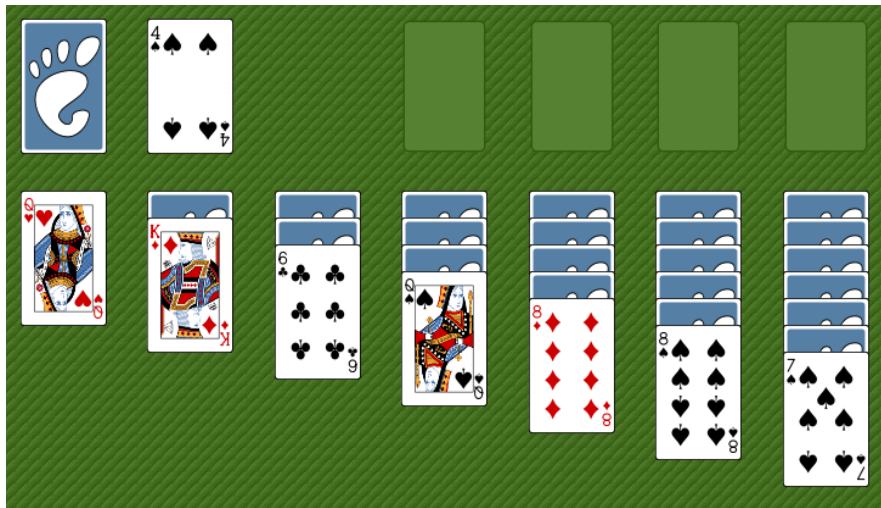


Introducción al Juego de Solitario en C++

Bienvenido al juego de Solitario implementado en C++ utilizando estructuras de datos básicas como pilas, colas y listas doblemente enlazadas, sin hacer uso de librerías externas. Este juego clásico de un solo jugador ofrece una experiencia entretenida y desafiante para usuarios de todas las edades.

El Solitario es un juego de cartas en el que el objetivo es mover todas las cartas a una serie de pilas ordenadas según su palo y valor. En esta implementación, utilizaremos una combinación de estructuras de datos para simular el tablero de juego y las reglas del Solitario.



Características del Juego:

Interfaz de Usuario Intuitiva: La interfaz de usuario proporcionará una experiencia fácil de usar, permitiendo a los jugadores mover las cartas con facilidad utilizando comandos simples.

Movimientos Legales: El juego garantiza que solo se realicen movimientos legales según las reglas del Solitario. Los jugadores podrán mover cartas entre las diferentes pilas y columnas de acuerdo con las reglas establecidas.

Funcionalidades de Deshacer y Rehacer: Se incluirán opciones para deshacer y rehacer movimientos, lo que permite a los jugadores corregir errores o probar diferentes estrategias sin penalización.

Estadísticas del Juego: Los jugadores podrán consultar estadísticas detalladas sobre su desempeño, incluyendo el número de movimientos realizados, el tiempo transcurrido y el porcentaje de juegos completados con éxito.

Estructuras de Datos Utilizadas:

Pilas (Stacks): Las pilas se utilizan para representar las columnas de cartas en el tablero de juego. Las cartas se apilan una encima de la otra, y los movimientos implican agregar o quitar cartas de estas pilas.

Colas (Queues): Las colas se emplean para almacenar temporalmente las cartas que se mueven entre las diferentes columnas del tablero.

Listas Doblemente Enlazadas: Se utilizan listas doblemente enlazadas para representar las cartas en las pilas y las colas. Esta estructura de datos permite un fácil acceso tanto a la carta siguiente como a la anterior en la pila.

Objetivo del Juego:

El objetivo del Solitario es mover todas las cartas del tablero a las pilas de destino, organizadas por palo y en orden ascendente. Los jugadores deben planificar cuidadosamente sus movimientos para evitar quedar atrapados sin opciones.

Herramientas Utilizadas:

Lenguaje de programación C++ con mingw64 version C++ 17.

El desarrollo de este juego se llevó a cabo utilizando el IDE CLion de JetBrains, que ofrece un entorno de desarrollo integrado completo para proyectos en C++.

Para compilar el juego, se utilizó un archivo Makefile que automatiza el proceso de compilación y enlazado de los archivos fuente.

Además, se empleó el sistema de control de versiones Git para mantener un registro del código fuente, facilitando la colaboración y el seguimiento de cambios durante el desarrollo del juego.

Como compilar el Makefile

Que es makefile

Un Makefile es un archivo de texto especial utilizado por el programa "make" en sistemas basados en Unix y similares, como Linux y macOS, para automatizar el proceso de compilación de programas y la gestión de dependencias.

En esencia, un Makefile especifica cómo construir un proyecto a partir de sus archivos fuente. Define reglas y metas que indican a "make" qué comandos ejecutar y en qué orden, basándose en las dependencias entre los archivos del proyecto. Esto permite que "make" compile solo los archivos que han cambiado desde la última compilación, ahorrando tiempo y recursos.

Un Makefile consta de una serie de reglas, cada una de las cuales especifica un objetivo, sus dependencias y las acciones necesarias para construirlo. Por ejemplo, una regla puede indicar cómo compilar un archivo .cpp en un archivo objeto (.o) y luego cómo enlazar todos los archivos objeto para generar un ejecutable.

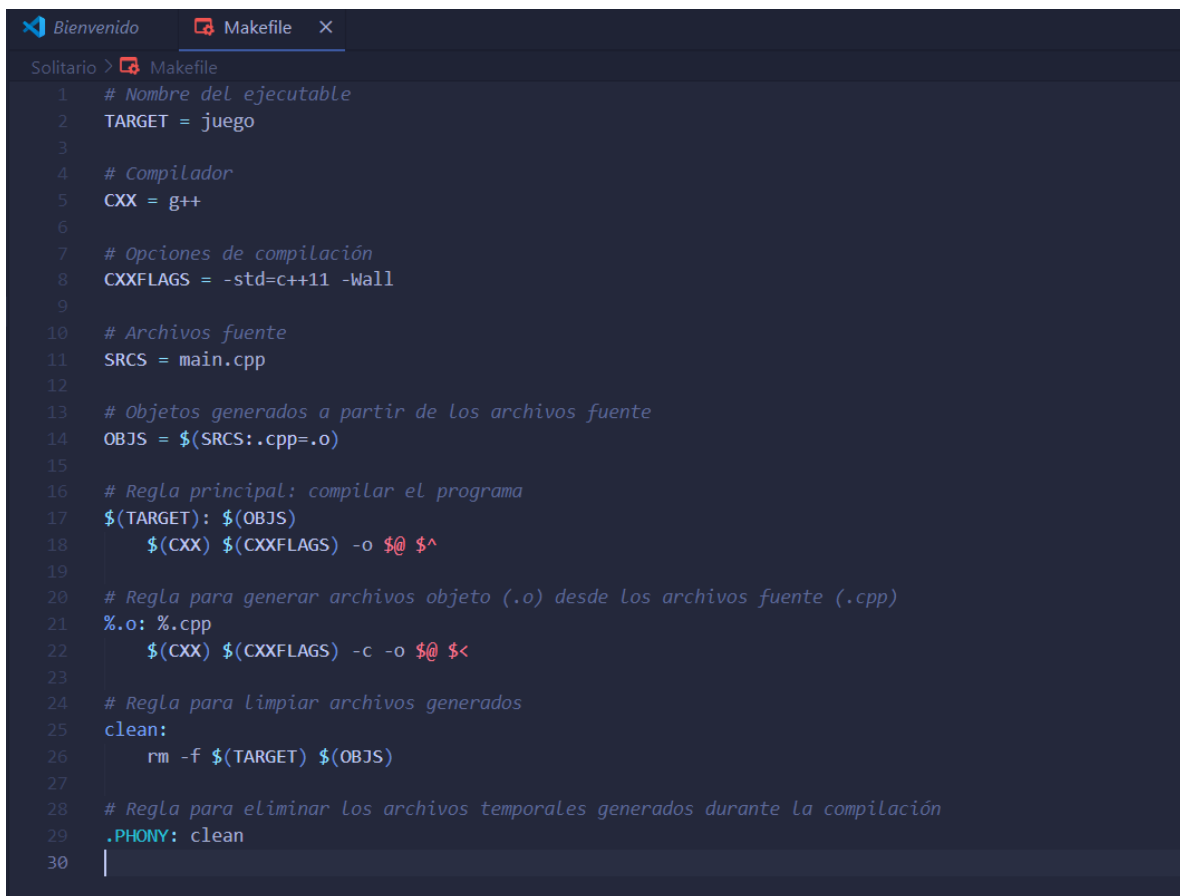
Abre una terminal.

Navega al directorio que contiene tu Makefile y el archivo .cpp que deseas compilar.

Ejecuta el comando make. Esto buscará automáticamente el archivo Makefile en el directorio actual y compilará el programa según las reglas definidas en el Makefile.

Una vez completada la compilación, se generará un archivo ejecutable con el nombre especificado en la variable TARGET del Makefile (en este caso, "mi_programa").

Si deseas limpiar los archivos compilados, puedes ejecutar el comando make clean, lo que eliminará el archivo ejecutable generado.



```
1  # Nombre del ejecutable
2  TARGET = juego
3
4  # Compilador
5  CXX = g++
6
7  # Opciones de compilación
8  CXXFLAGS = -std=c++11 -Wall
9
10 # Archivos fuente
11 SRCS = main.cpp
12
13 # Objetos generados a partir de los archivos fuente
14 OBJS = $(SRCS:.cpp=.o)
15
16 # Regla principal: compilar el programa
17 $(TARGET): $(OBJS)
18     $(CXX) $(CXXFLAGS) -o $@ $^
19
20 # Regla para generar archivos objeto (.o) desde los archivos fuente (.cpp)
21 %.o: %.cpp
22     $(CXX) $(CXXFLAGS) -c -o $@ $<
23
24 # Regla para limpiar archivos generados
25 clean:
26     rm -f $(TARGET) $(OBJS)
27
28 # Regla para eliminar los archivos temporales generados durante la compilación
29 .PHONY: clean
30 |
```

Especificaciones del juego

Datos del juego

xxxx representa las cartas volteadas

_____ representa el espacio donde van las cartas

---- representa espacio vacío

trabajaremos con las 52 cartas **A,2,3,4,5,6,7,8,9,10,J,Q,K**

así se verán las cartas en el tablero donde:

<3R (son corazones rojos)

<>R (son diamantes rojos)

E3N (son treboles negros)

!!N (son espadas negras)

Ejemplo

tres de corazones rojos

3<3R

10 de espadas negras

10!!N

Q de trebol negro

QE3N

J de diamantes rojos

J<>R

Columnas del tablero:

Bueno estas son las instrucciones de la filas que son lista doblemente enlazadas

columna1 guarda 7 cartas pero solo revela 1 y las demas se quedan volteadas

columna2 guarda 6 cartas pero solo revela 1 y las demas se quedan volteadas

columna3 guarda 5 cartas pero solo revela 1 y las demas se quedan volteadas

columna4 guarda 4 cartas pero solo revela 1 y las demas se quedan volteadas

columna5 guarda 3 cartas pero solo revela 1 y las demas se quedan volteadas

columna6 guarda 2 cartas pero solo revela 1 y la otra queda volteada

columna7 guarda 1 carta pero es revelada

Diagramas

