Hope this file helps you understand the code.

~Anandita

Problem Statement

Implementing a multi-player UNO game using C++.

Introduction

UNO is the classic card game of matching colours and numbers that is easy to pick up… impossible to put down and now comes with customisable Wild Cards for added excitement! Players take turns racing to get rid of all their cards by matching a card in their hand with the current card shown on top of the deck either by colour or number.

● play one card matching the discard in color, number, or symbol

● play a Wild card, or a playable Wild Draw Four card

● draw the top card from the deck, then play it if possible

● A player who draws from the deck must either play or keep that card and may play no other card from their hand on that turn.

● A player may play a Wild card at any time, even if that player has other playable cards.

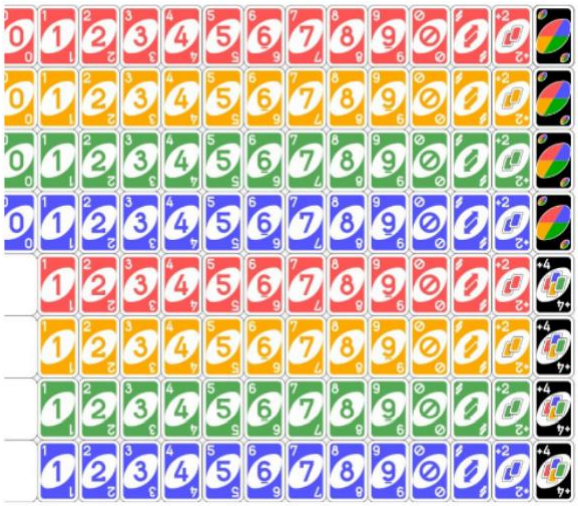
● A player may play a Wild Draw Four card only if that

player has no cards matching the current color. The player may have cards of a different color matching the current number or symbol or a Wild card and still play the Wild Draw Four card. A player who plays a Wild Draw Four may be challenged by the next player in sequence to prove that their hand meets this condition.

● If the entire deck is used during play, the game is stopped then and there

● It is illegal to trade cards of any sort with another player.

Well we all have played UNO at least once in our lives. But the idea to implement the game in the form of code seemed interesting. In the project, we tried to cover all the possible moves in the game so that the user gets the real feel to play UNO.



List of Data Structures used in the project

1.Linked list

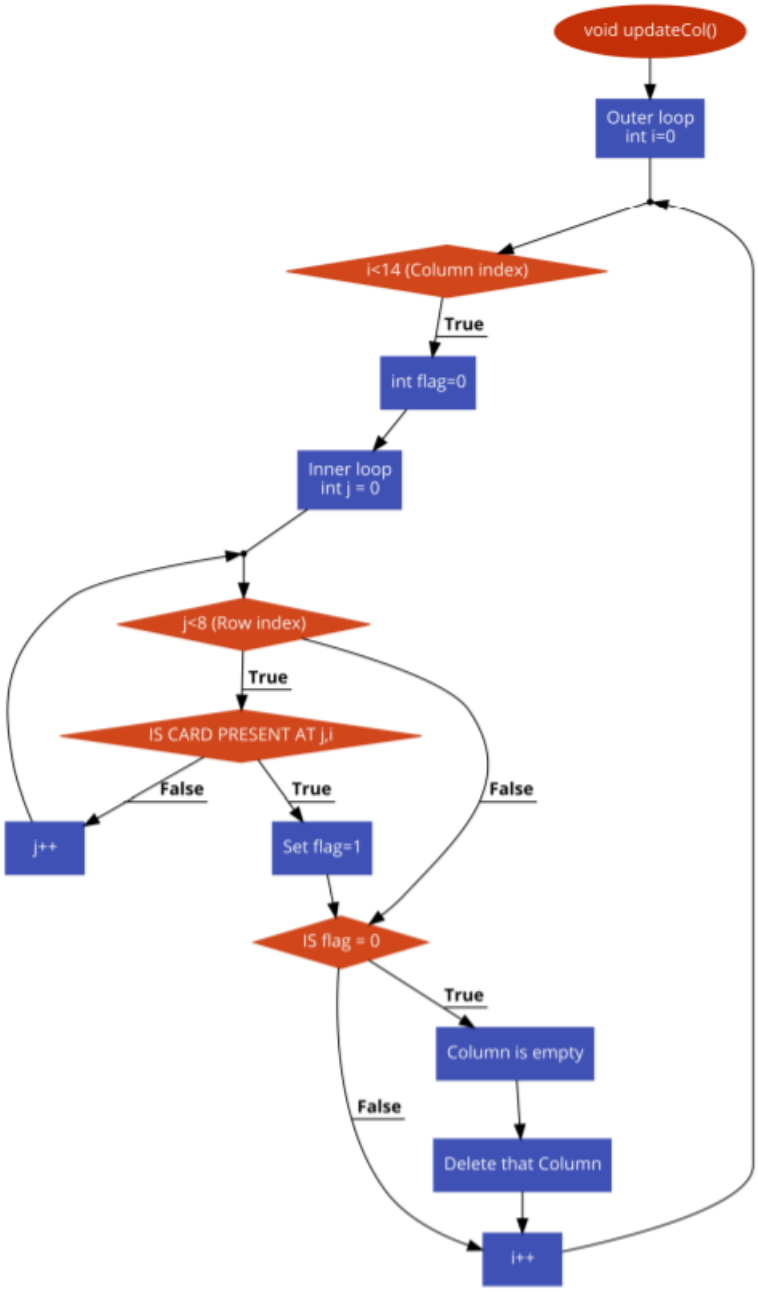
2. Vector

3. Double dimension array

Detailed Design of the project

Void updateCol ()

After a certain amount of draws, there is a good chance that some column have been emptied in the 2D Deck matrix. The purpose of this function is to delete such column, significantly reducing the time taken in selecting random cards.



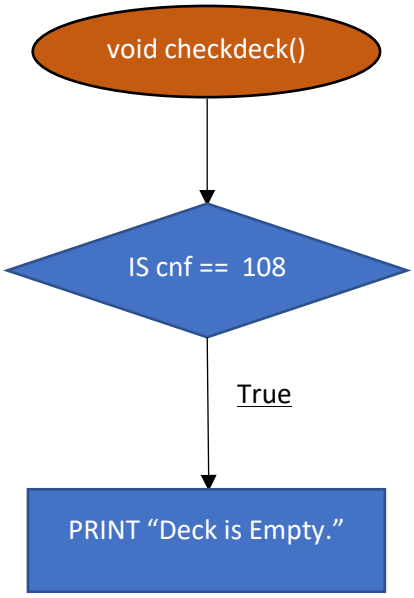
Void updateRow()

After a certain amount of draws, there is a good chance that some rows have been emptied in the 2D Deck matrix. The purpose of this function is to delete such rows, significantly reducing the time taken in selecting random cards.

Similar to function updateCol ()

Void checkdeck ()

The function checks whether deck is empty or not

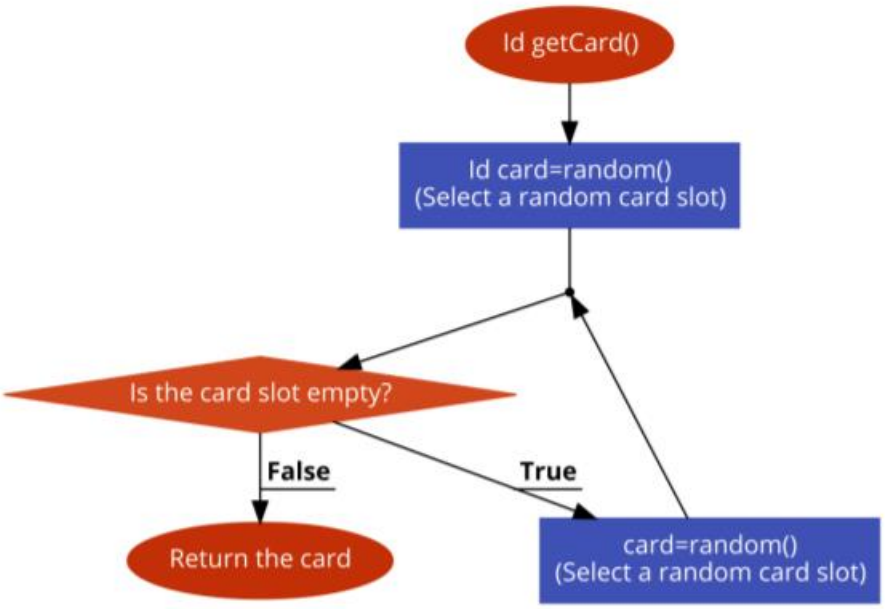


Id Random ()

This function generates the indices of a card slot, with the help of the rand () function, thus returning a random card from the deck. It is used while dealing hands, drawing a card etc.

Id getcard()

If random() returns a card that has already been drawn from the deck, getCard() calls it again until it returns an undrawn card, rectifying any possible errors.



Class Hand

The class hand declares a linked list which helps in taking of care of the cards drawn.

Linked List helps in insertion and deletion of cards in hand per player and can dynamically increase or decrease its size.

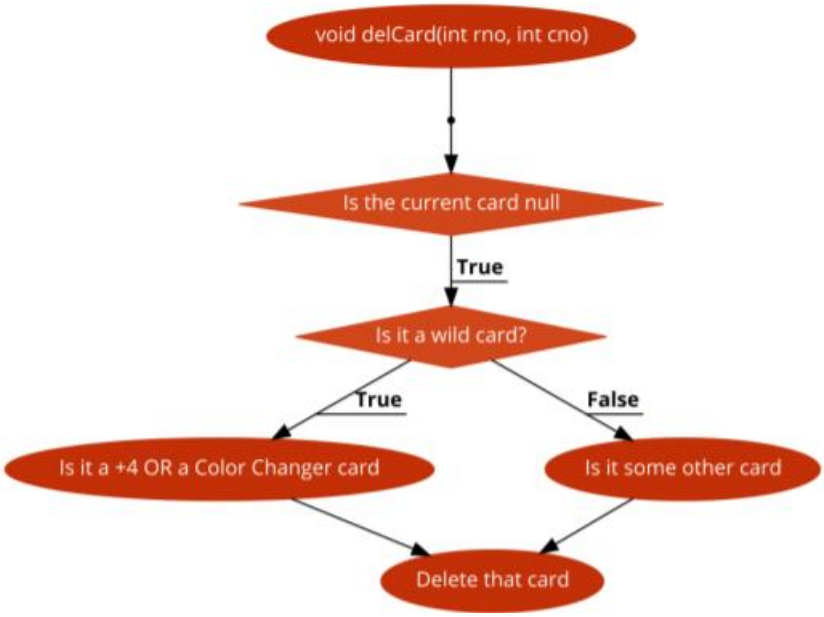
Functions inside the class

void addCard()

This function does the job of generating a players's hand by adding a random card called using getCard() in the end of linked list corresponding to that player.

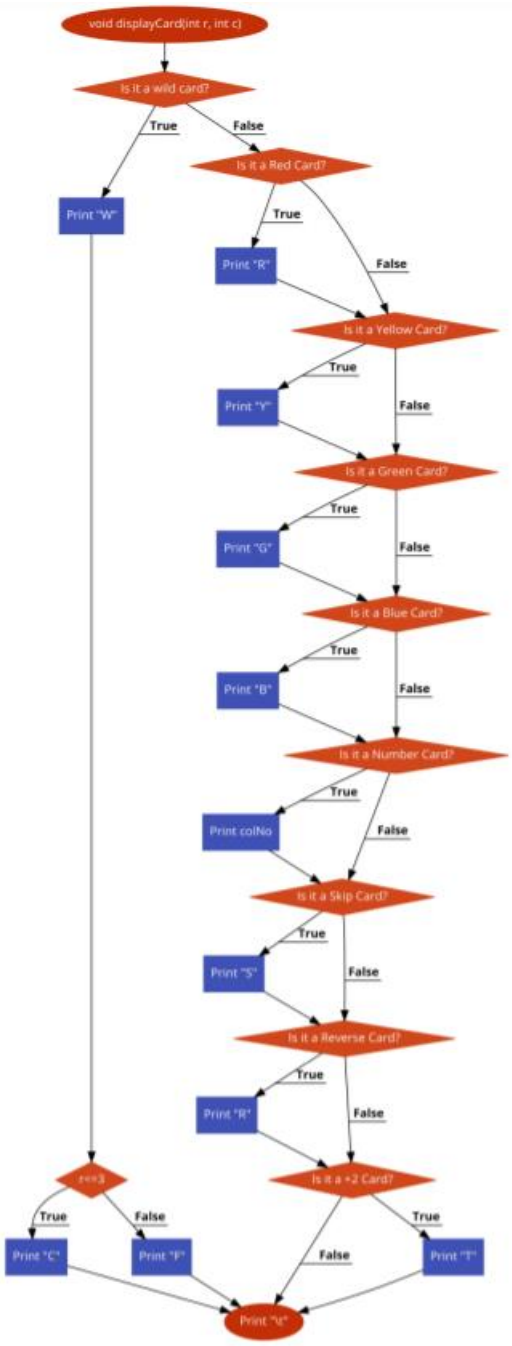
void delCard ()

Whenever a card is played by a player, it has to be deleted from that player's hand. This operation is done by delCard ().



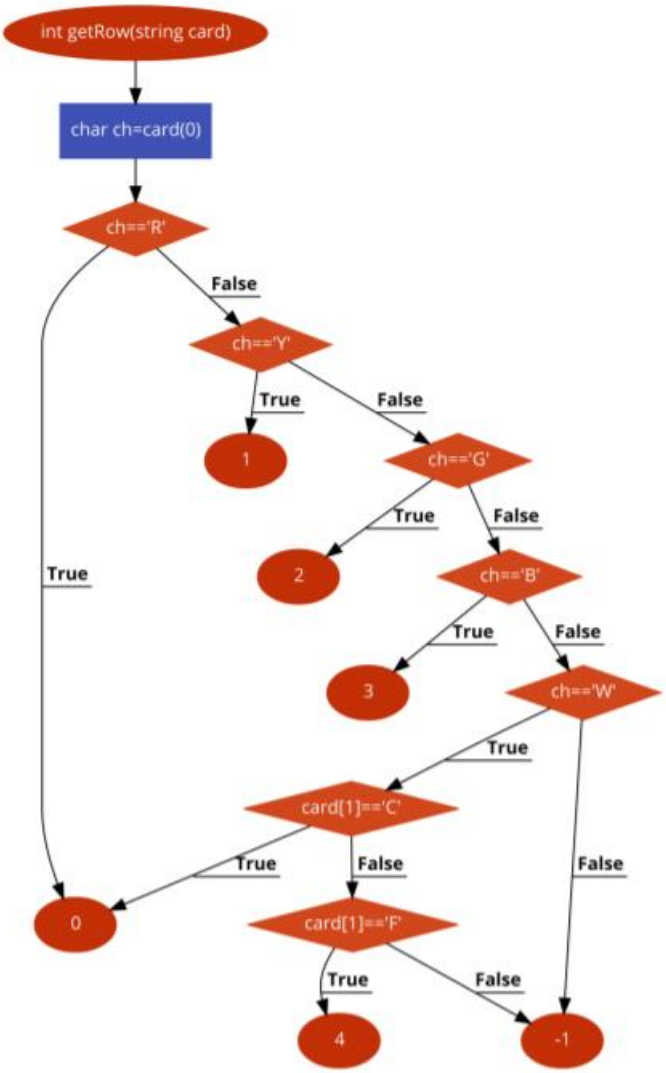
void displayCard ()

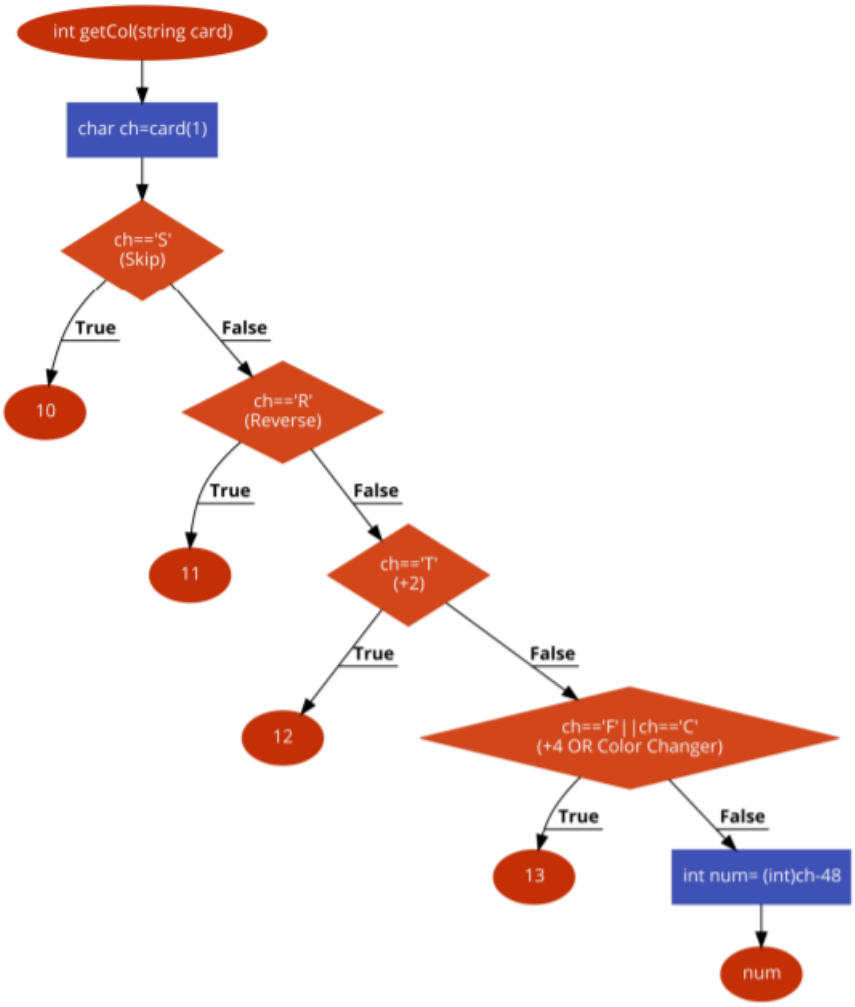
All the cards are stored using their row and column indices. To make it understandable to the player displayCard() is used. This function is called whenever a card is to be displayed in String form.


void displayCard(int r, int c)
Is it a wild card?
True
False
Print "W"
Is it a Red Card?
True
Print "R"
False
Is it a Yellow Card?
True
Print "Y"
False
Is it a Green Card?
True
Print "G"
False
Is it a Blue Card?
True
Print "B"
False
Is it a Number Card?
True
Print colNo
False
Is it a Skip Card?
True
Print "S"
False
Is it a Reverse Card?
True
Print "R"
False
Is it a +2 Card?
True
Print "T"
False
r==3
True
Print "C"
False
Print "F"
Print "c"

int getRow() and int getCol()

These functions have the opposite use of displayCard(). They convert user entered card choice in string form to corresponding row and column indices.




int getCol(string card)
char ch=card(1)
ch=='S'
(Skip)
True
False
10
ch=='R'
(Reverse)
True
False
11
ch=='T'
(+2)
True
False
12
ch=='F'||ch=='C'
(+4 OR Color Changer)
True
False
13
int num= (int)ch-48
num

Bool checkCard ()

For a card to be playable, it should have atleast one matching attribute with the last played card i.e. the top card(except in the case of a wild card, which has no restrictions) . bool checkCard() is used to verify the same.