Programming Fundamentals Project

Let's Play UNO!!

CLO 3 – Deadline Dec 1, 2023

1. Make sure that you read and understand each instruction. If you have any questions or comments you are encouraged to discuss your problems with your colleagues (and instructors) on google classroom.

2. If there is a syntax error in the code, zero marks will be awarded in that part of the project.

3. Keep a backup of your work always that will be helpful in preventing any mishap and avoid last hour submissions

4. Displayed output should be well mannered and well presented. Use appropriate comments and indentation in your source code.

5. Please ensure that only concepts covered in class are used for the project. Filing is a self learning concept that you will use in the project.

7. Only submit your project .cpp file. File name should be student1_RollNo-student2_RollNo

8. The student is solely responsible for checking the final file for issues like corrupt file, viruses in the file, or mistakenly exe sent. If we cannot download the file from Google Classroom, it will lead to zero marks in the project.

Note: Start early so that you can finish it on time.

Let’s Play UNO!!

UNO is a popular card game, typically for 2-10 players, it uses a specialized deck of cards and is based on the shedding-type gameplay mechanics.

Objective:

The goal is achieved by being the first to play all your cards.

Gameplay:

- Each player starts with **7 cards**.
- A deck consists of **coloured cards** (red, blue, green, yellow), each with numbers 0-9 and action cards like *Skip*, *Reverse*, *Draw Two*, *Wild*, and *Wild Draw Four*.
- Players take turns matching the top card of the discard pile by **colour** or **number**.
- Action cards introduce strategic elements, such as forcing opponents to draw cards or changing the color.
- Play the game on the following link to understand the game: **https://www.crazygames.com/game/uno-online**

Project Statement

This project requires developing a console-based, 2-player UNO game using C++. The game will simulate a real-time UNO experience between two players, following the standard rules of UNO in a turn-based format. The game interface is text-based, designed to be simple and intuitive for players to interact with.

Game Design and Requirements

1. **Game Components**:
   - **UNO Deck**: The deck should have 108 cards with standard UNO colours (Red, Blue, Green, Yellow) and types (numbered cards, Skip, Reverse, Draw Two, Wild, and Wild Draw Four). Each card will be uniquely identifiable by its colour and type.
     - **In one colour, let’s say blue we have**
       1. 19 number cards (1 zero and 2 of each number up to 9)
       2. 2 reverse cards.
       3. 2 skip cards.
       4. 2 draw 2 cards.
          Other than these, overall, we have
       5. 4 wild draw 4 cards.
       6. 4 wild cards.
   - **2D Array Representation**: Use a 2D array to represent the deck of cards, where rows indicate colours, and columns represent types.
   - **Player Hands**: Each player will have a "hand" represented by an array, with a maximum of seven cards initially (Can be more!!).
2. **Core Functionalities**:
   - **Deck Initialization and Shuffling**:

    - Create and populate the deck, ensuring that each colour and type is included in the correct quantities.
    - Implement a shuffling function to randomize the deck order before dealing cards.
  - **Card Dealing**:
    - Deal seven cards to each player at the start of the game.
    - Each player's hand should be updated and displayed after every turn.
  - **Turn-Based Play**:
    - Alternate turns between Player 1 and Player 2.
    - Each player should be able to see the top card in the discard pile and their hand to decide which card to play.
    - If a player cannot play any card, they must draw a card.
  - **UNO Call**
    - Allow the current player to declare "UNO."
    - A function to let the opponent catch the missing "UNO" call and impose a penalty. Add two cards to the current player's hand if they fail to call "UNO" in time.
  - **Card Play Validation**:
    - Enforce card play rules, ensuring a card can only be played if it matches the color or type of the top discard pile card.
    - Implement logic to handle special cards, such as Skip, Reverse, Draw Two, Wild, and Wild Draw Four, with their respective effects on gameplay.
  - **Special Card Functionality**:
    - **Skip**: The next player's turn is skipped.
    - **Reverse**: In a two-player game, Reverse acts as a Skip, so the current player takes another turn.
    - **Draw Two**: Forces the next player to draw two cards and skip their turn.
    - **Wild**: The current player can select any colour for the next card to match.
    - **Wild Draw Four**: Forces the next player to draw four cards and allows the player to choose the colour for the next card to match.
  - **Win Condition and Endgame**:
    - Implement a win condition where a player wins by playing all their cards.
    - Display a congratulatory message for the winner and save the game result in a file.
  - **Save Game History**:
    - Log each game, including the final result and the winner, into a text file.

        - Include details such as moves played, final scores, and the number of turns.

3. **User Interface and Interaction**:
    - **Hand Display**: Display each player's hand with their options (Play Card, Draw Card).
    - **Discard Pile**: Show the top card of the discard pile after each turn.
    - **Action Prompts**: Prov ide prompts to guide players in each action (e.g., "Play a card or draw one").

Functions and Code Structure

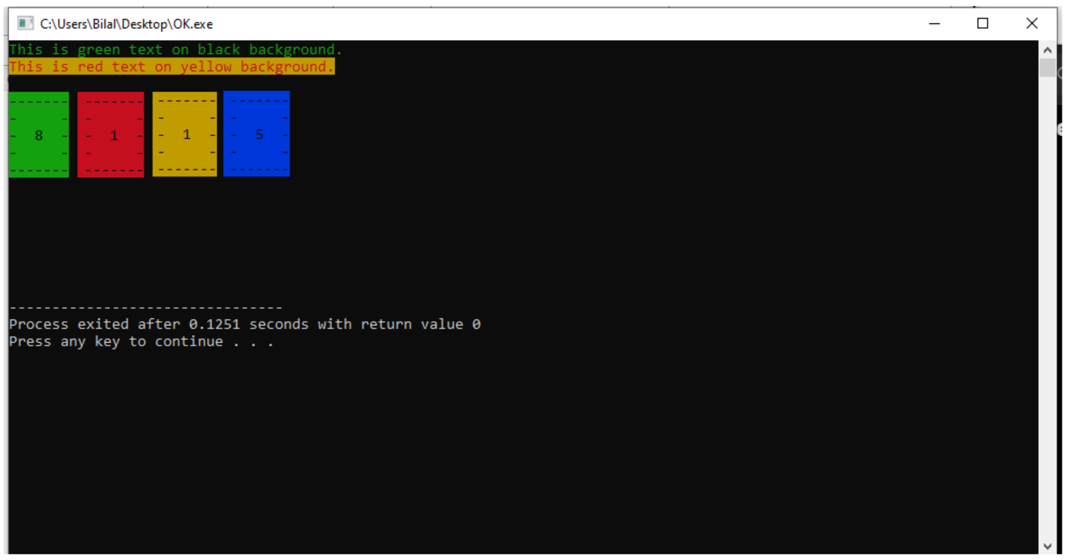
The following functions will structure the code for modular and efficient gameplay:

1. **void initializeDeck(int deck[][15]);**: Initialize the deck with all UNO cards.
2. **void shuffleDeck(int deck[][15]);**: Shuffle the deck to randomize the card order.
3. **void dealCards(//add your own arguments);**: Deal cards to players from the deck.
4. **bool isValidPlay(int playerCard, int topCard);**: Validate if a player's card can be played.
5. **void playTurn(//add your own arguments);**: Manage a single turn, including drawing or playing a card.
6. **void handleSpecialCard(int specialCardType);**: Handle the special effects of cards like Skip, Reverse, etc.
7. **void saveLoadGameResult((//add your own arguments);**: Save game history and result to a file and in the start load the history. (No of games won by player 1 and player 2)
8. **bool callUno(//add your argument);**: Before selecting second last card, if user inputs 'U', Uno is called. If player directly selects the second last card, add two cards as plenty. Uno function cannot be called before 2nd last card.
9. **void UpdateDiscardPile(//add your own arguments);**
10. vo**id PrintPlayerHand(//add your own arguments);** show all player cards on consoles

Student 1	Student 2
void initializeDeck	**void shuffleDeck**
void dealCards	**bool isValidPlay**
void handleSpecialCard	**void playTurn**
Void PrintBoard	**void UpdateDiscardPile**
void saveLoadGameResult	**bool callUno**

Note: Other than these, you can make more helping function to make your implementation easier.

The board should look as follows



C:\Users\Bilal\Desktop\OK.exe

```
This is green text on black background.
This is red text on yellow background.

 -------   -------   -------   -------
 -     -   -     -   -     -   -     -
 -  8  -   -  1  -   -  1  -   -  5  -
 -     -   -     -   -     -   -     -
 -------   -------   -------   -------



--------------------------------
Process exited after 0.1251 seconds with return value 0
Press any key to continue . . .
```

Game Flow is as follows:

- Main screen: Provides a welcoming screen for UNO. Player can select play option or see the score.
- If play is selected, game is displayed, at any time, user and enter ‘E’ to Exit the game. If user enter E, confirm if they want to go back to main menu. If Y is pressed, go to main menu, else if N is pressed, continue the game.
- Program should only exit if user selects exit option.

Note:

A sample help file is provided with following functions.

1. Change colours
2. Print Card
3. Clear Screen

😊 **Good Luck!!**