Project 1

Red-7

CIS-5 (43358) Name: Jovanny Alarcon

Date: 06/08/2021

Introduction

Title: Red7 Card Game

<https://www.ultraboardgames.com/red7/game-rules.php>

Game Play

The rule for winning at Red is simple: Have the best card! But will you still be playing the same game when your turn ends? If you're not winning by the current game's rule at the end of your turn, you're out, and the last person standing wins the round.

The deck consists of 49 cards numbered 1-7, in each of the seven rainbow colors. A 7 is always higher than a 6, but a Red 6 is higher than an Orange 6, and so on down the color spectrum (Red, Orange, Yellow, Green, Blue, Indigo, Violet). When comparing two cards, compare value first, then color.

To begin a game of Red7, deal out a seven-card hand to each player, and then deal one more card faceup in front of each player to start their Palette.

Start the Canvas (discard pile) with the You are Playing Red card. The top card of the Canvas pile determines what the rules are. The player with the highest card on their Palette is currently the best at Red, so the player to that player's left goes first.

Game Turn

On your turn, you must take one of the following actions:

1. Play a card faceup from your hand to your Palette.
2. Discard a card from your hand to the Canvas to change the game to the rule of the discarded card's color. You must be winning the new game after you do this.
3. Play a card from your hand to your Palette AND THEN discard a card to the Canvas. You must be winning the game after you do this.
4. Do nothing, and lose. You might want to do this intentionally when playing Advanced Red7 to limit the number of points an opponent will score. If your hand is empty, you must do this.

If you are not winning the game at the end of your turn, you lose and are out of the round. Place your hand cards and Palette cards facedown. If you are ever the last player in the game, you win the round!

You are currently winning a game if your Palette contains more cards that meet the current game's rule than any other player. To break a tie, look at each tied player's highest card that follows the rule, checking its value, then its color if necessary.

If you have no cards (for Green or Violet) that follow the rule, you are not winning. Examples are to the right for each rule. If you have no cards in your hand at the start of your turn, you lose since you are unable to do anything to win by the end of your turn.

**My Implementation:** This game version is only designed for 2 players. The game will start off by displaying who will go first based on the turn order designated by the rules. The program will display Player 1’s hand and Player 2’s hand. It will also display the 1st card for each players of their palette. It will tell which user to play first and will ask the user if they will play onto their palette or onto the canvas. If they play onto the canvas, they have the option to then play on the palette. After playing on the palette the player must tell the program with a ‘y’ that their turn is over. The program will check the conditions of the rules (Red, Orange, Yellow, Green, Blue, Indigo, or Violet) and show which player won that round. The Canvas rule will determine the conditions of winning that round. The program will then prompt the user to repeat this process. If yes it will be the next player’s turn. If not, that player loses and the other players is declared the winner.

**Summary:**

Project Size: About 800 lines

Number of variables: Over 100+

This project covers the first 5 chapters of the Gaddis textbook. Concepts covered are primarily data types, if-statements, switch statements, formatting output, logical operators, incrementors, etc. I tried to cover as many as I could. The program was incredibly repetitive, and it cemented the idea that arrays can cut down on so much of the labor of typing the code. I do wish it was cleaner and a bit more optimized, but I am looking forward to seeing how it is going to look with functions and arrays. There were still some logical debugging that needed to be completed but the majority of the code was compiled.

**Updated for Project 2:**

The program now runs for all game modes. In the last implementation the rule for Indigo was not running but not it is fully functional. There are 21 separate functions and the main function. All listed variables were added to the program and the program also checks whether the player inputted a card from their hand.

**Shuffle Function**

Previously I used the rand function to generate random numbers, now I created a vector and shuffled those numbers around 10^7 times.

**uniRand Function**

This creates a vector of size 16 that will hold values for the next function that uses a switch statement to initialize the players hands.

**strtHnd Function**

A switch statement nested in a for-Loop is used to initialize the players hands.

**firstPly Function**

Function determines which player goes first bases on random card designated on their Palette

**show Function**

Overloaded function that shows the players hands. Uses different arguments from the following function.

**show Function**

Overloaded function shows the players Palette

**plyrTrn Function**

**Description:**

The point of the game is to play 1 or 2 cards per turn and be winning by the end of it. The user must be careful in entering the input because the code is not completely validated for errors.

Did not finish in time…