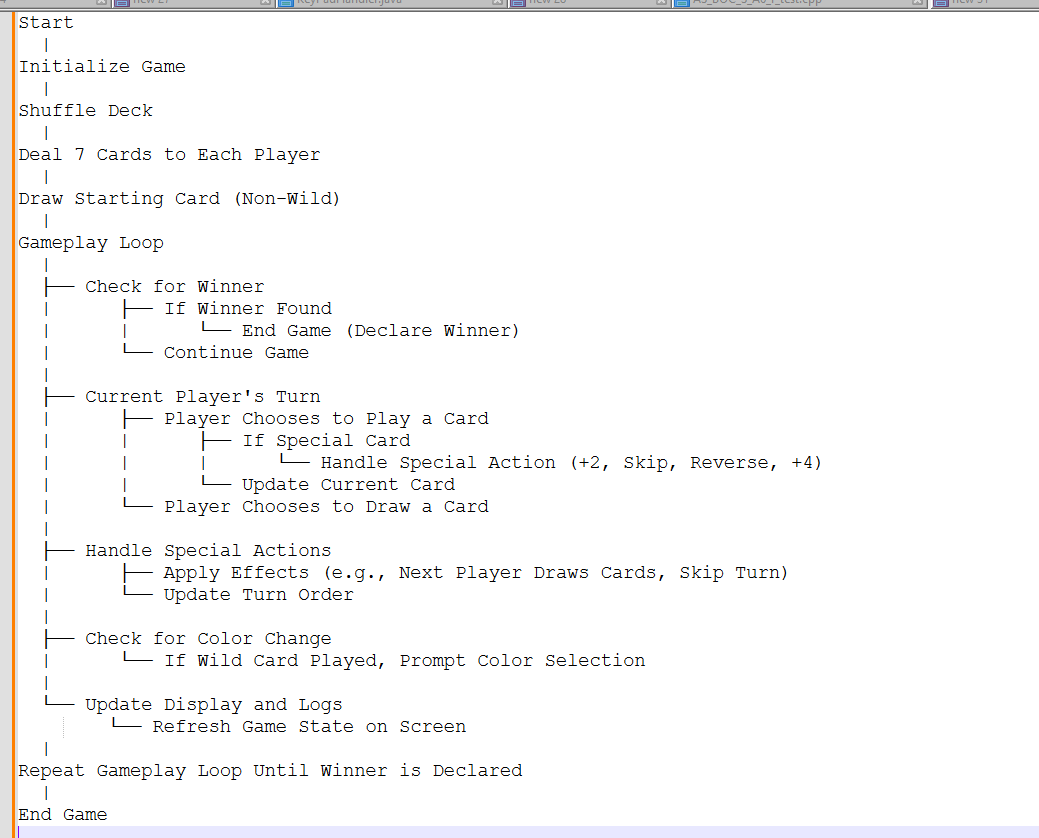
Flow of logic :  
  


To install the required libraries, open your terminal (Mac) or Command Prompt (Windows) and run the following commands:

pip install --upgrade   
pip install cmu\_graphics

If you are using Python 3 and the above commands don't work, try:

pip3 install --upgrade pip   
pip3 install cmu\_graphics

**About the game :**

This Python program implements the classic card game UNO using the cmu\_graphics library. The game features two players and includes all standard UNO functionalities such as drawing cards, playing cards, handling special actions like skip, reverse, and draw two/four, and determining the winner. The game interface is graphical, allowing players to interact with the game through mouse clicks.

The game defines several constants to represent the different attributes of UNO cards. CARD\_COLORS includes the standard UNO colors: DeepSkyBlue, ForestGreen, Tomato, and SlateGray. CARD\_NUMBERS is a list of integers from 0 to 9, representing the numerical cards. SPECIAL\_CARDS includes action cards like "+2", "skip", and "reverse". These constants are used to create and manage the deck of cards throughout the game.

The Card class represents an individual UNO card. Each card has a color and a value. The isPlayable method determines if the card can be played based on the current card in play by checking if the colors match or if the values are the same. The draw method visually renders the card on the screen at specified coordinates, with options to highlight the card or display it in a larger size. Special handling is included for wild cards and action cards to ensure they are displayed correctly.

The Player class models a game participant. Each player has a name and a hand, which is a list of Card objects. The drawCard method allows the player to draw a card from the deck, adding it to their hand. The playCard method enables the player to play a card from their hand by removing it from the list based on the provided index.

The UnoGame class encapsulates the core game logic. It initializes the deck by creating and shuffling cards, sets up the players, and manages the current state of the game, including the current card in play, the current player, and the game’s message log. The class includes methods to handle player turns, manage special card actions (such as skipping a player or reversing the turn order), and update the game state based on player actions. It also manages graphical elements, such as drawing the game interface, handling user interactions through mouse clicks, and resetting the game when necessary.

**App Functions**

The program includes several functions that integrate with the cmu\_graphics library to manage the application's lifecycle:

* onAppStart(app): Initializes the application window with specified dimensions.
* redrawAll(app): Continuously updates and renders the game interface, including the current state of the deck, players' hands, and any active game messages or popups.
* onMousePress(app, x, y): Handles mouse click events, allowing players to interact with the game by playing cards, drawing from the deck, selecting colors for wild cards, and resetting the game when a player wins.
* onStep(app): Updates the game countdown and manages delays between player turns to ensure smooth gameplay.

The runApp function at the end of the script launches the UNO game application with the defined width and height, enabling the graphical interface and interactive gameplay.