* 2 players enter their credentials and authenticate
* There is a deck of 30 cards, 10 of each colour each card is numbered 1-10
* Player 1 takes the top card and player 2 takes the next one
* Which ever player wins (see win table) takes both cards
* Keep playing until all cards are used.
* Store winner and no. of cards in external file
* Display a leader board of top 5.

|  |  |  |
| --- | --- | --- |
| Card1 | Card2 | Winner |
| Red | Black | Red |
| Yellow | Red | Yellow |
| Black | Yellow | Black |

Basic logic for gameplay loop:

Diagram

Description automatically generated

Text Based Program ------------------------------------------------------------------------------------ Documentation.

Functions, Classes, and global variables:

key\_gen(): generates a key for encrypting and decrypting sensitive data.

decrypt(message): decrypts a message using the key generated by key\_gen().

encrypt(message): encrypts a message using the key generated by key\_gen().

authorize(): prompts the user to enter a username and password, and returns the username if it is valid, or False otherwise.

Player(name): a class representing a player, with attributes for the player's name, score, and cards.

sign\_in(): prompts two users to sign in and creates Player objects for each user.

reset\_players(): resets the score and cards for each player.

win\_cond(): checks which player has the higher score, prints a message announcing the winner, and updates the leader board in the database.

randomize\_deck(): generates a deck of cards, shuffles them, and stores them in the database.

save\_game(): saves the current state of the database to a CSV file.

DataBase: a dictionary representing the database, with keys for the players, leader board, colours, number of cards, and keywords.

Player\_one: a Player object representing the first player.

Player\_two: a Player object representing the second player.

checker(player): takes a player class argument and updates its attributes whilst displaying that they won and updating the deck in the database to remove the two cards.

Play(): a function that is repeatedly called (by recursion) and prompts the user to play, if no more cards calls wind\_cond()

Main loop:

1st check is if the number of profiles in database is >= 2 if it is not then prompting the user to create more profiles and then skips the rest of the loop.

2nd check is if any profile has been logged in (due to the code, cannot have just one person signed in either 2 or none) if none are signed in then prompts the user to sign in

Then takes menu input.

1. New Game:
   1. Randomizes the deck.
   2. Resets the player objects.
   3. And starts the gameplay loop.
2. Leader Board:
   1. Displays a leader board of all profiles.
3. Change User:
   1. Prompts the user to re-sign in.
4. Save Game:
   1. Saves Data Base dictionary to DataBase.CSV.
5. Create New Player:
   1. Takes username.
   2. Takes password.
   3. Creates new profile in Data Base.
6. Delete Profile:
   1. Confirms they would like to delete a profile.
   2. Displays all profile names.
   3. Takes username.
   4. Takes password.
   5. Removes profile from data base.
   6. Prompts users to re-sign in.
7. Exit:
   1. Breaks from main loop.
   2. Ends program.

Code is in project file.

GUI based program ------------------------------------------------------------------------------------- Documentation.

Functions and classes

In order of appearance

|  |  |
| --- | --- |
| Code: | Function / class: inputs/output |
| Key\_gen() | only output key |
| Decrypt(message) | input text to be decrypted and outputs decrypted text |
| Encrypt(message) | Input text to be encrypted and outputs encrypted text |
| Reset\_deck() | Only changes deck list to be full and shuffled |
| Player | Creates player class with scores and holding cards |
| ImpButtons | Creates rects and formats image based on image file |
| Card | Creates rectangle and text based on card input |
| MiniCard | Creates rectangle and text based on card input but smaller |
| Save() | Encrypts data and saves to external file |
| Sign\_in() | Signing takes user input of username and password and signs them in |
| Print\_cards() | Displays all cards in either players cards |
| End\_cond() | Displays all cards of winner and displays who won |
| Play\_game() | Allows users to play the game by clicking 2 buttons |
| Leaderboard() | Displays a list of top 5 profiles in external file |
| Edit\_profiles() | Allows user to create or delete profiles if they have access to username and password |
| Main\_menu() | Allows user to pick between what they would like to do e.g. sign in or play |