**How to Run the Program:**

Need to use C++ 11 for compilation.

To compile please type the following in the command line: g++ game.cpp -std=c++11

To run the program please type: ./a.out

**Bug Report:**

No bugs found

**Feature Report**:

**Missing features**: No features missing

**Extra features**: No extra features

**Data Structures/classes**:

Card: Class that creates object with member variable suit and face. Represents a single card object that is used in a deck class.

Deck: Composed of vector of pointer to Card objects. Creates a deck of 52 unique cards and stores it in a vector of pointer to cards. Provides functionality to shuffle the cards that are present on the vector.

Game: Provides the options to start a new game or load a saved game. When a round ends, provides option to start another round. If user wants to quit the game, fetches the result of the game from player class and displays the winner.

Round: Represent a single round in a game. Contains logic that implements that determines turn of the player changes turn and provide option to play, serialize, ask for help for human player, and exit the game. Data structure used is vector of pointer to card and an array of players

Player: Represents a player in the game. Contains a virtual function play which is implemented by human and computer class. Also contains strategy for playing from the stockpile for human and computer. Provides all the necessary functionalities that are common to both human and computer class. Data structure used is vector of pointer to cards.

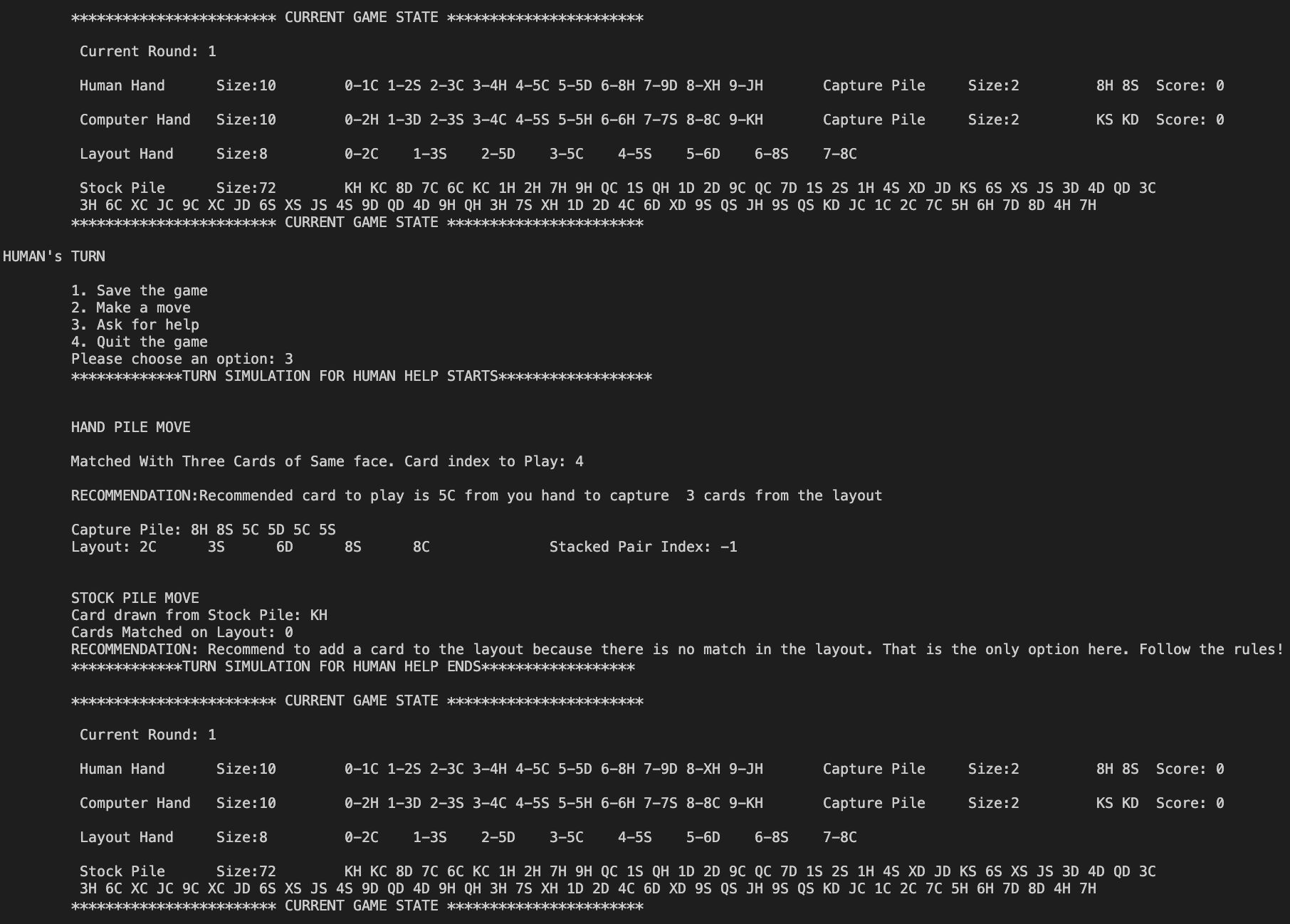
Human: Class that Inherits from the player class and implements the virtual function play which is in player class and also implements the strategy of play from hand for human player.

Computer: Inherits from the player class and implements the virtual function play which is in player class and also implements the strategy of play from hand for computer player.

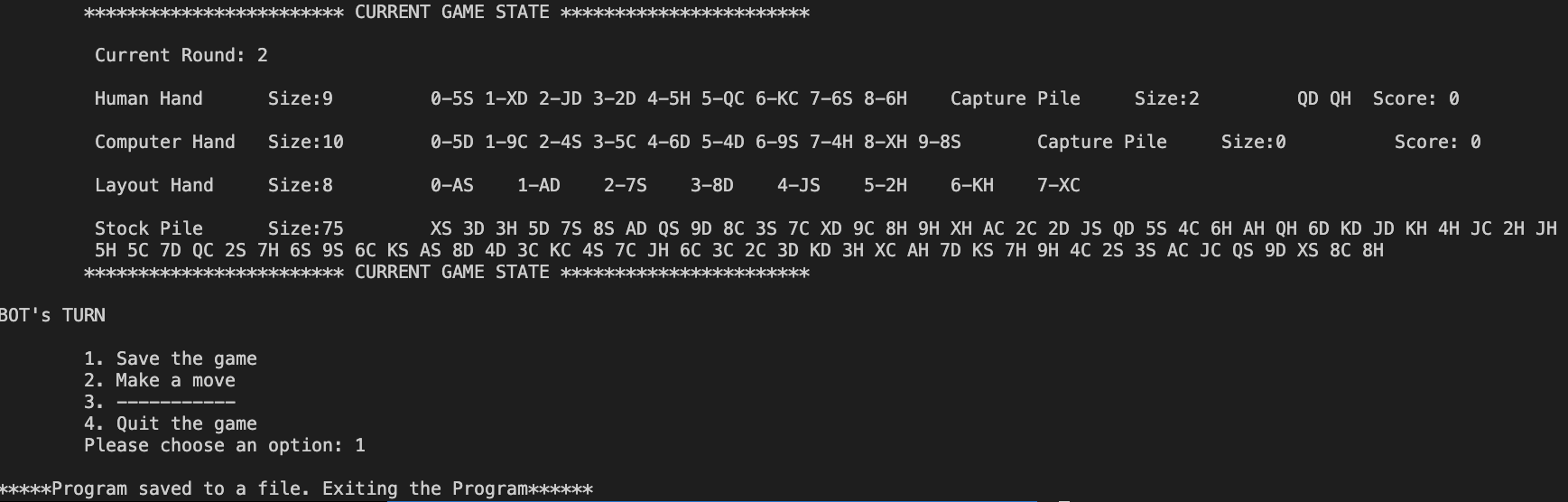
Serializer: Provides human user an option to serialize the state of the game at any point in the game to a text file and also provides an option to load a saved state of the game. Data structure used is vector of pointer to cards

**Screenshots:**

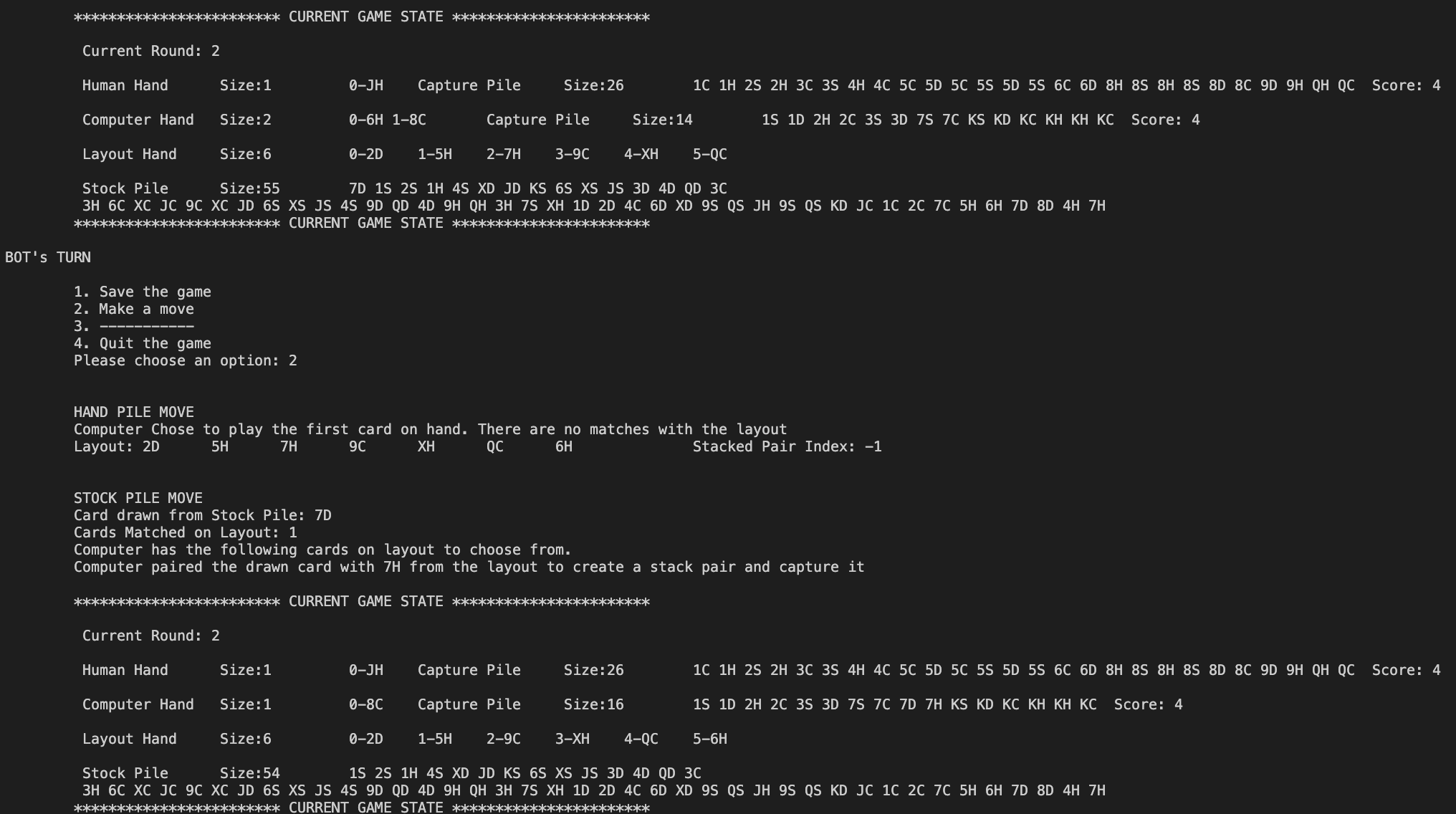
**Help**

****

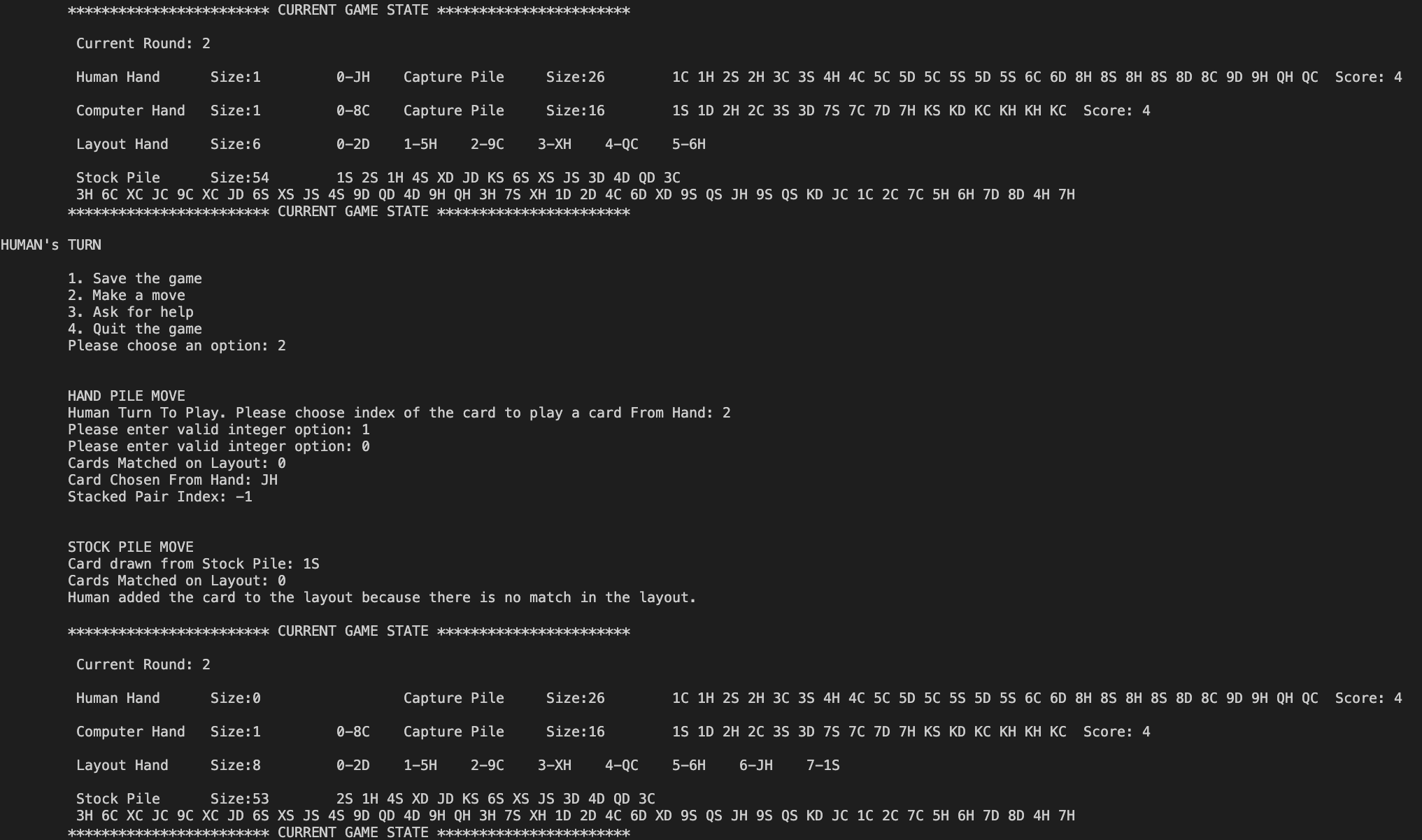
**Save Game**

****

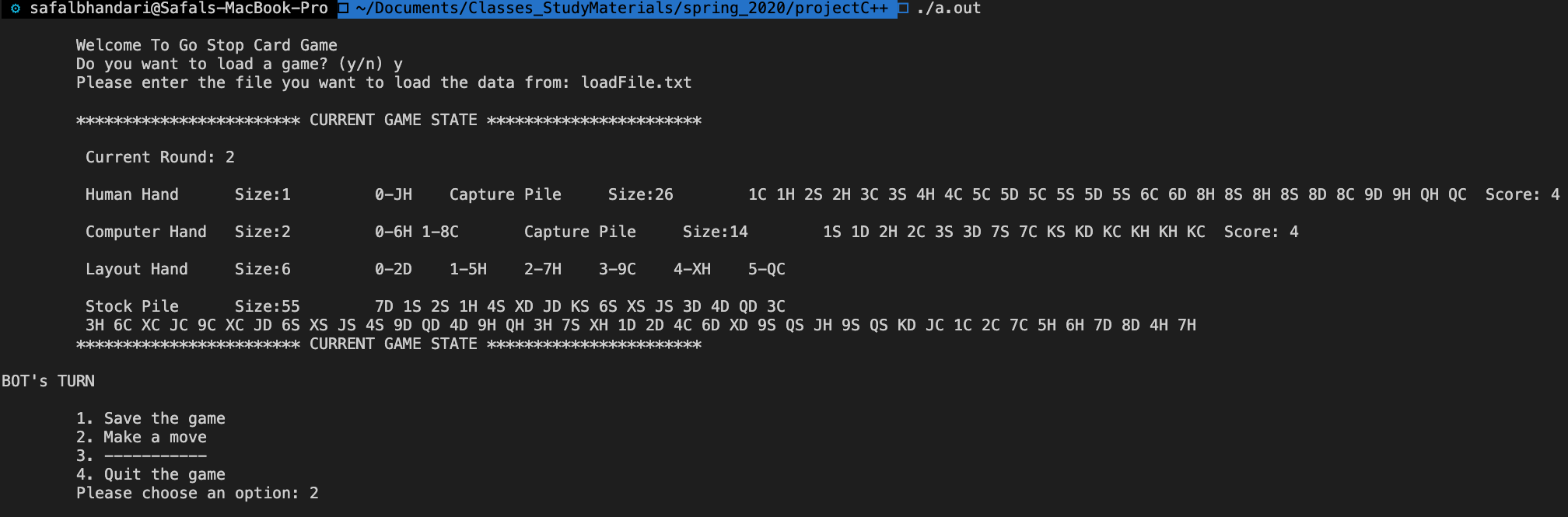
**Computer Move**

****

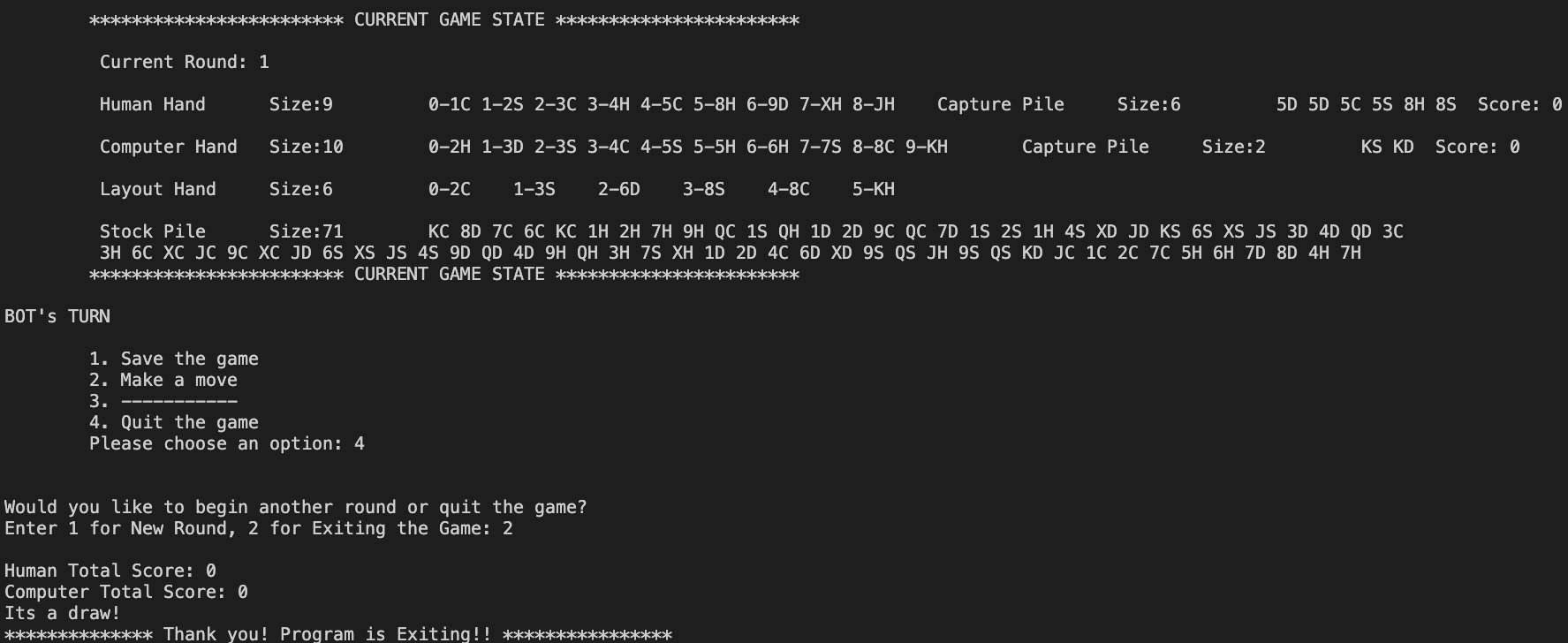
**Human Move**

****

**Load Game**

****

**Quit Game Screenshot**



**Log:**

Jan 26

Read the description of the project to get an understanding of what the program might look like and created a basic design on paper of what classes are needed for the program and how to use those classes.

1.5 hours spent

Jan 29

Wrote all the classes basic classes like Card, Deck, Player class basic functionalities, Human Class with no implementation, Computer Class with no implementation, Game Class basic functionalities, Round Class basic functionalities, and thought about how to link up classes and created a very simple prototype for basic program without any constraints of the actual program.

Checked all the classes using main in each class to make sure the classes are working fine before moving to other classes

5.5 hours spent

Feb 1

Implemented functionality for starting a new game in Round Class. Distributing Cards from Deck on Round to human hand, computer hand, layout and stockpile. Implemented function to determine the turn of the player in a new game

3 hours spent

Feb 3

Implemented strategy for play from hand for human player

Did not handle triple stack case

2 hours spent

Feb 4

Implemented strategy for play from hand for computer player

Did not handle triple stack case

Did not handle triple stack case rather distributed the triple stack to the layout as single cards and handled them as 3 separate cards

3.25 hours spent

Feb 5,6,7

Implemented strategy for play from stockpile for human and computer player

Did not handle triple stack case rather distributed the triple stack to the layout as single cards and handled them as 3 separate cards. Did this in 3 days

2+ 1.5 + .5 hours spent

Feb 11

Implemented logic for help option for the human player. Made use of computer class to recommend card to the player. Implemented functionalities such that help function made no changes to the state of the program but only recommended the best possible move to the human player making use of the computer class

1 hour spent

Feb 12

Implemented code to handle triple stack case and display it hyphenated and also implemented code to capture triple stack from hand of computer and human whenever possible. Removed code that distributed the triple stack to the layout as single cards

3 hours spent

Feb 16

Implemented the serialization class and implemented logic to upload the state of the game from a file and also implemented logic to save the state of the game that is being played to the file.

3.5 hours spent

March 2

Tested the code. Found out several bugs almost most of which had to do with vector out of index caused by erasing cards from the layout out and later trying to access the index which was previously saved before erasing certain cards from the layout which resulted in accessing wrong cards and sometimes resulted in index out of range error. Made necessary changes like first accessing the index before erasing cards which were captured to prevent the error

1 hour spent

March 3

Created understandable and readable display for the user by filtering out unnecessary lines which were mostly cout statements that were written for checking purposes and making sure program was right when working throughout the project. Only left the necessary outputs on the screen while playing the game so that it meets the projects specification for display and removed unnecessary checking output statements.

1.5 hours spent

March 9, 2020

Found an error while checking with serialization file. Found out the score was not properly calculated as it was only giving a point to player if there were 4 cards on the capture pile of the same face and did not gave any point if there were 6 or 8. Made necessary changes to display the correctly update the score after the end of each round. 25 mins spent