**HOW TO RUN THE PROGRAM**

Simply follow the prompts.

To select an option from the prompts, enter the numerical value associated with the desired option. For example, if I want to select option “2.) Build” I will type and enter “2”.

When prompted to select a card, please enter the card’s rank then its suit as they are displayed on the table and in your hand. For example, if I want to select the 10 of Spades, I would type and enter “XS”.

Whenever given an option to input, “-1” will always cancel the current operation at hand.

To resume game from a save file, select the option to continue from a save. Then when prompted to enter the safe file, enter the name of the save file followed by its “.txt” extension.

**PROJECT LOG**

**September 7th**

- Setup files for the project. (Main, card, deck classes)

**September 8th & 9th**

- Started working on the card class

- Card class holds suit and rank of a card

**September 11th**

- Started working on the deck class

- Generate a deck of cards using 52 card classes in a vector

- Created a shuffle function

- Created a display function so I knew the cards were being made properly

and that the shuffle function worked properly

**September 13th**

- Setup addition classes that I will and/or might need

- Started working on tournament, round, player, and derived classes

by filling them with information that I think they might need

**September 14th**

- Met with you in office hours to play Casino and better understand

the game and its mechanics

**September 17th**

- Added more to tournament and round classes

- Added more to table classes

**September 19th**

- Completed all online assessments

- Reorganized how classes and functions were being initialized and called

**September 20th**

- Added more to round, especially StartRound() function

- Deck now deals to both players

- Table reveals all cards

**September 21st**

- Added virtual play function

**September 23rd**

- Added functions to table class to check if player can build, capture, or trail

- Later scrapped this but might salvage it for AI

- Computer can now trail but can only trail

- Started working on build for human

- CheckBuild() now find all possible subsets of cards to make a build

**September 24th**

- Added function to check if player can make multi build

- Restructured build + user input so it is more simplified

- Salvaging code for AI potentially

**September 25th**

- Single Build is simplified and works

- ISSUE FOUND AND RESOLVED: does not want to build Aces for some reason

- Single build checks for builds and returns user to option menu if invalid attempt

- Made single build more reusable by breaking it down into functions

**October 2nd**

* Fixed an issue with serialization not recording builds properly
* Fixed small issue with AI explaining its decisions when playing

**October 4th**

* Fixed logic issue when checking if user can capture
* Fixed capture logic
* Fixed input validation for cards, yes or no, numbers

**October 10th**

* Reworked user input and data validation throughout the program

**October 20th**

* Fixed CPU AI so that it now builds properly.

**October 21st**

* Fixed multi-builds when loading game from save state
* Fixed logic for CPU builds that crashed the game

**MISSING FEATURES**

* No ability for player or CPU to extend build.
* No ability for the CPU to multi build.
* No ability to save the games current state, only read from save files

**BUG REPORT**

* When resuming game from save file, builds are loaded in but not able to be displayed
* Issue when capturing multi-builds as human player
* Last player does not get cards left on the table at the end of round for some reason.

**DESCRIPTION OF CLASSES**

// CASINO CLASS

// This class stores the entire instance of the game casino

// CARD CLASS

// Defines what a card is and maintains its attributes

// DECK CLASS

// This class is responsible for creating, shuffling, and dealing cards to players (managing the deck)

// PLAYER CLASS

// This class determines the properties of all players in the game as they are derived from this class

// HUMAN CLASS

// This class determines the actions of the human player by prompting the user what action

// they would like to perform throughout the game.

// COMPUTER CLASS

// This class is responsible for how the computer AI plays during its turn

// TOURNAMENT CLASS

// This class is responsible for holding the entire instance of the game

// TABLE CLASS

// This class controls everything that happens in the table's jurisdiction

// ROUND CLASS

// This class controls everything in the round’s jurisdiction, such as how the round starts, with what settings it starts, etc.

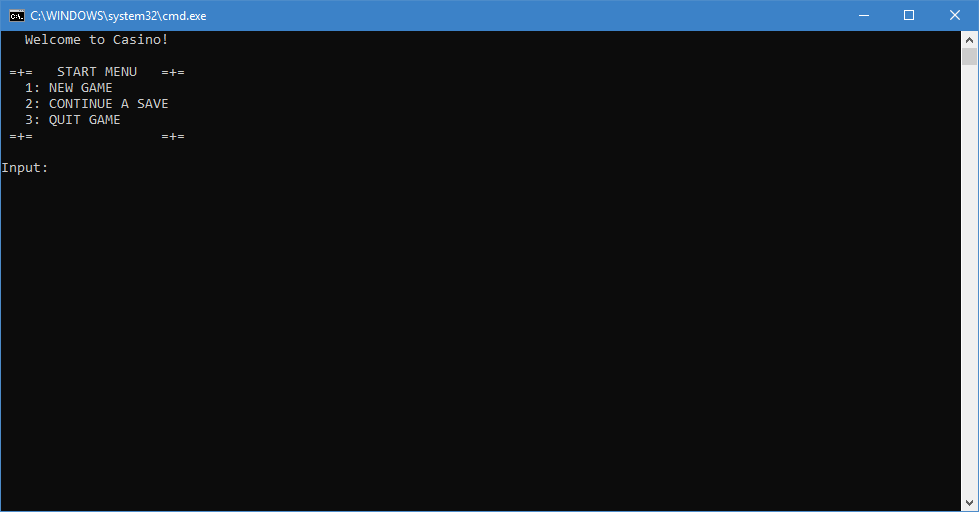
// BUILD CLASS

// This class was made to hold a build and the name of the player owning it

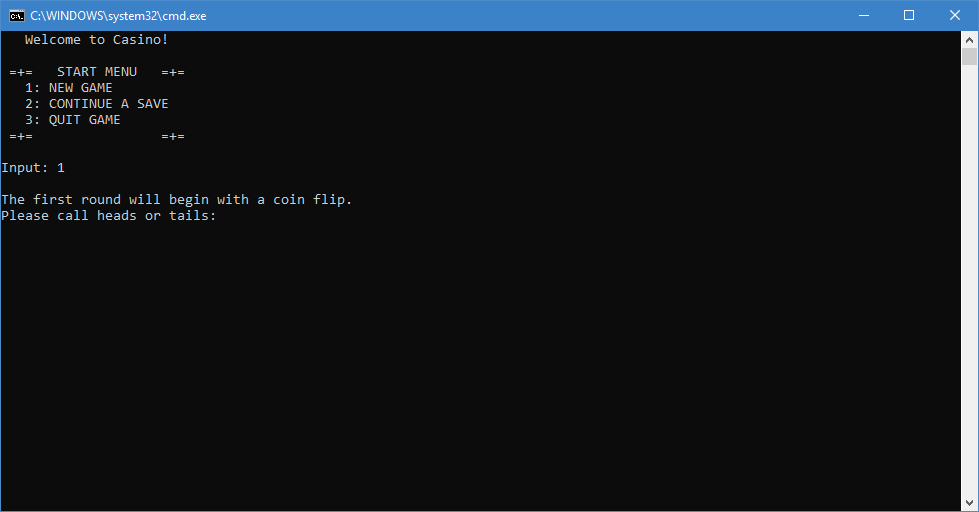
// SERIALIZATION CLASS

// This class was made to read and write to and from save files

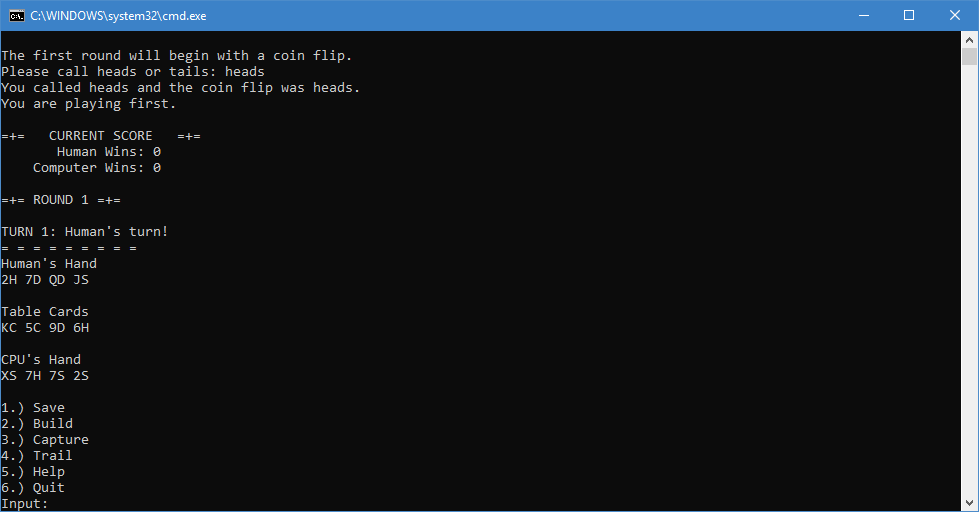
**SCREENSHOTS OF GAMEPLAY**



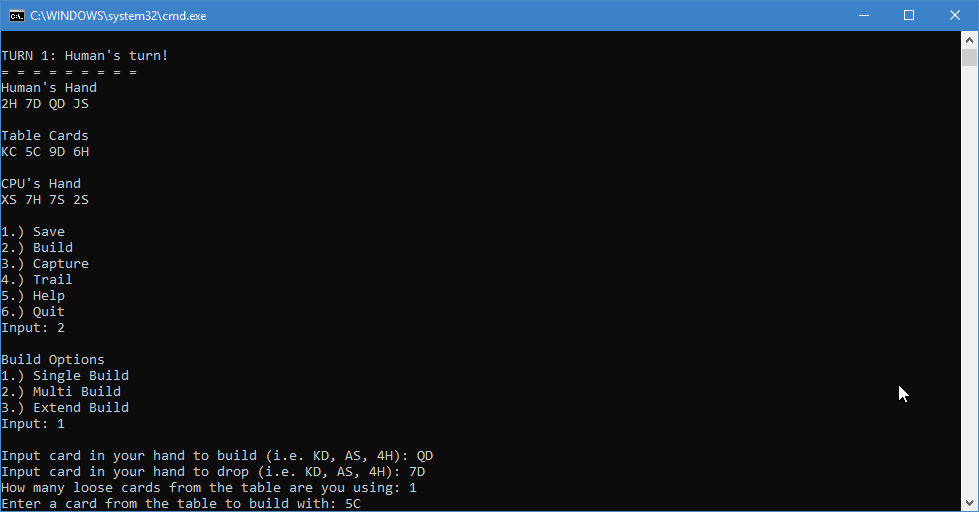
Enter in numeric value for your desired action.



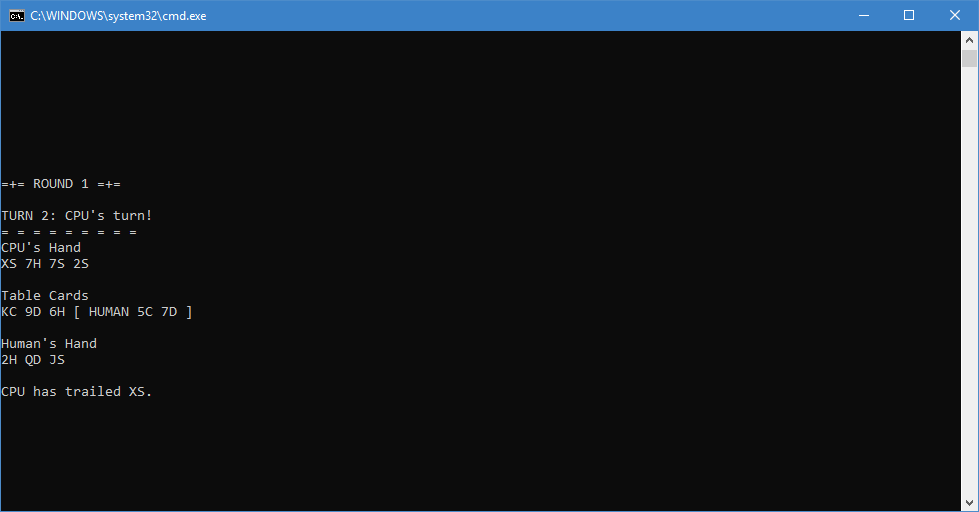
New game will start with a coin flip, enter either “heads” or “tails” to determine the first player.



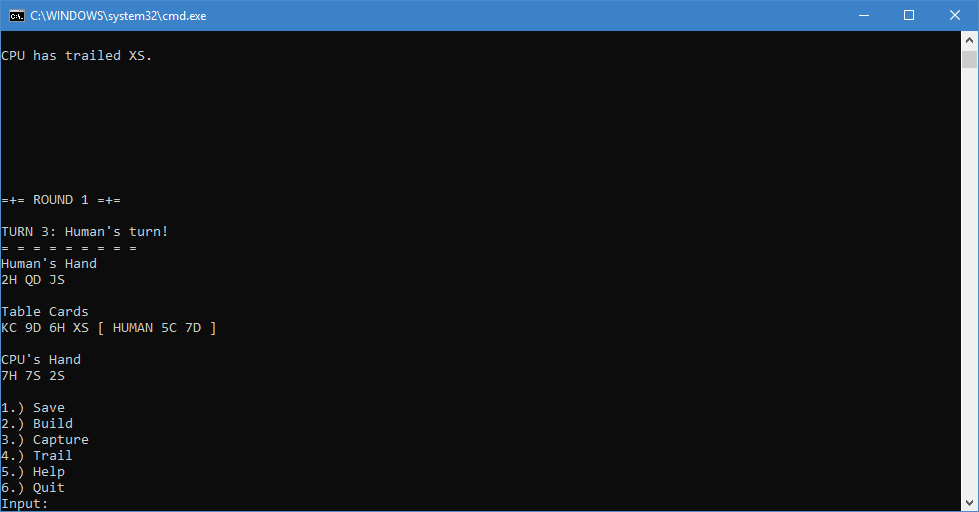
We got lucky and made the right call, we are playing first. You are not prompted with the games current score, the round, who’s turn it is, and what cards are in each players hand as well as the table. You may now select what you’d like to do at the bottom by entering a numerical value listed.



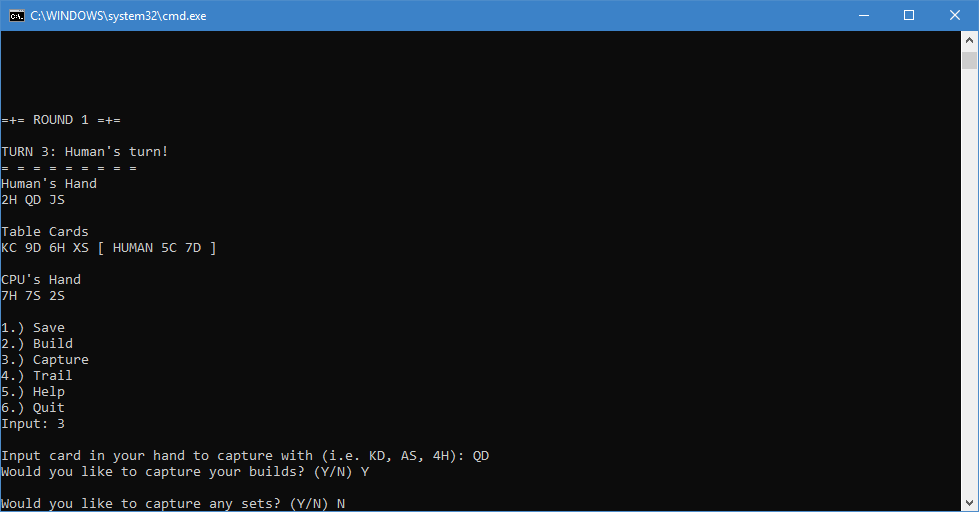
Here I am building my Queen of Diamonds by dropping the 7 on Diamonds onto the table’s 5 of Clubs.



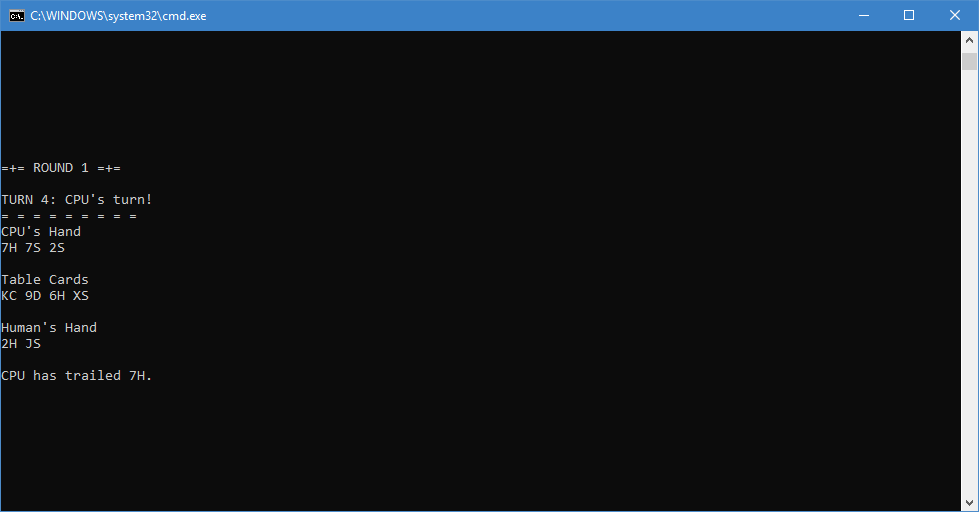
The CPU now takes it turn by trailing the 10 of Spades as it cannot build or capture.



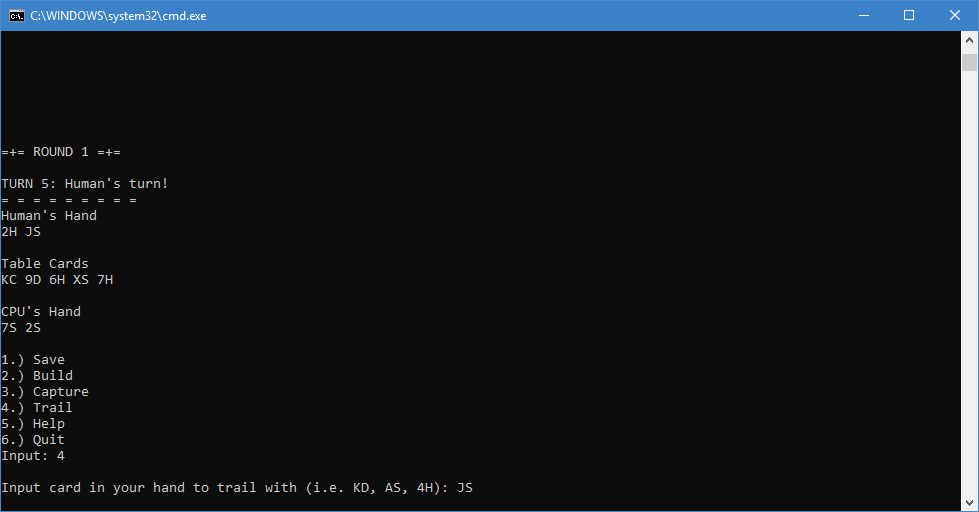
Our build now shows up along with the changes made by the CPU’s turn.



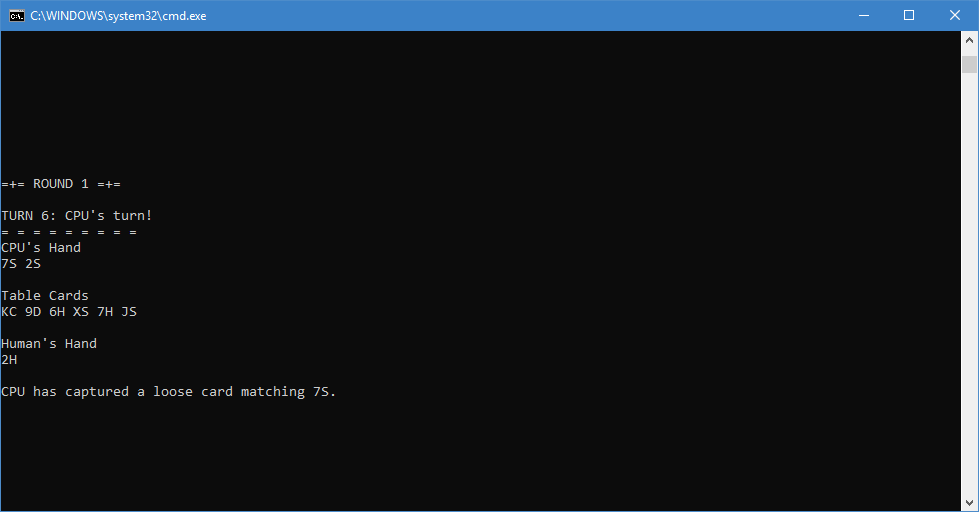
Now we will capture our build using our build card.



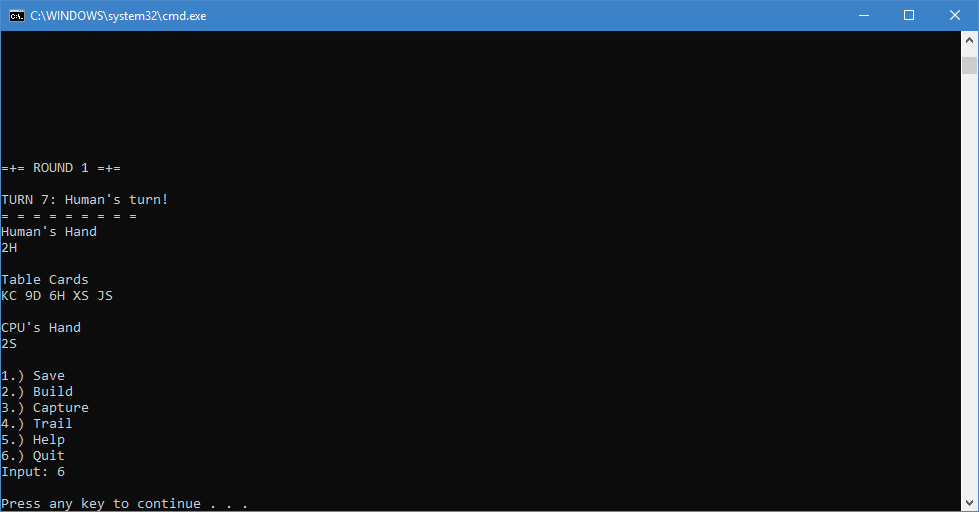
CPU took its turn by trailing again as it could not build or capture. Our build is gone as we have captured it.



We trail the Jack of Spades because we can’t build or capture.



Computer captures a loose card matching its 7 of Spades.



We quit the game without saving by using the quit option.