

Demo for Straights Game - Michael Zheng, 20885687

I have included the associated .in and .out files. All .in files will have respective .out files that represent the expected output of the .in files when run with the program.

Note: When I write “program argument”, I am referring to the optional command line argument that the user can enter which is a number representing the seed that they wish to use to shuffle the cards with.

Firstly, I will demonstrate the basic functionalities of the program:

Setting all players to computers. This just requires all players, when prompted, to be made as computers and the game will be run by itself. This functionality is tested in **test1.in** and the output is seen in **test1.out** when run with argument “1”.

Using the “play” command. I set one player to be the human, and at its turn takes a few play commands to demonstrate that it can properly play cards before eventually exiting with the “quit” command. This is found in **test2.in** and results are in **test2.out** with a program run with argument “2”.

Using the “discard” command. I set one player to be human, and play cards until I have no legal plays, then I will discard cards. This is in **test3.in** and results are in **test3.out** with a program run with argument “3”.

Using the “deck” command. I set one player to be human, and called the “deck” immediately before quitting the game. This is in **test4.in** and results are in **test4.out** with a program run with argument “4”.

Using the “quit” command. In previous tests, this command has already been used, so for this test case I will demonstrate that it works even in the middle of the game. In **test5.in**, I do use one human player to play the game, and after some commands, I exit the game using the “quit” command. Output is in **test5.out** run with argument “5”.

Using the “ragequit” command. I will create one human player and use it to play a few commands, before ragequitting and having computers finish the game. This is found in **test6.in** and results are in **test6.out** with a program run with argument “6”.

Now, we will test a few test cases to prove the program has an entire range of functionality.

In the case of multiple human players, I have included **test7.in** which is to be run with argument "7" and output is in **test7.out**. I initialize 3 humans and play command from all three perspectives, before eventually quitting.

In the case of multiple rage quitters, I have included **test8.in** which is to be run with argument "8" and output is in **test8.out**. I initialize 4 humans and periodically call rage quit until computers end up finishing the game.

In the case of trying to discard a card when there is a legal play in the hand, I have included **test9.in** which is to be run with argument "9" and output is in **test9.out**. I initialize 1 human and play a few cards, then try and discard a card when there is a legal play, in which I will not be able to.

In the case of trying to play a card that is not a legal play, I have included **test10.in** which is to be run with argument "10" and output is in **test10.out**. I initialize 1 human and play a few cards, then try and play a card that is not a legal play, in which I will not be able to.

In **test11.in**, I will demonstrate a full running of a human player, provided that the program receives an argument of "11". Output is in **test11.out**.

In **test14.in**, I wanted to test if I should choose a legal play or discard that was not the first card from the row of cards printed for the human player (if possible). This way, I could see if the program truly reads what card I enter and play / discard the correct one regardless of where in the list printed it is. The resulting output ran with program argument "14" can be found in **test14.out**.

In **test15.in**, I test to see if the game works properly with 2 human players, so I run it with 2 human players and play a few cards before ragequitting with both and having computer play out the game. The output with program argument "15" is found in **test15.out**.

In **test16.in**, I test gameplay with 3 human players, and play a few cards with them before ragequitting with two of the players, and then quitting the game itself. The output with program argument "16" is found in **test16.out**.

In **test17.in**, I test gameplay with all 4 players being human players, and I play a few cards with each before quitting the game. The output with program argument "17" is found in **test17.out**.

Extra Feature Testing:

One of the extra features of my program that the program document did not necessarily specify is the ability to handle end of file errors (doing ctrl + d when input is needed). Thus, you can test this out on my program either during the player setup (human or computer?) or during actual gameplay itself when a human player needs input, you can do ctrl + d and the program will exit and free its memory accordingly.

Another extra feature that I had implemented was the ability for the program to detect invalid commands and ask the user to try again. I demonstrate this ability in **test12.in** with the resulting output in **test12.out** (with program argument "12"), where I give misspelled commands to the input and it's able to detect that it is invalid input, and asks for the user to try again.

In **test13.in**, I demonstrate that if the user were to give input during the player type stage that is not either h or c, then the program will simply ask the user to try again. I give the program some random letters, which are not valid to initialize player types with. I then eventually initialize the players with valid inputs then quit the game. The output is in **test13.out** with program argument "13".