Demonstration Document

Ashna Jagadisan (ID# 20824887)

Commands

Executable	`./straights`

Optional Command-Line Parameters

Argument	**Description**
seed	Optional integer argument to initialize random number generator for shuffling.
`-gui`	Use to play game with graphical user interface.
`-houserules [seed]`	Use to play game with house rules. Can be followed by optional integer seed. **House Rules:** - Game ending score is 45 points (rather than 80) - 2 players (rather than 4) - Starting card is king of hearts (rather than seven of spades)
`-easy [seed]`	Use to play game with easy difficulty computer players. Can be followed by optional integer seed. **Easy Computer:** - Computer players use 'play low, discard high strategy' - Play lowest legal play (if any) or discard highest card in hand
`-hard [seed]`	Use to play game with hard difficulty computer players. Can be followed by optional integer seed. **Hard Computer:** - Computer players use 'play high, discard low strategy' - Play highest legal play (if any) or discard lowest card in hand
`-dynamic [seed]`	Use to play game with dynamic computer players. Can be followed by optional integer seed. **Dynamic Computer:** - If more than 9 cards in Computer's hand, use *Easy* strategy - If less than 5 cards in Computer's hand, use *Hard* strategy - Otherwise, use basic strategy detailed in program specification
filename	Optional argument to load game state from input specified in `filename.in` and arguments specified in `filename.args`.

Note that if no command-line parameters are specified (`./straights`), the program will run as specified in the Straights project document with no enhancements and a time-based seed.

In-Game Commands

Command	Description
`play <card>`	As detailed in program specification.
`discard <card>`	
`deck`	
`ragequit`	
`quit`	

Notes

- Using the quit command will output a summary illustrating the current game state and winners based on that state, as in the sample executable.
- Entering an invalid command (e.g. `plat 7S`) will not break the program.

Walkthroughs

As all save state files must end with a newline, an extra "`>`" will be output when loading them.

Basic Walkthrough

This walkthrough will run through the program as outlined in the program specifications (without any enhancements).

Command	`./straights 7`
Description	This test looks to show the format of the start of the game and that the game can have multiple winners.

List of Commands	Notes

`h`
`c`
`h`
`c`
	These commands will set Player2 and Player4 to be computer players.
	Note that Player4 has the starting card (7S) which they then play and is now on the table in the appropriate suit pile.

`deck`	Deck after shuffling with random number generator seeded with 7. Outputs
	`2C 9H 6C AD 3D KS AH 4C 9D AS 3H JS 4D`
	`8C 2H JD AC KC KH 9C QD 5H 8D 7H 8S JH`
	`7D TD KD 6D 6H 5S 5D 7C 4S QS 4H 2S QC`
	`3C 2D 8H 3S TH JC TS 7S 6S TC QH 5C 9S`
	Note that when the new round begins, it is Player4's turn to play since Player4 has the starting card, 7S.

`discard 2C`	The program outputs "`Player1 discards 2C.`"
`ragequit`	The program outputs "`Player3 ragequits. A computer will now take over.`" Note that the player does not increment, and the new computer player plays the first legal play.
`play 9H`	`9` is added to the `Hearts` suit pile.
`quit`	Note that in the final summary, Player1 has a score of 2 as we discarded 2C. No other players discarded any cards and therefore tie with a score of 0. Thus, Player2, Player3, and Player4 are *all* stated to be winners.

Command	`./straights state1`
Description	This game state uses the same seed as above (7) however now, the game at the end of the round. This test looks to show the end of round format as well as the deck shuffling. Moreover, the program's response to incorrectly discarding and using unknown commands will be demonstrated.

List of Commands	**Notes**

`play AS`	This is the last command input before the round ends. Note that when the round ends, the discards and score calculation of each player is outputted as follows
	`Player1's discards: 2C 6C AD 3D KS AH 4C`
	`Player1's score: 0 + 30 = 30`
	`Player2's discards: 2H AC KC`
	`Player2's score: 0 + 16 = 16`
	`Player3's discards:`
	`Player3's score: 0 + 0 = 0`
	`Player4's discards: 3C 2D 5C`
	`Player4's score: 0 + 10 = 10`

`deck`	Deck after shuffling with the random number generator seeded with 7. Outputs
	`2C 5D 7S 3H 6D KH 5S AD 3S 4S KC 8H 8C`
	`TH 3D 7D 4C 5C 7H QC 4H KD AH 8S 9H KS`
	`JD 6S 9S 2H 6C 7C TD 3C 9D JS QS 5H AS`
	`8D JC TS QD 9C 4D QH AC 6H 2D JH TC 2S`
	Note that when the new round begins, it is Player1's turn to play since Player1 has the starting card, 7S.
	Moreover, the deck is different from the previously output deck as it has been shuffled.

`discard 7S`	Note that there is a legal play, thus a card cannot be discarded. The following message is output "`You have a legal play. You may not discard.`" and the program waits for a legal move to be made.
`flurble`	Note that the program doesn't terminate when it receives an unknown command. Instead, it outputs "`Invalid command. Please try again.`" and waits for valid command to be input.

`quit`	Outputs the following:
	`Player1's discards:`
	`Player1's score: 30 + 0 = 30`
	`Player2's discards:`
	`Player2's score: 16 + 0 = 16`
	`Player3's discards:`
	`Player3's score: 0 + 0 = 0`
	`Player4's discards:`
	`Player4's score: 10 + 0 = 10`
	`Player3 wins!`

Command	`./straights state2`
Description	This game state uses the same seed as above (7) however now, the game is nearing completion. This test looks to show the end of game format as well as the programs response to incorrectly playing.

List of Commands	**Notes**
`play QD`	Note that QD is not a legal play, thus the following message is output "`This is not a legal play.`" and the program waits for a legal move to be made.

`discard QD`	As this is the final command input before the game ends, the following message with player discards, player scores, and winning player(s) will be output:
	`Player1's discards: 2C KC QC TC QD`
	`Player1's score: 62 + 49 = 111`
	`Player2's discards:`
	`Player2's score: 55 + 0 = 55`
	`Player3's discards: 8C JH AC KD QH`
	`Player3's score: 27 + 45 = 72`
	`Player4's discards: 9C KH JC`
	`Player4's score: 66 + 33 = 99`
	`Player2 wins!`
	Note that the game ended as Player1 and Player4 accumulated scores of 80 points or more and that Player2 is the winner as they have the lowest score.
	After this message is output, the program terminates.

Command	`./straights state10`
Description	This test looks to show that the game can finish with multiple winners.
Notes	The game ends as `Player2` accumulates greater than 80 points. `Player1` ***and*** `Player3` are listed as winners as they both have a score of 32, which is the lowest among all the players.

This concludes the walkthrough for the program as described in the project specification with no enhancements. For the following walkthroughs, we will only explore new features relevant to that enhancement rather than repeating the functionality already covered in this one.

For instance, the program's deck shuffling and response to unknown commands will not be tested again, it will just be assumed that these baseline features are already implemented correctly (of course, the marker can run commands to verify this).

House Rules Walkthrough

This walkthrough will run through the program with the `-houserules` enhancement.

Command	`./straights -houserules 7`
Description	This test looks to show the different starting card and number of players under the house rules.

List of Commands	**Notes**
`h` `c`	These commands will set Player2 to be a computer player. Note that the player type of only two players needs to be specified as the game only has 2 players under the house rules. Also observe that the starting card is now KH rather than 7S, which you will have as the starting player. Moreover, notice that the hand size is doubled.

`quit`	The following output is obtained:
	`Player1's discards:`
	`Player1's score: 0 + 0 = 0`
	`Player2's discards:`
	`Player2's score: 0 + 0 = 0`
	`Player1 wins!`
	`Player2 wins!`
	Note that the discards and score info is output for only two players now.
	Since no moves were made, both players are winners with a score of 0.

Command	`./straights state3`
Description	This game state uses the same arguments as above (`-houserules 7`) however now, the game is nearing completion. This test looks to show the different game ending score under the house rules.

List of Commands	**Notes**

`discard AD`	As this is the final command input before the game ends, the following message with player discards, player scores, and winning player(s) will be output:
	`Player1's discards: 8S 7S 7D 6D 5D 5S 4S 3S 3D AD`
	`Player1's score: 8 + 49 = 57`
	`Player2's discards: JD 6S TD 9D AS 8D 4D 2D 2S`
	`Player2's score: 20 + 53 = 73`
	`Player1 wins!`
	Note that the game ended as both Player1 and Player2 accumulated scores of 45 points or more and that Player1 is the winner as they have the lowest score.
	After this message is output, the program terminates.

This concludes the walkthrough of the house rules enhancement.

Easy Computer Walkthrough

This walkthrough will run through the program with the `-easy` enhancement.

Command	`./straights state4`
Description	This game state uses the arguments "`-easy 7`" with all 4 players being human. The game is still in its very early stages. This test looks to show the different play strategy of 'easy level' computer players.

List of Commands	**Notes**
`ragequit`	Prior to typing the command, observe that the legal plays of Player3 (the current player) are listed as follows: `Legal plays: 7D 6H 7C` Upon rage quitting, a computer player takes over and plays 6H, which is the lowest ranked card in the legal plays. Note that the basic computer player outlined in the program specification would simply play the first legal card, 7D.
`ragequit`	Prior to typing the command, observe that the legal plays of Player4 (the current player) are listed as follows: `Legal plays: 8H 6S` Upon rage quitting, a computer player takes over and plays 6S, which is the lowest ranked card in the legal plays. Note that the basic computer player outlined in the program specification would simply play the first legal card, 8H.
`ragequit`	Prior to typing the command, observe that Player1 (the current player) has no legal plays and a hand of cards listed as follows: `Your hand: 9H 6C AD 3D KS AH 4C 9D AS 3H JS 4D` Upon rage quitting, a computer player takes over and discards KS, which is the highest ranked card in the player's hand. Note that the basic computer player outlined in the program specification would simply discard the first card, 9H.
`ragequit`	After typing this command, the game will be completed by computer players. Notice that the discards of each player are very highly ranked, with each on being at least a Jack (which has rank 11).

This concludes the walkthrough of the easy computer player difficulty enhancement. Note that by implementing the 'play low, discard high' strategy, these computer players accumulate scores very quickly and are thereby easy opponents to play against.

Hard Computer Walkthrough

This walkthrough will run through the program with the `-hard` enhancement.

Command	`./straights state5`
Description	This game state uses the arguments "`-hard 7`" with all 4 players being human. The game is still in its very early stages. This test looks to show the different play strategy of 'hard level' computer players.

List of Commands	**Notes**
`ragequit`	Prior to typing the command, observe that Player1 (the current player) has no legal plays and a hand of cards listed as follows: `Your hand: 9H 6C AD 3D KS AH 4C 9D AS 3H JS 4D` Upon rage quitting, a computer player takes over and discards AS, which is the lowest ranked card in the player's hand. Note that the basic computer player outlined in the program specification would simply discard the first card, 9H.
`ragequit`	Prior to typing the command, observe that the legal plays of Player3 (the current player) are listed as follows: `Legal plays: 5H 8S` Upon rage quitting, a computer player takes over and plays 8S, which is the highest ranked card in the legal plays. Note that the basic computer player outlined in the program specification would simply play the first legal card, 5H.
`ragequit` `ragequit`	After typing these commands, the game will be completed by computer players. Notice that the discards of each player are very low ranked, with each one equal to or below rank 4.

This concludes the walkthrough of the hard computer player difficulty enhancement. Note that by implementing the 'play high, discard low strategy, these computer players accumulate scores very slowly and are thereby more difficult opponents to play against.

Dynamic Computer Walkthrough

This walkthrough will run through the program with the `-dynamic` enhancement.

Command	`./straights state6`
Description	This test looks to show that 'dynamic' computer players use the 'easy' play strategy when they have more than 9 cards in their hand.

List of Commands	**Notes**

`ragequit`
 	
`quit`
	Prior to typing the command, observe that the legal plays of Player3 (the current player) are listed as follows: `Legal plays: 7D 6H 7C`
	Also note that Player 3 has more than 9 cards in their hand.
	Upon rage quitting, a computer player takes over and plays 6H, which is the lowest ranked card in the legal plays. That is, Player3 uses the 'easy' play strategy.

Command	`./straights state7`
Description	This test looks to show that 'dynamic' computer players use the basic play strategy when they have less or equal to 9 and greater than or equal to 5 cards in their hand.

List of Commands	**Notes**

`ragequit`
 	
`quit`
	Prior to typing the command, observe that the legal plays of Player4 (the current player) are listed as follows: `Legal plays: 9C 8S JH`
	Also note that Player 4 has exactly 9 cards in their hand.
	Upon rage quitting, a computer player takes over and plays 9C, which is the first card listed in the legal plays. Since this is the middle card in terms of rank, it would not be discarded had the computer used the 'easy' or 'hard' play strategy. Thus, Player4 uses the basic play strategy described in the program specification.

Command	`./straights state8`
Description	This test looks to show that 'dynamic' computer players use the 'hard play strategy when they have less than 5 cards in their hand.

List of Commands	**Notes**

`ragequit`
 	
`quit`
	Prior to typing the command, observe that the legal plays of Player1 (the current player) are listed as follows: `Legal plays: 3D JS`
	Also note that Player 3 has less than 5 cards in their hand.
	Upon rage quitting, a computer player takes over and plays JS, which is the highest ranked card in the legal plays. That is, Player3 uses the 'hard play strategy.

This concludes the walkthrough of the dynamic computer player difficulty enhancement. Note that by implementing both the easy and hard play strategies, these computer players are more advanced and less predictable, making them more difficult opponents to play against.

GUI Walkthrough

This walkthrough will run through the program with the `-gui` enhancement.

Command	`./straights -gui`

Action	**Instructions**
Select player type	Click the "Human" button before starting the game to change the initial player type to "Computer" for any specific player. This can be switched any number of times prior to starting the game.
Input a seed	Enter an integer in the entry box located at the top centre of the window before starting the game. If no seed is specified, the game will use a time-based seed.
Start the game	Click the "Start Game" button.
Play a card	Click on the card in your hand (bottom of window) that you would like to play. If the card cannot be played, select another card to perform an action.
Discard a card	Click on the card in your hand (bottom of window) that you would like to discard. If the card cannot be discarded, select another card to perform an action.
Rage quit	Click on the current player's "Rage Quit" button once the game has started.
End the game	Click on the "End Game" button. This will terminate the window and the program.

Notes

- Legal plays in the current player's hand are signified with cards highlighted green.
- Widgets that should not be pressed are set to be inactive.
- "Human"/"Computer" button sizes are fixed and will not change with the labels.
- The initial text on the entry box is only placeholder text.
- Player type select become rage quit buttons once the game begins.

Command	`./straights state9`
Description	This test looks to show that the output of the GUI aligns with that of the Text UI

Output	`Player1's discards: 2C KC QC TC QD`
	`Player1's score: 62 + 49 = 111`
	`Player2's discards:`
	`Player2's score: 55 + 0 = 55`
	`Player3's discards: 8C JH AC KD QH`
	`Player3's score: 27 + 45 = 72`
	`Player4's discards: 9C KH JC`
	`Player4's score: 66 + 33 = 99`
	`Player2 wins!`

Command	`./straights -gui`
Instructions	Click the "Human" button for each player to set them as Computer players. Enter "7" as the seed. Click the "Start Game" button. Click the "OK" button to close any dialog windows that come up. **Note that the final game state and winner aligns with the Text UI output above.** Click the "End Game" button to exit.

Command	`./straights -gui`
Instructions	Click the “Start Game” button. **Note that the legal play (7S) is highlighted in green.** Click the “End Game” button to exit.

Screenshots

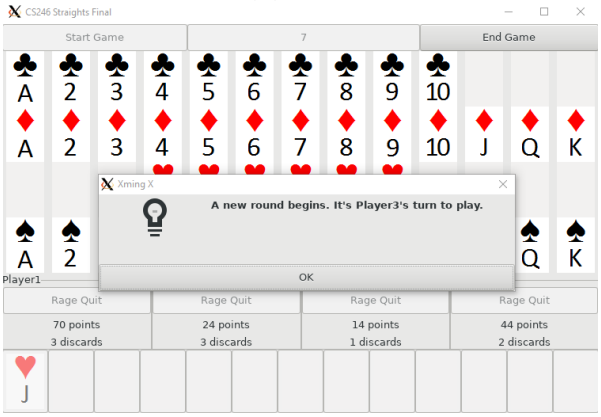
These are random screenshots of gameplay with the GUI.

PLAYER2 AND PLAYER4 SET TO BE COMPUTER PLAYERS.

CURRENT PLAYER (1) HAS NO LEGAL PLAYS.

CURRENT PLAYER (3) HAS THREE LEGAL PLAYS.

DIALOG WINDOW DISPLAYING ROUND SUMMARY.



DIALOG WINDOW DISPLAYING NEW ROUND MESSAGE.

DIALOG WINDOW DISPLAYING GAME WINNER.