HW 03 REPORT

Date: *03/09/20*
Group: *Wednesday Group 08*
Group Members: *Alan Joshua, Isaac Kirsch, Zaina Qasim*

USER INSTRUCTIONS AND DEVELOPER NOTES

- .h and .cpp files for Deck, SidePile as well as main.cpp should be in the same file-directory.
- Card values are determined by hierarchy and nit the face value. Also, hierarchy starts at 2 for simplicity. Hence, cards 2-10 have hierarchy values of 2 -10, Jack = 11, Queen = 12, King = 13, and Ace =14.
- Some expected outputs are meant for numerical/logical accuracy only. Ignore formatting in these cases.

TEST CASES, DISCUSSION AND SCREENSHOTS

TEST CASE 1

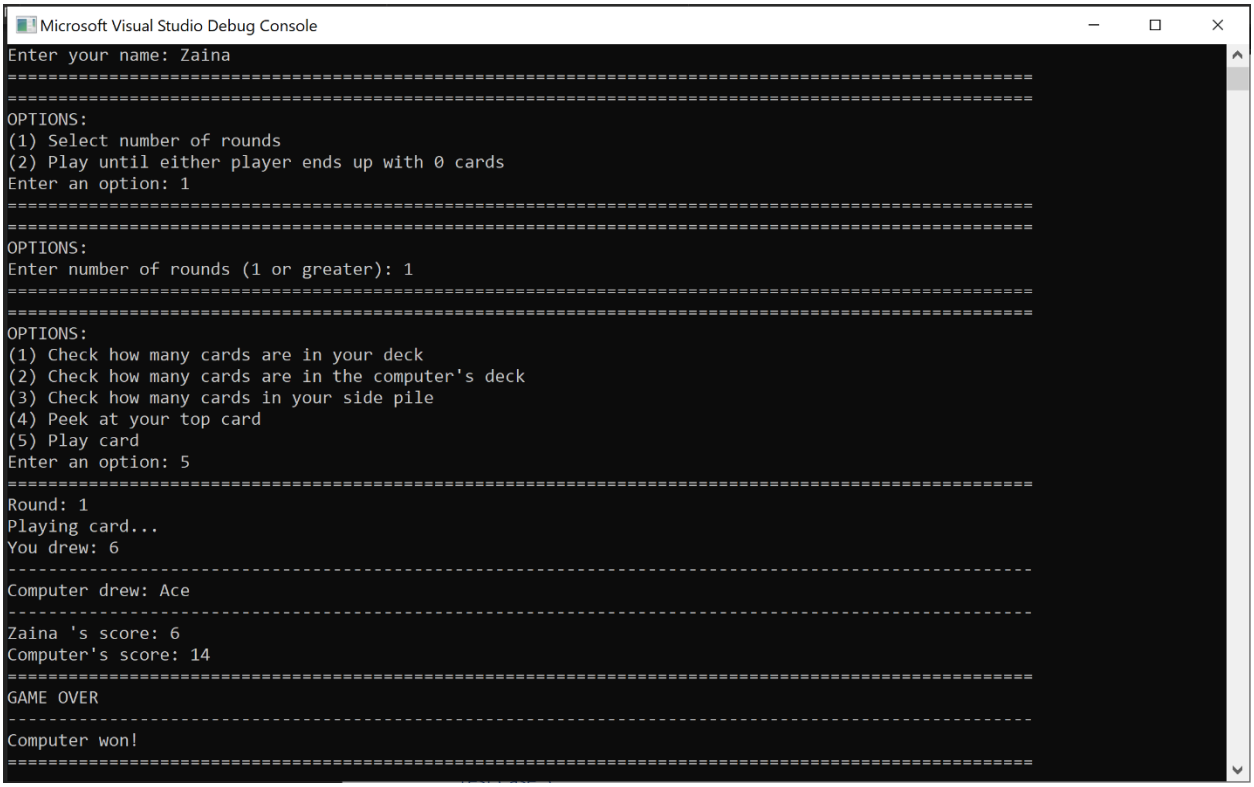
DESCRIPTION:

This test is designed to test a one round game (edge case testing) wherein the winner is decided based on a single card draw.

Test Designed by	Isaac Kirsch	**Pre - condition**	**In Main.cpp:** Line 39 – Uncomment this line Line 40 – Comment out this line
Test Executed by	Zaina Qasim	**Dependencies**	None

S. No.	Action	Inputs	Expected outputs	Actual outputs	Test Result	Test Comments
1	Enter Name	Zaina	Prompts for game condition *(refer to screenshot)*	Prompts for game condition *(refer to screenshot)*	Passed	No comment…
2	Choose game condition	1	Prompts for number of rounds	Prompts for number of rounds *(refer to screenshot)*	Passed	Works great!
3	Choose number of rounds	1	Prompts for user choice	Prompts for user choice *(refer to screenshot)*	Passed	Testing edge case successful.
4	Play card	5	User draws 6; Computer draws 14 (Ace). Computer wins.	User draws 6; Computer draws 14 (Ace). Computer wins.	Passed	Lost to a computer :(

SCREENSHOT



Results of 1 round with a standard deck with predetermined seed

TEST CASE 2

DESCRIPTION:

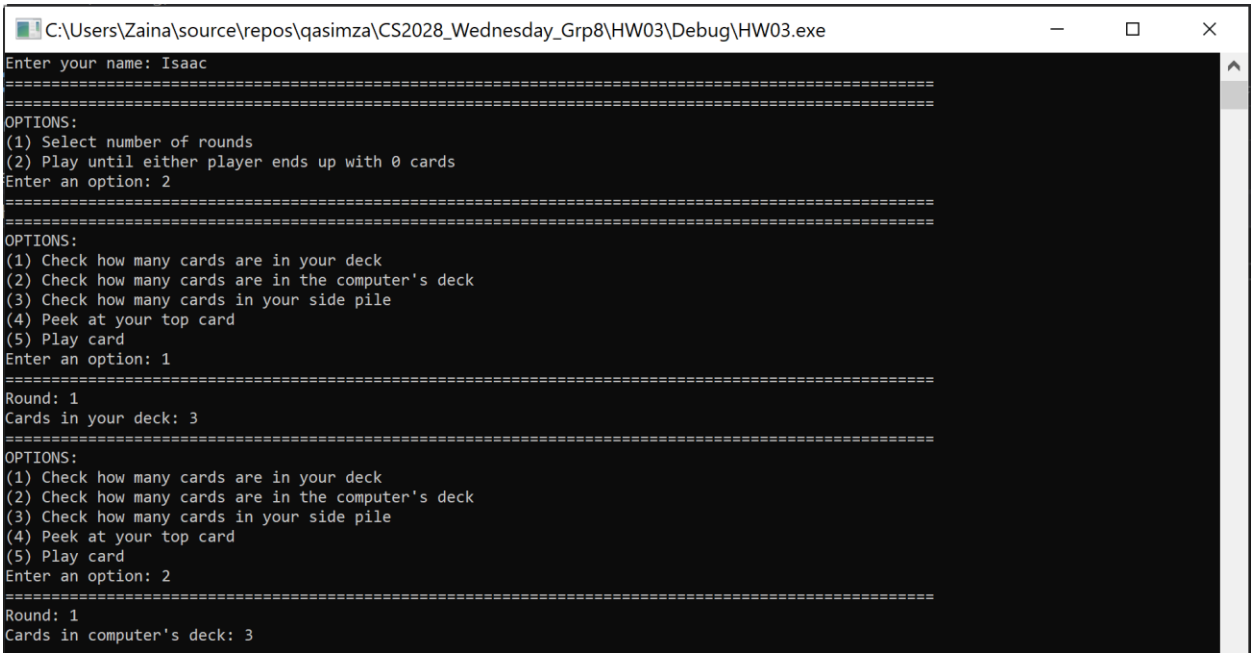
This test is designed to test playing until one of the players ends up with 0 cards. For simplicity, instead of using a full deck of 52 cards, a deck with 6 cards is used (2 suits – each with card values 2,3,4). Since we are changing deck size, this test also serves to confirm the accuracy of the total card counting mechanism. Similarly, the number of rounds is also tested.

Test Designed by	Alan Joshua	**Pre - condition**	**In Main.cpp –** Line 10: Set WHOLE_DECK = 6 Line 39 – Uncomment this line Line 40 – Comment out this line Lines 47 – 50: Comment out these lines Lines 56 – 57: Uncomment these lines
Test Executed by	Isaac Kirsch	**Dependencies**	None

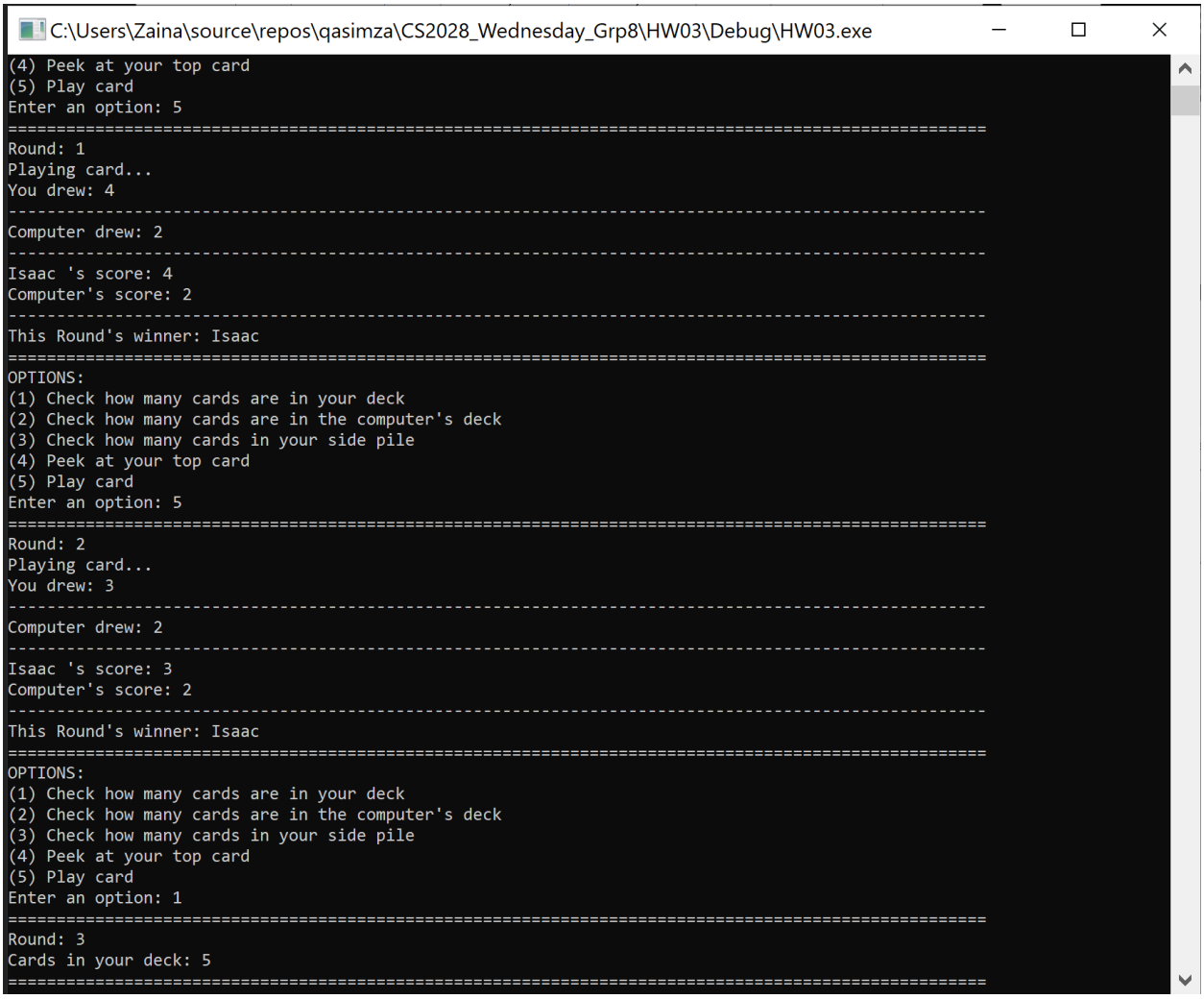
S. No.	Action	Inputs	Expected outputs	Actual outputs	Test Result	Test Comments
1	Enter Name	Isaac	Prompts for game condition	Prompts for game condition *(refer to screenshot)*	Passed	None
2	Choose game condition	2	Prompts for user choice	Prompts for user choice *(refer to screenshot)*	Passed	None
3	Check how many cards are in your deck	1	Round: 1 Cards in your deck: 3 Prompts for user choice	Round: 1 Cards in your deck: 3 Prompts for user choice *(refer to screenshot)*.	Passed	Initial count is correct
4	Check how many cards are in the computer's deck	2	Round: 1 Cards in computer's deck: 3 Prompts for user choice *(refer to screenshot)*.	Round: 1 Cards in computer's deck: 3 Prompts for user choice *(refer to screenshot)*.	Passed	Initial count is correct
5	Play card (2 times)	5	User should win both rounds Card Draws (Player, User): Round 1: (4, 2) Winner: User Round 2: (3, 2) Winner: User	User won both rounds *(refer to screenshot)* Card Draws (Player, User): Round 1: (4, 2) Winner: User Round 2: (3, 2)	Passed	Counter for number of rounds updates correctly *(refer screenshot)*

			(NOTE: This output is mentioned in brief and does not quote character for character – Pass/Fail should be determined based on the accuracy if the cards drawn, and whether the winner for that round was determined correctly)			
6	Check how many cards are in your deck	1	Round: 3 Cards in your deck: 5 Prompts for user choice	Round: 3 Cards in your deck: 5 Prompts for user choice *(refer to screenshot)*.	Passed	Total number of cards updates correctly.
7	Play last round	5	User draws 4; Computer draws 3. User wins this round. ***(NOTE: This output is mentioned in brief and does not quote character for character – Pass/Fail should be determined based on the accuracy if the cards drawn, and whether the winner for that round was determined correctly)*** Computer has 0 cards now, game should end.	User draws 4; Computer draws 3. User won round 3.	Passed	
8	Determining final winner	None	GAME OVER User won	GAME OVER Isaac won	Passed	Yay 😊

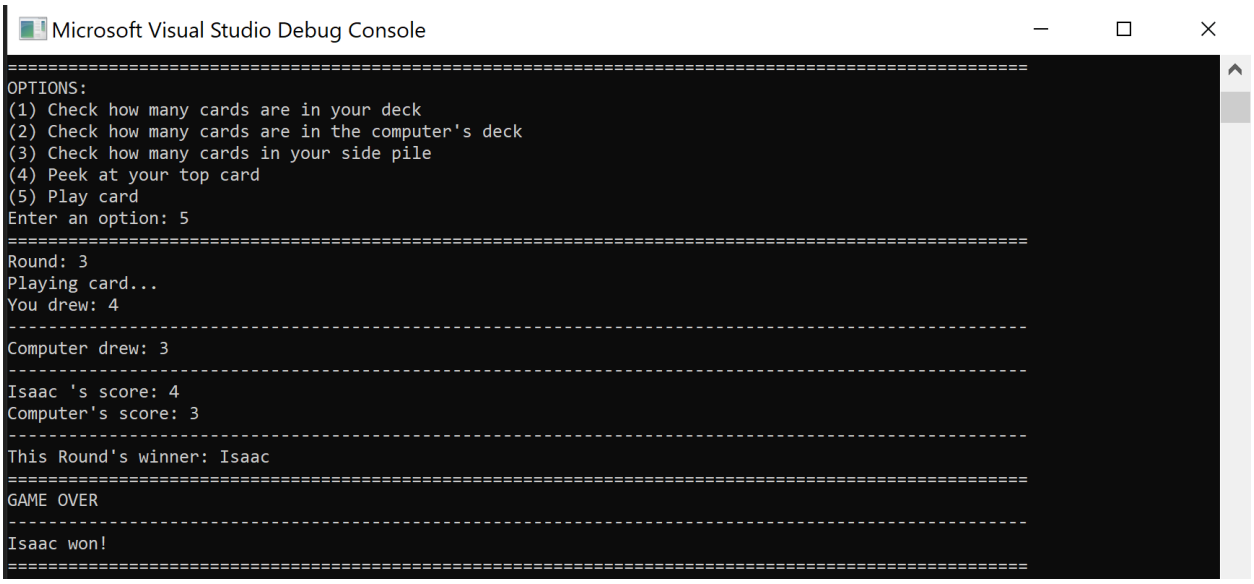
SCREENSHOTS



Selecting game condition and confirming initial number of cards



Rounds 1 and 2, Total number of cards in user's deck after 2 rounds (in round 3)



Round 3, Final winner

TEST CASE 3

DESCRIPTION:

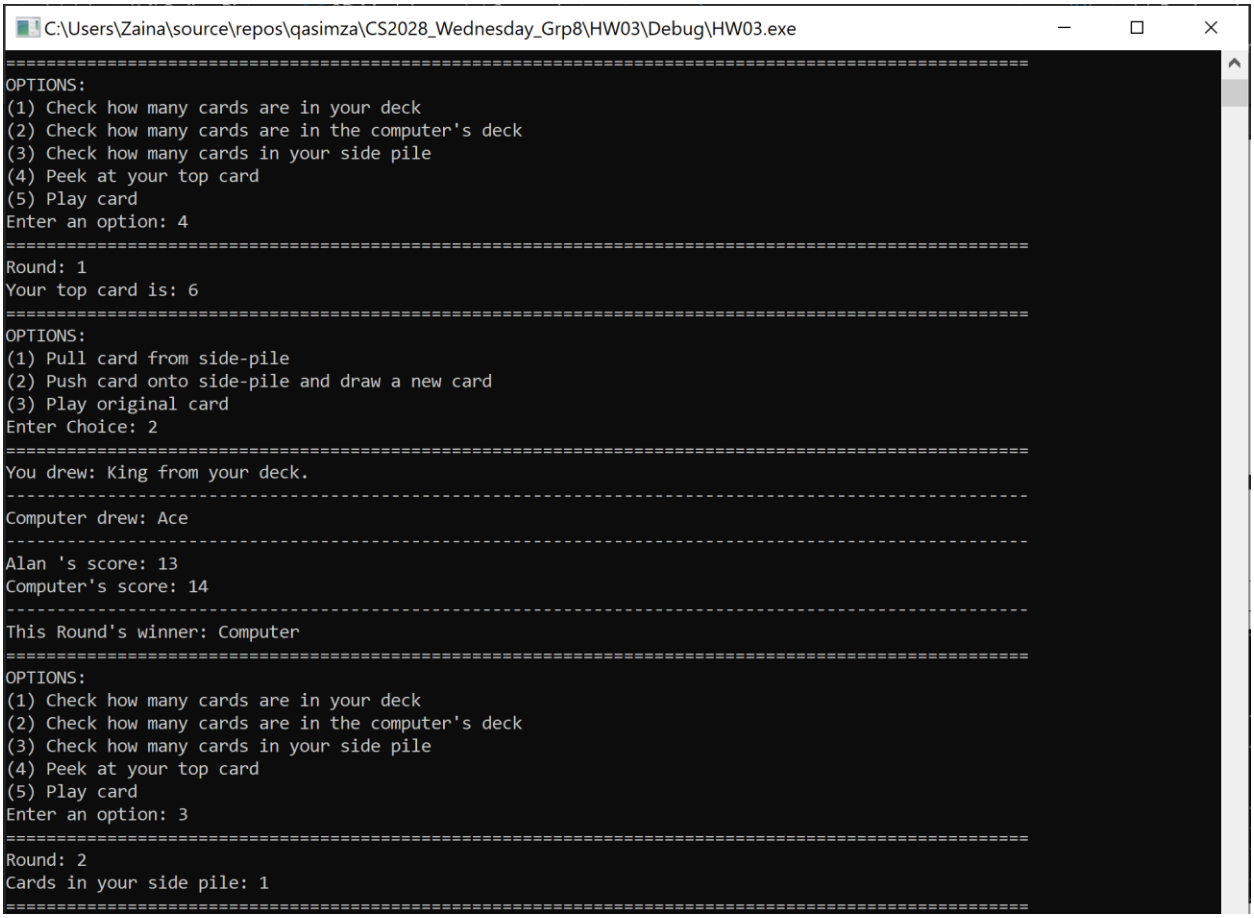
This test is designed to test the functionality of the sidePile in a normal test run. (Adding card to side-pile/ Removing card from side-pile, Underflow exception)

Test Designed by	Zaina Qasim	**Pre - condition**	**In Main.cpp –** Line 39 – Uncomment this line Line 40 – Comment out this line
Test Executed by	Alan Joshua	**Dependencies**	None

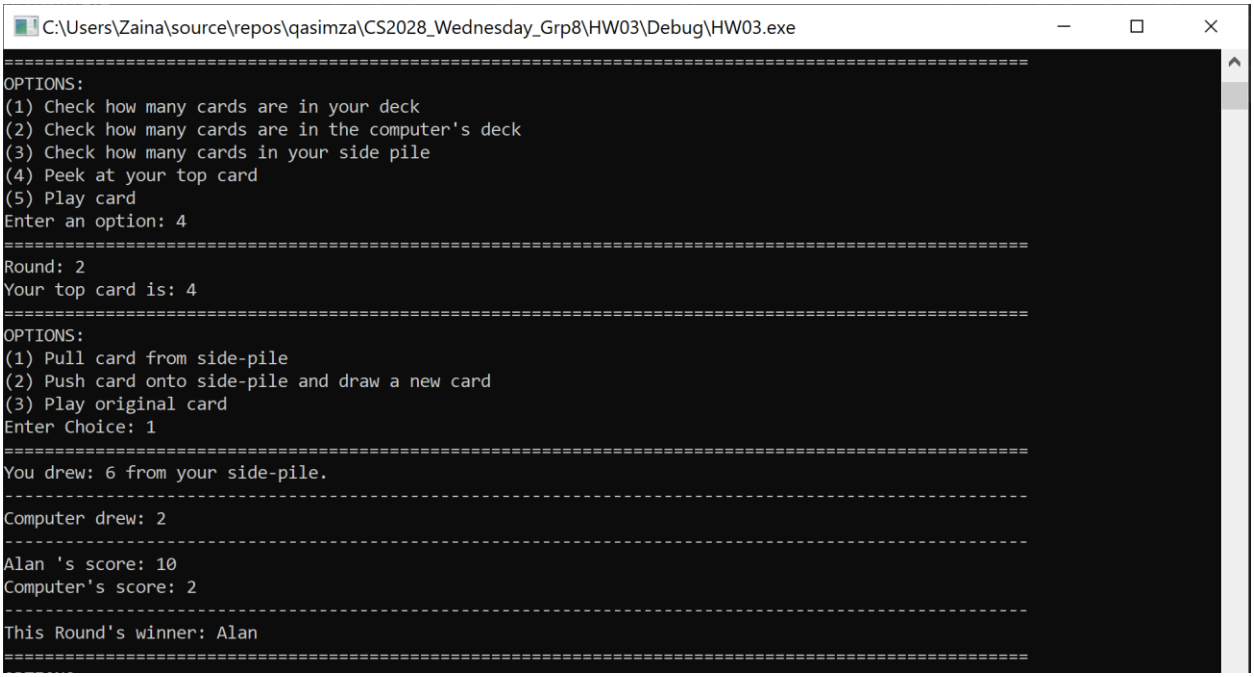
S. No.	Action	Inputs	Expected outputs	Actual outputs	Test Result	Test Comments
1	Enter Name	Alan	Prompts for game condition	Prompts for game condition *(refer to screenshot)*	Passed	None
2	Choose game condition	1	Prompts for number of rounds	Prompts for number of rounds *(refer to screenshot)*	Passed	None
3	Choose number of rounds	5	Prompts for user choice	Prompts for user choice *(refer to screenshot)*	Passed	None
4	Peek at top card	4	Your top card is: 6 Prompts user for choice	Your top card is: 6 Prompts for user choice *(refer to screenshot)*	Passed	None
5	Push card onto side-pile and draw new card	2	User draws King, Computer draws Ace, Computer Wins	User draws King, Computer draws Ace, Computer Wins *(refer to screenshot)*	Passed	None
6	Check number of cards in your side pile	3	Cards in your side pile: 1 Prompts user for choice	Cards in your side pile: 1 Prompts user for choice *(refer to screenshot)*	Passed	None
7	Peek at top card	4	Your top card is: 4 Prompts user for choice	Your top card is: 4 Prompts for user choice *(refer to screenshot)*	Passed	None
8	Draw card from side pile	1	You drew 6 from your side-pile.	You drew 6 from your side-pile.	Passed	None

			User's Score: 10 Computer's score = 2 This round's winner: Alan Prompts user for choice	User's Score: 10 Computer's score = 2 This round's winner: Alan Prompts user for choice *(refer to screenshot)*		
9	Peek at top card	4	Your top card is: Queen Prompts user for choice	Your top card is: Queen Prompts for user choice *(refer to screenshot)*	Passed	None
10	Draw card from side pile	1	Your side-pile is empty. No cards to remove. Reselect menu option. Prompts user for choice.	Your side-pile is empty. No cards to remove. Reselect menu option. Prompts user for choice. *(refer to screenshot)*	Passed	Underflow exception thrown by SidePile class caught in main.cpp and handled appropriately.
11	Play original card	3	Alan's score: 12 Computer's score: 14 This round's winner: Computer Prompts user for choice.	Alan's score: 12 Computer's score: 14 This round's winner: Computer Prompts user for choice. *(refer to screenshot)*	Passed	None
12	Play 2 cards without peeking	5	Card Draws (Player, User): Round 4: (7, 14) Winner: Computer Round 5: (13, 6) Winner: User	*(refer to screenshot)* Card Draws (Player, User): Round 1: (7, 14) Winner: Computer Round 2: (13, 6) Winner: User	Passed	None
13	Final winner	None	GAME OVER Computer won	GAME OVER Computer won	Passed	Winner determined based on the total number of cards each player has after 5 rounds.

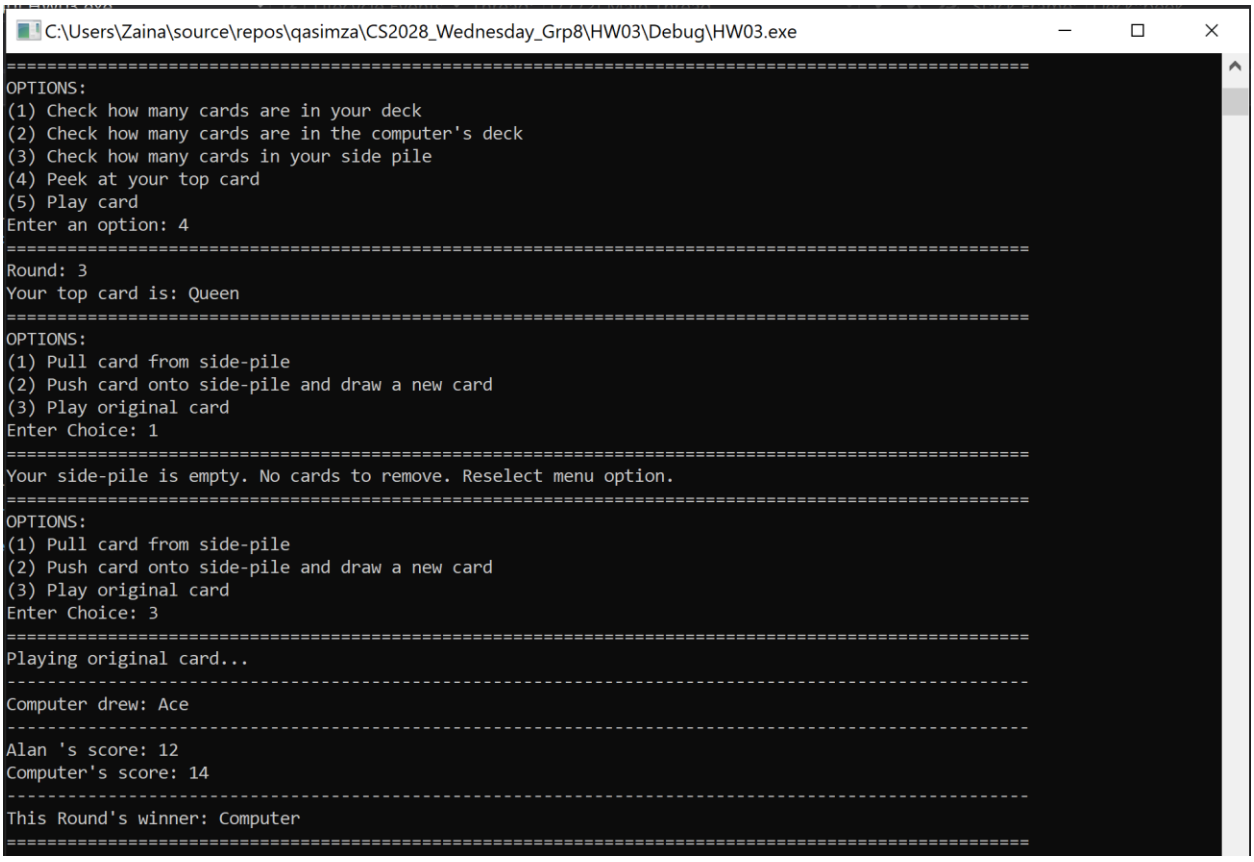
SCREENSHOTS



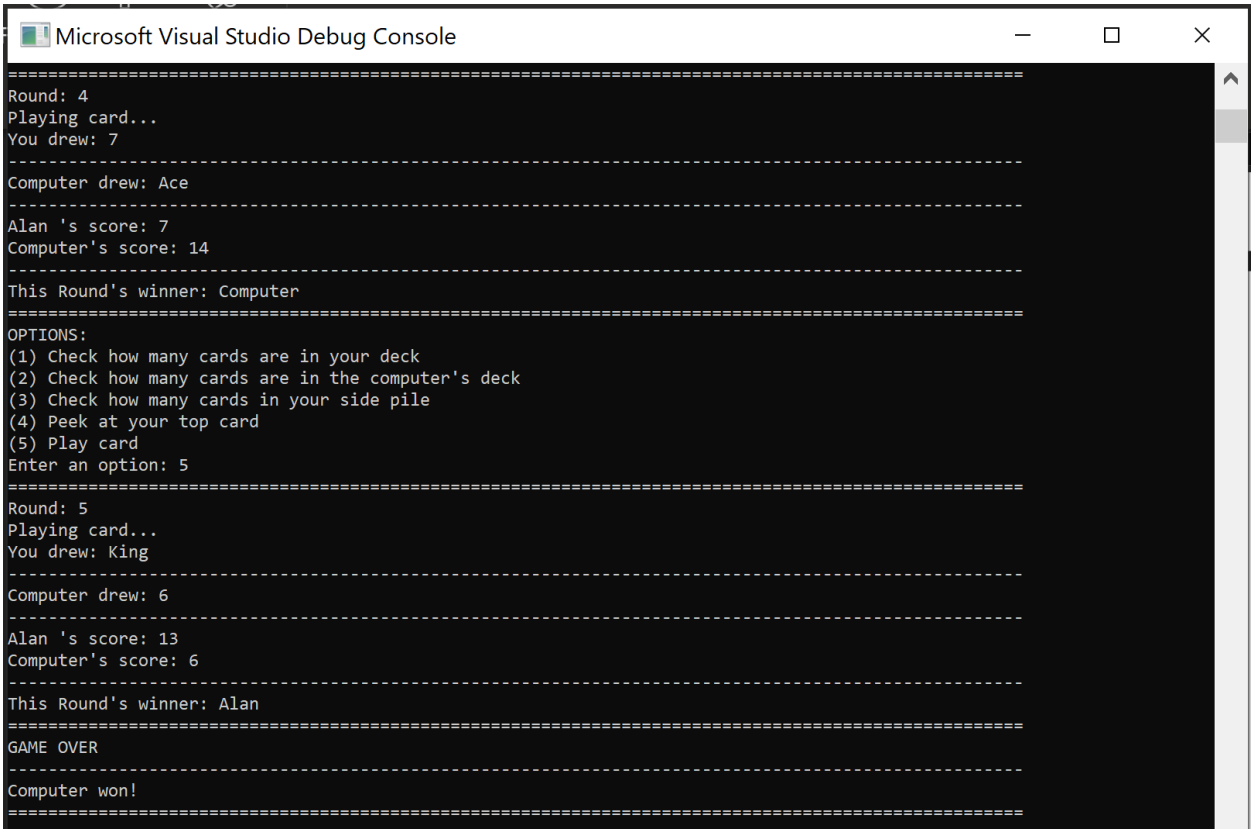
Round 1: Peek at top card, push card to side-pile, draw new card, check number of cards in side-pile



Round 2: Peek at top card and pull card from side pile



Round 3: Peek at top card and try pulling card from an empty side-pile



Rounds 4,5 and Final winner

CONTRIBUTIONS

ALAN:

- Implemented the game functionality through the main program.
- Designed test case 2 and executed test case 3.

ISAAC:

- Implementation of .h and .cpp for the Deck Class and the SidePile class.
- Designed test case 1 and executed test case 2.

ZAINA:

- Planned out the implementation of the game and wrote supporting functions.
- Designed test case 3 and executed test case 1.