

Ryan DiRezze
 CS 162 400
 Lab 3: War Game (with Dice)
 October 14, 2018

Test Case	Input Value(s)	Driver Functions	Expected Outcome(s)	Observed Outcome(s)
Incorrect menu selection (special chars)	Input = %, &, *, (,), #, \$	menu() if(tolower(input) == 'a') else if(tolower(input) == 'b')	Loops back to the question, prompting the user for input (once)	Loops back to the question, prompting the user for input (once)
Incorrect menu selection (positive integers)	Input >= 0	menu() if(tolower(input) == 'a') else if(tolower(input) == 'b')	Loops back to the question, prompting the user for input (once)	Loops back to the question, prompting the user for input (once)
Incorrect menu selection (negative integers)	Input < 0	menu() if(tolower(input) == 'a') else if(tolower(input) == 'b')	Loops back to the question, prompting the user for input (once)	Loops back to the question, prompting the user for input (once)
Incorrect menu selection (wrong letter)	Input = (letters != 'A', 'B', 'a', or 'b')	menu() if(tolower(input) == 'a') else if(tolower(input) == 'b')	Loops back to the question, prompting the user for input (once)	Loops back to the question, prompting the user for input (once)
Correct menu selection to begin program (uppercase char)	Input = 'A'	menu() if(tolower(input) == 'a') else if(tolower(input) == 'b')	Starts the program; Asks for the number of rounds	Starts the program; Asks for the number of rounds
Correct menu selection to quit program (uppercase char)	Input = 'B'	menu() if(tolower(input) == 'a') else if(tolower(input) == 'b')	Quits the program	Quits the program

Correct menu selection to begin program (lowercase char)	Input = 'a'	<pre> menu() if(tolower(input) == 'a') else if(tolower(input) == 'b') </pre>	Starts the program; Asks for the number of rounds	Starts the program; Asks for the number of rounds
Correct menu selection to quit program (lowercase char)	Input = 'b'	<pre> menu() if(tolower(input) == 'a') else if(tolower(input) == 'b') </pre>	Quits the program	Quits the program
Number of rounds input validation (Incorrect input type)	Input != <int>; Input = \$, %, &, ...; Input = a, B, c, D, ...;	<pre> cin >> maxRounds; intValidation(maxRounds); </pre>	Loops back to the question, prompting the user for input (once)	Loops back to the question, prompting the user for input (once)
Number of rounds input validation (correct input type - very small)	Input = 1	<pre> cin >> maxRounds; intValidation(maxRounds); </pre>	Accepts input, then asks the user to select the type of dice for player 1 & 2 When game executes, only one round is be played	Accepts input, then asks the user to select the type of dice for player 1 & 2 When game executes, only one round is be played
Number of rounds input validation (correct input type - very large)	Input = 1,000	<pre> cin >> maxRounds; intValidation(maxRounds); </pre>	Accepts input, then asks the user to select the type of dice for player 1 & 2 When game executes, 1,000 rounds are played	Accepts input, then asks the user to select the type of dice for player 1 & 2 When game executes, 1,000 rounds are played

Wrong input (special char) for type of die	Input = #, \$, %, ^, & *	game() if(tolower(input) == 'a') else if(tolower(input) == 'b')	Loops back to the question, prompting the user for input (once)	Loops back to the question, prompting the user for input (once)
Wrong input (integers) for type of die	Input >= 0; Input < 0;	game() if(tolower(input) == 'a') else if(tolower(input) == 'b')	Loops back to the question, prompting the user for input (once)	Loops back to the question, prompting the user for input (once)
Correct (uppercase) input for 'Die' type of die (player 1 & 2)	Input = A	game() if(tolower(input) == 'a') else if(tolower(input) == 'b')	Accepts inputs for players 1 & 2, then asks the user for the number of sides of each die	Accepts inputs for players 1 & 2, then asks the user for the number of sides of each die
Correct (lowercase) input for 'Die' type of die (player 1 & 2)	Input = a	game() if(tolower(input) == 'a') else if(tolower(input) == 'b')	Accepts inputs for players 1 & 2, then asks the user for the number of sides of each die	Accepts inputs for players 1 & 2, then asks the user for the number of sides of each die
Correct (uppercase) input for 'LoadedDie' type of die (player 1 & 2)	Input = B	game() if(tolower(input) == 'a') else if(tolower(input) == 'b')	Accepts inputs for players 1 & 2, then asks the user for the number of sides of each die	Accepts inputs for players 1 & 2, then asks the user for the number of sides of each die
Correct (lowercase) input for 'LoadedDie' type of die (player 1 & 2)	Input = b	game() if(tolower(input) == 'a') else if(tolower(input) == 'b')	Accepts inputs for players 1 & 2, then asks the user for the number of sides of each die	Accepts inputs for players 1 & 2, then asks the user for the number of sides of each die

Number of dice sides input validation (Incorrect input type)	Input != <int>; Input = \$, %, &, ...; Input = a, B, c, D, ...;	cin >> numSides1 (or numSides2); intValidation(numSides1/2); player1/2 = new Die/LoadedDie(numSides1/2)	Loops back to the question, prompting the user for input (once)	Loops back to the question, prompting the user for input (once)
Number of dice sides input validation (correct input type - very small)	Input = 1	cin >> numSides1 (or numSides2); intValidation(numSides1/2); player1/2 = new Die/LoadedDie(numSides1/2)	Accepts inputs for dice for players 1 & 2, then begins the game, executing the number of rounds specified by the user	Accepts inputs for dice for players 1 & 2, then begins the game, executing the number of rounds specified by the user
Number of dice sides input validation (correct input type - very large)	Input = 1,000	cin >> numSides1 (or numSides2); intValidation(numSides1/2); player1/2 = new Die/LoadedDie(numSides1/2)	Accepts inputs for dice for players 1 & 2, then begins the game, executing the number of rounds specified by the user	Accepts inputs for dice for players 1 & 2, then begins the game, executing the number of rounds specified by the user
Menu repeats until player quits	N/A (No user input)	while(selection == true) { menu(&selection); ... Game game = Game(); ... }	When the program starts, open menu. Then, open menu after each game is played until the user "quits" the program.	When the program starts, open menu. Then, open menu after each game is played until the user "quits" the program.