Classes:

Game (has a)
members:
number of rounds player objects die objects number of sides draw counter
functions:
who won round start game

Menu
functions:
display menu get choice

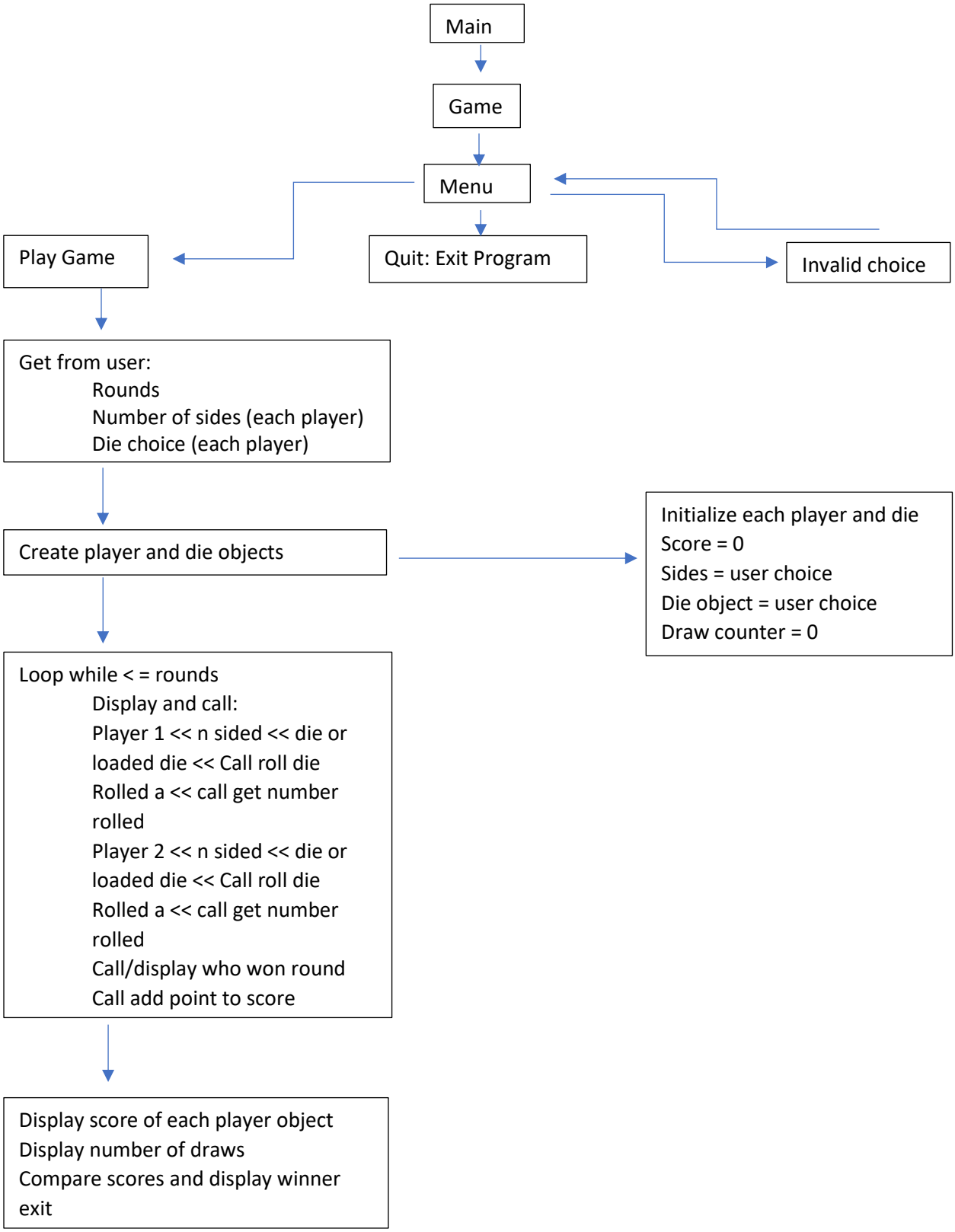
player(has a)
members:
die/loaded die object number of sides score
functions:
Roll die add points score

loaded die(is a)
members:
inherit number of sides
functions:
override rolling

or

die
members:
number of sides
functions:
rolling

Design Plan:


Main
Game
Menu
Play Game
Quit: Exit Program
Invalid choice
Get from user:
Rounds
Number of sides (each player)
Die choice (each player)
Create player and die objects
Initialize each player and die
Score = 0
Sides = user choice
Die object = user choice
Draw counter = 0
Loop while < = rounds
Display and call:
Player 1 << n sided << die or loaded die << Call roll die
Rolled a << call get number rolled
Player 2 << n sided << die or loaded die << Call roll die
Rolled a << call get number rolled
Call/display who won round
Call add point to score
Display score of each player object
Display number of draws
Compare scores and display winner
exit

Location	Test Case	Input	Expected Output	Actual Output

Start Menu
displayStartMenu();
menuChoice();
	1	1	Start Program	Start Program

Start Menu
displayStartMenu();
menuChoice();
	2	2	Quit Program	Quit Program

Start Menu
displayStartMenu();
menuChoice();
	Integer not 1 or 2	8	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid

Start Menu
displayStartMenu();
menuChoice();
	Character	t	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid

Start Menu
displayStartMenu();
menuChoice();
	String (only char)	abc	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid

Start Menu
displayStartMenu();
menuChoice();
	String(char digit)	a7	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid

Start Menu
displayStartMenu();
menuChoice();
	String (digit char)	5d	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid

Start Menu
displayStartMenu();
menuChoice();
	String(char digit char)	s5s	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid

Start Menu
displayStartMenu();
menuChoice();
	float	3.3	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid

Location	Test Case	Input	Expected Output	Actual Output
Get Number of Rounds temp2 = inputValid(temp1);	Integer greater than 0	5	move to next step	move to next step
Get Number of Rounds temp2 = inputValid(temp1);	0	0	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Rounds temp2 = inputValid(temp1);	integer less than 0	-10	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Rounds temp2 = inputValid(temp1);	Character	a	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Rounds temp2 = inputValid(temp1);	String (only char)	abc	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Rounds temp2 = inputValid(temp1);	String(char digit)	d4	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Rounds temp2 = inputValid(temp1);	String (digit char)	8d	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Rounds temp2 = inputValid(temp1);	String(char digit char)	y5x	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Rounds temp2 = inputValid(temp1);	float	1.1	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Location	**Test Case**	**Input**	**Expected Output**	**Actual Output**
Get Number of Sides (p1) temp2 = inputValid(temp1);	Integer greater than 0	2	move to next step	move to next step
Get Number of Sides (p1) temp2 = inputValid(temp1);	0	0	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Sides (p1) temp2 = inputValid(temp1);	integer less than 0	-88	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Sides (p1) temp2 = inputValid(temp1);	Character	h	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Sides (p1) temp2 = inputValid(temp1);	String (only char)	dsf	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Sides (p1) temp2 = inputValid(temp1);	String(char digit)	q7	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Sides (p1) temp2 = inputValid(temp1);	String (digit char)	9r	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid

Get Number of Sides (p1) temp2 = inputValid(temp1);	String(char digit char)	w5j	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Sides (p1) temp2 = inputValid(temp1);	float	5.5	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Location	**Test Case**	**Input**	**Expected Output**	**Actual Output**
Get Number of Sides (p2) temp2 = inputValid(temp1);	Integer greater than 0	6	move to next step	move to next step
Get Number of Sides (p2) temp2 = inputValid(temp1);	0	0	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Sides (p2) temp2 = inputValid(temp1);	integer less than 0	-55	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Sides (p2) temp2 = inputValid(temp1);	Character	a	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Sides (p2) temp2 = inputValid(temp1);	String (only char)	abc	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Sides (p2) temp2 = inputValid(temp1);	String(char digit)	f4	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Sides (p2) temp2 = inputValid(temp1);	String (digit char)	3s	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Sides (p2) temp2 = inputValid(temp1);	String(char digit char)	n8m	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Get Number of Sides (p2) temp2 = inputValid(temp1);	float	7.7	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid

Location	Test Case	Input	Expected Output	Actual Output
Die Choice (p1) temp2 = inputValid(temp1);	1	1	create die object (use temp cout to check)	create die object (use temp cout to check)
Die Choice (p1) temp2 = inputValid(temp1);	2	2	create loaded die object (use temp cout to check)	create loaded die object (use temp cout to check)
Die Choice (p1) temp2 = inputValid(temp1);	Integer not 1 or 2	8	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Die Choice (p1) temp2 = inputValid(temp1);	Character	t	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Die Choice (p1) temp2 = inputValid(temp1);	String (only char)	abc	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Die Choice (p1) temp2 = inputValid(temp1);	String(char digit)	a7	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Die Choice (p1) temp2 = inputValid(temp1);	String (digit char)	5d	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Die Choice (p1) temp2 = inputValid(temp1);	String(char digit char)	s5s	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Die Choice (p1) temp2 = inputValid(temp1);	float	2.3	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid

Location	Test Case	Input	Expected Output	Actual Output
Die Choice (p2) temp2 = inputValid(temp1);	1	1	create die object (use temp cout to check)	create die object (use temp cout to check)
Die Choice (p2) temp2 = inputValid(temp1);	2	2	create loaded die object (use temp cout to check)	create loaded die object (use temp cout to check)
Die Choice (p2) temp2 = inputValid(temp1);	Integer not 1 or 2	-55	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Die Choice (p2) temp2 = inputValid(temp1);	Character	a	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Die Choice (p2) temp2 = inputValid(temp1);	String (only char)	abc	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Die Choice (p2) temp2 = inputValid(temp1);	String(char digit)	f4	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid

Die Choice (p2) temp2 = inputValid(temp1);	String (digit char)	3s	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Die Choice (p2) temp2 = inputValid(temp1);	String(char digit char)	n8m	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Die Choice (p2) temp2 = inputValid(temp1);	float	7.7	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid

Location	Test Case	Input	Expected Output	Actual Output
Roll Die	Rounds	4	prints all rounds with rolls and result	prints all rounds with rolls and result
	Player 1 Sides	same - 5	does not roll greater than sides or less than 1	does not roll greater than sides or less than 2
	Player 1 Die	die		
	Player 2 Sides	same - 5	does not roll greater than sides or less than 1	does not roll greater than sides or less than 2
	Player 2 Die	die		
	score	n/a	score adds 1 to correct player and displays it use temp cout statements to check	score adds 1 to correct player and displays it use temp cout statements to check
	final score	n/a	correct scores and winner displayed based on game	correct scores and winner displayed based on game
Roll Die	Rounds	6	prints all rounds with rolls and result	prints all rounds with rolls and result
	Player 1 Sides	same -23	does not roll greater than sides or less than half sides	does not roll greater than sides or less than half sides
	Player 1 Die	loaded die		
	Player 2 Sides	same -23	does not roll greater than sides or less than half sides	does not roll greater than sides or less than half sides
	Player 2 Die	loaded die		
	score	n/a	score adds 1 to correct player and displays it use temp cout statements to check	score adds 1 to correct player and displays it use temp cout statements to check
	final score	n/a	correct scores and winner displayed based on game	correct scores and winner displayed based on game
Roll Die	Rounds	5	prints all rounds with rolls and result	prints all rounds with rolls and result
	Player 1 Sides	same - 14	does not roll greater than sides or less than half sides	does not roll greater than sides or less than half sides
	Player 1 Die	loaded die		
	Player 2 Sides	same -14	does not roll greater than sides or less than 1	does not roll greater than sides or less than 1
	Player 2 Die	die		
	score	n/a	score adds 1 to correct player and displays it use temp cout statements to check	score adds 1 to correct player and displays it use temp cout statements to check
	final score	n/a	correct scores and winner displayed based on game	correct scores and winner displayed based on game

Roll Die	Rounds	3	prints all rounds with rolls and result	prints all rounds with rolls and result
	Player 1 Sides	same - 7	does not roll greater than sides or less than 1	does not roll greater than sides or less than 1
	Player 1 Die	die		
	Player 2 Sides	same - 7	does not roll greater than sides or less than half sides	does not roll greater than sides or less than half sides
	Player 2 Die	loaded die		
	score	n/a	score adds 1 to correct player and displays it use temp cout statements to check	score adds 1 to correct player and displays it use temp cout statements to check
	final score	n/a	correct scores and winner displayed based on game	correct scores and winner displayed based on game
Roll Die	Rounds	4	prints all rounds with rolls and result	prints all rounds with rolls and result
	Player 1 Sides	different -9	does not roll greater than sides or less than 1	does not roll greater than sides or less than 1
	Player 1 Die	die		
	Player 2 Sides	different -21	does not roll greater than sides or less than 1	does not roll greater than sides or less than 1
	Player 2 Die	die		
	score	n/a	score adds 1 to correct player and displays it use temp cout statements to check	score adds 1 to correct player and displays it use temp cout statements to check
	final score	n/a	correct scores and winner displayed based on game	correct scores and winner displayed based on game
Roll Die	Rounds	2	prints all rounds with rolls and result	prints all rounds with rolls and result
	Player 1 Sides	different - 40	does not roll greater than sides or less than half sides	does not roll greater than sides or less than half sides
	Player 1 Die	loaded die		
	Player 2 Sides	different -3	does not roll greater than sides or less than half sides	does not roll greater than sides or less than half sides
	Player 2 Die	loaded die		
	score	n/a	score adds 1 to correct player and displays it use temp cout statements to check	score adds 1 to correct player and displays it use temp cout statements to check
	final score	n/a	correct scores and winner displayed based on game	correct scores and winner displayed based on game

Roll Die	Rounds	3	prints all rounds with rolls and result	prints all rounds with rolls and result
	Player 1 Sides	different - 100	does not roll greater than sides or less than half sides	does not roll greater than sides or less than half sides
	Player 1 Die	loaded die		
	Player 2 Sides	different -3	does not roll greater than sides or less than 1	does not roll greater than sides or less than 1
	Player 2 Die	die		
	score	n/a	score adds 1 to correct player and displays it use temp cout statements to check	score adds 1 to correct player and displays it use temp cout statements to check
	final score	n/a	correct scores and winner displayed based on game	correct scores and winner displayed based on game
Roll Die	Rounds	11	prints all rounds with rolls and result	prints all rounds with rolls and result
	Player 1 Sides	different - 70	does not roll greater than sides or less than 1	does not roll greater than sides or less than 1
	Player 1 Die	die		
	Player 2 Sides	different -40	does not roll greater than sides or less than half sides	does not roll greater than sides or less than half sides
	Player 2 Die	loaded die		
	score	n/a	score adds 1 to correct player and displays it use temp cout statements to check	score adds 1 to correct player and displays it use temp cout statements to check
	final score	n/a	correct scores and winner displayed based on game	correct scores and winner displayed based on game

CS 162

Brittany Dunn

April 22, 2018

Lab 3 Reflection

As I was working on this lab where I ran into issues was with the differences between composition and inheritance. From the readings I thought I had a decent grasp on the concepts. However, when I began trying to implement the code to go with the concepts I had some struggles.

Overall, I understood that classes own objects through composition as a has-a relationship and through inheritance classes where specialized versions through a is-a relationship. Where I ran into issues was trying to figure out when a class held an object of a class versus when it held a point to an object of a class. To better understand this, I first turned to the book. When the examples in the book were limited then I tried searching the internet. There I found a website called learncpp.com which had many examples. This made it much easier to understand how to implement code for composition and inheritance. From here I was able to begin coding has-a relationships without too many major errors.

For the die and loaded die classes, I knew inheritance was necessary. I noticed in the book that smart pointers were being used frequently in the book examples of inheritance. I had to go back and review smart pointers before I could write any code. After reading the smart pointer section of the book I was able to use smart pointers to create my die and loaded die objects. I did run into one main issue during this which was forgetting to include <memory>. Once I found the mistake is was a quick fix.

The other issue I had with inheritance was overriding member functions. I knew I need to override the rolling function of die for the loaded die. After looking over the section in the book called overriding member functions and trying to apply it I was unable to get my code to work. When I could not figure out the issue I went to piazza to see if anyone else had this issue. I found a post about using virtual functions. Once I changed the functions to be virtual I was able to get my code working.