Initial design:

**Main**

Calls Game in a do-while loop, looping until the user decides to quit.

**Game**

(Blue text was added as I expanded the design)

Constructor:

* Displays menu: Play, or quit. Quit returns to main and ends the program immediately.
* If play is selected, prompt for:
  + Number of rounds (up to 10000)
  + Type of die for each player
  + Number of sides of dice for each players (Must be greater than 2. can be different for both)
* Creates the necessary Die/LoadedDie objects
* Loops the number of rounds
  + Plays one round
  + output the detailed result of each round, including:
    - the side and type of die used for each player
    - the number each player rolls
    - the score result
* display the final score and the final winner of the game
* ~~Prompt to play again~~

Private:

Die for player 1 - pointer

Die for player 2 - pointer

Counter for player 1 score

Counter for player 2 score

MainMenu()

GameSetup()

PlayGame()

**Die**

Protected:

Integer number of sides

Bool loaded or not

Public:

Roll die – returns a random number between 1 and numSides

**LoadedDie**

Public:

Roll die – returns a random number between 2 and numSides

**Test Plan**

Input validation (user enters too high, too low, zero, decimal, letters)

Loaded die wins long term

You need to include test results using different combinations of Die and LoadedDie for each player and different numbers of sides for each player in the test plan table in the reflection document

**Reflection**

This program turned out to be more difficult than I initially anticipated.

I needed to add:

Die: flag for whether loaded or not

Game: Die to Die pointer

Game: flag for checking memory allocation for die

I had trouble getting the overloaded function to work with a pointer: die was getting created, and the getLoaded flag was correctly reporting whether it was loaded or not, but locaded die were still using the regular Die rool function. Solved with virtual.

describe changes from original design - describe problems encountered and solutions