CS162 – Lab2

Khoa Phan

April 16, 2017

Requirements:

- Single user
- 2 players for the user
- Odd number of rounds
- Multi-sided dice
- Regular or loaded dice
- Use inheritance for Die and LoadedDie classes
- A Game class that runs the game itself
- Test plan for game

Design:

Default menu to be displayed

- Menu
    - Start
        - Default game (regular vs regular, 1 rounds)
        - Modified value game (if options have changed)
    - Game Options (*options that have been modified for easy ID)
        - Player 1/2 options
            - Type of Dice
                - Regular or loaded
            - Number of sides
            - Player Names
        - Number of rounds
        - Rules
            - Modify high vs low score
            - Die restrictions
                - Regular vs regular, regular vs loaded, etc.
            - Round type
                - Regular vs regular
                - Regular vs loaded
                - Loaded vs loaded
                - User selected
        - Reset options
            - Changes all values back to default (remove * from options)
    - Quit

Main Game Program Design

The main program will call the menu class, which will determine which function in the game class will be called. It will default to an instance of the game class. Through this menu will the user be able to access the rest of the functions.

- Create dice based off values (default or modified)
    - Die/LoadedDie class
        - Regular variables
            - # of sides = 6
            - Values = 1-6
            - Chance to roll are equal for all values
                - Random number function
        - Loaded
            - Inherit variables/functions from regular.
                - Modify variables with special conditions in function
                    - # of sides can be changed
                    - Values for sides may be changed
                    - Random number function can favor certain range
- Create default interface with a single screen prompting user to press a key to start the roll.
    - Game class
        - display interface
            - default borders
        - load user variables options
            - # of rounds
            - User names(?)
            - Game type
            - High score/Lower score winner
            - Type of dice per round(?)
- Pause for 2-3 seconds between rolls to add an element of dice roll.
    - Game class
        - Real time screen update with text animation
            - ...[ ]..........
            - ..../ /.......
                - All of these on a single line displayed one at a time(?).
- Output dice roll values. Determine winner (default: high)
    - Game class
        - Display # of sides and type of die for player X
            - Retrieve info from die class
        - Replay round if dice have same values
- Display counter for each player victories, (Player 1: 0 | Player 2: 1)
    - Game class

- Keep track of values in variable
  - If victories = # required based on # of rounds = winner

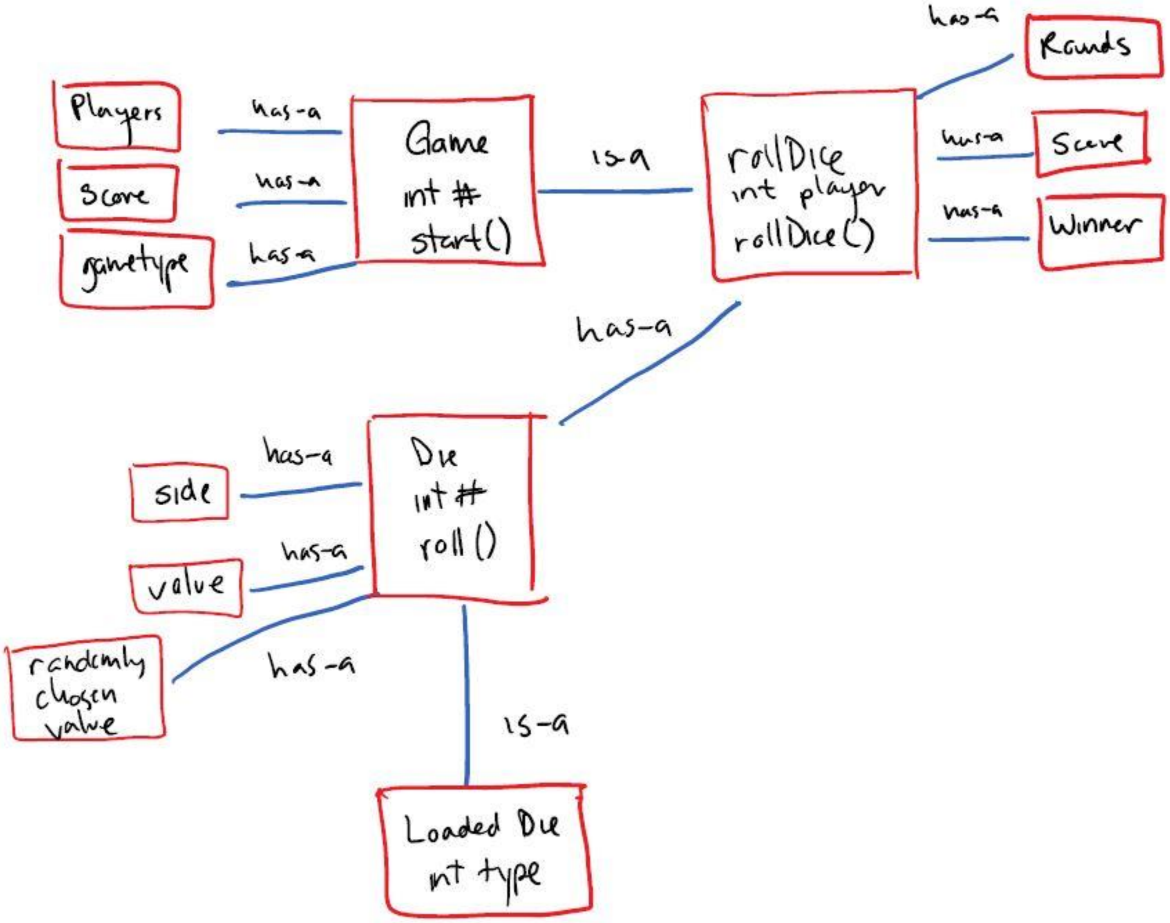
- Pause after each dice roll and wait for user key input
- Loop until # of rounds specified (default or modified) are met
  - Game class
    - Function in loop

Die class design (detailed)

- Die class
  - Regular variables
    - # of sides = 6
    - Values = 1-6
      - Array with values 1-6 to represent sides
    - Chance to roll are equal for all values
      - Random number function
        - Using time srand function and modulus to randomly select a value within array. Modulus % 6(?)

LoadedDie class design (detailed)

- Loaded Die class
  - Inherit variables/functions from regular.
    - Modify variables with special conditions in function
      - # of sides can be changed
        - Modify this value, higher number of sides = more chance to roll higher number
      - Values for sides may be changed
        - Array with high(or low) values doubled and remove/reduce non-winning values
      - Random number function can favor certain range
        - Using time srand function and modulus to randomly select a value within array. (maybe using a different value modulus?)



Testing plan:

GAME

Check Options	Start Game	Roll Dice	Check Values of Dice for Player X	Compare Dice Rolls	Determine Winner of Round	Check # of Rounds	Check for Winner When # of Rounds Reached

DIE/LOADEDDIE

Check Die Values	Roll Dice	Return Die Value

Input validation will be used throughout program. Most values must at least hold a value of 1. A single line string input will be used for player name(maybe).

- Menu
    - Start
        - Call Game class to start
            - Retrieve variables from game
            - Use variables for game functions
                - Player names
                - Calling die and retrieve values
                - Score counter
                - Round counter
    - Game Options (*options that have been modified for easy ID)
        - Player 1/2 options
            - Type of Dice
                - Call die class, if Loaded is chosen, call loadedDie class
            - Number of sides
                - Modify value in die class
            - Player Names
                - Modify value in game class
        - Number of rounds
            - Modify in game class
        - Rules (maybe)
            - Modify high vs low score
            - Die restrictions
                - Regular vs regular, regular vs loaded, etc.
            - Round type
                - Regular vs regular
                - Regular vs loaded
                - Loaded vs loaded
                - User selected
        - Reset options
            - Changes all values back to default (remove * from options)
                - Calls default constructor class to return values
    - Quit