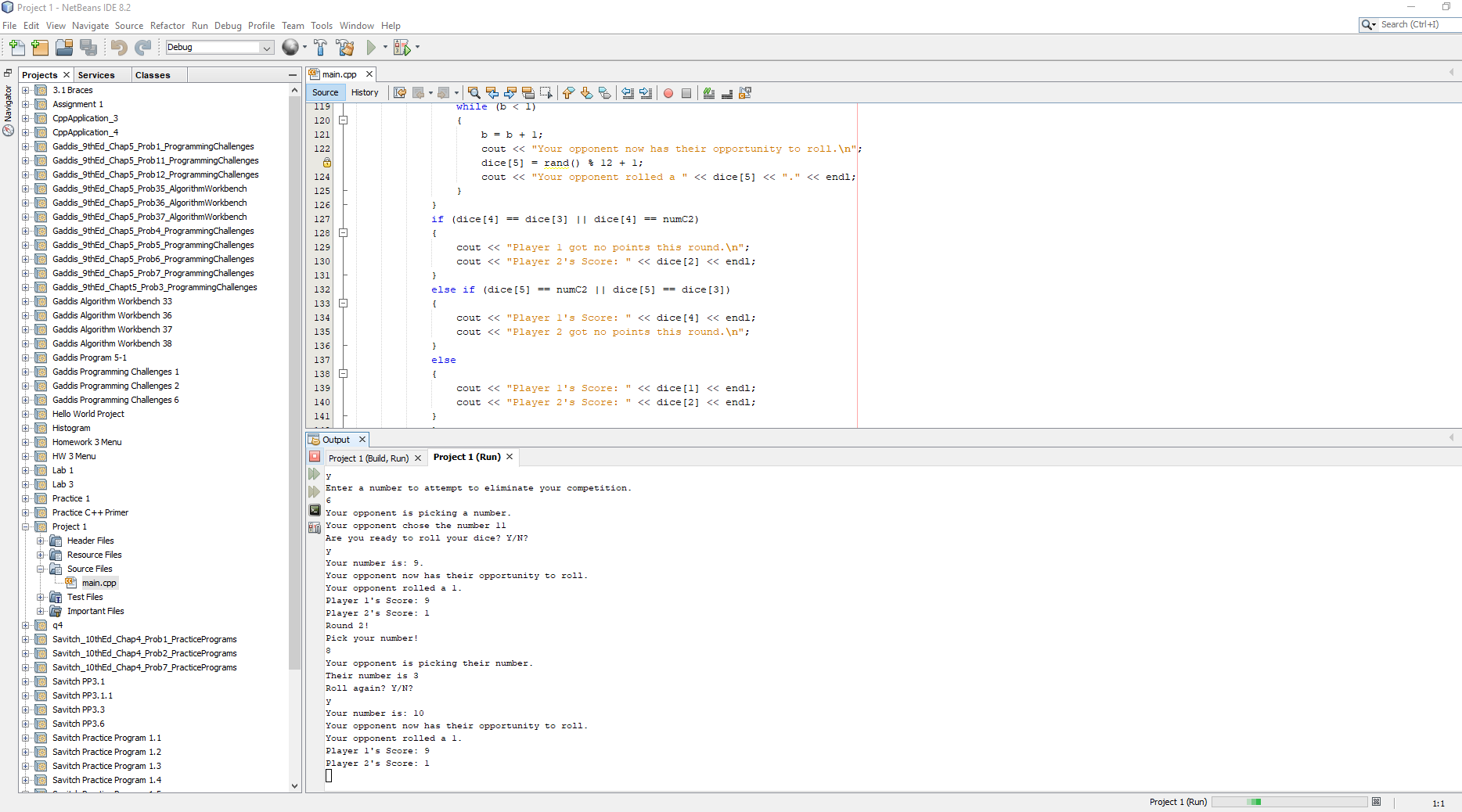
Project 1

217 lines

Zachary Romero | CIS - 5 | 07 November, 2018

# The Game

The game I decided to try and make is called “Knock Out”. A relatively easy looking game to replicate considering it is designed to make children do simple arithmetic.

* The game is played with two dice.
* Players call out which numbers will be considered “Knock Out” numbers for that round. If anyone rolls those numbers with the given dice, they are knocked out for the round, they get no points.
* Whoever has the most amount of points after a few rounds wins, there was no given round limit from the rules that I looked up.

## The process

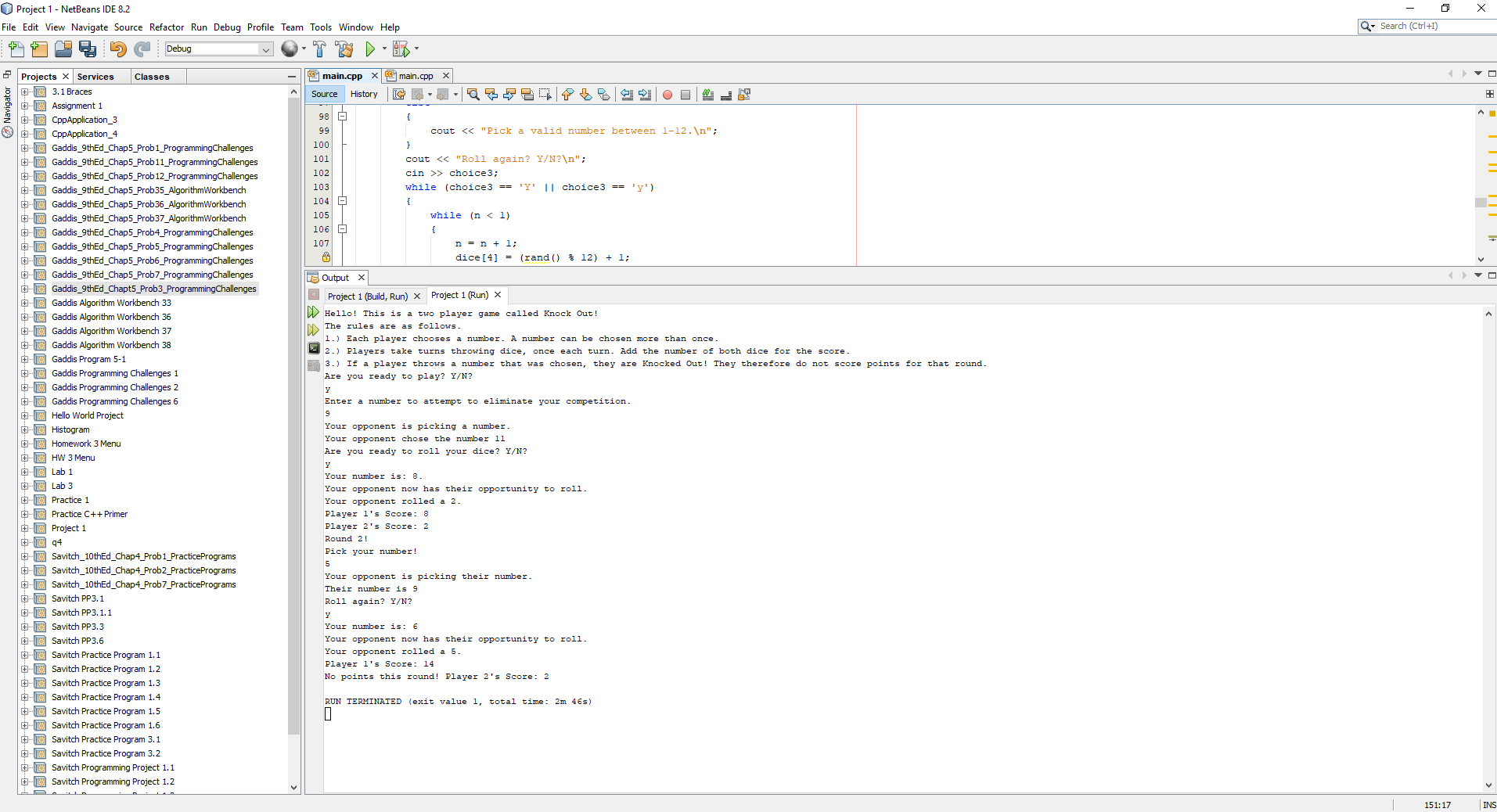
This game was (in concept) a very easy thing to code, as long as I figured out how to roll some dice, have a scoring system that was accurate to each game, and knew how to write code that knew when knock out numbers equaled rolled numbers. Over the course of about a week I realized that I didn’t know how to do most of those things, if statements, if else statements, while statements, do-while statements sure I knew those but they only take you so far.

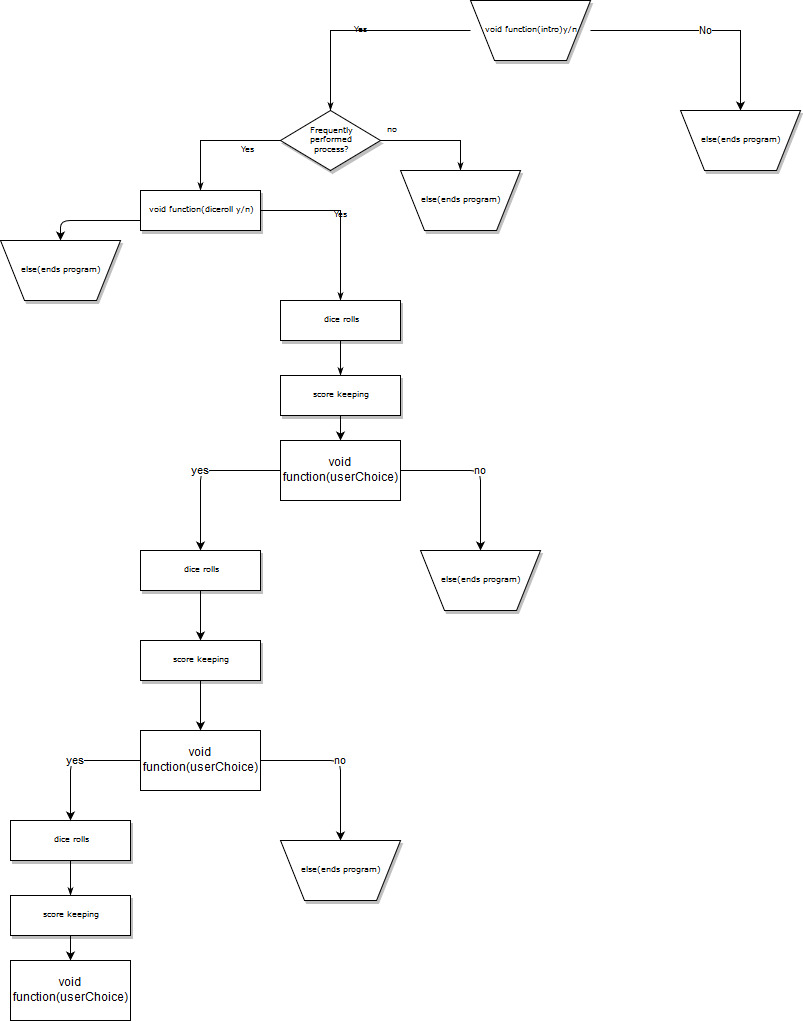
I was determined to make some of what the game required work so I kept at it for a few days. My project really isn’t at where I thought it would be when turning it in, it really seemed quite simple at first. It’s a simple game but it turned out to be so much more, I definitely have learned a lot and want to improve it more, although I do recognize that the game might be too simple to really improve it all that much.

I did want to make it a bit more interactive like a video game rather than just simulating real life dice throwing. I thought asking the player to continue and if they wanted to roll would make the dice throwing a little more interesting.

I thought that I could make the game using the new concepts we learned over the past few weeks, with that being said I definitely did not fit all of them in there because I did not know how to effectively use them and also figure out a way to put them in the game without it breaking. I thought it might be a relatively easy endeavor but it turned out that my code would break with every single little change I made so changing the code turned out to be extremely time consuming and more difficult that I could have expected (especially for a beginner).

In the end I think I was not very successful in demonstrating my knowledge of the material that we went over in the past few weeks, this is because I quite frankly ran into a lot of trouble with it. I did include a flowchart like you asked me for the last one, I did not quite understand how to do one so I hope it fulfills what you were asking, I included my logic for each section of the code.





Cross-List\_From\_Proj1

**Cross Reference from Project 1**

**You are to fill-in with where located in code**

**Chapter Section Topic Where Line #''s**

**Pts**

**Notes**

2 2 cout

3 libraries 8-11 5

iostream, iomanip, cmath, cstdlib, fstream, string, ctime

4 variables/literals No variables in global area, failed project!

5 Identifiers

6 Integers 19-21 1

7 Characters 24 1

8 Strings 25 1

9 Floats No Doubles 1 Using doubles will fail the project, floats OK!

10 Bools 1

11 Sizeof \*\*\*\*\*

12 Variables 7 characters or less All variables <= 7 characters

13 Scope \*\*\*\*\* No Global Variables

14 Arithmetic operators

15 Comments 20%+ 8-11,19-25 2 Model as pseudo code

16 Named Constants All Local, only Conversions/Physics/Math in Global area

17 Programming Style \*\*\*\*\* Emulate

Emulate style in book/in class repositiory

3 1 cin

2 Math Expression

3 Mixing data types \*\*\*\*

4 Overflow/Underflow \*\*\*\*

5 Type Casting 1

6 Multiple assignment \*\*\*\*\*

7 Formatting output 1

8 Strings 264 1

9 Math Library 1 All libraries included have to be used

10 Hand tracing \*\*\*\*\*\*

4 1 Relational Operators

2 if 36 1 Independent if

4 If-else 36,61-85 1

5 Nesting 62-72 1

6 If-else-if 169-195 1

7 Flags \*\*\*\*\*

8 Logical operators 1

11 Validating user input 1

13 Conditional Operator 1

14 Switch 1

5 1 Increment/Decrement 1

2 While 1

5 Do-while 1

6 For loop 163 1

11 Files input/output both 2

12 No breaks in loops \*\*\*\*\*\* Failed Project if included

\*\*\*\*\*\* Not required to show Total 30

Cross\_List\_For\_Proj2

**Cross Reference for Project 2**

**You are to fill-in with where located in code**

**Chapter Section Topic Where Line #''s**

**Pts**

**Notes**

6 Functions

3 Function Prototypes 14-15 4 Always use prototypes

45454454

5 Pass by Value 4

8 return 4 A value from a function

9 returning boolean 4

10 Global Variables XXX Do not use global variables -100 pts

11 static variables 4

12 defaulted arguments 4

13 pass by reference 4

14 overloading 5

15 exit() function 4

7 Arrays 19-20,26

1 to 6 Single Dimensioned Arrays 3

7 Parallel Arrays 2

8 Single Dimensioned as Function Arguments 2

9 2 Dimensioned Arrays 21 2

12 STL Vectors 2

Passing Arrays to and from Functions 5

Passing Vectors to and from Functions 5

8 Searching and Sorting Arrays

3 Bubble Sort 4

3 Selection Sort 4

1 Linear or Binary Search 4

\*\*\*\*\*\* Not required to show Total 70