**Written Report**

A brief discussion after the completion of the program and presentation. I reflect upon my processes and end products and evaluate myself critically to see how I can improve.

**Self Evaluation:**

-Program Logic and Design [8/10]

- I used simple, and consistent choice of structures and names. I Made it easy to navigate, and grasp the logic behind the program. The design is mostly modular where individual methods and inputs and aesthetics can be added. It can be more modular and be more split into individual classes though.

Program Execution and usage [9/10]

- Does not crash, after many beta testings and self testings.You cannot input something that breaks the system. The only custom input is the name, and that accepts all characters and truncates if it’s too large. Gives you error message if you try to do the test with no name selected yet. It prompts you to input. It very occasionally crashes on laggy school computers after spamming buttons for a long time though.

Program Specifications [10/10]

- The program can teach, and is able to reinforce key skills. In terms of coding requirements, it was interactive having many buttons, and an option to play a game. It is simple in design and clean in aesthetics. It demonstrates object oriented concepts by using multiple methods outside of the main method to keep organized. It also is very modular as there are multiple classes and the methods are very prominent in the code while being very flexible. The concept of constructors and passing parameters was included aswell in the music classes.

Coding style [10/10]

-Different methods, variable declarations, and actions are consistent in groupings and organization. The program is overall logical and has small explanations for unintuitive structures, such as the scoring system, and the game system, although the program is dominantly understandable and navigable as functions are grouped based on similar functionalities.

Documentation [10/10]

- The code has good commenting, and explains the use of methods and functions accurately and concisely. For example, in the game running section of the code, the scoring system is a little vague in how it calculates your score, but as you read the comments it becomes clear how each variable is affected and how they affect each other.

Creativity [10/10]

- Clean and user friendly design. Simple naming and simple process. Nice language use, and professional style structure and language uses. Easy to figure out interface, satisfying feedback sounds (buttons). The explanations of the lesson are complete and easy to understand, but includes all of the necessary information plus more in that unit of the lesson, based on YRDSB mathematics curriculums.

**Detailed Analysis of process**

The process:

First I began with the template for an applet. With the applet class, and inside, the init and paint methods, along with some action listeners

Second, I split the paint class into sections for each page, with a conditional statement for each page it can goto. I set up all the pages with text, and added backgrounds.

Third, I worked on the game program. I made many methods for modularity. One for creating random numbers that always worked out. One for organizing the numbers into random positions. One for generation a question based on the numbers, and one for displaying the numbers.

Fourth, I worked on the stats and the leaderboards. I modified the game system to store values throughout the program into text files and temporary arrays.

Fifth I added sound effects and feedback information like color change when button press. I also added quality of life things, such as a name input popup along with the static name text field in the corner. I also added logos and some more pictures.

Lastly I commented my code thoroughly and went back to optimize many methods and cut down my total lines from 1800 to 1500.

Rationale:

I believed it’s better to start from the base up. I build it in such a way I could test each section after another, where the new stuff added doesn’t interfere heavily with the previous additions. It is also forces me to take a simplistic design as I have to make things branch from the menu, and work on each part individually. This is also easiest to understand to someone else who is reading the program as things are in chunks.

Things I would change:

I would make my program even more modular. I would make the questions based on classes too. I would make a method for printing, and try to make everything as simple to read as possible by keeping everything in black boxes. I would try to multithread in game music more cleaner as to not cause the music to play in slow motion if it was lagging.

**The three beta test reports**

**Beta testers:**

*Patrick W. (Neighbor)*

*Angeline C. (Neighbor)*

+Title screen is very creative

**-Bland buttons, make them better (match background and stuff)**

**-Challenge button should be named "PLAY" and stand out more.**

**-Some terminology errors in the lesson**

**-Make positive and negative feedback during the game, but not a popup because it may obscure view**

**-Make leaderboard and stats work**

**-Don't repeat the questions (make a list and save already used ones)**

+Seems promising to young child, but doesn't appear finished as half of the lessons weren't included in the challenge

+Colorful but not too distracting

+Clean design overall

**- Name is kind of hidden, not noticeable at first**

+the previous next buttons is cool, and should be used more with other sections too

+Neat, little window as the game is like a pocket practice game, instead of the game being oversized

**-The window that shows can't be closed without closing the rest of the game**

+-Should Add music and sound effects

Beta Test John W. (Neighbor)

The interface was easy to figure out, with limited buttons to click. All of the buttons worked and took me where I needed to go. The game itself was also easy to figure out, and fun to play. I kept getting higher scores and it made me feel like I remembered (some of) my high school math courses. Tim showed me all of the difficulties, and I surprisingly did better on the harder difficulties. I liked it when the guy screamed “Yeah” when I got a question right. It’s cool when I go on a hot streak, and get many questions correct. I’m glad Tim asked me to beta test his game. It was a fun fifteen minutes.