**Proposal**

**Purpose:**

The purpose of the program is to teach and reinforce critical mathematical concepts about exponentiation. The goal is to address the problem that increasing numbers of grade 6 to 8 students are struggling with math. (This program also provides interactive practice to those who have exhausted their supply of homework!) This program focuses specifically on exponents, but also introduces estimation. These are key concepts in the Ontario Grade 7 and 8 mathematics curriculum, while also being essential to high school mathematics, as they will continue to be relevant.

The program provides a greater intuition towards foundational skills needed in high school mathematics, as these concepts are less naturally intuitive to younger students. These mathematical concepts were specifically selected because students usually struggle with them, and they will be more important in grades 9 through 12.

*Ontario’s 2016 EQAO test results showed that the number of students who are at or above the Provincial Standard--meaning a Level 3 (70%) or higher--has been declining steadily since 2012. Currently, only 50% of grade 6s are at or above the standard, compared with 58% in 2012. Students at the applied level in grade 9 are even lower, at 45%. Even students taking academic math have roughly only 10% achieving Level 4 scores. Clearly, not only do many students need the extra help, everybody could take the extra practice.*

**Target Audience**

The program is primarily for use by elementary school students nearing high school, grades 6 to 8, and those who are in high school who need reinforcement with the basics. This does not mean it is limited to them, as younger students who are excelling are also able to use the program, as the explanations are far simpler than those with a standard grade 6 reading comprehension.

**Interface and User Capabilities**

The program will overall have overall calm and simple colors and shapes. It will be simplistic in design but complex in algorithms regarding the “challenge”. It will be an applet and be run in java. It will include some images throughout. It will have buttons in predictable locations, evenly spaced on the left side for primary buttons and bottom for the specialized buttons.

***Main menu****:* The initial screen the user begins at when the program is launched. It leads to all other programs. The user can also get back from any page to main menu.

***Credits*:** Displays all citation to pictures used, and the creator of the program. Also shows programmer information in the case a freelance programmer is needed.

***Lesson:*** Allows the user to learn the material intended to be taught by the program. They will be also able to look at some example questions, of which explanations of the answer will be given. Each lesson will be composed of three pages, and assisted by diagrams.

***Challenge*:** Goal: Get as many points as possible in the given time

Rules:

-Start with 60 seconds

-Points are given by answering questions correctly and quickly

-Points are deducted by answering incorrectly.

-Time is added by getting correct streaks.

* Difficulty will amplify:
  + the points given per correct answer
  + the complexity, and comprehension required for the question
* Questions will be multiple choice A to E with randomized order and fake answers.
* Top scores are shown in the leaderboards, stats in the stats section

***Help*:** Describes briefly how to use the program, and simple debugging if certain problems appear. This also links additional teaching websites.

**Proposed Timeline**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Goals | Date | Goals |
| 2016-11-28 | Make drafts and blueprints of design and ideas. Make a mock up algorithm. Blueprints will include drawings of the basic layouts. | 2016-12-16 | Make goals sheet for break, but don’t be overly expecting, |
| 2016-11-29 | Christmas break | Work a bit on program and finish up all other non program stuff like cleaning up and color labeling IPOS and DFD’s and other things. |
| 2016-11-30 |
| 2016-12-1 | Write and make proposal, and a guideline to follow the proposal along with some preliminary online diagrams to describe the plan. Begin to structure this proposed timeline. Make a secondary calendar for goals. |
| 2016-12-2 | 2017-1-2 | Create formal algorithm, using pseudo java style coe. Include basic comments so I get whats going on after. |
| 2016-12-5 | 2017-1-3 |
| 2016-11-7 | 2017-1-4 |
| 2016-12-7 | IPO charts and other DFDs initial. Make using google doc’s draw picture option. This may be further detailed later. | 2017-1-5 | Complete and comment program. Make sure to stick with the initial plan, with exceptions of minor changes to combat bugs and inherent and unforeseen problems. Include METHODS, MODULARITY CONSTRUCTORS AND PASSING PARAMETERS |
| 2016-12-8 | 2017-1-8 |
| 2016-12-9 | Create algorithm in plain english | 2017-1-9 |
| 2016-12-12 | Begin on program, gather images and crop them, make them suitable etc. Find the dimensions and make sure things can fit together. | 2017-1-10 |
| 2016-12-13 | 2017-1-11 | Finish up proposal and everything else. Make things look prettier. |
| 2016-12-14 | 2017-1-12 | Presentation #2 & hand in proposal |
| 2016-12-15 | IPO charts and other DFDs. Check them over, and make sure the algorithm/ real code is beginning to follow it | 2017-1-13  To end | Finishing touches, make sure things are titled, names and class codes are included and check for grammar. |

**Citation:**

<http://www.eqao.com/en/assessments/results/assessment-docs-elementary/provincial-report-junior-achievement-results-2016.pdf>

<http://www.eqao.com/en/assessments/results/assessment-docs-secondary/provincial-report-grade9-results-glance-2016.pdf>



A: Credits

* Accessible from all pages except in game
* Static, non interactive page

B: Main Menu

* Accessible from all pages except in game
* Static, non interactive page

C: Help

* Accessible from all pages except in game
* Static, non interactive page

D: Lesson

* Accessible from all pages except in game
* Next and Previous buttons enabled

E: Challenge

* Accessible from all pages except in game
* Contains private access to F - K pages

F: Difficulty

* Not an actual page, simply the set of buttons that control the page display I, J, and K

G: Session stats

* Only accessible from Challenge
* Uses temporary storage

H: Leaderboards

* Only accessible from Challenge
* Uses permanent storage from text file.

I: Easy

J: Medium

K: Hard

* Difficulties only accessible from Challenge
* Share same base page layout from each other (in challenge)

L: THE Challenge

* Actual game page
* Only accessible from Challenge
* Interactive and contains the game black box

M: Lesson 1

N: Lesson 2

O: Lesson 3

P: Lesson 4

* Only accessible from Lesson page
* Same base page layout as each other shared from Lesson.

Linguistic Algorithm

-------------------------------------------------------------------------------------------------------------------

Display menu

→ Print the following

* Print background & images
* Print text in all required locations
* Top right corner text field for their name
  + Save name.
* 2 buttons on the bottom right for left and right
  + Make usable if the current page = lessons
* 5 buttons on the left side labeled menu, credits, help, lesson, challenge
  + Each of the buttons set change the page number and repaints
  + Menu returns you to the title page
  + Credits gives you info on sources and authors
  + Help gives you an idea of what the program does
  + Lesson teaches you the required knowledge
    - 4 pages total, can use the left and right buttons to navigate
  + Challenge Lets you test yourself to see how good you are
    - On the challenge page there are 6 more buttons
      * Difficulties (3 different buttons, easy medium hard)
        + Changes the toughness of the game
      * Stats
        + Shows recent stats
      * Top Scores
        + Shows total best scores all time
      * Play
        + Plays the game
    - If name exists, continue if not, prompt for name
    - During game generate an expression that is full solved
      * Take away a variable and call it x
      * Ask use what x is
      * List 5 numbers, 4 of them wrong, 1 of the correct
      * Correct answers add points
      * Wrong answers deduct points
      * Winstreak is increased for correct streaks
      * Winstreak points are added to total stats every answer
    - Score is saved into text doc.
    - Leaderboards print in order of highest to lowest
      * Uses simple bubble sort
      * Truncates names to 10 characters
      * Shows maximum of top 15 scores.
  + Repaints and depend on the page number prints different things

**A.C.E. LOGS**

|  |  |  |  |
| --- | --- | --- | --- |
| DATE | Accomplishments | Concerns | Expectations |
| 2016-11-28 | Developed preliminary ideas  Developed basic algorithm how to do demonstrations | Organization may be difficult  Picking a fun topic that is sufficiently interesting in grade 6-8 | Finalize the template algorithm |
| 2016-11-29 | Researched about curriculum  decided the order of teaching  Decide a topic | Organization may be difficult  How to demonstrate without being boring | Begin proposal ideas, and topics to talk about  Find some visuals and sounds |
| 2016-11-30 | Found sounds and visuals for the presentation and program itself | Too difficult to add all detail without not being concise | Begin on code and the introduction. Attempt to make an AI that does this for me |
| 2016-12-1 | Create a sample gui on paint | How to make the questions challenging but also doable for all audiences, also too much repetition often | Create question functions  And make the gui |
| 2016-12-2 | Finish proposal + attempt at helpbot | Making the proposal thorough and interesting | Begin on advanced algorithms while making DFD |
| 2016-12-5 | Complete algorithm in depth | Doing it so that i can understand what i’m doing later | Create the template for the program in terms of shapes and positions of buttons and stuff |
| 2016-11-7 | No accomplishments | Wifi down | Check algorithms and make it practical |
| 2016-12-7 | Complete algorithm in depth | Doing it so that i can understand what i’m doing later | Create the template for the program in terms of shapes and positions of buttons and stuff |
| 2016-12-8 | Revise and go over algorithm and make more similar to java code and prepare the images | Images may take a long time and may not align with invisible buttons. May have to use coordinates for mouse | Implement testing system and leaderboard listing. And profanity detection |
| 2016-12-9 | Keep revising and optimize code for redundancies | No concerns, ahead of schedule | Prepare to write the real algorithm (or just start writing) |
| 2016-12-12 | Keep working on actual program | - | Keep working on actual program |
| 2016-12-13 | Keep working on actual program | - | Keep working on actual program |
| 2016-12-14 | Keep working on actual program | - | Keep working on actual program |
| 2016-12-15 | Keep working on actual program | May need to use game threads, because i need to run multiple while loops at once in parallel and in both real and in game time. | Keep working on actual program |
| 2016-12-19 | Present Keep working on program | Make program work better on school computer | Adjust to work for school computer |
| 2016-12-20 | Keep working on actual program | - | Keep working on actual program |
| 2016-12-21 | Keep working on actual program | - | Keep working on actual program |
| 2016-12-22 | Keep working on actual program | - | Make a to-do list for the break |
| 2017-01-10 | Optimize program, comment more, make leaderboards | - | Find something to do or finish leaderboard system |
| 2017-01-11 | Look for more people to test it | - | Improve based on advice |
| 2017-01-12 | Implement tips | - | Finish soon |
| 2017-01-13 | Implement tips | - | Finish soon |
| FINAL DAY | | | |
|
|

-------------------------------------------------------------------------------------------------------------------

**Written Report**

**Self Evaluation:**

-Program Logic and Design

- 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 modular where individual methods and inputs and aesthetics can be added.

Program Execution and usage

- 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.

Program Specifications

- 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

-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

- 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

- 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 (steps, rationale, things i would do the same or differently)**

-

**The three beta test reports**

**Beta testers:**

*John W. (Neighbor)*

*Patrick W. (Neighbor)*

*Angeline C. (Friend)*

+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):

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.

------------------------------------------------------------------------------------------------------------