**Software Planning**

This is the general flow chart of the program’s interface. The letters are there for labels so to be further described below.



Linguistic Algorithm

This algorithm would describe the projected program in the style of an explanation broken up into point form. Instead of describing every step of the logic behind algorithms, it allows users to assume black boxes for the actions it says it performs.

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

(B) Menu:

Purpose: The title and the cover of the program.

→ Can navigate to help, credits, lessons, play

* Paint an image of a superhero

(A) Help:

→ Can navigate to menu, credits, lessons, play

* Paint the set of lines that has to do with introductory information

(C) Credits:

→ Can navigate to menu, help, lessons, play

* Paints the set of lines that has to do with credits and extra help websites

(D) Lessons:

→ Can navigate to menu, help, credits, play

* Begins on page 1:
* Contains next and previous buttons that changes the page number of the lesson
* Each page paints sets of lines for teaching a lesson about the topic
* Lessons in order: Estimation, Exponents, tricks with powers, and real life application

(E) Challenge:

→ Can navigate to menu, help, credits, lessons

* Contains the following buttons:
* Easy medium hard: they change they set the difficulty modifier for the game
* Stats: Shows the stats since the launch of the program recently, shows session best, session worst, highest streak, total games played and total seconds spent in game.
* Leaderboards:Shows the top 15 scores ever, along with name, displayed from textfile.

Function Algorithms

This algorithm will include certain processes such as the game display, or the leaderboard systems. It goes it to detail and takes a form more similar to a real algorithm than a linguistic one.

**Page change:**

Paint method

{

Paint(background)

Paint(logo)

Paint(nameTextfield)

If(page != menu)

Set menu’s buttons invisible

If(page != help)

Set help’s buttons invisible

… etc - *This applies to all pages*

If (page = menu)

Display(menutext)

Paint(Menu pictures)

Else if (page = help)

Display(helptext)

Paint(Menu pictures)

… etc - *This applies to all pages*

}

**Game display :**

Set timevalue = 602

While (timevalue > 0)

{

Set game buttons visible (a, b, c, d, e buttons)

If (question is finished)

{

Create new numbers

Change up question format

}

Display question (new numbers, question format)

Check scores → save for stats if it’s a career high

Time = time - 3

Display time

repaint ();

}

Else if (timevalue <= 0)

{

Save score (user name, score)

Display feedback, and stats

Set screen to session stats

GameStat = notRunning; //This makes shuts off other functions running in the background

Repaint;

}

**Leaderboards:**

Open text file for leaderboards

String line

While (line = current line isn’t blank)

{

If line is only spaces

If it’s an odd line, make it “<empty>”

If it’s an even line make it score 0

Store value into array

}

Bubblesort (array)

Print the leaderboard onto page

**Question generation:**

Int num1, num2, num3 = 1;

Generate random number for num1, and num2.

For (int i = 0, i < num2, i++)

Num3 = num3\*num1;

Randomly select a number from 1-3

If 1:

num1 is replaced with x

If 2

Num 2 is replaced with x

If 3

Num 3 is replaced with x

Int correctAnswer = which ever was replaced with x

Generate 4 more non-duplicate numbers that isn’t the same as x;

Display (the 5 numbers on the buttons) //one of the numbers being correct

The correct button gives points

The incorrect buttons deduct points

**Input Output charts**

**Input | process | output ← format**

|  |  |  |
| --- | --- | --- |
| Scoring | | |
| -Answer choice  -currentPoints  -currentStreak  -difficulty | If answer = correctanswer  {  Points = points + difficulty + current Streak + 1  current Streak += difficulty  }  Else  Points = points - difficulty \* 2 - 1  Current Streak = 0;  } | Print:  Points  Whether or not it was correct (sound feedback) |
| Name input | | |
| Name from textfield | If name length > 10  Truncate to 10 characters  If there are special characters  Set errortest = true;  Set errorreason = “Unknown character”; | Display (errortest, errorreasons)  Display name |
| Menu screens | | |
| Button choice | Change currentScreen to most recent button clicked | Repaint with new current Screen variable |