**Bold** = function / module Hyung Park

*Italics* = variable

**6/6/18**

* Created a rough flow of **main** module for the program

**7/6/18**

* Created a rough flow for math skill modules (Addition, subtraction, multiplication, division)
* Asked teacher for a check on the rough flow and it was good enough to be used as plan so changed the name to Planning instead of rough flow. Added bit more detail into the planning such as loops or other specific modules

**11/6/18**

* Modified **Main** Planning so the program would not have any structural errors (in theory).
* Added planning for **greetUser** function to show how the program is supposed to greet user when the program is first opened.
* Removed Greet User on the planning for **Main** and changed its name to **selectSkill** so when each **learn(math skill)** function ends, it can refer to skill selection rather than from the start.
* Started coding program.

**14/6/18**

* Finished making a very first window of greet using a module **greetUser**. (IMG-1)
* Created Skill Selection window that will allow users to choose the type of mathematical skills they want to practice using a module **selectSkill**. (IMG-2)
* In the planning, I planned it to close window of itself and then run the program but it is difficult to close the window and then run the program so as a result, the following program will be opened which will include line of code above all to close the previous window.

**15/6/18**

* Because it was easier to have one window open per module, learn”Math Skills” modules had to be separated into three modules. Eg) **learnAddition** → **learnAddition** (Purely learning) + **tryAddition** (Question given) + **checkAddition** (Checks the answer).
* Programmed **learnAddition** window to tell the users what Addition is and how it works. (IMG-3)
* Programmed **tryAddition** window to give the users an addition question. (IMG-4)
* Programmed **checkAddition** window to check whether the user has inputted a correct answer or not and display corresponding window. For example, if the user has got the the right answer, the program will say “That is correct! Keep up the good work.” (IMG-5) and if the user has got the wrong answer, the program will say “Sorry, (*number1*) + (*number2*) = (correct answer). Keep practicing.” (IMG-6)
* For *number1* and *number2*, a random number generator randomly generates two numbers in a range of 1 to 10. And then they are stored in a variable called *number1* and *number2* which is then used on a question to give user the question and added to find the correct answer.
* Used a geometry function in every window to have program window open in a consistent place instead of having them open in a shifted location which was what was happening before using geometry function. Also, by using the geometry function, empty label functions that were used at the end of code to make margins could be removed because geometry function can decide the window size which will allow the margins to be created manually by having bigger sized window for the program.
* Tried to run the program but there was a slight error where *userAnswer* would not be able to be compared with *actualAnswer*. Did a research and could not come up to a conclusion. (If I have researched right, when I use *answer*.get() to get the *userAnswer* from input box in the window, the value should’ve been an integer which should be able to be compared with *actualAnswer.*) However, when I have added **int()** in front of variables to convert the values into integer when it is being compared, it did work and produced correct value. It is still mystery why it has to be converted into integer value when they are suppose to be integer but that has done the job and fixed the problem.
* After checking that everything works perfectly in Addition part of the program, the section was copy-and-pasted into subtraction, multiplication, and division section. There had to be changes made such as changing addition to corresponding math skills and changing the math formula to a corresponding one instead of addition for everything but after that, everything worked perfectly.
* Roughly finished the program but there are still some places to work on. For example, because I have used **int** function in division modules, the answers come out as integer. This means if the question gives us “what is ⅗?”, then the answer would be 1 because 3 / 5 = 0.6 and 0.6 rounds to nearest integer of 1.

**18/6/18**

* Asked the teacher to view the program and received a suggestion to make the program look more user friendly. Even though the program is already quite user friendly, considering the math skills this program covers, it must be year 4 student. (Knowing how to do addition, subtraction, multiplication, and basic idea of division) In order for them to successfully follow teacher’s guidance, it would be more helpful to have more user friendly program by having green buttons and bigger font which can lock the student’s attention and make them focus into the program which will be helpful to the learning.

**19/6/18**

* Changed the color of the buttons so the buttons are more intuitive (eg exit buttons are red and OK buttons are green) and attractive (bright colors on other buttons too).
* Tried to change the font and height for individual labels and buttons but quickly have realized that it is inefficient to do this way and it is way more efficient to create a variable that includes those values (**font** and **buttonHeight**) so later on if anything has to be fixed, it can just be done by changing that value.
* Have decided to remove division section out of program simply because the target users for this program is Year 4 students and they only learn basic ideas of division. (Eg, we have a total of 12 candies and there is four of us. How many candies should we each have?)

**20/6/18**

* Made the buttons and fonts bigger so it is much easier to read and focus for year 4 students.
* Changed the fonts of the buttons so it is easy to read.
* Changed the colors of the buttons so it is easier to distinguish between buttons which will allow teachers to easily guide students.

**21/6/18**

* Went back the diary and filled up any missing notes that was done on the program but not written in the diary. (Especially **15/6/18** because basic fundamentals of the program was completed in that date without writing notes.)

**22/6/18**

* Finished making fonts bigger and buttons bigger with color. I have used a color chart to find optimal color for each buttons (for example, redish for quit, greenish colors for ok buttons, etc…)
* Now that the buttons and the labels have gotten bigger, adjustments to the size of the window had to be made. Initially, window.geometry() function was manually added on each window where dimensions would be manually calculated. However, there was a function where tkinter can identify the screen resolution of the computer which then can be used to calculate where the screen would need to be in order for it to be in the center of any screen. So I have used it and now, only window width and height is required because the code can identify the screen resolution and identify where the application window should be.
* After testing with couple of people, there was a common behaviour appearing where they would all press enter key after they have entered their calculation values. In order to make the program more user friendly, I have found a function of window.bind(“<Return>”, **function**) which allows the users to press enter key and trigger function inside the bracket to operate. Because pressing enter to go to the next window is quite intuitive, it will be applied to every window to a most reasonable choice such as Ok except *selectSkill* window where a user selection is mandatory.

\*Forgot to take screenshots on this date so there is no photos on this date.

**28/6/18**

* Ran the program over and discovered a problem where the user would type a string value into the entry box and the program would just not do anything. In order to inform the user that they should not input strings but only integers, I have added a code where the program would make a window saying “Please enter a valid number” instead of doing nothing and confuse the user. This is done by detecting the TypeError and ValueError as when a string value is inputted, the lines of code which uses the value inside the entry box will make a ValueError or TypeError. By using this, detection of any values that aren’t integer became possible which would make the program more user friendly.
* Because I have used window = Tk() on both of my alert window for wrong input type and the actual question window, the program would either make the original window malfunction or close that window. So, I have changed *window* to *win* so Python doesn’t get confused on which window to close. This is applied on every math skills present.
* Adjusted any window sizing errors such as windows that are too small to fit every text in a line.

**29/6/18**

* Started writing the testing table for this program to check whether or not every function and features work as expected. Also, I was able to discover that this program does not have any responses for string inputs through this testing.

**1/7/18**

* Finished the testing table with 8 testing topics. Mostly, it was about whether the functions inside the program actually work.
* Wrote down the comments for this program to make sure anyone that is trying to read this program or modifying this program can easily identify where each modules or functions are.
* Finished the program & journal.

IMG (Images)

|  |  |
| --- | --- |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |