Project Report for Comp 2800: Mental Math Game

By: Ethan Saunders (photo here: <https://github.com/esaunder?tab=repositories>)

Student Number: 110079496

Date: April 7th, 2024

Project Report for Comp 2800: Mental Math Game

**Project Outline**

The objective of this project is to address the growing problem in our schools when it comes to keeping students engaged in their learning. Nowadays more and more teachers are saying that their students are performing much worse than they should be for their grade level due to the massive impact Covid-19 had on an entire generation of school kids. This is a serious problem as these kids not only struggle with the content but struggle to focus as social media, although it has many benefits, has negatively impacted their attention span significantly. This project aims to address this major issue in specifically the mathematics field which many find boring and uninteresting. By having kids improve their math skills through the medium of a game, we can potentially keep them more actively engaged in their learning, improving both their ability to focus on math problems, to think quickly, and to improve their basic arithmetic skills to at minimum the expected level for their grade.

Now with the medium of a game decided upon, the core gameplay loop had to be determined. Solving math problems can take some time so any kind of action game, platformer, or shooter was off the table as interrupting the flow of gameplay to answer math problems would result in many students most likely losing interest in the game and realistically math problems wouldn’t be able to be incorporated in any meaningful way in these games. The only realistic option was a turn-based game where the user chooses an action to take while fighting an enemy, answers a math question that is generated by the program, and then the resulting action is impacted based on whether they answered correctly or not. This incentivizes players to correctly answer the questions so they can more effectively progress through the game. A combo system and a max time limit would also reward consistency and speed so as students improve playing the game, they are further rewarded for their consistent success.

The program should have a simple turn-based combat system with each action requiring a math question being answered to successfully perform that action, a save/load feature so progress can properly be maintained, sound effects and music, a simple and easy to use GUI, a combo system, and a time limit for the questions.

All in all, the goal of this project was to create an engaging way for students to practice their mental math skills to help improve their engagement in their learning. With the general design of the game explained, the major use cases were then developed to determine the functionality that the game would need to successfully solve the problem that the program is meant to solve. From there, each feature was introduced until the project ended up in its current state.

**Features:**

*Saving the Game:*

The game can be saved any time and it will write important data to Save.txt in the same directory as all the files.

*Loading the Game:*

The save can be loaded upon startup and will return you to the previous point you were at.

*Difficulty Setting:*

The difficulty can be chosen upon starting a new game with the options Easy, Medium, and Hard with each subsequent difficulty having more operands in each question presented and the maximum/minimum value of those operands being increased (time limit is increased for harder difficulties to account for difficulty spike). This means that each difficulty is exponentially harder than the last.

*Sound Effects/Music:*

There is a wide variety of sound effects and music that plays throughout the game that can be heard in the demo video provided.

*Mute/Unmute Sound:*

The sound can easily be muted or unmuted by pressing the mute button (Figure 1: blue circle) which will subsequently turn into the unmute button.

*Player Health and Mana:*

The player’s current health and mana display in the top right and change based on damage taken or healed.

*Enemy Health and Image:*

The enemies have their health above their head and their sprite underneath that moves back and forth.

*Math Question Generation:*

The program generates math problems based on parameters provided to generate the appropriate math problem for the user’s difficulty level and what action they chose. It also generates the correct answer to compare to the user’s input later.

*Actions:*

The run action simply has a set chance to skip a fight with no math question asked. The other 3 commands will prompt the user with a question before they do the action they chose. Attack commands are only addition and subtraction questions, magic commands add in multiplication and division as well, and potion commands do addition and subtraction of squared values of the operands. This provides harder questions for more impactful actions but with an equally higher payoff. Magic use is limited by mp and potions are limited by the number owned (they are purchased from shops). If they answer the question, their action will succeed while if they are close to the answer they will succeed partially (half the usual effect with no combo bonus) and otherwise, the action will fail completely.

*Basic Animations:*

Very primitive animations have been implemented for the enemies and the attacks. They are something I would like to improve in the future.

*General Progression:*

The game is basically a text-based adventure as no open world to explore has been developed in this iteration. So, the program displays some text about what is happening, the user presses the continue button until the next battle begins and this repeats. There are shop sections which appear every now and then which allow the player to buy potions with gold they acquire from enemies. The player heals health and mana here as well (not a full heal). (Figure 5: the whole picture)

*Time Limits:*

There is a limit for how long the user can take to answer the math questions and if they surpass it, their action will fail as if they got the question wrong (the program does tell the user they timed out though). (Figure 6: red circle)

*Combo System:*

A simple combo counter has been implemented that increases by 1 for every correct answer a user gets in a row. It carries over between fights and is saved in the save file. When the user gets a question completely wrong, times out, or fails to run away their combo will be reduced to one quarter of its current value rounded down. If the user is close to the correct answer, they will have the combo reduced to one half its current value rounded down. This combo value increase damage from attacks by 0.25 the value, magic by the value, and healing from potions by 0.1 of the value (0.5 intervals).

*Randomization:*

After the game has been completed, the game can still be played, and a random enemy or shop will be chosen for the player. The enemies have their hp multiplied by 10 to challenge the player further. This allows for further playability until a more concrete story and open world are developed.

**Playing the Game**

Playing the game is very straightforward as it involves simply clicking buttons and entering numbers into text boxes so anyone can pick it up immediately and play without needing to learn complex controls.

**Running the Game**

As outlined in the GitHub repo, you need Java SE 8 to run this program and optionally an IDE like VS Code. You can simply compile and run the program from the command line with Java SE 8 installed by using the following commands:

javac mentalMathGame.java Music.java

java mentalMathGame

Alternatively you could open the project folder in the IDE of your choosing and run it with your IDE.

**Future Improvements**

Although this program does not have all the features that are required for it to be a successful solution to the problem proposed as many aspects are still in a preliminary stage such as the animations, it still accomplishes the goal of providing a more engaging way for students to practice mental math when compared to standard school practices such as doing worksheets. In the future, I plan to further expand on many aspects of this program to further expand its desirability as a game and to further expand the mathematics that are tested to widen the scope of the game’s usefulness.

One major aspect that needs to be developed is an open world where players can find things like dungeons and explore them to find treasure. This would expand the game from an on-the-rails text-based adventure into a more fleshed out turn-based RPG, making it significantly more engaging to players.

Another major aspect that needs to be changed and further developed are the assets and animations used. To actually distribute this program officially, non-copyrighted materials would need to be used for the music and images, so development of my own creative assets would be crucial (or finding copyright-free). The current animations are also basically placeholders until I develop my own sprite sheets with detailed animations in the future.

Further expanding the turn-based battle system and shop to allow for more progression with equipment and new attacks/magic would further gamify the math game to make it more enjoyable and would make it seem much more like a true turn-based RPG.

**Conclusions**

In conclusion, the program was successfully developed to solve the problem it was targeted to solve and has become a fairly engaging turn-based RPG. It provides an opportunity for students to develop their mental math skills by challenging them with a variety of questions at varying difficulties in a gamified format and it endeavors to make math more enjoyable to students who are currently struggling with mathematics (specifically arithmetic) or simply find math boring. I hope to further develop this program into a more polished version of itself in the future so that it might actually become a free tool for students to use to improve their math skills.