CS683 Project Assignment   
Math and Slash  
Joseph Monk

Instructions

* This is your final project report. I should include all information from the previous report as well as from this iteration.
* Please name your report as CS683\_<Last Name><First Name>\_<ProjectTitle>. It can be either a PDF or Word document.
* Please submit your source code in a zip file named CS683\_<Last Name><First Name>\_<ProjectTitle>.zip. You should create a zip file in AS ( using the Menu item File -> Export to zip file … ).
* Please provide your feedback in the “Add comments” section when submitting your lab report. Thanks!

[**Overview**](#_g6igqliy7rm) **2**

[**Requirement Analysis and Test Result**](#_9dheewbiht5g) **2**

[**Design and Implementation**](#_312k3b3li0xh) **2**

[**Project Structure**](#_hkcglxnjhrt2) **2**

[**Timeline**](#_qws9eve21c07) **2**

[**Future Work (Optional)**](#_92xr55qh0x78) **3**

# Overview

In trying to come up with an idea for a project I was helping my son with his math homework and thought that it would be nice if he had a fun way to practice more.  He actually enjoys math, so wouldn't really need to hide that it's math, just gamify it a little.  Based off older RPG/hack&slash type games with Pokemon style combat (sort of).

# Related Work

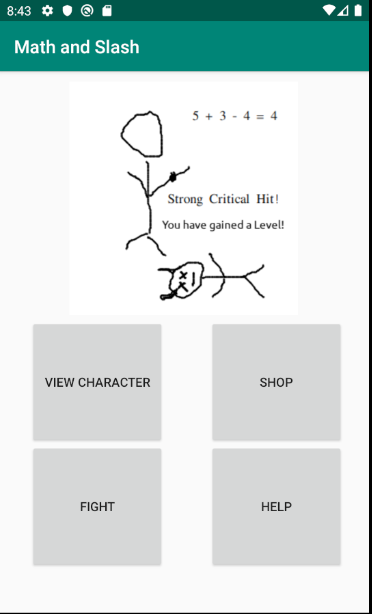
Math and Sourcery – Besides being much more developed than I am planning for this project, their battle system in similar in concept but quite different in application. I am planning on having each attack give a random problem, where this game assigns each enemy a number and operand.

<https://play.google.com/store/apps/details?id=one.gangof.mathsandsorcery&hl=en>

Fantasy Math Quiz RPG - Math Fantasia – Very polished game with a similar combat system to what I’m planning. This one appears to have both weapons and magic as well as multiple characters to your party. It appears to always use multiple choice, where I am planning that as the player progresses they no longer get the hints and instead have to type in their answers.

<https://play.google.com/store/apps/details?id=jp.lrstudio.mathfantasia&hl=en_US>

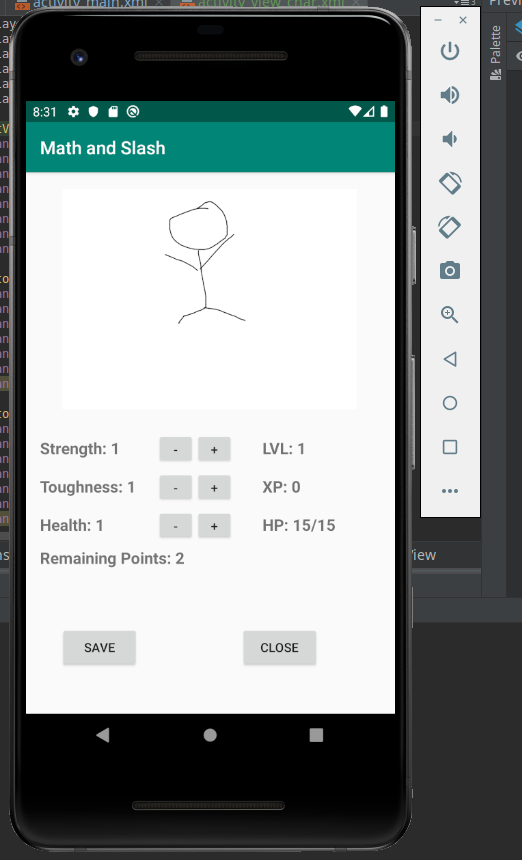
# Requirement Analysis and Testing

Base screen:

* 1. Essential Features:

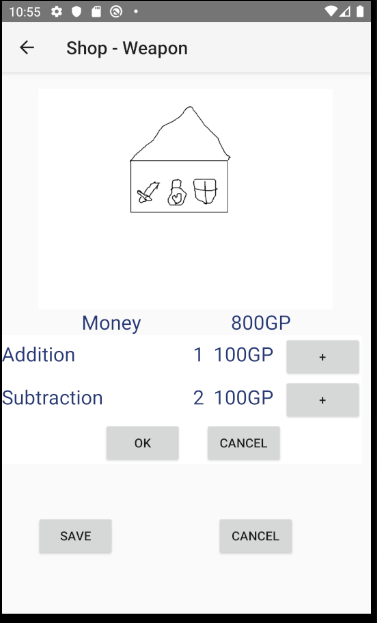
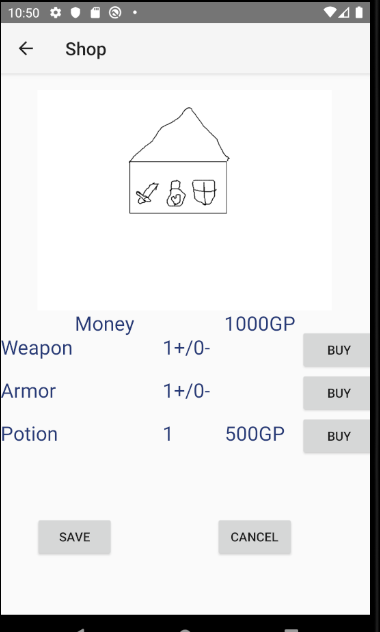
1. Player character with stats, health points, defense, level, xp, etc.

Testing: Player can press View Character button to see the various statistics. Player should be able to use “Points to Use” to increase individual stats (Strength, Toughness, Health). Additional points should be gained when XP >= MaxXP and a new level is gained.



2. Store to purchase attack (weapon) upgrades as well as defense upgrades and health potions.

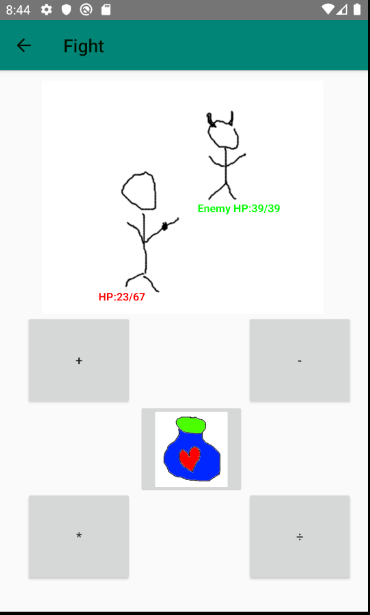
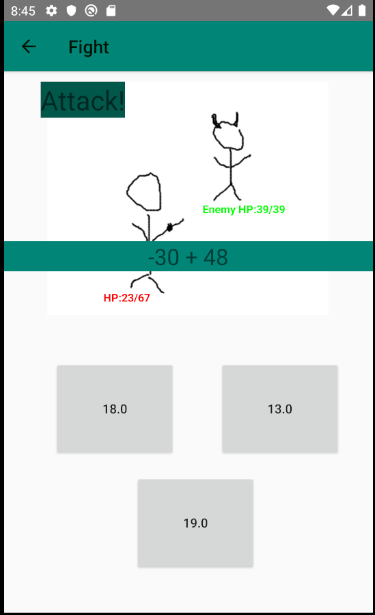
Testing: Player can click the Shop button to view the Weapon and Armor stats as well as purchase potions. Clicking the Buy button next to each item type should purchase that item (for potions increase by one, and subtract gold). For weapons this should open a new fragment allowing the user to purchase individual “elemental” stats (Addition/Subtraction).



3. Attack and defense will have “damage types” of addition and subtraction.

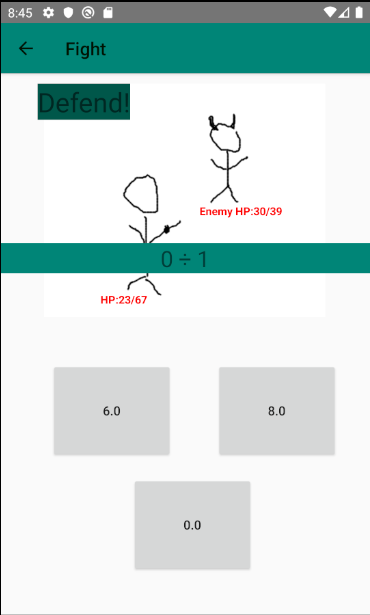
Testing: Displayed on Shop activity, and used during battles. On the Fight page the user should be able to choose the attack type they wish to use by clicking the appropriate button. Only attack types (Addition/Subtraction/etc) that the user has learned can be used. Learning is done through the Shop page.

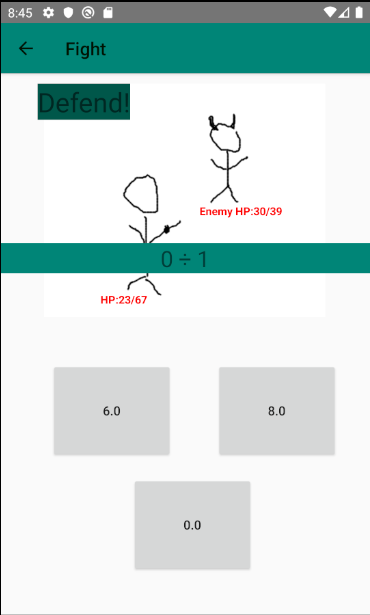
4. Combat will consist of you choosing a damage type then be given a problem of that type. Difficulty of the problem will be based off upgrades (bigger numbers, multiple numbers, etc). At lower levels you will be given 3 options to pick the right answer, at higher levels you will have to type.

 Testing: Partially covered in E3. As the user trains higher levels in individual attack types the problems should become more complicated. For example, if Subtraction reaches a certian level then negative numbers become available to use during combat. Every 7 levels the problem gains an additional component (ie 1 + 1 + 1 instead of just 1 + 1), to a maximum of 5.

After Player character level 5 (not item level) the player will have to enter the answer using the keypad. Level 5 and below will be given 3 answers to choose from by clicking appropriate buttons.

5. Defense will work in the same way, given a math problem to solve for defense.

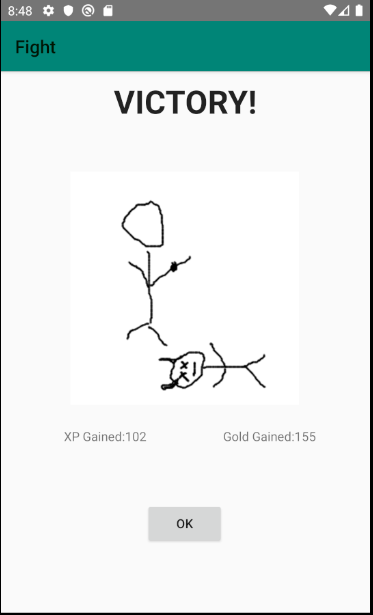
 Testing: After a player attacks the enemy should attack them. Enemy will attack with a random attack type, but only ones that the player also knows. Player will be given another math problem to complete, an incorrect answer leads to a “Critical Hit”.

6. Potions will be rapid fire simple math problems, given 3 options to pick the right one. Faster you answer, the more it heals.

Testing: Pressing the Potion button on the Fight screen pops up a simple math problem (max level 5, or the highest level the player has obtained if lower). Regardless of player character level this will always give 3 choices to pick from.

7. Winning a battle gives xp and money (to be used in the shop).

Testing: After winning a fight the player should get a victory screen showing the XP and gold they gained. If the XP gained gives them enough to gain a level they should be notified of this and the level gain processed.



8. Xp will level up the character, giving stat points to increase stats.

Testing: Combined with E7. Each level gives 2 additional stat points to use, and HP is based off level (and Health stat).

9. Attack/defense will flash the “battle screen” to show hit/miss and give the correct answer if you were wrong.

Testing: After answering a problem the player should see Miss (Correct answer: X) if they got the question wrong. In the case of defense this will be Critical Hit (Correct Answer: X). Other hit types include Miss, Glancing Hit, Hit and Critical Hit (all without showing an answer).

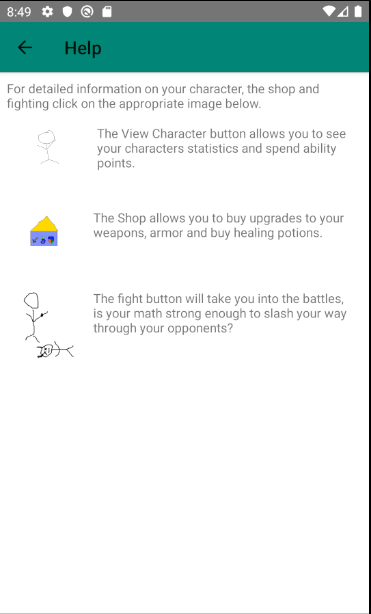
10. Full heal after battles.

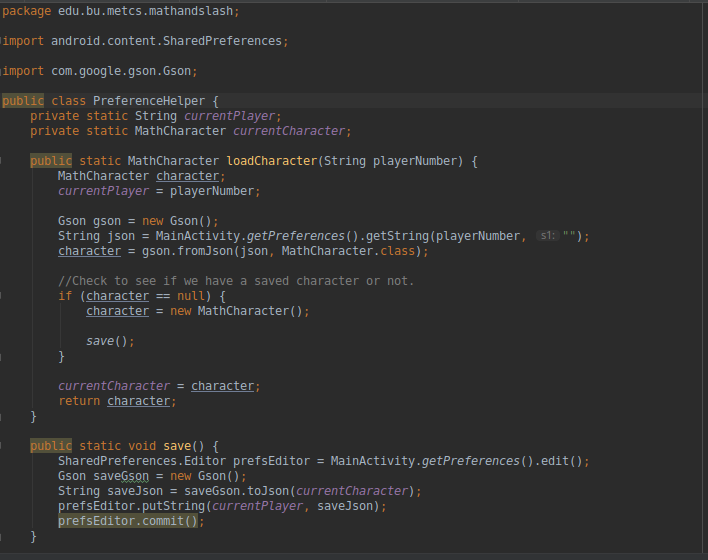
Testing: After a battle is completed, win or lose, player character is restored to full health. If the player “runs” from a fight (uses the back button) NO heal is given.

11. Saving game.

Testing: Game should save automatically as changes are made to the player character. Character should be loaded automatically upon launch.

12. Help files

 Testing: Pressing the Help button should display the help screen. This will include pictures and text for the major feature screens, linking to detailed help for those topics.

* 1. Desirable Features

1. Add multiplication and division as damage types.

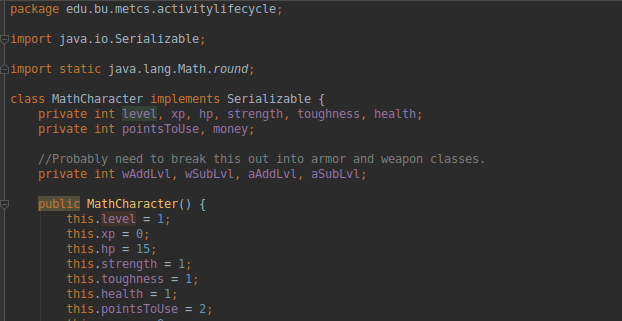
Testing: Additional attack types available. Testing the same as E2, 3, 4, 5, and 6.

2. Higher levels of upgrades can give combined damage types (addition and subtraction).

Testing: Once a player has 2 attack types at level 10 then combo attacks will randomly happen. Math problem will be constructed using the chosen attack plus at least 1 other type. For example, 1 + 1 – 1 or 1 + 1 – 1 \* 1. Order of operations is respected.

# Design and Implementation

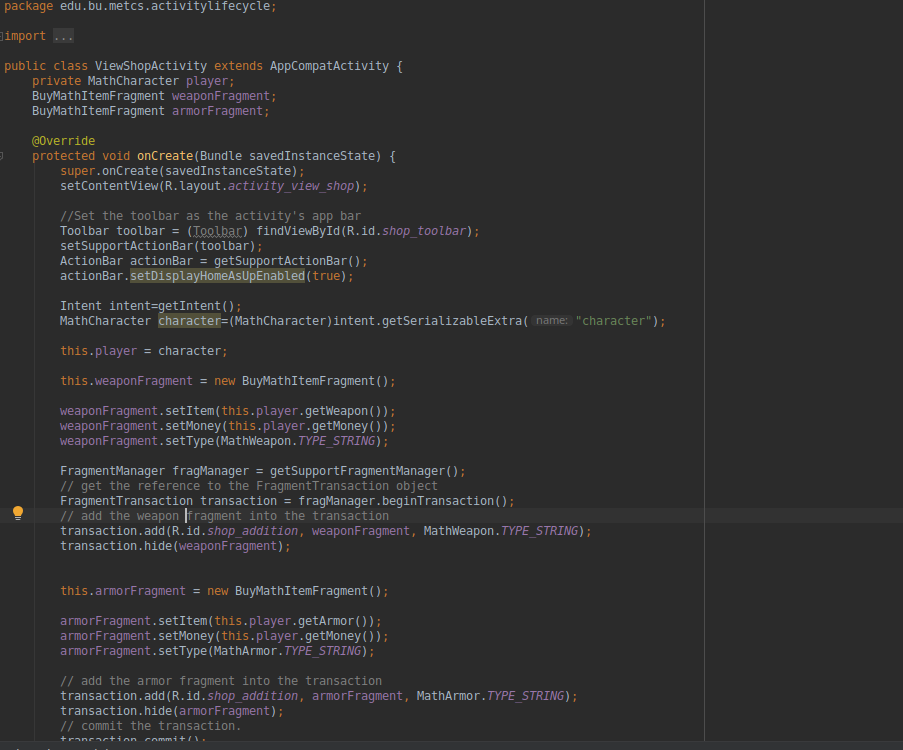
Searlize was used on the MathCharacter class to allow saving the character through SharedPreferences. PreferenceHelper uses Gson to store and provides a uniform way for acitivites and Java code to access the saved character.

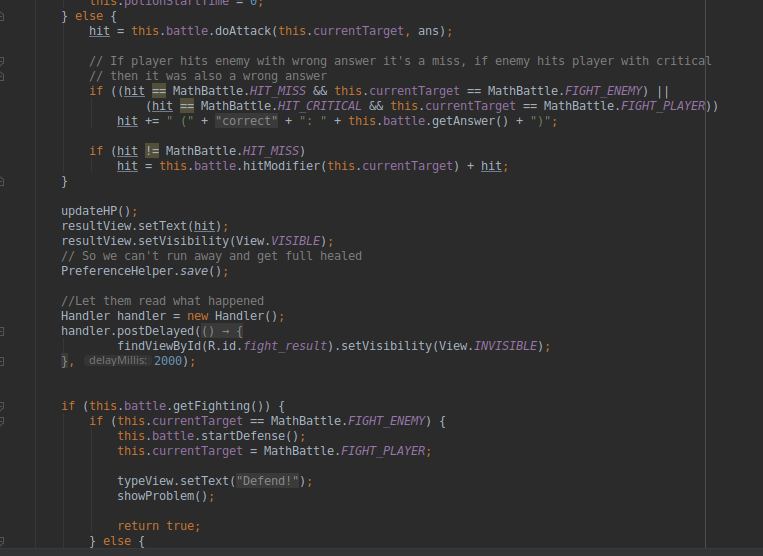


The main shop was done as an activity, with the weapon/armor upgrades as a fragment. There is a single fragment for any items (currently only weapon/armor) that has additional aspects (addition/subtraction currently) to purchase. It holds a type of MathItem (of which both MathWeapon and MathArmor are children) to keep it flexible.

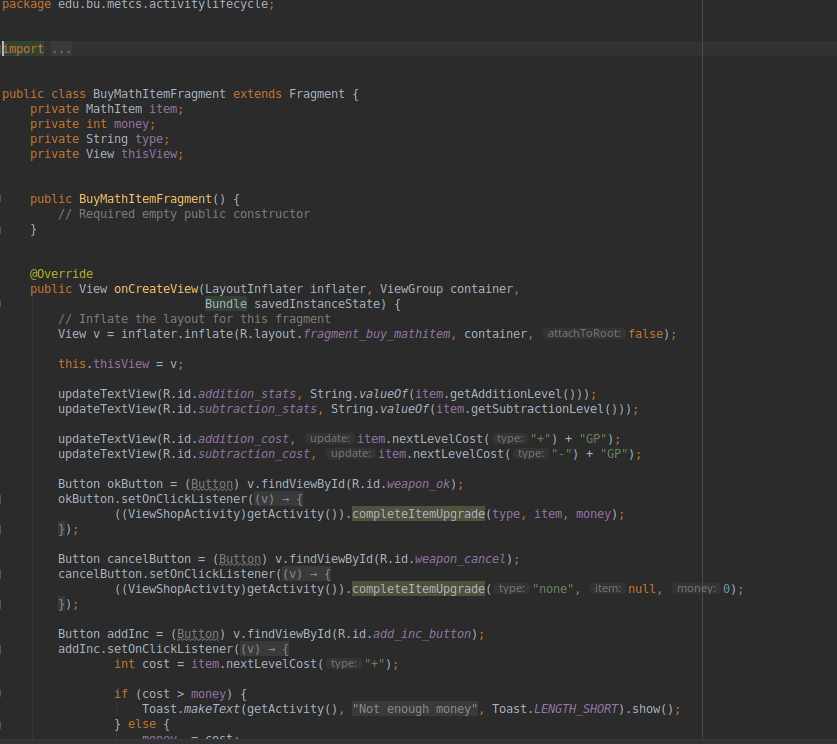
I also used toasts if the player did not have enough money for the purchase.

Here’s the top of ViewShopActivity:

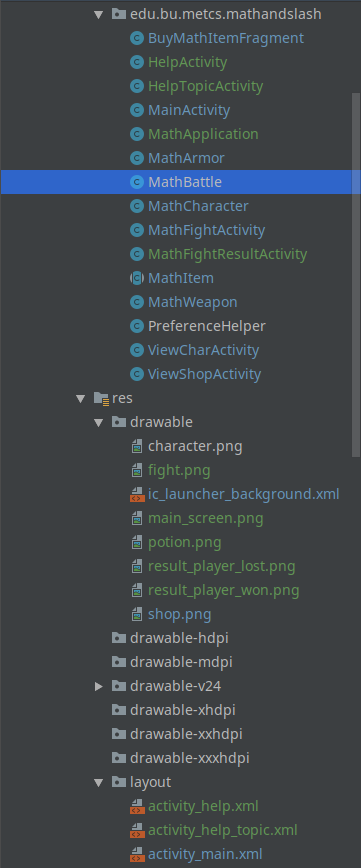


And this is the top of BuyMathItemFragment:

MathFightActivity utilizes a Handler to show the result of an attack for 2 seconds, allowing it to “flash” on the screen long enough to be read but without cluttering the screen permanently:



# Project Structure



This is the full project structure and as I you can see in this last iteration basically everything was touched in some way. The main work was done in MathBattle, MathFightActivity and MathFightResultActivity. But bug fixes and balance adjustments required changing something in nearly every aspect of the project. Additionally HelpActivity, HelpTopicActivity, and MathFightResultAcitivity were newly created to handle the results of fights and the help files to explain how the game works.

This last iteration completed E4/5/6/7/8/9/10/12 as well as D1/2.

# Timeline

|  |  |  |
| --- | --- | --- |
| Iteration | Application Requirements  (E/D/O) | Android Components and Features |
| 1 | E1 | ConstraintView, GridView, Searlize, Intent |
| 2 | E1/2/3 | Activies and functional code. |
| 3 | E11/E4 | Activities, Files  Finishing up any aspects incomplete from previous iterations. Saving game. Code cleanup. |
| 4 | E4/5/7/8/10 primary, E6/9 secondary  D1/3 | Activities and graphics, polishing and tweaking the numbers to make sure progression is correct.  – Ensure all essential features are complete, then work on expanding and further polishing |
| 5 | Combine to finish E4/5/6/7/8/9/10/12  D1/2 | Integrate everything to complete the fights, create functionality for portions. If possible do combined problems and add mult/div.  Activities, Threads(Handler), HTML. |

# Future Work (Optional)

These are some features that I had hoped to implement but was not yet able to do. One major piece of future work is to improve the UI/UX, as it is functional but very basic.

* 1. Desirable Features

3. Animations for the fights instead of static pictures.

4. Healing as time (real time) based instead of full heal after battle.

5. History for problems so you can see which types you need to work on.

6. Time usage feature (so parents can see how much time was spent playing).

7. Multiple enemies available instead of just 1v1.

* 1. Optional Features  
     1. Algebra as an unlock at higher upgrade levels.

2. Add in more RPG like features (story, map, etc).

3. Mutilplayer so you can battle your friends.

4. Parental controls

# Project Links

**Project Presentation Kaltura Capture Link**

Github: <https://github.com/CS683/cs683-project-valsacar>

Video is loaded to GitHub as Kaltura does not support Linux which I use.

# References