



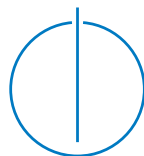
DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master Thesis in Informatics: Games Engineering

**Mathstation - A Serious Game for
Mathematics**

Jonathan Borowski





DEPARTMENT OF INFORMATICS

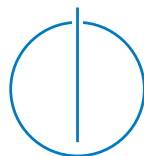
TECHNISCHE UNIVERSITÄT MÜNCHEN

Master Thesis in Informatics: Games Engineering

Mathstation - A Serious Game for Mathematics

Mathstation - Ein Serious Game für Mathematik

Author: Jonathan Borowski
Supervisor: Prof. Gudrun Johanna Klinker, Ph.D.
Advisor: Dr. David A. Plecher
Submission Date: July 15th 2021



I confirm that this master thesis in informatics: games engineering is my own work and I have documented all sources and material used.

Munich, July 15th 2021

Jonathan Borowski

Acknowledgments

In loving memory of Miki Borowski.

I would like to thank Ms. Daniela Lederer and Ms. Hao Xu for their support.

Abstract

Mathstation is an educational serious game in the field of mathematics. It contains three distinct game modes and features math problems for the players to solve while playing. The problems are generated by a configurable math problem generator, that covers math topics relevant for the first through fourth grade of elementary schools in Bavaria, Germany.

Due to the recent COVID-19 Pandemic, schools have had to adapt in order to keep providing education for their pupils, resulting in distance learning and classroom restrictions on site. These result in a learning loss for pupils affected by it. Serious games have been shown to have significant learning yields in scientific school subjects when compared to conventional teaching methods when applied properly. As such, *Mathstation* makes itself available as a supplemental learning activity, utilizing the educational benefits of serious games as well as the availability of online distribution.

Although it was not possible to test *Mathstation's* learning yields in proper conditions due to the pandemic, methods of testing have been studied and prepared for eventual future use.

Contents

Acknowledgments	iii
Abstract	iv
1. Introduction	1
1.1. Background	1
1.2. Serious Games	1
1.3. Proposition	2
1.4. Plan	3
1.5. Motivation	3
2. Related Work	6
2.1. Serious Games as Teaching Tools	6
2.2. Math Content	7
2.3. Inspirational Games	8
3. Game Concept	13
3.1. Content	13
3.2. Art Style	15
3.3. Story	16
3.4. Progression	17
3.5. Technical	18
4. Math Editor	19
4.1. Introduction	19
4.2. Problem Generation	19
4.3. Input & Evaluation	20
4.4. Class Overview	21
4.5. Expandability	24
5. Game Modes	26
5.1. Fighting	26
5.1.1. Introduction	26

Contents

5.1.2. Controls & UI	27
5.1.3. The Bosses	28
5.2. Mining	30
5.2.1. Introduction	32
5.2.2. Controls & UI	32
5.2.3. The Gadgets	33
5.2.4. The Hazards	35
5.3. Exploration	36
5.3.1. Introduction	36
5.3.2. Controls & UI	38
5.3.3. The Artifacts	39
5.3.4. The Anomalies	40
5.3.5. Levels	41
6. Evaluation	45
6.1. Reason	45
6.2. Method	45
6.2.1. Game Assessment	45
6.2.2. Math Assessment	46
6.3. Testing Conditions	47
6.4. Unavailability of Tests	48
6.5. Summary	49
7. Future Work	50
7.1. Customization	50
7.2. More Base Features	51
7.3. Story	52
7.4. Multiplayer	53
7.4.1. Fighting	54
7.4.2. Mining Game	54
7.4.3. Exploration game	55
7.5. Hub Area	56
7.6. External features	57
7.6.1. Instructor Interaction	58
7.6.2. Class Integration	58
7.6.3. Data collection	59
7.6.4. Evaluation	60
7.7. Beyond Schools	60
7.8. Summary	60

8. Conclusion	62
A. Math Questions	64
A.1. Base Configuration	64
A.1.1. 1st Grade	64
A.1.2. 2nd Grade	64
A.1.3. 3rd Grade	65
A.1.4. 4th Grade	66
A.2. Multiplication	68
A.2.1. 2nd Grade	68
A.2.2. 3rd Grade	69
A.2.3. 4th Grade	69
A.3. Division	71
A.3.1. 3rd Grade	71
A.3.2. 4th Grade	72
A.4. Fractions	73
A.4.1. 3rd Grade	73
A.4.2. 4th Grade	74
Bibliography	75

1. Introduction

1.1. Background

The inception of the thesis occurred during the COVID-19 pandemic, in the summer of 2020. It was during that time, that the indirect damage caused to school systems around the world made itself abundantly clear [20]. All across the globe, pupils were confined to stay at home. Alternatives did make themselves available, such as home schooling and online learning sessions via internet technologies [20], however, it has been widely perceived to be inferior to on site studying [20]. The damage caused by absence from school has also had some additional effects, such as an impact on a generation's mental health [10].

1.2. Serious Games

In order to fully delve into the proposition of this thesis, it is important to thoroughly grasp the concept of what exactly a serious game is. This section will explain what serious games are, and what they are capable of in the educational sector.

"Serious games are generally defined as digital software the primary purpose of which is learning rather than entertainment" [33]. It is an approach in education that makes use of technological advances (within the context of video games) in order to teach via game like experiences, allowing the traditional class room to be supplemented with engaging experiences for learning, practice and skill retention of any imaginable discipline.

By combining the technological prowess of modern video games with educational content relevant to studies or practices, real life simulations can be built which would place the player in an environment similar to a real life situation, where they can train. These solutions have been found to be successful with science based subjects in particular [33], and it is for that reason along with their accessibility and availability, that they are considered a potential candidate for the relief of the aforementioned situation.

1.3. Proposition

Leading up to this crisis, the learning benefits offered by serious games in the field of education have been increasingly recognized as valuable [33]. In order to best withstand the pandemic, while keeping the educational gaps caused by it to a minimum, making use of this technology will provide alternative learning activities available within safe social distancing circumstances.

The aim is not to completely replace the educational activities provided by the school systems, hindered as they may be during the crisis, but rather to offer supplemental learning activities to keep the pupils engaged and trained to better combat the downsides of distant learning, or otherwise inconsistent school activities.

Hence, I would like to introduce the math serious game which is the subject of this master thesis: *Mathstation*. This learning game has been conceived as a direct response to the school outages of the COVID-19 pandemic. It will feature mini games with engaging activities, and math solving as part of the core gameplay.

The novelty therein lies not within the concept of a serious game in the educational field, as those have been around for a while now [33]. Where this game aims to set itself apart is by striking an optimal balance between the fun aspect of playing an engaging video game and being a math exercising tool, all in one. As we can later on see in the related work section of the thesis, while educational games are abundant, they often lack the aforementioned balance, which is set as a goal in this case.

Additionally, some of the features which are considered for future work on this project will attempt to push the limits of what a math oriented serious game can achieve. The most notable of the features considered are social elements. These features, which have been present in some of the related works, are able to elevate the game experiences and enhance the learning yields. Be it through peer to peer interaction, multiplayer modes with cooperative or competitive play, the engagement [34] earned through the experiences may result in a greater learning result. Instructor interaction, on the other hand, can be a powerful tool in such an environment. It's integration within a serious game allows the instructor to carefully monitor a pupil's progress, to flag any subjects with which they might be struggling, assist by means of in-game tools, and set in-game rewards and achievements for their groups. Truly, the possibilities for this feature alone are limitless, and it might offset the shortcomings of the distant learning methods.

1.4. Plan

During the working time of this thesis, a math based serious game will be developed via Unity game engine.

The game will have at least three separate and unique game activities to offer, each one leaning on a formerly established staple genre in the video game industry.

It will feature a configurable math problem generator, which can generate math problems from a wide assortment of subjects and difficulties. Players will be able to solve math problems as an organic part of the game play loop in their chosen activity.

Originally, a study has been planned for this thesis. It would have shed some light on the game's assumed ability to provide some relief to the situation, as well as it's lasting value thereafter. Alas, it was not possible under the circumstances. Please refer to chapter 6 for more information.

Each of the separate game modes has had a month allocated to it's development time. This includes the coding of game logic as well as art asset creation. In addition, the math problem generator and interface, henceforth referred to as the math editor, had a further month allocated for it's development time. The final two months are then allocated to the possible study (which did not take place) and for writing the thesis, as well as some improvements, bug fixes and polish to the game.

More features that are perceived to be useful for such an undertaking have been accumulated for the purposes of future work documentation and recommendations. These can be found in the future work chapter, chapter 7.

Research question: Will the introduction of this serious game to a school class have positive effects on the learning outcomes during the school outages?

1.5. Motivation

Personally, it was the restricting of the school systems in two countries, Germany and Israel, which was the most notable during the preparations for this project, eventually leading to the suggestion of this subject. Myself, being a half German half Israeli student, I am very concerned, at a personal level, for future generations. Those who are subject to the pandemics restrictions, and suffer interruptions to their school education as a consequence, will need a way to continue their studies in spite of COVID-19.

The months leading up to the start of the thesis were filled with news about changes in plans regarding school accessibility and availability in both countries [16] [37]. With little time from announcement to execution, the educational system of either country

would close down the schools as a last resort, or reopen them when the conditions permitted it. It happened frequently over that time period.

This was truly unsettling to witness. As the events of the year 2020 unfold, parents have to explain to their kids every so often why they would now be unable to go to school or why they should return there.

Education is the cornerstone of civilized modern life [8]. It is through education that we are able to progress and advance in life, find a place for ourselves, acquire a profession, hobbies, and many more things that life has to offer. It is my personal opinion that the preservation of the educational systems is a high priority during these trying times. The future of the next generations may as well depend on the school systems being able to pass on their teachings to the current pupil generation, despite being confined to remote learning methods. Any compromise made to their education could have disastrous repercussions.

Acknowledging all of the above, it is my personal wish to provide this situation with some relief and alleviate future consequences in any way possible. Putting to work the knowledge and education acquired with the games engineering studies, I do firmly believe that the implementation of the educational serious game is at the very least a step in the right direction. Teaching math is fundamental, as math is considered part of the core education system worldwide [28]. While it is not the only important subject that deserves an undertaking of this sort, it is the basis for all sciences, and should receive top priority amongst the various subjects that need to be addressed.

While less concerned with providing entertainment for children, considering that the gaming industry keeps growing steadily [11], I do believe that the serious game solution is a decent attempt to bring more learning options to the spotlight. If the game proves to be fun enough on its own, it might even compete with other game products for the children's attention.

During the earliest stages of inception, the perceived solution took the form of a lightweight mobile application containing math questions and solutions. A self exercise tool that would somewhat resemble *Duolingo* [26]. In this version of a product, more akin to an educational software than a serious game, a user would be able to pursue different math skills and exercise them. Extra nice to have features would have been achievement badges and daily exercise streaks and the like.

After further elaboration however, that vision has changed substantially. In order to best pursue a solution for the education problem, considering the target audience, the success of serious games so far and my personal project experience with game engines, the solution should take the form of a more immersive experience. A serious game. It's entertainment and education, bundled into one package. Something a bit more exciting

1. Introduction

for the kids to educate themselves with while stuck at home.

Bearing this in mind, my instructor and myself have agreed that the game must cater to a wide target audience. After all, games come in many genres, shapes and sizes. It would be difficult to call it a job well done if the resulting game is fun for a select few. Further discussion has led to the iteration of the game featured here. That is, a collection of mini games. A game containing several modes of play, each it's own small version of a well known game play loop that has had tried and true products released over the years to great acclaim, financial success or both.

In order to connect the games with an overarching theme, a game setting and a world at the background, they will all feature sci-fi elements. I believe that its a matter of flavour, it could also have been something else. I would, however, argue that the subject of sci-fi and space in general is a good fit for educational games, doubly so if the education candidates concern themselves with natural sciences. More so still if it features math, which is the base of other sciences.

The games themselves will entertain, the math in them will teach and train, and it is with this firm belief in mind that developing is continued. It would be an exciting experience to have had a positive effect on the educational gaps torn wide open by the pandemic.

2. Related Work

This chapter denotes any significant pieces of related work. When undertaking a project such as this, it is important to form a strategy that will ultimately achieve the goal. In order to succeed, and to avoid unnecessary false assumptions, related work from all relevant areas and disciplines is studied and observed. These shape the base of knowledge that guides the decision making process, lays the conceptual foundation of the implementation, and can even quantify the learning benefits of the product, which would roughly measure how valuable a future work could be.

2.1. Serious Games as Teaching Tools

The main piece of related work, which has not only helped establish most of the design decisions concerning the game, but has also reaffirmed the success of serious games in the educational sector, is **Impact of serious games on science learning achievement compared with more conventional instruction: an overview and a meta-analysis** [33]. This data rich meta analysis has carefully collected and analyzed different configurations for serious games as teaching tools. It spans over decades, and has an impressive variety of research hypothesis that reach out to the elements comprising the serious games themselves.

Questions addressed include, but are not limited to the following:

- The learning yields of serious games compared to conventional methods [33, p. 176].
- The variance of the above for some scientific disciplines [33, p. 195].
- The duration of the play session's effects on learning outcome [33, p. 199].
- Passive vs active learning outcome [33, p. 184].
- The effects of the degree of control of the player for learning [33, p. 200].
- Game's visual art style effects, as well as ludic or story related content [33, p. 196].

A summary of the conclusions taken into account for *Mathstation* are:

- Duration of play should be flexible.
- Art style should be kept to a mostly schematic visual (basic visual fidelity).
- The player should have a large amount of freedom to choose the content, and control over the level of difficulty.

Mapping learning and game mechanics for serious games analysis [4], another noteworthy article, studies the interrelations of the serious game mechanics and their related game mechanics and learn mechanisms. The resulting product is a learning mechanics game mechanics framework, which serves the purpose of verifying that the planned game elements match the underlying learning aims [4, p. 16]. The findings are made use of in order to break down *Mathstation's* components in order to acknowledge the usefulness of it's game design patterns in the educational environment.

Another considerable related work, **Wearable Learning: Multiplayer Embodied Games for Math** [5], exemplifies the value of multiplayer experiences for educational serious games. Additionally, the pre- and post test method of evaluation has been adopted for this theses' (theoretical) study. It has proven to be invaluable in the determination of the quantification of the learning value of the game activity, and is therefore recommended for that reason [5, p. 209].

2.2. Math Content

Mathstation's actual math content is based off of the regional school books. In order to support the game's goal, teaching math, the source from which the question's content and style are derived should adhere to the official school program.

Firstly, "**Welt der Zahl 2**" [32], a German elementary school math book. This book's content in particular is a good source material with the game's intended math content, as the target audiences overlap significantly. Hence, problems in the game should be in the scope and spirit of the book's contents.

Secondly, "**Lambacher-Schweizer. Mathematik für Gymnasien, Bayern**" [6], a Gymnasium grade math book. While the actual math content within is considered to be out of scope for this project, this book contains plenty of examples of self test assignments, that are used to test the pupil's understanding of the previous chapters. It is for that reason that this book is also considered when implementing the game's math content.

In addition to school text books, the website Help Teaching [13] has been quite helpful in finding a rich variety of math related content. The site, owned and operated

by Sunstone Education, contains a plethora of questions of K12 programs, including but not limited to mathematics. Conveniently, the mentioned questions are categorized by their class and type of content. Accessible to premium membership accounts only, the premium questions offered on the website follow the common core math standards, an education initiative in the US [31].

The questions found in appendix chapter A have been taken from this site. In order to group the different tests according to different math configurations within the game, having a large catalogue of questions that adhere to some established standard at the ready has been beneficial. More info about testing can be found in chapter 6, evaluation.

2.3. Inspirational Games

The following games, educational or otherwise, have acted as an inspiration in some way, shape or form to *Mathstation's* inception. The reasons range from aesthetics to learning mechanics. They each bring their own unique attributes to consider.

While **Papers Please** is not an educational game, it does present a very minimalist art style [24]. Rather than realistic visuals, the game opted for pixel art style graphics. It is a prime example for serious games that retain their sense of purpose throughout the game's graphic direction. Especially when confronted with the game's narrative setting: A dystopian yet realistic world, featuring immigration and bureaucracy as its game elements.

When considering the game's content, one could assume that the sense of realism would be supported by a realistic art direction, attempting to depict photo realistic people and situations. Not such case though with *Papers please*, as the game's pixel graphics were chosen for the presentation. While pixel graphics are somewhat stylized, and as a consequence, shy away from realism, they are not necessarily cartoon like. And such is the case with this game, as the sense of realism is not at all inhibited by the pixel art. Please refer to fig. 2.1 for a visual representation of the discussed art style.

What makes this game serious, though, is the moral dimension in which it operates. The player assumes the role of an immigration officer stationed in a fictional yet busy border between countries after a war. There are clear immigration protocols that the player must follow when approving immigration documents and granting access, or otherwise when declining documents and denying access to the country. While this concept may sound simple to the unassuming player, making the choices will have implications on the in-game characters. While it may give the impression that following the rules is the correct procedure, it might also inflict suffering on the characters at times. Choosing to bend the rules may alleviate some of their distress at times, however,

2. Related Work



Figure 2.1.: Papers Please

it must be done carefully, as a violation of the protocols will come at the player's expense.

As the game progresses, the protocols and instructions become an ever more complex guideline to follow, and the players are faced with harder to make decisions than before. This invokes the altruistic morality aspect of the game. Do I risk my own well-being for others' sake? The game's ability to invoke action through feeling rather than instruction is what sets it apart as being serious.

What this means, concretely, for *Mathstation* is: Even though the game revolves around serious topics, the art style manages to convey it. It is possible to do something serious, with a minimalist art approach. In turn, this conforms to the findings of Riopel et al. [33], that schematic like graphics are the best for the purposes of educational serious games. It has an additional interesting side effect, however. Since pixel art is relatively easier to produce, it can assure a greater focus on the game's content and functionality, rather than consuming precious development time on elaborate art assets that are of lesser educational value.

Transitioning away from pure aesthetic resemblance and into more educational content related substance, the serious game **HieroQuest** [30] is the first of several games to contribute to the inception of *Mathstation* as an educational serious game.

HieroQuest is a serious game in the field of ancient Egyptian Hieroglyphs. It places the player in the Story of the Shipwrecked Sailor, an ancient Egyptian tale [22], bringing it to life as a 3D maze like environment filled with Hieroglyphic puzzles to interact with. Solving the puzzles rewards more learning content, such as additional hieroglyphs to study, which in turn are used in the upcoming puzzles as problem solving elements. This loop of learning value as a reward, which is required in the subsequent levels, makes *HieroQuest* stand out as an example of a serious game blending it's game activity with the educational content in a seamless, non intrusive, and immersive manner.

The study conducted with this game had set out to find out whether *HieroQuest* would have any meaningful learning results within it's own realm of hieroglyphic studies. Utilizing a combination of pre- and post test questionnaires, as well as the game experience questionnaire [17], the study yielded the conclusion that the game was able to teach ancient hieroglyphs. Perhaps most notably, students of the subject expressed the tool's usefulness as well as their wish to make use of it during their studies - "Further, many of the Egyptology students stated that they considered the game to be useful for their course of study and they want to continue to use it throughout their semester as an additional tool" [30, p. 18], hinting at the potential that serious games can have as a supplemental learning tool.

The game's presentation is in 3D, but still simplistic. It opts for a low polygon style graphics, which keeps the visuals clean and allows the player to focus on the tasks at hand. Additionally, the game features a plethora of ancient Egyptian themed props and styles. This intensifies the game's world building, and in turn, sense of immersion.

Of all the non math games covered in this section, **TwilioQuest** [38] is the closest of the bunch to the vision of *Mathstation*. The reasons for this are numerous, spanning from it's simplistic art style, to the focus of education in the field of science specifically. That is despite the fact that it deals with computer science, rather than mathematics.

TwilioQuest is a coding based serious game with computer science related topics. It allows the players to interact with the space themed world around them, go on missions and utilize an in-game text editor to solve code themed puzzles by coding appropriate solutions, simulating an actual programming environment. The game also incorporates elements of the role playing game genre, such as avatar customization and equipment.

The style and incorporated elements seem to, again, align themselves conveniently with some of the findings of Riopel et al., namely:

- Schematic art style has a positive effect on learning outcome for serious games, again [33]. Please refer to fig. 2.2 for a visual reference to this visual style.

2. Related Work

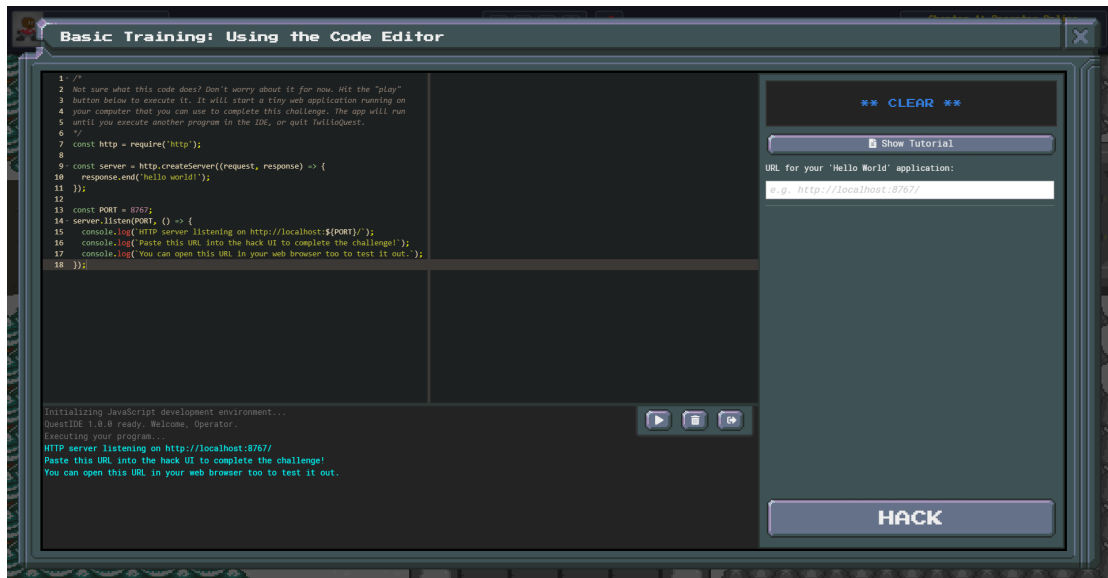


Figure 2.2.: TwilioQuest

- A degree of control also contributes positively to the learning yields [33].

Notably though, *TwilioQuest* also involves a decent amount of ludic content, such as a story and custom character, which does not conform to the findings of Riopel et al. [33]. It does, however, make the world and the game activity itself more immersive, which could be valuable in its own right.

The most notable single piece of educational serious game software, however, is **Prodigy** [18]. Whilst searching for math based games, the yields were initially not very impressive. Although it is true that there is a lot of variety to be found, some of the games left much to be desired, as these can often be limited in their scope, offering bite sized activities with limited room for substantial immersion, interaction and overall depth of the experience. While capable of exercising in their own right, the abundance of these smaller experiences, reminiscing of the flash browser game era, does invite the games of the bigger scope variant, offering more than a temporary distraction.

The reason for *Prodigy's* noteworthiness is that the game may as well have been created for the same purposes as *Mathstation*. A math based educational game with the aim of teaching through fun and engagement, aiming to balance the fun and immersion with the education. This game, however, does not cut back on the sense of scale or production values.

Prodigy can be classified as an online fantasy role playing game that is run on a

2. Related Work

browsers and mobile devices, and is also free to play. Anyone can sign on and make a student account. In the game world, the player controls a customizable avatar.

The game itself is a 2D top down view role playing game with a turn based combat system, colorful characters, dialogues with full voice over narration and multiplayer elements. The story tells of a new student in a school like institute that must now stand up to a powerful puppet master, who seeks to keep students from learning as a way to weaken them to their mind control spells.

The math makes a debut during combat, where the player must solve math problems in order to successfully execute their attacks. The math featured in the game is derived from actual curricula, and is subject to change from country to country, which the game itself accommodates during account creation.

Additionally, the game has accounts for parents and instructors, each serving a different purpose in the game when they interact with the student accounts. These interactions take this game a step forward from the rest of the relatively smaller experiences that are serious math games, and elevates it's potential. The possibility to interact with not only peers but also instructors and parents to a certain degree is of great value to the pupil's learning capabilities according to the game's site, be it through setting in-game goals or through monitoring, as well as an abundance of other artifacts, such as printable exercise sheets and video lessons [18].

The game does have a typical free to play monetization in place [18], wherein some game features can only be accessed with a premium status, obtained via a paid subscription. This model has advantages and disadvantages. It is inclusive, anyone with access to an internet browser can pick it up and start playing, immediately getting immersed in the world of the game and practicing math while doing so. It does, however, split the player experience into free players and premium players, which can have adverse effects on the enjoyment and the sense of fairness [23, p. 5] in a multiplayer environment. The influence thereof may or may not seep into the educational value, which might have a greater impact on the effectiveness of this game as an educational serious game.

Apart from being space themed with sci-fi-esque activities, *Mathstation* can set itself apart from Prodigy by avoiding any such subscription based models, leveling the playing field to all users.

3. Game Concept

This chapter details the game's initial design concept as well as the overarching elements found within the different game modes.

3.1. Content

Mathstation is a minigame collection. The game offers three minigames as playing activities:

- Fight - Action boss fighting game
- Mine - Strategic resource management game
- Explore - Exploration game and collectathon

The reason for this design: It should cater to a wider audience than any of the games could have done so on their own. Teaching math should not be hindered by personal taste, and so, the game's initial plan had it split into multiple activities, smaller in scope than a fully fledged one.

The activities are diverse enough in their nature and play elements to warrant such a design. These are, in essence, small experience from corresponding genres [3]. The fighting game, for example, features action style game play elements that are associated with action games and on rails shooters. As such, it will appeal to the audience most enjoying this kind of play, prioritizing quick reaction times and aiming capabilities rather than strategic thinking or elements found within the other modes.

All of the activities, which will be covered in greater detail in the following chapters, will have math problems interwoven in them. For this, a math editor component will be a core part of the game. This math editor is capable of generating math problems and allows users to solve them interactively. This editor will be described in greater detail in its own chapter. Additionally, the math editor has, since inception, been planned to be available as a standalone experience. Since it is already implemented with the incorporation into other game modes in mind, the product itself should also be available to anyone wishing to hone their math skills free of any game elements around them.

3. Game Concept



Figure 3.1.: Game Loop

If the player decided that they would like to engage in some straight to the point math practicing, without the game activity around it, the math editor lends itself for that purpose. Selecting to practice in the game's main menu will allow them to practice math on the editor with no other spectacles, distractions or restrictions. The idea behind a feature such as this is to allow a greater amount of autonomy in the learning experience, as well as a greater degree of control. This decision does conform to the findings of Riopel et al. [33], stating that a higher degree of control has had a positive effect on learning yields of science based serious games.

The point of this combination of diverse game activities and interwoven math problems, is that the former enables a fully immersive game play experience, and facilitates a reasonable in game narrative reason for the latter. This attempts to train the math of the players as an organic part of a game activity while avoiding intrusions to the game immersion, and maintain a sensation of flow [9]. This assumption is deeply related to *Mathstation's* goal of setting itself apart from existing math games. As has been disclosed in previous chapters, some of the existing games in the educational field, specifically in math, fail to immerse the player in a deep game experience, due to scope or otherwise lack of refinement.

It is by design that any of the mini games featured in *Mathstation* involve several instances of math solving during play. This method ensures that the sessions not only have the math subjects involved in the games, but that through repetition, the subjects can be learned as a means to pass the mode's levels.



Figure 3.2.: Featured art style

3.2. Art Style

The game will feature a simple art style, comprised of pixel art assets. The art style decision is derived of the findings of Riopel et al. [33], stating that schematic like graphics have had positive effects on the learning yields in scientific serious games. This discovery is a blessing for another reason: creating art assets in this art style has proven to substantially less effort when compared to elaborate graphics, be those realistic 3D ones or stylized 2D creations. This translates into a smoother implementation process, wherein art assets are created in a rapid pace, and allow the bulk of the allocated time to be invested into the programming of the game's mechanics.

Keeping in mind the intent of widespread availability through digital distribution, it is a great boon to be able to create original 2D pixel art graphics free of royalties for the game's potential future publication.

The game has a space theme, as it's story and settings for all mini games are set in space. Though this was largely due to personal taste, the intent was also to bring the setting to a scientific environment, even if that is more loosely based on science fiction rather than actual science. Having a science resembling theme to the game may increase the immersion and retain it while solving actual math problems, as it is loosely associated.

The game features electronic music that is set to make sense with the game's space

theme and aesthetics.

In the previous chapter 2, related work, we have examined numerous games, educational or otherwise, that boast a simple graphic style. These are not alone in the trend. Recent entertainment oriented games have adopted the simplicity into their graphical presentation to wide acceptance and acclaim. Some quite popular games as examples of this direction in art style are: *Minecraft* [27], *Terraria* [25], and *Among Us* [19].

These games do not need to balance educational values with fun and spectacles, as they are entertainment oriented. Still, an example of simple graphics is presented. Not necessarily through compromise, but by choice these games opted to look cartoon-like or pixel and or voxel-like, and these choices have made the games highly distinguished from their competitors and have managed to sustain the flow and immersion that they offer.

Returning to the subject of serious games though, we can now combine the findings of Riopel et al. [33] with these recent trends in video game graphics. To reiterate, according to Riopel et al., the greatest amount of learning yield has been associated with low fidelity graphics, with schematic graphics being the most influential when considering scientific based serious games.

In summation, the simple art style serves the learning purposes of educational serious games, and it does conform to recent trends in the entertainment games industry mentioned above. Please refer to fig. 3.2 for a screenshot of the game in it's current state, representing the above mentioned art style. The screenshot is taken from the fighting game mode during the encounter with the boss *Polarius*. The encounter's main objects, such as math signed orbs and the boss itself are visible in the center, with some UI features to the top on the sides. More details on this game mode in chapter 5.

3.3. Story

The game experiences are tied together by a story, that aims to bring a narrative reason to the table and enhance the immersive aspect of the experience. The story is present in the game's tutorials and is also present in the game's menu navigation. This goes against the findings of Riopel et al. [33], since it concludes that ludic content, such as a narrative, characters or dialogue, are not of significant contribution to the learning contribution in educational serious games. Nonetheless, a story does provide a richer experience, which could lead to further retention and enjoyment [7].

Having a story component in the game has another important implication. It invites the use of tutorials to ease the player into the game, and explain how some of the mechanics work. For example, the tutorials implemented in the game levels are

presented as a dialogue between the player and their captain. This corresponds to a cut scene / story LM-GM mechanic [4], through text based dialogue windows, and is used in order to inform the player of the game's rules before play.

The story: The player assumes the position of a new officer on board of a space ship. The ship is tasked with missions such as exploration, fending off monsters and resource collection. The player is being shadowed by the ship's captain, as they settle into their position. They are initiated in the ways of the weapons officer, the mining "*Astromancer*", and the navigator respectively, each corresponding to the game's modes. Not long into the trial phase, the captain of the ship vanishes. When chaos and distrust ensues as a consequence, it is up to the rookie officer to keep the crew together and maintain the ship's ability to carry out its tasks. It is now up to the player to gain the trust of the fellow crew mates through their accomplishments with the ship's assignments. Through progression and demonstration of skill, the officer eventually replaces the missing captain on his post. During early stages of inception, the story used to have more elements to it. These were sized down or inevitably cut in order to prioritize other aspects of the game. Please refer to chapter 7, future work, for more details.

The story is conveyed through text based dialogue bubbles that appear in the levels. The story elements retain a background position compared to the formal information that is presented to the player during these sequences, such as what to click and how to operate the game's systems.

3.4. Progression

When starting out, the game will have tutorial levels available. These serve as an introductory level, each explaining their respective game's rules and interactions.

Beating the game's stage unlocks a subsequent level of the same type, increasing the difficulty slightly each time. The level's mechanical difficulty is disassociated with that of the math problems, which can be configured individually, depending on the player's own learning preferences. With this system in place, players can learn the game mechanics unique to their respective game mode of choice with an increasing difficulty curve without compromising on the level of math they wish to practice.

Once configured, the math editor will generate the problems according to its current rule set. These problems will then make appearances through in game interactions, such as dealing damage or activating artifacts. As a consequence, the same type of math problems will be generated throughout the level. Doing so will allow the game to make the often mundane task of repeating math problems as an exercise a bit more exciting through its game design, and serves the purpose of repetition in the LM-GM map [4] on the learning mechanics side, it serves the purpose of skill training and

retention. The corresponding game mechanics used to realise this vary from game mode to mode, and are covered in greater detail in their respective chapters.

While it is not currently implemented, the math editor could be configured to scale the difficulty of generated questions. This will allow the game's math content to have adaptable difficulty, and encompass more subjects the further the game is played. Doing so directly binds the progression to the math subjects being practiced during play, and would therefore run into the issue of possibly out scaling the player's curriculum.

Later levels introduce ever more increasing mechanics to the game, such as new boss fights with different attack patterns and weaknesses, new types of obstacles to plan around strategically, or new elements to collect.

Figure 3.1 visualizes the game's progression loop, and hints at some of the planned future work. More information regarding future work can be found in chapter 7.

3.5. Technical

The game is developed using the Unity game engine version 2020.1.14f1.

The choice to work with unity has two main advantages. Namely, personal accumulated experience with the engine itself along with C# scripting, and the engine's ability to seamlessly create versions of the game for other platforms. The second point in particular is a huge consideration point for the goal of this thesis. If the game is made available on a plethora of platforms, each through their own digital distribution channel, it might reach it's target audience more easily.

The game will make use of well established mouse click controls for both menu navigation as well as interaction during play in each of the modes. This design choice was made to keep the game in a state from which it can easily be ported over to the touch screen family of devices, which include mobile phones and tablets. Touch controls are synonymous with mouse clicking controls for the games, which makes that transition an intuitive one to realise. The reason for the pursuit of touch screen devices stems from their abundance. While a game console might not be found in every household, smartphones and tablets are an ever increasing presence in households [2][39], which in turn makes the game accessible to the largest audience possible, in combination with a PC version.

Apart from this, the game logic will be implemented using C# scripts, and will make use of software engineering principles such as object oriented principles and code reuse. Keeping the code base from becoming larger then it needs to be will make future development easier, as well as lessen the opportunity to unintentionally introduce bugs and other issues to the game.

4. Math Editor

This chapter covers the Math editor in detail. It explains the reasoning behind the implementation of this component, its purpose in the grand scheme, as well as more technical information such as class hierarchy, capabilities, and extent of configuration.

4.1. Introduction

What this serious games attempts to achieve is the training of math skills. The aim is to make the pupils mitigate the learning deficit of school outages. In order to provide such training, the math in the game will be presented in a way that mimics the school textbooks in the related work section. If the problems are presented in a familiar form, then the learning results should directly be applicable in a school environment, like in the learning books [32].

In order to realize these specifications, a math editor has been implemented. The editor, a unity prefab, functions like a window inside the game, that can appear during game play. This will pause the game and keep it in the background, while the player is prompted to answer the presented questions. It is completely agnostic to the current game in session. It is able to generate math problems and accept answers from the player, as well as evaluate these to be correct or false.

In the spirit of the game's overarching space theme, the math editor presents itself as a minimalist yet fitting window. It has a color scheme to match the sci-fi setting, and presents problems in plain text. All feedback to the player's interactions are conveyed by a feedback lamp that will glow based on the correctness of the answer given: Green for correct and red for wrong. Figure 4.1 shows a simple problem being presented to the user via the math editor interface.

4.2. Problem Generation

The math editor's problem generation subsystem is able to generate problems at random. The choice to rely on random problem generation is derived from the added replay value that it offers, compared to static problems mapped to specific stages of the game.

However, since not all players are learning the same math subjects, the math editor should be able to generate a wide variety of problems. These differ in difficulty as well as topic. The solution to this requirement is the ability to configure the math editor's problem generation. This way, the players can tailor the experience to match their current study goals.

The configuration is comprised of a set of rules to be included or excluded in the problem generation process. Those include the following:

- Multiplication
- Fraction Division
- Remainder Division
- Fractions
- Negative Values
- Text Questions
- Long Statements
- Negative Values

In order to avoid the combination of rules that would result in unwanted artifacts, such as remainder division with negative values, some combinations will not be allowed to be active at the same time. Additionally, addition and subtraction are always included in the problem generation process, and as a result, can make appearances with any possible configuration.

Please refer to figure 4.3 for a visual representation of the configuration menu offered within the game. Using this interface, a player is able to specify whether a policy will be included or excluded for math problems within the game. An illegal combination of policies will cause the apply button to grey out, blocking the user from committing that rule set to the math editor prefab object.

4.3. Input & Evaluation

The editor offers an on screen keyboard to be used with a mouse. It makes use of Unity's built in GUI functions and assets to emulate the functionality of a calculator while entering the solution's value. Alternatively, the user may use the keyboard to enter the solution. The reason for the onscreen keyboard is to accommodate the different platforms that this game could be brought to, which support mainly touch

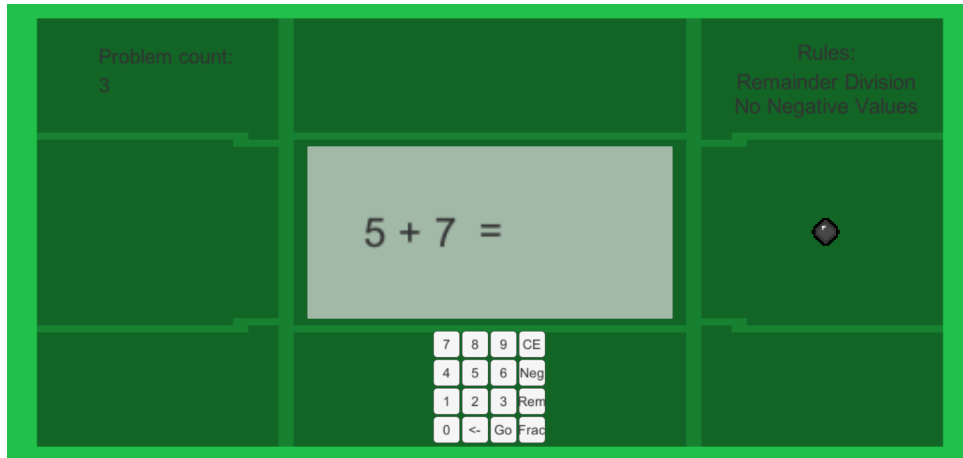


Figure 4.1.: Math Editor Appearance

screen controls, such as tablets and smartphones.

Evaluation is straight forward. Making use of the class *Number*, any form of statement can be evaluated. This not only internally solves the generated problem and stores a solution for comparison with the user given solution, it also handles different forms of numbers internally. The forms are whole numbers, rational fractions and numbers with remainder division. The sort of solution expected from the statement does, therefore, depend on the current rule set.

A fraction of any form will be evaluated by dividing it's numerator by it's denominator via floating point division. This method ensures that the user can enter any scale of the same value and receive a correct solution comparison, such as a quarter equating to two eights.

4.4. Class Overview

This section details some of the classes in the game's math editor subsystem. For a complete class hierarchy denoting inheritance and associations, please refer to figure 4.2.

The following describes the individual classes' responsibilities and relations to other classes:

- **EditorMaster** - Top level class for the management of the math editor and it's activities. Calls *ProblemGenerator* to set the current problem.

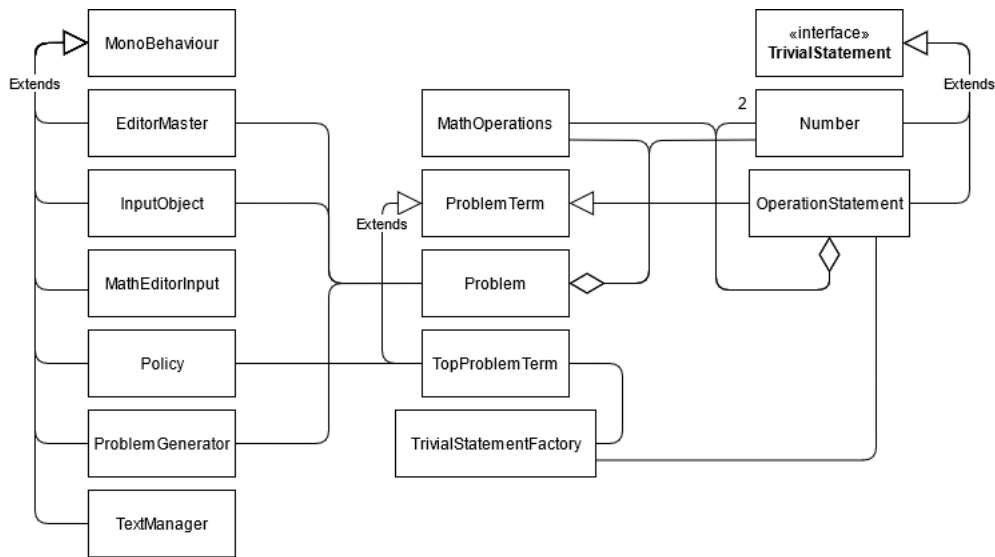


Figure 4.2.: Math Editor Class Hierarchy

- **InputObject** - The aggregate object that compounds all of the users input so far into a single entity, to be submitted and evaluated. Solution value stored as a string, compared to the given problem's solution value.
- **MathEditorInput** - Manages actual input, such as button pressing on screen or keyboard, to then pass the information into the *InputObject* class.
- **MathOperations** - Contains an enumerator field for the mathematical operations accommodated in the game, being addition, subtraction, multiplication and division, as well as their interactions. Since numbers are not only used as primitive data types throughout the game, the interactions had to be coded in order to take fractions and remainder values into account. This, in turn, is used for problem generation as well as solution evaluation.
- **Number** - Object capable of representing numbers that are rational with an optional remainder value. Contains int fields for a numerator, a denominator with a default value of 1, and a remainder with a default value of 0. Additionally, this class contains multiple methods for comparison with numeric values, which can either be of type int for integer comparison, or with an object of type *Number* for a comparison of other groups, such as rationals and remainder division products. Comparison of fractions uses floating point values internally. This different forms of fractions into account.

- **OperationStatement** - Represents the minimal amount of elements required to have a math problem in the math editor: Two objects of type *Number* to represent the numeric values of the problem, and a single operation value from *MathOperations* to bind them. Provides a *ToString* method used for representation on the GUI as well as an evaluation method which returns an object of type *Number*, modified to reflect the mathematical operation performed on the operands.
- **Policy** - Class contains Boolean fields describing the current rule set engaged. These include the following: Remainder division, negative values, text problems, include multiplication, include division, include fractions, single operations, simplify fractions and negative values. It pushes the current rule set into a GUI object for visualization.
- **Problem** - Contains problem term of type *ProblemTerm* as well as a solution object of type *Number*.
- **Problem Generator** - Generates new problems. The policy is handled in the underlying classes.
- **ProblemTerm** - A flexible structure for the problem part of any math problem. It has two fields of type Trivial statement (interface), which can either be of type *Number* or operation statement, as well as a math operation. This allows a problem term to consist of two smaller single operation problems, or be a single operation problem itself, or any of the options in between. Contains evaluation corresponding to the math operations.
- **TextManager** - Dedicated script for UI text management. Pushes problem text to the interface, generates text based problems.
- **TopProblemTerm** - Subclass of *ProblemTerm*. An instance of *TopProblemTerm* is used in problem generation as the first element, it's two operand fields are generated via the *TrivialStatementFactory* class.
- **TrivialStatement** - Interface that requires it's implementations to have an evaluate method with the return type *Number*.
- **TrivialStatementFactory** - Generates a trivial statement, which can either be an object of type *Number* or of type *OperationStatement* depending on the policy.

4.5. Expandability

This section will describe in detail how to expand the math systems in the game for a greater choice of problem generation and exercises.

At the time of writing, the math editor configurations allows for problem generation that covers a considerable amount of the curriculum of the target audience [32]. The variety of problems generated can be expanded through some of the following:

- Expanding the **range of possible numeric value to be generated**, as it currently generates single digit values. To do this, increase the range in the class' constructor, or alternatively, call it with an int value to enforce that value to be generated (already implemented).
- Expanding the problem generation to include an **arbitrary amount of trivial statements**, or better yet, follow a configured amount. Currently, the top level statement can be trivial (two values and a binary math operation) or contain two trivial statements bound by a binary operation itself. This can be implemented in the classes *Problem* and *TopProblemTerm*.
- Expanding the **text based problems**. At the time of writing, text based problems are supported for any math problems as long as they are short problems, consisting of two numeric values bound by a binary math operation. In order to expand the existing amount of questions, which are called at random if the policy is set, one would have to add a question node in the questions XML file, under the correct math operation tag. For example, if an addition question is to be added, navigate to the file *Assets/Text/TextQuestions.xml*, and add a question node between the <addition> tags.
- Expanding the classes **number** and **policy** to accommodate **irrational values**. Currently, the class *Number* contains the fields necessary to describe a rational number, but not an irrational one. Moving to irrational values will require floating point values. These have so far been avoided in the code in order to keep integers for whole numbers as well as integer values for the components of rational fractions, with the exception of fraction evaluation being calculated as a floating point value behind the curtain for comparison reasons. Including a floating point field will allow a number to be expressed as an irrational value. This will require some adjustments in the math operation classes, which will now need to accommodate such values. Additionally, these should be "infectious" in the sense that any of the operations combining irrational values with rational ones would need to transform the rational values to floating values in order to

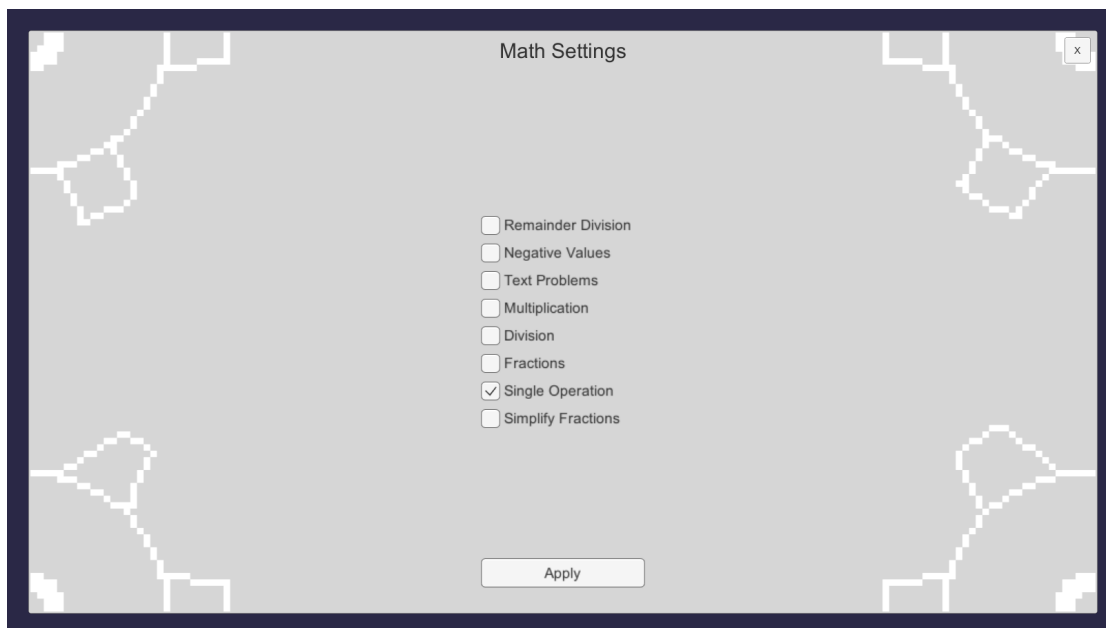


Figure 4.3.: Math configuration menu

correctly perform math operations. Thus, a *Number.ToIrrational()* method should be implemented, that returns the same value in the floating point format.

One of the main subjects covered by the target population's curriculum that has not yet found its way into the implementation by the math editor is geometry, as the current math problem generation process only takes numeric values and operations into account.

The game's design initially leaned to the direction of random problem generation rather than fixed problems. There are several reasons for this design choice. Most importantly, though, the problems encountered with such a solution are not exhausted as quickly. Random problem generation will allow players to exercise the same techniques with new values by slightly changing the number values here and there. While this is quite effective with the subjects implemented so far, this approach will require more supervision and less randomness when it comes to the topic of geometry. Due to the complexity and constraints, and above all else, the visual aspects of geometrical problems, it would be more efficient to separate geometry from random problem generation, and store an amount of fixed geometry problems that can be encountered.

To achieve this, the math editor prefab object will need to accept geometry as a policy and be able to display its images instead of the usual plain text problems.

5. Game Modes

This chapter covers all of the game modes currently present within *Mathstation*. Information about game design, elements, objectives, learning mechanics, user interface, controls as well as some visual references are found within.

Please refer to the LM-GM map of the respective game for a mapping of game mechanics to learning mechanics [4].

5.1. Fighting

The fighting game pits the player in a fight against a space monster. The form and abilities of the enemy change from level to level, as do its vulnerabilities and tricks. Math problems have to be solved in order to deal damage to the opponent in order to defeat it.

5.1.1. Introduction

The fight game mode is a downsized version of a game from the on rails shooter genre. It offers the player a shootout experience with a sci-fi themed opponent. It requires dexterous clicking or tapping capabilities, and rewards quick reaction and pattern learning. It is the player's goal to defeat their opponent in every encounter, figuring their abilities and weaknesses out along the journey.

The player has a variable amount of hit points (hp) with which they begin each level. These points can be reduced if the player fails to defend from an incoming attack. Attacks are telegraphed using visual elements such as color interpolations and flashes, and the player can deflect those by clicking on the respective areas. Reducing the hp to zero will result in a game over, which means that the players will have to restart the encounter to beat it.

Just like the players, the opponents also have a certain amount of hp that can be reduced to zero in order to beat the level, though these are not displayed in any formal manner to the player. Rather than being completely hidden from the player, some subtle clues are available for the sharp observers to learn of their opponent's current condition, and how far it places them from victory.

Reducing the opponents hit points will require the player to click on an enemy vulnerability, which are often available after a sequence of deflects or minor attacks have been performed successfully, while fully charged. Charging opportunities and vulnerabilities may vary from monster to monster, though the application of damage (reducing the hp) will be uniform across all levels; once a vulnerability has been clicked, the player is prompted to solve a set amount of math problems. Upon successful completion, the opponent takes a hit to their hp, often enough resulting in some visual changes such as cracks and other forms of degradation.

This game mode puts an emphasis on the frantic fun of an action game with quick reaction times and skillful aiming abilities, which is meant to appeal to the target audience of the same genre.

Please refer to table 5.1 to learn about the learning mechanics and game mechanics in place.

5.1.2. Controls & UI

The game mode is played solely with the mouse, with the intention of expanding it to a touch screen interface for other devices. A player can click anywhere on their screen in order to shoot the target. This is bearing the portability to touch enabled devices in mind, as it would easily translate into tap controls.

The game's user interface will display information regarding the player's remaining hit points and their charge rate.

The player's current hp is displayed on the top left corner of the screen, while the current charge of the cannon is displayed to the top right side. A glowing math symbol will let the player know when a charge shot is ready to fire, in order to deal damage to the opponent. The math takes place in a separate pop up window which can be interacted with the on screen buttons or with keyboard shortcuts. As mentioned before, there is no health indication for the opponent, other than subtle signs.

The first level of the fighting game introduces the player to the various game mechanics in motion via a tutorial. The tutorial is its own subsystem that is made reusable, with the intent of having it implemented into a first level of all game modes. In this case, however, the player is greeted with a message from the captain, explaining that they are currently explaining the ship's weapon system to the player. The captain proceeds to explain about the energy gauge, about how to ramp its value up and how to deal a damaging blow to an opponent. The player is then invited to try the gun controls for themselves, by testing these on an unsuspecting rock in their visor. This rock is harmless during this stage of the game.



Figure 5.1.: Rockstorm

After dealing the first blow though, the captain will notify the player that the rock is actually a sleeping monster, and that they will engage it in battle. From this point on, the player will have to defend against incoming attacks from the opponent, and strike them down with charged shots and math solutions.

5.1.3. The Bosses

The bosses found in this game are listed in detail in this section.

Rockstorm

The first tutorial monster encountered in the game, *Rockstorm* is a living rock monster. Initially it is slumbering and docile, but the tutorial practice shooting awakens it. From that moment on, the monster will fling rocks in the ship's direction, and the player must destroy these to prevent taking damage. Easily dispatched with a few charged shots. Please refer to figure 5.1 for a visual representation.

Polarius

This monster is a floating orb that can take on a mathematical sign, such as a plus or a minus. This represents the first boss with some mathematical flavour to it, as it conveys the concept of opposite math signs. It has accompanying orbs floating around it of both signs, and these can be shot by the player. Shooting the opposite sign will

weaken *Polarius*, opening the boss's vulnerability and allowing a player to inflict damage via charged shot. However, if the player shoots a sign that matches *Polarius*' current sign, then the favour is returned, and the boss will attempt to strike at the player. This attack can be deflected if interrupted with a shot.

This boss will be taken out after three charged shots, but there is a catch. Every point of damage dealt to *Polarius* causes the fight to become more complicated. After dealing the first point of damage, the fight will resume with more orbs in play, as well as *Polarius* changing their mathematical sign from positive to negative. The same rules are still in play, only now the player must shoot the orbs that they were avoiding before, and vice versa.

After the second point of damage is dealt, the amount of orbs increases yet again. This time, however, *Polarius* actively switches their math sign during the fight. Players must now carefully pay attention to the current math symbol, for if they shoot the wrong one, they risk being attacked by the boss.

Please refer to 3.2 for a screenshot, as this boss fight is depicted as an example of *Mathstation*'s art direction.

QuizShow

The final boss encounter featured in the game's current state is that of *QuizShow*, a TV quiz themed experience.

Once facing this opponent, the player sees before them a TV set with a stereo sound box system. These sound boxes will flash on occasion, warning the player that they are attacking. Like the previous fight, the player can deflect those by clicking on them during their warning time window. The interesting part of the fight will take place on the TV monitor: *QuizShow* will present the player with a simple math equation, involving two integers and an inequality sign. The player may shoot one of two buttons which also appear on the monitor, below the presented quiz. These buttons are a green correct button, denoted with a \checkmark symbol, and an incorrect red button, denoted with an X symbol.

As one may guess from this, it is up to the player to figure out if the presented inequality is a true statement or a false one, and shoot the button that corresponds to their assumption. This marks the second boss that involves a mathematical flavour to their encounter, this time venturing beyond the aesthetic into actual fighting relevance.

A correct answer will yield a bars screen, revealing *QuizShow*'s vulnerability where the player can now freely shoot at to gain charge, or unleash damage once charged (and after correctly solving a mathematical problem via the math editor just as before). Choosing wrongly, by contrast, will instead prompt an immediate attack from one of the amps, and cause the monitor to display a mocking face. Please refer to figure 5.2 for an image of the encounter.

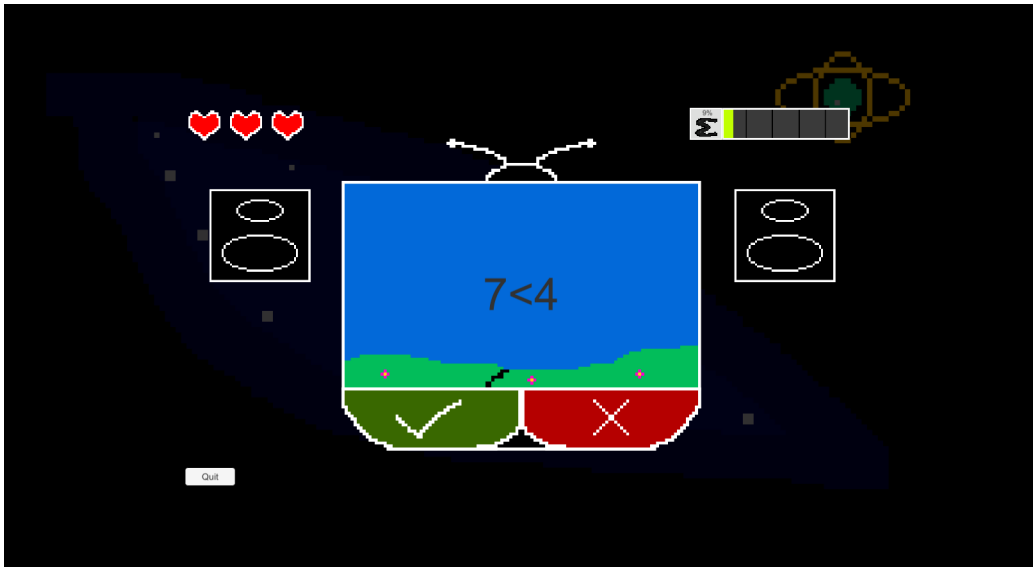


Figure 5.2.: QuizShow

The fight progresses each time damage is dealt to *QuizShow*. The second round will introduce an additional sub-woofer below the monitor, and the final round brings in an additional set of speakers for a full 5.1 surround experience, meaning that the player must carefully monitor all of these objects simultaneously in order to deflect incoming attacks. As an extra means to challenge the player, the positions of the true and false buttons are swapped after each quiz question in the final round. Careless players will find that their solution is wrong exclusively due to the now reversed position of the buttons.

Making new levels in this mode will require new art assets as well as scripts, as each of the above mentioned encounters brings their own specifically tailored assets to the table.

5.2. Mining

The mining game puts the players in charge of harvesting the raw materials of asteroids in order to construct devices on it, which in turn enable the delivery of the asteroid. The game features resource management and planning elements to mostly resemble itself to real time strategy games, with their respective target audience in mind.

5. Game Modes

Game Mechanic	Learning Mechanic	Implementation	Usage
Tutorial, Story	Guidance, Demonstration	Interactive tutorial, scripted dialogue with step by step tasks. Boss fight evolves out of the tutorial	Players learn about controls, game mechanics via guided, interactive dialogue with story elements
Elimination	Action, Repetition	Clicking to attack or defend. Player health as hearts, opponents health via visual effects and difficulty escalation	Click to deflect or deal damage after solving math problems
Action Points	Feedback	Hit points as hearts, charge shots as progression bar are shown via UI elements	Limited hit points, game over when out. Charge shots to deal damage
Questions / Answers, Feedback	Feedback, Assessment	Math question popups, feedback light bulb. Math editor window contains question text and offers input via soft keypad or keyboard	Math editor pops up when inflicting damage. Feedback through the light bulb - Green for correct, red for wrong. Damage dealt after successfully solving
Levels	Discover, Incentive	Progressing boss encounters. Further levels include more complex attack and defense sequences	Fights are contained within a level, difficulty escalates

Table 5.1.: LM-GM Map - Fighting Game

5.2.1. Introduction

Every level in this game mode consists of an asteroid ripe for mining, and a predetermined set of hazards. The player's first action is (as per the game's recommendation) to deploy a drill gadget onto the asteroid's surface. This gadget is deployed for free and yields resources from the asteroid over time. Once the mining has begun, it is up to the player to decide which gadgets to construct next.

Once enough resources are collected, players can choose to construct and deploy their gadget of choice. This is where the math comes in. As an abstraction from the tried and true research pattern of real time strategy games, the process is abstracted to math solving instead of a traditional progress bar. The player selects the available gadgets according to their current resource budget, and once the gadget button is pushed, a math editor will pop up and present questions according to the editor's current configuration. With this procedure, the familiar research process of strategy games has been swapped with active math solving problem.

In a vacuum of hazards, the only necessary gadget to beat the game would be the engine, as it is required in order to travel the journey towards the destination, where the level ends. However, the journey will prove to be more treacherous than that. Along the voyage, hazards such as asteroid showers and pirates may appear to block the player's progress, or even to wreak havoc on the gadgets constructed thus far. A certain combination of items can protect the asteroid and prevent the journey from taking a bad turn. It is up to the player to be prepared for the hazards when they make an appearance, making use of the available memos left by the previous miner and the handy radar gadget, which is capable of revealing the hazards ahead of their arrival. When in a pinch, the engine can be toggled off in order to halt progression of the journey, buying time for more resources to be mined and gadgets to be deployed.

When a journey is completed, the player is presented with a damage report. The report details all of the flaws that occurred during the session, or, if the player was able to properly prepare for all hazards, will commend them for the flawless journey that they just completed.

Please refer to 5.2 for a table of learning mechanics and game mechanics found in the game [4], and refer to figure 5.3 for a visual of the game mode.

5.2.2. Controls & UI

This mini game's controls are point and click. This design keeps the controls simple, and retain the ability to expand to a plethora of touch screen devices. All of the interactions with the gadgets are initiated by clicking the respective elements, and

construction as well as menu navigation is handled with unity button functionality.

The mode's tutorial will explain the game's core mechanics. It will let the player know that it's about resource collection and asteroid delivery, and will recommend the most sensible priority of gadget research, though the players may ignore this advice at their own risk to freely explore the possibilities. The tutorial makes use of the in game dialogue system.

5.2.3. The Gadgets

This subsection lists the available gadgets and explain their role, their possible interactions with hazards, and other unique features.

Drill

The most basic and necessary of them all. The drill can be acquired free of materials and deployed at the start of the level in order to gather resources. This is the only way to accumulate resources necessary for the construction of other gadgets. Interacting with the drill will increase the resource yield temporarily.

Engine

The engine will transport the asteroid to its destination. It needs to be constructed to finish any of the levels, as they all are completed once the journey counter reaches 100%. The engine can be interacted with in order to toggle it on or off, which will result in resuming the journey or halting it, respectively. Since some hazards can be discovered at a point in time which is too late for the player to prepare for, halting the engine can be a tool to buy the necessary time in order to gather the resources needed for the gadget that will protect the asteroid and the on-board gadgets from the incoming hazard.

Radar

An optional yet strongly recommended gadget. Installing the radar on the asteroid does not shield it from any of the hazards per se, but it does enable the ever crucial ability to peek into the distance and detect incoming hazards. Once deployed, the player can interact with the antenna by clicking it. Doing so opens a pop up window containing the radar interface, showing the distance from the incoming hazards once those are within its detection range. The radar interface window can then be closed to resume playing. This element is the key to foresight, as it allows the players who utilize it to form a correct build order in order to circumvent the given level's obstacles. Please refer to 5.4 for a visual representation of the radar interface.

5. Game Modes

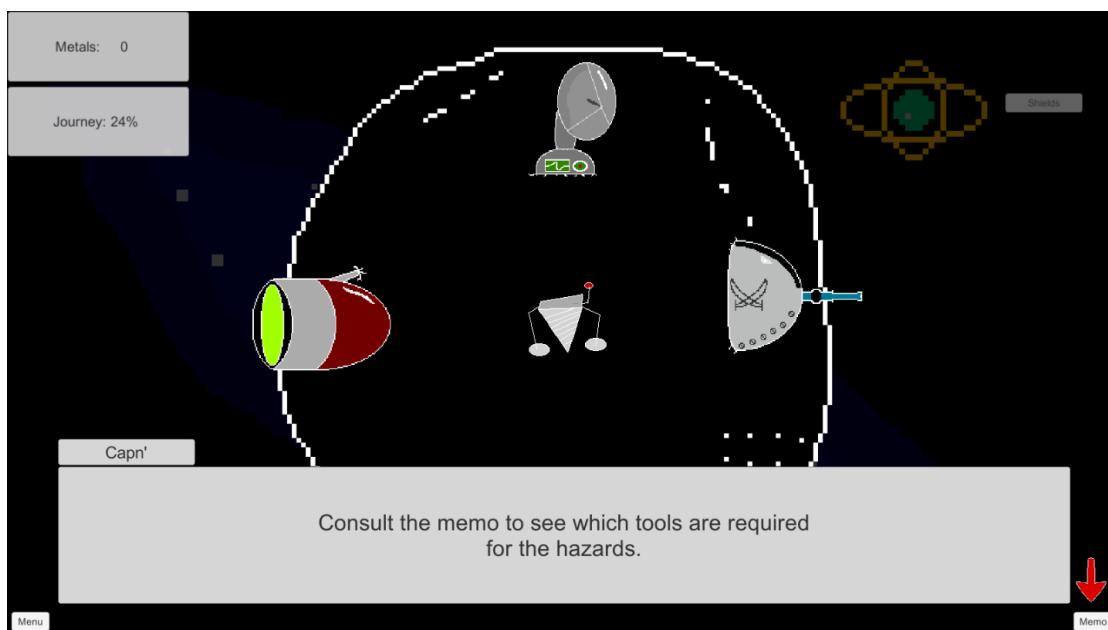


Figure 5.3.: Mining an asteroid

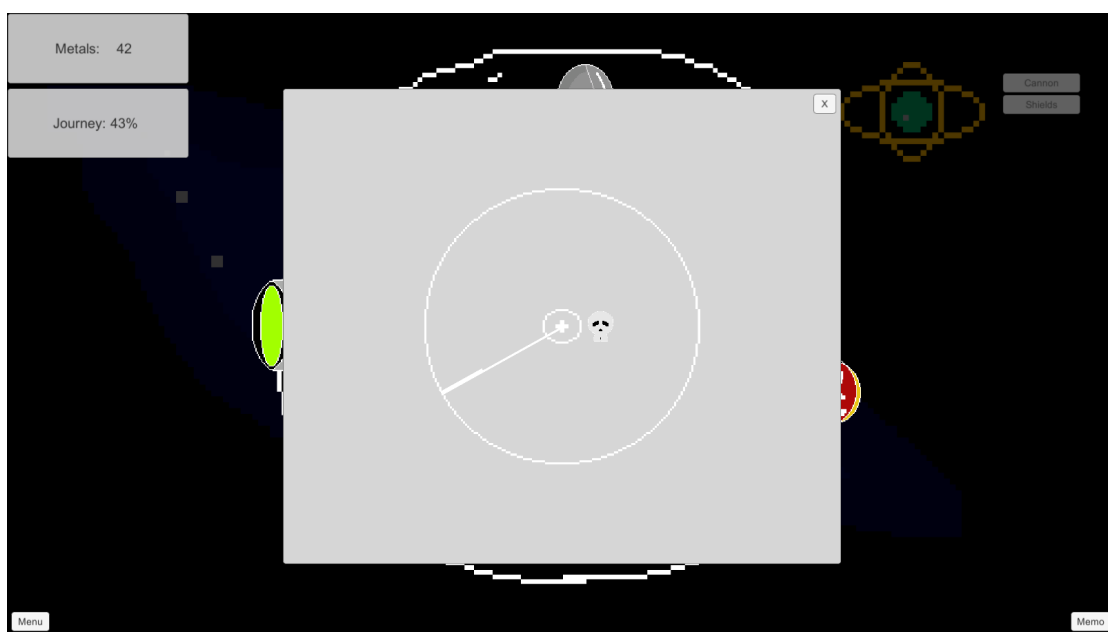


Figure 5.4.: Radar Window

Cannon

An offensive gadget that can be used to blast open barricades or defend the asteroid from an incoming pirate wave. This gadget requires no player interaction, and is activated automatically when the environment calls for it.

Shield

Similar to the cannon, but in reverse. It possesses no offensive capabilities, but it does boast a strong enough defensive abilities to protect the asteroid from a rock shower, which the cannon would not be able to do. Just like the cannon gadget, however, this requires no input from the player once deployed, and will automatically be used when the time is right.

5.2.4. The Hazards

Listed below are the hazards of the mining game, and how to handle them.

Barricade

It's a big space door. It doesn't exactly do damage, but it completely halts the progression. The asteroid cannot pass through a barricade while it's up. The cannon will blast it open, allowing the asteroid to pass and continue it's journey.

Pirates

A pirate swarm will appear and steal resources and damage the gadgets. A shield and a cannon are both necessary to defend from them.

Rock Shower

A hail of rock objects. It will damage the deployed gadgets. The shield gadget will protect the asteroid from the damage.

There have been numerous other concepts for hazards which sadly did not make the cut for this version of *Mathstation*. The most notable ones were: A toll station, where the player would have to pay resources to pass or use an invisibility field to avoid it.

The game has been developed in a way that allows the new levels to be introduced easily. More gadgets and hazards will need to be coded and implemented, but any future levels can be duplicated from the current ones and have the fields of the *Obstacles* component of the *MineGame* object adjusted. This script stores references to two lists: the obstacles contained within the level and their corresponding appearance

locations, represented in journey percentage. In essence, these fields of this script are the difference between the levels in this game mode.

5.3. Exploration

The exploration game puts the player at the ship's helm, maneuvering it through the expansive game world. It is the objective of each stage to explore the surrounding areas, while managing the limited fuel resource.

5.3.1. Introduction

This game mode is inspired by games that feature exploration and item collection, otherwise referred to as *collectathons*. The core components of the game build a spatial puzzle for the players to solve, which they can do through exploring, trial and error, and accumulation of items to enhance their capabilities.

In order to navigate the environment of each level, the player may move the ship by clicking anywhere around it. Once the fuel runs out, they ship will reset it's location to the stage's starting point marked by an anchor, fully refueled and ready for the next expedition. Interacting with the derelict artifacts will enhance the ship's capabilities, increasing speed and effective exploration range. In order to reactivate the artifacts, the player is prompted to solve math problems, cementing the math component as an organic part of game play.

Once the map has been thoroughly explored, the player should have the necessary travel range to solve anomalies found on the map. These can take several forms, representing space themed phenomena such as satellites and black holes. Solving an anomaly requires the player to transport a key item, such as a plug in the case of a black hole, to the anomaly site. Bringing the key into contact will resolve the anomaly. Resolving the level's required anomalies will reveal the goal artifact, which will present a final problem before the levels completion. The player can now navigate to it and complete the level, or continue to explore at their leisure.

The experience offered by this game mode is significantly more relaxing when compared to the other two game modes by design. There is no time pressure from an enemy's assault, nor is there a schedule of events that the player needs to prepare for in advanced. The Mechanics, level design, spatial puzzles and controls are all in place for any user to take their time and enable exploration at a laid back pace. This in turn lends itself to a more lenient style of math questions, such as text based questions. Those also match the puzzle nature of this game mode.

Please refer to 5.3 for a LM-GM map of the game mode [4].

5. Game Modes

Game Mechanic	Learning Mechanic	Implementation	Usage
Tutorial	Guidance, Instructional	Instructional dialogue. Makes use of arrows and step by step tasks. Can be ignored at own risk	First level explains controls and mechanics
Resource management	Plan	Mining and construction. Objects visually added to the field of play as constructions advances	Mine resources over time, construct gadgets in the correct order
Time Pressure	Simulation	Obstacles and solutions. Can be detected, will pop into play from the sides and obstruct the players progress	Hazards appear along the way, requiring specific gadgets
Cascading Information	Participation	Memos, Radar gadget. Clicking on them will reveal if hazards are inbound, which type they are, and how to handle them	Radar can reveal hazards ahead of time, memos decipher the radar readings
Strategy / Planning, Feedback	Action / Task, Repetition, Feedback	Research gadgets via math popup, feedback when solving math problems and when encountering hazards	Math editor light bulb shows correctness of solution, correct gadgets are required for specific hazards

Table 5.2.: LM-GM Map - Mining Game

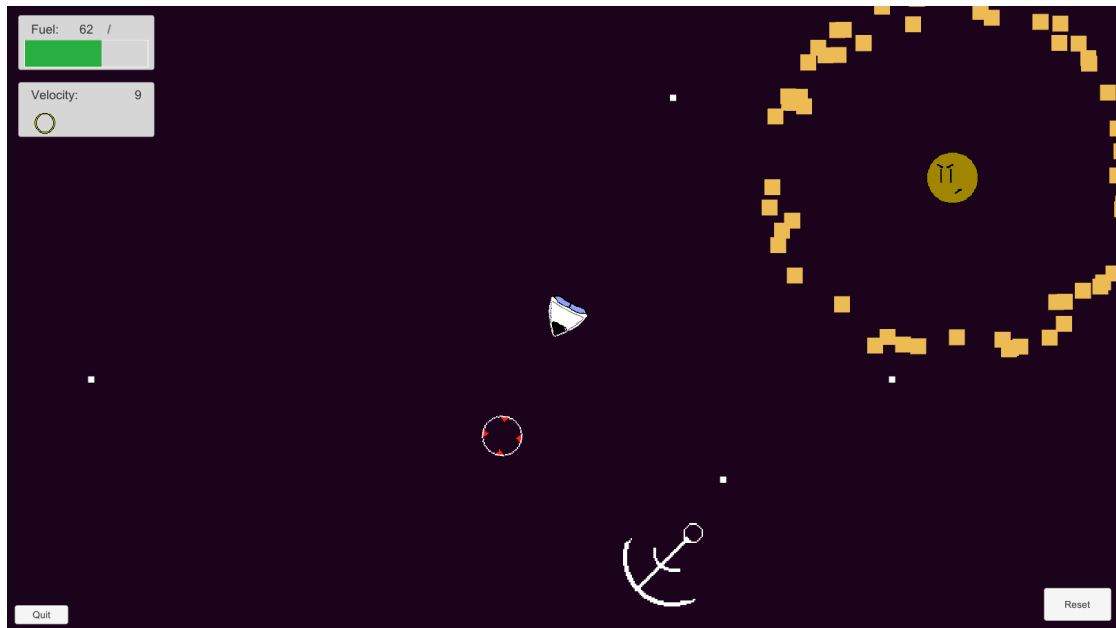


Figure 5.5.: Explore mode overview

5.3.2. Controls & UI

The game features a click to move 2D control scheme. This design keeps interaction simple and adaptable to touchscreen devices. Anywhere the player clicks will place a target marker on the game world in that location, causing the ship to turn to that heading and advance at its current velocity. The player's point of view is top down. The ship will always cruise at the static velocity value noted in the heads up display. The only way to change the ship's velocity is to interact with the speed upgrade ring artifacts. After solving the presented math problems, these rings will be consumed, permanently increasing the ship's speed for the remainder of the stage.

Any of the interactions with items will be invoked when the player enters their trigger collider area. Entering the colliders of artifacts will prompt the math editor to pop up with a generated question, causing the controls to freeze until the math problems are resolved.

The UI is kept to a clean minimum, displaying the ship's current fuel status via text and fuel gauge as well as the velocity, marked by a numeric values and the amount of speed rings the player has consumed in the current session. Please refer to 5.5 for a visual representation.

5.3.3. The Artifacts

This subsection lists all of the in game objects that the player can interact with, otherwise known as artifacts, their roles and how they work.

Anchor

The anchor is a static prop that denotes the starting point of the stage. The player will always be placed next to it, and is free to explore in any direction from it. Once the fuel runs out, the ship will reset to this position with full fuel, in order to keep exploring.

Speed Ring

An artifact that can be consumed by the player in order to permanently increase the ship's velocity for the remainder of the stage. Touching it will initiate the activation sequence, prompting math problems to appear which the player needs to solve. Solving these allows the player to fly through the ring, consuming it and applying its increased speed effect, as visualized by a "+Speed" notification.

Increasing the ship's velocity increases its effective range for cruising, as the consumption rate of the fuel remains the same but it covers more distance. It is therefore expected of the player to find these artifacts and consume them in order to expand the available exploration locales within the levels.

Radar Station

The Radar Station artifact acts as the players detection tool in the vast expanse of space which they will be exploring in this mode. Much like the Speed Ring, it is derelict at first. This means that the player will have to touch it in order to solve math problems, and then activate it by solving them.

Once solved, the radar ring will reveal the direction of other objects, while not revealing how far away they are. The player can use the provided information in order to narrow the search for valuable artifacts in order to progress in the level, without handing out the exact location.

Refuel Station

The Refuel Station will, after activation, completely refuel the ships, effectively doubling its cruising range from that point. It is a vital element in the spatial puzzles that the game offers.

Goal Ring

The Goal Ring is the final element in the game's puzzle. It will always appear near the origin of the level, so that the player may always reset their position to get close to it.

The ring acts as an artifact, it will present the player a math problem that they need to solve. After which the player passes through it, consuming it, and clearing the level. A report is then presented to the player, denoting what collectibles they were able to obtain, and how many were left behind.

5.3.4. The Anomalies

This subsection denotes all of the in-game anomalies, presenting the problems to be solved by the player in order to uncover the goal ring artifact, allowing the level to be completed and the player to advance to the next one. Also denoted are the respective key objects required for the anomalies. Bringing the key item to the anomaly will resolve the puzzle.

Lonely Planet

The lonely planet anomaly makes its appearance as a sad planet. It does not feature any interactions with the player other than sitting sadly in the background, waiting for the player to bring its key item to it.

The key item is a moon, which will orbit around empty space before the player interacts with it. Once picked up, the moon will orbit the player and follow it. Deliver this item to the sad planet to make it happy, as it will from that moment on have a moon companion orbiting around it.

Vortex

An ever sucking black hole, the Vortex will reset the player if they dare to venture too close to it. It features a particle system effect to intuitively convey it's dangerous nature to the player. The player will not get diverted from their course because of it, it will only affect the player if they touch it. Other objects are not affected by it.

The key item here is the cork. A big cork that can be picked up by the player in order to plug the vortex with. Once the vortex is plugged, it no longer poses a threat, and the ship may freely fly over it.

Hot Star

The hot star will reset the player if they fly too close. It is an angry faced planet burning with rage, exhaling fire and hot materials in all directions. A particle effect is in place to warn the players and show an approximate burning range of this anomaly.

An ice cube acts as a key to the hot planet. It is sitting cool with sunglasses while waiting for the player to pick it up. Once delivered to the hot star, the star cools off. Its face transforms from angry to timid in an instant, and it stops exhaling dangerous

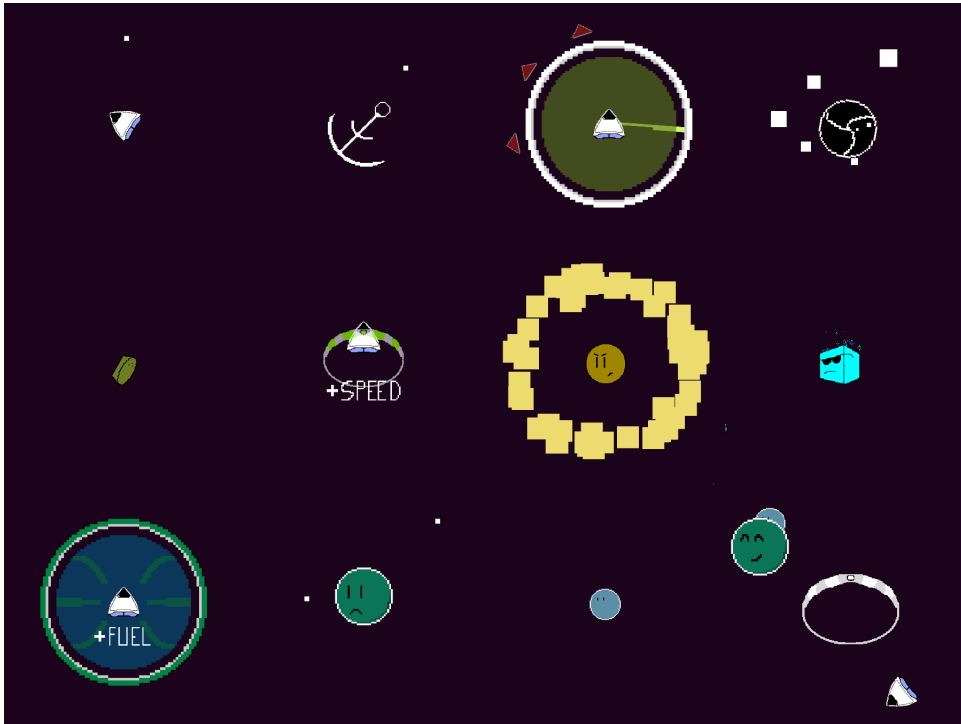


Figure 5.6.: Explore Game Elements

materials. It is now safe to fly over it and around it.

Please refer to figure 5.6, all of the explore game elements are displayed in order of appearance from the top left: The player's ship, the anchor, radar station, vortex, cork, speed ring, hot star, ice cube, refuel ring, lonely planet, moon, and the goal ring.

5.3.5. Levels

This subsection goes over the levels present in this game mode as well as an instructional part about how to compose future levels within the game engine's editor.

Level 1

This is the introduction to the game mode. As such, it features a tutorial explaining how the basic game mechanics work. Included are the fuel management as well as reset mechanics, and how to maneuver the ship around.

The level does not feature any of the advanced artifacts, opting for a very basic level design with a radar station and speed boost ring.

The player may freely explore around in any direction and discover the reset mechanic after expending their fuel reserves, at which point they will have observed some of the other objects in the level. They will, however, only be able to interact with the objects in a correct sequence, as the rest is out of reach at every point in the stage.

The anomaly present in the level is the vortex. The player can learn about it resetting them when flying too close, without any significant penalty due to its close proximity to the reset anchor. Closer yet to it, however, is the radar station that can be activated by the player. Doing so reveals the location of the cork and a speed ring. The player must obtain the speed ring in order to have the necessary exploration range to reach the cork. Once this is achieved, the player can plug the vortex, which will reveal the goal ring.

Level 2

The second level brings more elements into play for the player to interact with while they explore, without a prefacing tutorial this time around. The anomaly present is the hot star, which is placed near the player's original position. Exploring closer to it will reset the player, and teach it not to fly too close to the sun.

The corresponding ice cube will sit in a position that is out of reach for the player. They will need to expand their reach with a speed boost ring, in order to then reach their first refuel station. At this point, they will learn about the value of the refuel station, and gain access to the area behind it for more exploration.

On the other side of the refuel station, the player will find a radar ring, revealing the second speed boost ring that will allow them to reach the ice cube. Proceed to reset, drop the ice cube on the hot star, reveal the goal ring, consume, and the level is cleared.

Level 3

The final included level is substantially larger than the previous ones. It makes use of all the elements used in the previous levels in order to test the player's exploration prowess. It also features all of the anomalies, including the not before seen lonely planet and moon.

The level starts out with the player's position between 3 radar rings and a refuel station, any direction where they choose to explore will uncover a valuable artifact. The refuel station placed at the bottom of the starting position will need to be activated for the player to extend their range and eventually start to solve the anomalies. Beyond that, the player is free to explore the stage at their leisure.

The final section of the stage involves a chain of refuel stations and radar stations that will point the player in the direction of the moon and resupply them with precious fuel on their voyage. The moon is the key to the final anomaly in the level, the lonely planet. Obtaining it after all the math solving along the way will allow the player to

solve the anomaly, consume the goal ring, and advance to the next stage.

Level Creation

In order to create further levels in this game, one must place the in game objects into a scene to form the spatial puzzles. All forms of artifacts and anomalies are prefabs stored within the unity project, and the only rules to follow are to carefully monitor the ship's effective range for the spatial puzzles.

In order to make the goal ring appear at the end of the level, please make use of the *Objective* script component of the player object. It contains a counter for the objectives - how many anomalies were solved thus far - of type *int*, as well as a reference to the goal ring prefab and the location where it is instantiated. Every anomaly solved will increment this counter from their receiver superclass script. The designer may now choose if all present anomalies are required to generate the goal ring or if some of them are optional, in which case there can be more anomalies to solve then the counter's value.

The ship's speed values can be configured within the explore player control script component on the player object. It contains an *int* array of velocities to set, these may vary from stage to stage. Each speed up ring will increase the current value to the next one. Overflowing will cause the latest speed value to be in effect still.

Game Mechanic	Learning Mechanic	Implementation	Usage
Tutorial	Guidance	Tutorial dialogue. Does not force the player, they are free to explore and learn afterwards	Explains controls, basic game mechanics
Resource Management	Limitation, Plan	Fuel via UI element, Reset mechanic moves the player back to the start when out of fuel or by button press	Movement consumes fuel, requires planning. Reset mechanic as punishment
Movement	Action / Feedback, Repetition	Click / Tap to move controls. Tapping leaves a cursor to which the ship maneuvers automatically	Explore through movement. Moving into artifacts will activate them
Collecting	Discovery, Incentive	Artifacts: Upgrades and stations, math problems	Players activate collectibles with math solving
Questions / Answers	Feedback, Repetition, Reflect	Math popups, input via soft keypad or keyboard	Pops up when interacting with an artifact
Feedback	Feedback, Assessment	Feedback light bulb, math editor	Correct answers - Green light. Wrong answers - red light
Levels	Motivation, Explore	In game levels. Increasing complexity with subsequent levels	Progression & Spatial puzzles, in level speed progression, goal rings

Table 5.3.: LM-GM Map - Exploration Game

6. Evaluation

This chapter explains the theoretical evaluation plan for *Mathstation*, as well as enlighten why the methods were not implemented at this time.

6.1. Reason

An evaluation is a necessary process of serious games in order to determine their ability to instruct and educate. In other words, the only way to reliably determine the value of such an undertaking, is by testing it. We can refer to other serious games from the related work chapter in order to find evidence of this in the studies that they have conducted [30] as well as the numerous other related works following through with the same methodology. The yields of such evaluation processes are what meta analysis such as the one by Riopel et al. [33] rely on to reach their conclusions about science based serious games over the span of decades.

6.2. Method

Depending on the purpose of the serious game in question, different tools for evaluation should be utilized during studies. In the case of *Mathstation*, the game is intended to be both an entertainment software as well as an educational tool to practice math skills with. As such, tools that should be taken into account are game evaluation tools as well as math training evaluation tools. Ideally, these can be assessed with studies and questionnaires.

6.2.1. Game Assessment

There are several questionnaires to evaluate the experience of video games, such as the **Game Experience Questionnaire** (GEQ) [17], and the **Player Experience Inventory** (PXI) [1].

The GEQ attempts to quantify the game's entertainment value. It has been "widely applied by games researchers and practitioners to a broad scope of game genres, user groups, gaming environments, and purposes" [21]. According to Abelee et al., the questionnaire has been applied to games of various genres, including "immersive virtual

learning environments for students", making the GEQ a highly regarded candidate to measure *Mathstation's* entertainment factor.

In addition to the GEQ, there is the PXI. This questionnaire builds on the means end theory and games user research in order to link game design to both functional consequences and psychological consequences, corresponding to immediate experiences and second-order emotional experiences respectively. It is with this method that the PXI "aspires to provide actionable insight, enabling a better understanding of how game design choices impact the player actions during the run-time of the game, and how they shape emotional responses" [1, p. 11].

For the purposes of this serious game, either questionnaire would suffice. The GEQ, however, does have some citation inconsistencies, as has been discovered by Law et al. in their systematic review and validation of the GEQ [21]. The PXI is therefore considered to be slightly favored, and will serve as this game's questionnaire regarding all play experiences.

6.2.2. Math Assessment

As mentioned before, the game as an entertainment product is only one of the two dimensions needed for evaluation. The other would be the learning yields as an educational math game. Math pretest and post tests should be conducted in order to determine *Mathstation's* value in this regard. This practice is seen put to work in a multitude of related works in various forms, such as **Games for Math's** conducted pretest and post tests [5] and **Lambacher-Schweizer's** self test assignments at the end of chapters [6]. It is considered to be a decent tool in finding out what the student has learned since the start of the learning process. It is for this reason that the learning yield part of the *Mathstation* experience will be quantified via a math pretest and post test.

The game's flexibility of math related content does invite another important point of discussion with regards to evaluation and testing. Since the math subjects making an appearance during the play of *Mathstation* are at the player's mercy through configuration, how does one construct a pair of reasonable math tests to discern learning yields? To answer this requirement, the pretest and post test pair must match the pupil's intended curriculum and, by extension, configured math settings. Only through carefully picking the relevant subjects, and creating a combined test with questions from all of the relevant fields, will it be possible to determine the improvement in math skills for every test subject in their respective math level. Acknowledging the above, the math tests that would have been used for the evaluation are comprised of

components from different math subjects that are included or excluded based on the math configuration used during play.

The math tests are recommended therefore to be composed of a pool of questions in order to meet its flexibility requirements. The questions included in the appendix chapter A cover the extent of the math content included in the game, and can be selected for each testing session as is needed. The questions are sorted by subjects and classes, and their abundance reflects the amount of freedom and flexibility that is given to the player when determining the materials to practice within the game. The amount of time allocated for testing sessions, which is a variable too, can also be addressed by selecting the right amount of questions. It is recommended to select a suitable amount of questions from the catalogue that correspond to the class of the group in the evaluation. A test subject group from the 4th grade, for example, should be presented with questions from the subjects that they are currently learning in the 4th grade section.

The questions are from a the website *helpteaching.com* [13], which contain a large amount of diverse K12 education related questions. Premium questions from the site conform to the common core standards, a US educational standard that has been adopted in over 30 states [31]. For that reason, if a strongly established math standard is to be used in the assessment process, it is strongly recommended to use the premium questions offered by the same site.

Reasons to use a well established math standard are numerous. Most importantly though, a proper benchmark is useful to compare the learning yields of *Mathstation* to that of educational systems adopting the benchmark, providing a standardized comparison. A candidate that comes to mind for such bench-marking is the program for international student assessment - PISA [29]. However, due to the target age of PISA being 15, it does not match *Mathstation's* intended target audience age group of about 6 to 11 (targeting first to fourth grade). It is still worthy to consider PISA for bench-marking in case the content of the game is expanded to that age group.

6.3. Testing Conditions

The proper testing conditions for this tool are explained in this section.

Testing *Mathstation* in a school environment would be ideal. If enough students could participate, then the tests could achieve statistical significance. Testing with an entire class in a computer lab would therefore be the most desired condition.

Optimally, the gathered class could begin the session with a math pretest that includes questions that are relevant to their current math subjects. Afterwards, the game is

played for a non exhausting amount of time (depends on the time allocated for the tests and the questionnaire), followed by a math post test and the player experience inventory questionnaire.

The data of such a session would then be relevant to the used math settings of both game and tests, and would reveal the game's ability to improve math solving skills. Tests of different classes with different math subjects would therefore be warranted in order to expand the scope of data to further math content. The PXI will reveal how the game feels like. Precious feedback of testers would be used in order to facilitate a plan to improve existing features, mechanics and content in preparation for future work.

As has been demonstrated by Plecher et al. [30], a sensible practice before conducting the actual study of this sort would be to conduct a smaller scale study in order to validate the most fundamental assumptions regarding usability and duration of play. This trial study will provide information, shaping the structure and length of the actual studies that are to follow. Crucial feedback may also be implemented as a consequence of this trial study in preparation for a bigger undertaking in the actual study, in order to improve the user experience as well as to cut down on the time needed to play through the study build of the game.

Alternatively to the above described testing conditions, the game will also be suited for a long term study, provided that data collection is implemented at some point in the future, as well as the amount of game content found within swells to warrant a long term study. For this kind of study, though, it would not be necessary to conduct a trial study before it to learn about critical improvements that need to take place. Allowing users to engage with the game regularly will provide data over time, showing the learning value of the game over the same time period. This form of study can also determine the game's longevity and ability to retain the attention of the players.

6.4. Unavailability of Tests

During the time of writing, the pandemic is still about. In Germany, we find ourselves constantly changing the availability of non crucial services and infrastructure during the third corona wave.

During this time it is, however unfortunate and ironical, not possible to conduct this kind of evaluation under ideal conditions. Schools are not accessible for this, as they have strict pandemic restrictions in place and wish to reduce the presence at the schools to a necessary minimum. Conducting tests for a master thesis such as this, is unfortunately out of the question.

Remote testing, while possible, would not be as empirical as testing in lab condi-

tions. Not being able to provide assistance with the game, as well as unreliability to confirm that the entire allocated section has been played through as intended with no unexpected bugs or interference contribute to this fact. Furthermore, conducting the test with a relatively small amount of test subjects would lead to results that are not statistically significant enough to bother with. Testing individuals is, as a consequence, not advised. Which is the reason why testing did not take place during the time window for the thesis.

The only thing left in order to eventually determine the value of *Mathstation* is to conduct the described tests at a time when it is possible. For this reason, all of the testing protocols and materials included in this thesis are to be considered as recommendations for a future work effort. Hopefully before long, testing this serious game in school environments will become a reality. Once it does, a tester will be able to conduct a proper and thorough study, which will provide results that are both reliable and statistically significant.

6.5. Summary

In summation, *Mathstation* being an educational serious game can be evaluated in two dimensions: Educational value, and game quality. The recommended tools of evaluation are math pretest and post tests and the PXI questionnaire respectively.

Though long term studies will offer insight to both of the aforementioned aspects as well, the game will need some additional technical abilities and significantly more content to warrant this.

At the time of writing, it was not possible to conduct a study involving the intended target audience (grade school pupils) playing the game and filling the questionnaires. It is recommended to conduct the study at a safe time in the future.

Once the study has been conducted, the data it would have provided would reveal if the concept has been successful or not, and help explain what exactly worked and what didn't. This in turn would determine the answer to the thesis' research question.

7. Future Work

This section thoroughly discusses the possibilities of future work considered for *Mathstation*. Considering that this undertaking is heavily limited by its duration and circumstance, it would not have been able to achieve the game's full potential within these limitations. As such, it was the goal to achieve a state that could be considered a sound foundation which can be improved in the future. Using this groundwork, some if not all of these future work features can be integrated into the game in the long term, in order to benefit from them both in the educational regard as well as in the game aspect. Please refer to figure 7.2 for a diagram of features either implemented in the game or considered for future work.

7.1. Customization

The former prototypes of *Mathstation* considered having greater role playing elements [3], such as a custom character avatar, a class to choose from, and custom visuals for the ship and crew. The character class would have determined the character's specialization in the game world, which should have placed the player as a replacement for one of the ship's officers. A weapon officer would have specialized in the fighting game, a mining *astromancer* in the mining game, and finally, a navigator in the exploration game. This system would have added more flavour to the in game world as well as served the learning mechanics of ownership and planning. The rest of the activities would have not been locked out for any of the classes. This would correspond on the role play element game mechanic on the LM-GM map, as well as ownership and status to some degree. The officer to be replaced in the game's narrative can then, in turn, be used as a guide figure, matching the guidance as well as observation elements in the learning mechanics side of the LM-GM map [4].

Other points of customization would have been some of the visual elements of the game, such as appearances in the game modes. This would have varied from the spaceship being navigated in the explore missions to the gadgets in the mining game, to different visual effects in the boss fight game. The Hub area considered for the game (see section 7.5) would have been an ideal setting for the self expression brought forth by customization options. Different layouts, pieces of equipment and furniture would

have all come into question, as well as a fully customizable player's room which could have been modeled to the players content.

The customization serves the purpose of ownership in the LM-GM model, as well as rewards and incentive [4].

7.2. More Base Features

Developing three game modes, which in essence means developing three separate games with separate mechanics and assets, has been a challenge to achieve within the allotted time frame of the project. It also meant spreading the features thin, as a single game would have enjoyed a greater depth of game mechanics and visuals than three separate ones.

Because of this, as a future work opportunity, more of the same should be taken into consideration. This means fleshing out existing features within the already standing games, extending some of their assets and mechanics to have a more meaningful experience and abundance within their respective realms.

Pragmatically, this means adding more bosses and levels to the fighting game, along with more ways to interact with the in game elements. Things to be considered are different ways to attack and defend, upgrades, shields, weakness identification mechanisms and more.

The mining game deserves to have a bigger variety of gadgets which can be used to overcome a greater variety of obstacles. Ideally, some of the gadget's functions should overlap so that they can defend against common hazards, forcing the player to make decisions regarding their build priority in accordance to their predictions and radar readings. This will give the game a strategic edge which it does not currently possess.

With regards to exploration, a huge amount of in game elements can be added in order to increase the possible complexity of the puzzles formed by the environment. This yields interesting puzzle solving processes that improve upon the player's current experience with the game mode, enhancing the math problem along the way.

This covers the games that have been implemented so far, within the limited time window. Introducing more game modes is also possible. Many others can be implemented to broaden the range of activities available for pupils to practice math with, and they should be implemented if there is a reason to believe that the genres they are based on retain the attention of the pupils for longer. Adding more game modes does not only expand the available activities, it also raises the ceiling for achievement collectors and completionists, and results in more selection and collecting as per LM-GM.

Last but not least, the math editor can be improved immensely. From the problem generation process to the displaying capabilities, it's current form is shy of what it could be. Increasing the problem pool with more subjects, such as geometry, is a top priority. More elaborate rules and conditions for the editor's configuration will allow the players to tailor the game more specifically to their immediate math training requirements, and tweaking the visual aspects will greatly improve it's clearness and usability in turn.

A feature considered since the game's inception is the ability to have the math editor interact dynamically to problem solving. This means that while solving some formula which includes different mathematical operators and parenthesis, for example, a player who would get stuck or otherwise confused would have been able to push a button to simulate which operations should be performed in the correct sequence, and how the numeric values contained in the problem's body flow accordingly. This includes the learning mechanics guidance, observation, shadowing and possibly discovery on the learning mechanics side. It does invite action and repetition LM for further problem solving, as the acquired knowledge will be applied in subsequent problems. Relevant game mechanics: Tutorial, feedback, simulate and cascading information.

7.3. Story

Since the finding of the ludic content's diminished value for the learning yields in serious game [33], a massive amount of story content has been cut from the game. This is not necessarily due to time constraints, but due to careful allocation of available time to other aspects that had a higher priority.

Having said that, a greater presence of story has been there during the game's inception. According to the evaluation of the game, it would not be out of the realm of possibility to invite the greater story back into the game. That would be up to the evaluation data to reveal, if the existing story elements are valuable enough for it, or not.

The original story would have involved the previous section's character class system. The player deciding the destiny of their character would have impacted the story: They are a new officer of the same type on the ship, shadowing one of the senior officers from the same department. The captain will have still disappeared after some tutorial levels, however, the ship's A.I, which would have played a substantial role in this version of the story, will have calculated that it is the player's character who should replace the captain as the new commanding officer of the ship, stating that this strategy has the highest survival rate for the entire crew. Understandably, the crew is outraged. It's just

not feasible to get behind the logic of the ship's computer, why on earth should this rookie replace?

After these introductory story events have set the narrative in motion, the game's story would have reached its main point. The crew members are characters to interact with throughout the adventure, be it during the levels via dialogues, through an in game appearance to assist with the events and challenges, as well as being available to chat with in the game's omitted hub area. Again, these elements are inspired from the genre of role playing game [3] with the ability to connect to their respective target audience.

The interaction with the crew members would have been impacted by the player's performance on the diverse playing levels. The aim of this would have been to acquire some of the character's trust over by showing them competence as the ship's commanding officer, winning them over and gaining their acceptance. These would in turn translate into more game elements, such as customization options, hidden levels, achievements and badges and other trophies of accomplishment. This would have corresponded to the game mechanic of status [4].

Playing through the game's levels and winning over the trust of the crew would have eventually resulted in the game's climax, in which the player confronts the most difficult levels of the different disciplines. Beating those would have revealed that the captain that had vanished earlier, along with the crew members, were all a simulation of the ship's computer. They were put in place for the player in order to hone their math skills, which serves a higher purpose according to the computer.

The game could have had a post game where the player already learned of the nature of their circumstance and the simulation of their crew mates. Additional levels could have been played from this point on without progressing the story further.

LM-GM mechanics involved with the story: Guidance, feedback, plan, repetition, motivation and story / cutscene [4].

7.4. Multiplayer

According to Kristin Siu et al. [34], games with a purpose that feature multiplayer modes, either competitive or collaborative, have been noted to score higher in terms of player engagement than their single player counterpart. Among the two types of multiplayer, the competitive scored higher than the collaborative. In addition, it is recommended to implement both types of multiplayer modes as the engagement may vary and is ultimately subject to personal preference. The addition of multiplayer features could boost *Mathstation's* longevity and player retention substantially, due

to the increased engagement. This corresponds to the competition, collaboration and cooperation on the LM-GM map [4].

The following subsections explore the possibilities on how to incorporate any of the two variants into the respective game modes, extending their assets and functionality in order to enhance engagement through multiplayer features.

7.4.1. Fighting

The normal single player variation of this game has a player facing off against one of the game's bosses. A multiplayer cooperative mode could have teams of two or more players cooperating in order to bring down an adapted version of the same boss, or a new monster entirely. In order to effectively work together, players must execute attacks and solve math to deal the damage, thereby relying on each other's performance in order to resolve the level successfully.

Competitive multiplayer, on the other hand, lends itself to two interesting interpretations whereas this mode is concerned; either pitting players against each other to see who can down a boss first, or letting the players duke it out in a one on one setting.

The former is a race, with some minor interactions added so that the players may affect each other's performance, such as adding extra attacks that the other player must defend from, hindering their progress in unexpected ways that stay true to the action genre [3].

The latter, in contrast, will have none of the featured bosses interfere with an intense duel carried out by the players. The same game mechanics of math loaded attacks and defenses are now used to fight an opponent player. As a consequence, the game will not freeze in the background while carrying out attacks. Suddenly, it is not only important to have good aim and fast reaction time, it is now imperative to solve the math questions quickly in order to outperform the opponent. This particular vision for a competitive multiplayer game also ties into some of the customization options discussed earlier, as the player's avatar and ships will make an appearance in game, along with other custom visual or sound effects expressed by the players.

7.4.2. Mining Game

The mining game is a strategic resource management game, wherein the challenge correlates to the growing disorganization of the given situation. For an example of a multiplayer version for this game, one must look no further than some of the staple

strategy games available today, such as *Starcraft* [15] or *Age of Empires* [35].

The most obvious of the two is competitive, wherein the players compete over the same resource pool in order to achieve victory. Now, in the context of the mining game's original objective, which is transporting an asteroid using its mined resources, players can compete for the resources of the same asteroid to attempt to push it into their side. This tug of war is then dictated by the player's ability to mine more materials, research the required modules faster (which in turn means solving the math problems quicker) and managing their equipment more efficiently than the opponent. A game mode as such will offer intense math based competition with a strategic element. Players will be able to show off their achievements gained in game thus far by deploying custom looking variants of the existing gadgets of the original mining game.

Moreover, the game is also suited for a cooperative setting. A prime example of how to achieve this is with *Starcraft's Archon* mode, which allows two separate players to control the same units. When translating this game concept into the mining game of *Mathstation*, two players will collaborate in order to deliver the asteroid to a common destination. Due to both players working together in order to construct the gadgets and manage them, as well as mining the resources, the challenge put up by the levels should be higher. It is then up to the players to come up with some division of labor in order to successfully pass the hazards. They might even complete their journey without a strict mapping of responsibilities, as long as they are efficient enough at gathering and researching.

7.4.3. Exploration game

The exploration game, which is a set of spatial puzzles navigated through by cruising the space ship and collecting upgrades, can have a similar treatment to it as the games described above.

A cooperative version of the same game could go in one of two routes. First option: Both players control separate vessels within the same space, upgrades will apply to both players. In such a setting, players work with each other to uncover the upgrades faster than they would alone. Additionally, some more puzzle elements will need to be introduced in order to establish some tougher tasks for the players to tackle as a team.

The other possibility for collaboration in this game mode does seem like an interesting take on the single player variant as well: asymmetric cooperative play. An example of this would be that one player controls the ship in the level, and the other has to control other elements in the level, such as the artifact location or solutions to puzzle

elements, in such a way that makes the solution of the level possible for the ship player. In this configuration, players will have to rely on their peer's ability to construct logical puzzles as well as solving these. Solving math problems will then be part of each of the roles.

In a competitive setting, players can compete for the artifacts within a level in order to race to the goal first. Adding a wide range of artifacts to the mix will expand the tactical opportunities players will have in game to gain the upper hand in the race, or actively mess with each other's progress. Solving math suddenly becomes a tool of interference and destruction in this mode, applying pressure on the opponent the more proficient a player is at it.

Both variants still retain the core mechanics of the original game, which includes exploration and discovery with some math solving frosting on top. Custom ships are able to shine through in this mode as well, as is discussed in the customization section.

Lastly, it's worth to mention that competitive multiplayer games are a point of contingency. According to [36], while competitiveness does have a positive effect on scores, it can also lead to less time spent socializing or for leisure, which in turn translates into less happiness. It is important to keep this in mind when planning competitive multiplayer variants of this educational game. Additionally, it is a concern that any such competition would lead to certain students being at the top permanently while others struggle to ever get there. In severe cases, this might even become a form of bullying, which is the opposite of the sense of community and social behaviour that we wish to establish in the first place. If it creates an unhealthy environment, the responsible feature should not be bothered with. It would be desirable to achieve the positive learning effects while keeping the negative side effects to a minimum.

In summation, the above descriptions depict possible multiplayer versions based on the original game modes of *Mathstation* in order to capitalize on the enhanced engagement, in accordance with [34], and the potential longevity in retention properties associated with it. These are merely speculative suggestions loosely based on existing game design patterns, and are in no way the proven way to implement these in game until further research justifies those as such.

7.5. Hub Area

Originally, in the earlier stages of the game design development process, the game would feature a hub area. The area would have filled the role of menu navigation and mission selection, but it would also conveniently realize role playing game elements [3]. These would include non player characters with which the player could interact through

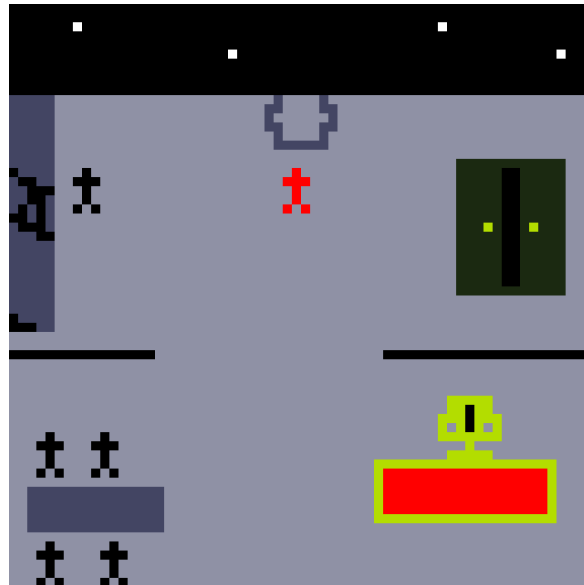


Figure 7.1.: Early Hub World Concept

their avatar. This would have echoed events in the game's narrative, unlocked secret missions and allow the game world to feel more alive and flavourful. The hopes with these are that the increased sense of depth of the world and it's inhabiting characters will increase the player's immersion, and in turn interest and learning yields. This corresponds to the simulation mechanic on the LM-GM map [4].

Figure 7.1 depicts a coarse design to how the layout of such a hub environment would look like. It includes non player characters as the black figures, the player as a red figure, utilities such as a closet to customize the avatar, as well as an achievement and trophies section. The console on the top center would have served as a mission selection terminal.

7.6. External features

The following future work subjects are considered external to the game. They are not part of the game's content per se. Rather, those can be considered as external enhancements that will make the serious game a more capable tool with rich extra game features. These features will be denoted in this section.

7.6.1. Instructor Interaction

The ability for an instructor, such as a math teacher, tutor or a parent or guardian, to interact with a player's in game environment should have a big impact on the player's ability to learn math from the game. After all, that is the main selling point of *Prodigy's* premium membership [18]. In the parent account description, linking these to a child's account enables progress tracking, so that the supervisor can monitor the player's performance in the math disciplines, offer assistance and supplemental activities.

On the game side of things, the parent account can set an in game reward for completion of math skills. Offering in game achievements and status in tandem to practicing the most distressed skills boosts the fun aspect of the game, hopefully resulting in better quality of learning than a traditional grade report.

Extending the game's math editor to intelligently identify in which areas the player is struggling is a key to making this feature a reality. A supervisor can detect a distressed pattern from a player if they are playing to their ability, and offer in game missions with a corresponding math configuration in order to help the pupil train the necessary skills more frequently.

Other such supplemental activities could be taken into consideration, such as story elements, achievements and symbols of status, and even further game modes or custom created levels that all aim to tackle a player's struggle with a math subject.

7.6.2. Class Integration

A class wide integration facilitates a small community based around the same math skills that should follow the class's curriculum. This combines the benefits of the instructor interaction section with an environment in which pupils can play together, compete or collaborate, and compare their achievements and progress in the game.

Building a community around the class makes sense because of the common learning goals of the class. Math editor configuration should in theory be identical to that of any other peer of the same group.

Finding people to play with is also made easier with the class integration, and if the game modes were extended, this could even be considered as a possible group assignment for pupils to carry out together, training selected math skills while doing so. This goes a step beyond *Prodigy's* [18] classroom integration, which serves the purpose of monitoring and tracking above facilitating a multiplayer community for the class, rather than finding a fine tuned balance for both capabilities.

Another benefit of classroom integration, apart from those listed above, is the sense

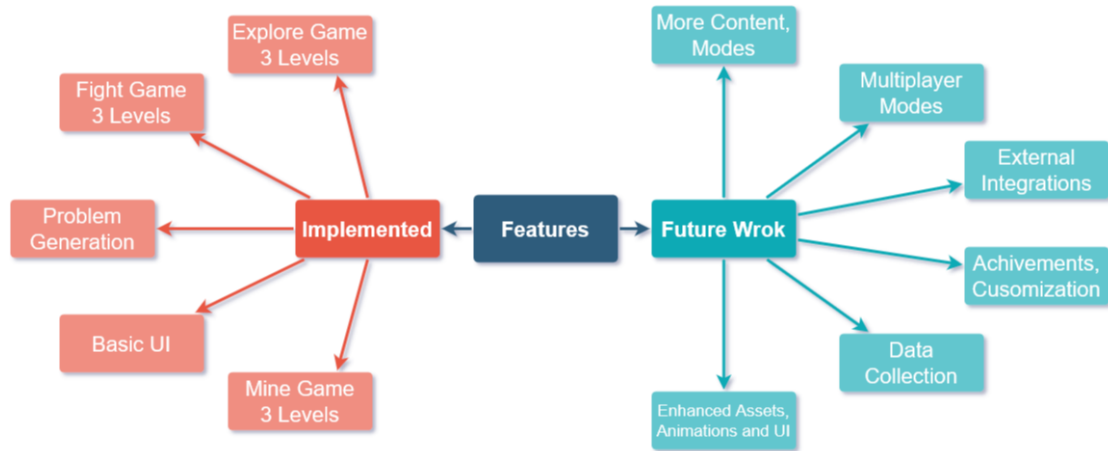


Figure 7.2.: Features of the thesis, implemented or considered for future work

of community. Once removed from the school and restricted to distance learning, a large part of the children’s daily social interactions have been removed, or severely limited in some sense. Having a common activity between classmates might be able to restore some sense of class social life during social distancing. This should be positive by itself, yet it might even motivate the kids to study further.

7.6.3. Data collection

As mentioned in the previous chapter, *Mathstation* possesses some qualities that would make the game a good fit for long term studies, wherein a large amount of data is collected over the course of a predetermined time window. Such tests could provide data that is richer than testing the game in short play sessions in computer labs of schools.

However, where this falls short is in two key points. The first being the volume of content currently included in the game, which will need to increase substantially in order to make a long term study feasible. The second, is a technical capability of collecting in game metrics about the interaction with the game, with the user’s consent. Having the ability to do so will provide precious data about the game’s usability, retention quality and math related data. The data in question can explain how a user fairs with their current subjects, and whether any distressed topics are identified.

In order to realize this, data collection components would have to be implemented within the game. Connect these with any other interesting components found within the game, such as the math editor’s problem generation and solution evaluation components, and allow users the option of sharing this data.

7.6.4. Evaluation

As stated in the evaluation chapter 6, due to the pandemic's restrictive circumstances, it was not possible to conduct a worthwhile study in order to determine *Mathstation's* assumed positive effects on learning outcome during school outages. Because of this, it is considered for future work.

7.7. Beyond Schools

Mathstation so far has only been considered as a math exercising game within the context of school pupils, but the tool might have potential with other audiences as well.

Consider all of the math games that are intended to a general target audience, such as the timeless classic Sudoku or more recent games such as the *Brain Age* series and number based mobile games [12]. These activities are often times enjoyed by all age groups, and can prove to be quite invaluable among the elderly to maintain a healthy mind. According to English [14], "Brain Age software use led to enhanced cognitive performance over time, but it did not do in a manner that exceeded the effects achieved by general cognitive stimulation". Even though it does not surpass general cognitive simulation, offering this option can be of value as an alternative tool for brain exercise. We can learn from this that *Mathstation* does have a place among these games for this target audience.

The adaptation to new audiences might not be a difficult one at all. *Mathstation* already prides itself with it's plethora of game activities and a wide range of included math content with which to play. If the game is handled correctly, it might have some success with these audiences, even though it was not made for it originally. Studies in this direction may therefore be warranted, as they would express the value of the product with other age groups.

7.8. Summary

In summation, while the basis for an educational math serious game has been achieved with *Mathstation*, many of the intended features are not currently included in the game. Some of the features are crucial in order to affirm the game's teaching capabilities, via game related content or features that are closely related to education, testing and community.

The game's current state, however, has been implemented to be extendable, and does allow for new features to be introduced with relative ease. This is reflected in numerous aspects of the game, by the ability to easily produce more levels with the

provided building blocks, and by the math editor related class hierarchy, which can be extended with new policies, rules and values to enrich the mathematical content thereof.

The following lists present the features in the game as well as those considered to be valuable future work assets, and where they lay on an importance scale.

Existing features

- Math problem generation, editor, and configuration
- Fighting Game with three levels
- Mining Game with three levels
- Explore game with three levels
- Basic menus and UIs, serviceable state

Important future work features

- More level content, math subjects (Geometry)
- Community features, multiplayer, classrooms
- Data collection, evaluation tools
- More game modes

Other future work features

- Enhanced UI, animations, sound effects
- Content for other audiences (elderly)
- Enhanced story elements
- Achievements, customization

Going forward, it is recommended to research the potential value of each of the yet to be implemented features, in order to prioritise these correctly for implementation. While some evidence of significance is provided to all of the considered features, their relevance may be context dependant, and change based on the eventual testing environment.

8. Conclusion

This chapter contains a discussion of the results followed by a hindsight reflection. It offers insight to the development process and enlightens about what could have been done better. It denotes the lessons learned from the experience as well.

The project itself was a challenging one to undertake and provided a plethora of interesting tasks to perform in order to form a functioning serious game that, to the best of our knowledge, might achieve its goal of providing relief for distance learning pupils.

In summation, *Mathstation* is a mathematics based serious game, intended to be a supplemental source of learning through game activity for pupils who have to distance learn, either due to current restrictions or in general. Especially during times where distance learning is the norm, but above all else, as an alternative tool to be used by whom it would benefit under any circumstances.

The game makes use of a mini game collection structure, enabling different activities for players to choose from. Each of these activities is based on some well defined genres of the video game industry, and is bundled with math solving as an organic part of the game. Theoretically, a player who prefers to play only a single activity can practice the entire available curriculum with their mode of preference, or spread their interest among all of the offered activities.

Moreover, the game includes a math editor interface that is used to convey the math problems in text book similar manner. It allows the configuration of preferred content for problem generation, and offers input and feedback based on the player's correctness. Included math subjects range from basic arithmetic to fraction division based on math curricula of grades one to four [32], making it suitable for kids in that age group. It is most notably missing geometry.

Though it is lamentable that COVID-19 related restrictions did not allow for ideal testing conditions at the schools, it is safe to say that the effort is not in vain. At the time of writing, Germany is exiting it's third wave, and as life starts to slowly find it's way to the daily routines of early yesteryear, surely an opportunity to put *Mathstation* to the test will arise before long.

All of the relevant testing methods and recommendations have been carefully curated

and prepared for that day, and are available in chapters 6 for evaluation, 7 for future work as well as the appendix A containing relevant materials.

The game's code base has been thoroughly prepared for submission. Comments are widely available to explain in detail what the classes are responsible for as well as their components. Code reuse techniques have been implemented wherever possible. Through usage of object oriented principles, such as inheritance and polymorphism, it was possible to achieve the most compact form of files with little to no unnecessary duplication.

Surely there was plenty of room for improvement though. Notes about visual elements aside, the code base of the game has evolved substantially from the initial tasks to the final form in which it is presented. Admittedly, there was a lot of learning about both the Unity game engine's intricacies as well as programming with the C# language during implementation. As a result, the artifacts created later in the lifetime of the project's time window have enjoyed a greater amount of experience and proficiency. As such, starting the project over would have enhanced the quality of some of the earlier artifacts implemented.

The game does currently only contain self created art assets and effects. While it does make it safe for publication, incorporating some higher fidelity effects for the game will surely make for a greater user experience, as the basic nature of current effects sorely lacks the polish of industry standard with which it would ultimately compete against for the kid's attention.

In hindsight, though it was daunting to provide game logic for three separate and unrelated game modes, the limited depth of the modes themselves along with the fact that only three of them made the cut, kept the backlog to a manageable size within the working time for the thesis. Whether the design choice of quantity of modes over the depth of a single one was worth it or not remains to be seen in the game's eventual evaluation.

Additionally, it would be most interesting to observe any studies meant for different target audiences. As mentioned in chapter 7, this kind of solution might have hidden potential with the general population and the elderly in particular.

In conclusion, due to the unavailability of schools, there has not been an opportunity to test the game with it's intended target audience in satisfactory testing conditions. The most important next step is the testing of *Mathstation* in these conditions, in order to reach a conclusion for the research question of this thesis, which remains unanswered as of now.

A. Math Questions

This appendix includes math questions to use for testing, grouped by the possible configuration states of the math editor and the user's class. Sections can be added and omitted as necessary for the group at hand.

All questions are from the website Help Teaching [13]. Please refer to the evaluation chapter, chapter 6, for more info about this source.

A.1. Base Configuration

A.1.1. 1st Grade

1. $6 + \underline{\quad} = 11$

Solution: 5

2. $2 + 5 =$

Solution: 7

3. What is $2 + 10$?

a) 20

b) 12

c) 10

d) 14

Solution: b

4. $8 - 7 = \underline{\quad}$

Solution: 1

5. Santa wants to make sure that he has enough toys for all the children. So far he has 10 toys made. He has to make 15 toys in all. How many toys does he still need to make?

a) 2

b) 3

c) 4

d) 5

Solution: d

A.1.2. 2nd Grade

1. What number is 100 less than 998?

a) 980

b) 999

c) 988

d) 898

Solution: d

2. 270 is how many more than 230?

- a) 30
- b) 40
- c) 20
- d) 10

Solution: b

3. $12 + 15 =$

- a) 1215
- b) 12
- c) 3
- d) 27

Solution: d

4. What is 10 more than 384?

- a) 394
- b) 484
- c) 374
- d) 284

Solution: a

5. Solve.

$$25 + 50 + 50 =$$

- a) 122
- b) 125
- c) 124
- d) 126

Solution: b

A.1.3. 3rd Grade

1. Joaquin had 187 markers. His friend Xavier gave him 101 markers for his birthday. How many markers does Joaquin have now?

- a) 288
- b) 290
- c) 281
- d) 295

Solution: a

2. $51 + 84 = ?$

Solution = 135

3. Select the correct equation needed to solve this problem.

Katlin had some bubble gum in her purse. She chewed 3 pieces. Then she still had 10 pieces left in her purse. How many pieces did she have in her purse to start with?

- a) $10 - g = 3$
- b) $10 + 3 = g$
- c) $10 - 3 = g$
- d) $g + 3 = 10$

Solution: b

4. Find the sum.

$$427 + 152 =$$

- a) 379
- b) 588
- c) 335
- d) 579

Solution: d

5. The sum is the answer to an addition problem.

- a) True
- b) False

Solution: a

6. Daryl is 23 years old. His brother Larry is 11 years younger. Which number sentence can you use to find how old Larry is?

- a) $23 - 11 = 12$
- b) $23 + 11 = 34$
- c) $34 - 23 = 11$
- d) $23 - 12 = 11$

Solution: a

7. Which operation should be used to solve the following problem?

If Gary's ruler is 12 inches, and Perla's ruler is 36 inches, what is the length difference between the two rulers?

- a) Addition
- b) Subtraction
- c) Multiplication
- d) Division

8. One hundred twelve subtracted from a certain number equals six. What is the number?

- a) 159
- b) 179
- c) 169
- d) 118

Solution: d

A.1.4. 4th Grade

1. What is the total distance traveled in one triathlon?

Parts of an Olympic-Distance Triathlon:

- Swimming: 1,500 meters
- Running: 10,000 meters
- Bicycling: 40,000 meters

Solution: 51,500 meters

2. Write the expression in words. $9 + 2$

Solution: Nine plus two

A. Math Questions

3. Last week the snack bar sold 1,024 hot dogs. This week it sold 1,155 hot dogs. What was the total number of hot dogs served for the two weeks?

- a) 131
- b) 1,179
- c) 2,079
- d) 2,179
- e) None of the above

Solution: 2179

4. The answer to an addition problem.

- a) Quotient
- b) Sum
- c) Difference
- d) Product

Solution: b

5. $2 + G = 6$

$G = \underline{\quad}$

Solution: $G = 4$

6. What is the value of b in the equation below and why?

$$b - 32 = 46$$

- a) The value of b is 88 because $46 + 32 = 88$

- b) The value of b is 78 because $46 + 32 = 78$

- c) The value of b is 15 because $46 - 32 = 15$

- d) The value of b is 14 because $46 - 32 = 14$

Solution: b

7. What is the value of the variable in the following equation? $x - 62 = 21$

- a) 41

- b) 30

- c) 83

- d) 73

Solution: c

8. The distance from San Francisco to Chicago is 2,314 miles. The distance from Santa Fe to Seattle is 1,466 miles.

What is the difference, in miles, between these distances?

- a) 848 miles

- b) 942 miles

- c) 1,152 miles

- d) 1,858 miles

Solution: a

A.2. Multiplication

A.2.1. 2nd Grade

1. How many groups?



- a) 2 groups of 3 is 6
- b) 2 groups of 4 is 8

Solution: b

2. How many groups?



- a) 3 groups of 2 is 6
- b) 2 groups of 4 is 8

Solution: a

3. Which is correct?



a) $2 + 2 = 2 \times 3$

b) $3 + 3 = 2 \times 3$

Solution: b

4. Write how many on each line.



There are ___ rows of Santas.

There are ___ Santas in each row.

Add. $__ + __ = __$

Multiply. $__ \times __ = __$

Solution: in order of appearance –
2,3,3,3,6,2,3,6

5. Write how many on each line.



There are ___ rows of trees.

There are ___ trees in each row.

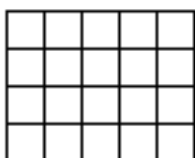
Add. $__ + __ = __$

Multiply $__ \times __ = __$

Solution: in order of appearance –
2,4,4,4,8,2,4,8

A.2.2. 3rd Grade

1. What multiplication sentence is shown by the array?



- a) $4 \times 4 = 16$
- b) $4 \times 5 = 20$
- c) $3 \times 4 = 12$
- d) $3 \times 5 = 15$

Solution: b

2. Paco has nine bags of toy cars. Each bag has 11 toy cars. How many toy cars does Paco have in all?

- a) 11
- b) 9
- c) 99
- d) 20

Solution: c

3. What is the missing number?

$$5 \times \underline{\quad} = 20$$

- a) 4
- b) 5
- c) 10
- d) 15

Solution: a

4. A group of people took 3 cars to a pool. Each car held 5 people. How many people were there in total?

- a) 10
- b) 25
- c) 15
- d) 50

Solution: c

5. Ellie looked at the fish in the pet store. She counted exactly 6 fish in each aquarium. There were 7 aquariums. What was the total number of fish in the aquariums?

- a) 63
- b) 42
- c) 54
- d) 72

Solution: b

A.2.3. 4th Grade

1. If 35 is 7 times as many as 5, which other statement is true?

- a) 35 is 5 more than 7.
- b) 5 is 7 more than 35.
- c) 35 is 5 times as many as 7.
- d) 5 is 7 times as many as 35.

Solution: c

A. Math Questions

2. Carol received 189 votes for class president. Emma received twice that number of votes. How many votes did Emma receive.

- a) 278
- b) 368
- c) 378
- d) 345
- e) Non of the above

Solution: c

3. Romeo watched 5 television shows that were each 26 minutes long. How much time did he spend watching television?

- a) 31 minutes
- b) 21 minutes
- c) 100 minutes
- d) 130 minutes

Solution: solution: d

4. $302 \times 5 =$

- a) 1,505
- b) 1,510
- c) 1,500
- d) 1,502

Solution: b

5. Marcus has 3 apples, but Molly has 6 times more apples. How many apples does Molly have?

Solution: 18

6. $49 \times 34 =$

Solution: 1666

7. Joseph sold 24 tickets to a school play. The tickets cost \$7 each. Which shows how to calculate the total cost of the tickets Joseph sold?

- a) $(20 \times 7) + (20 \times 4)$
- b) $(24 \times 7) + (20 \times 7)$
- c) $(20 \times 7) + (4 \times 7)$
- d) $(24 \times 7) + (4 \times 7)$

Solution: c

8. A bus driver travels 234 miles every day. How many miles does the bus driver travel in 5 days?

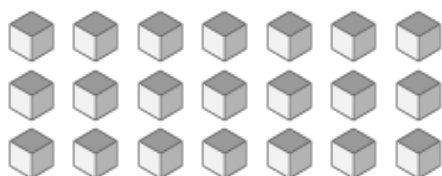
- a) 1,050 miles
- b) 1,150 miles
- c) 1,170 miles
- d) 1,520 miles

Solution: c

A.3. Division

A.3.1. 3rd Grade

1. Count the blocks:



Into how many equal groups of 3 can the blocks be divided?

- a) 3
- b) 4
- c) 7
- d) 21

Solution: c

2. What is the missing number?

$$15 \div \underline{\quad} = 3$$

- a) 3
- b) 5
- c) 12
- d) 8

Solution: b

3. What is the missing number?

$$\underline{\quad} \div 6 = 3$$

- a) 6
- b) 9
- c) 12
- d) 18

Solution: d

4. The entire fourth grade class is going to the zoo. There are 3 buses for the field trip. Each bus has the same amount of kids. If there are 90 kids in the fourth grade, how many are on each bus?

- a) 30 kids
- b) 60 kids
- c) 25 kids

Solution: a

5. Which story could be solved with $64 \div 8$?

- a) Tim used all but 8 of the 64 nails. How many nails were used?
- b) Mrs. Pemberton bought 64 bags of hotdog buns for the picnic. Each bag had 8 buns. How many buns did she buy?
- c) Alex made 64 placemats. If he makes 8 more, how many placemats will he have made?
- d) Jeff has 64 ounces of juice. Each glass holds 8 ounces. How many glasses can he have?

Solution: d

A.3.2. 4th Grade

1. What number is a factor of 6?

- a) 5
- b) 3
- c) 15
- d) 19

Solution: b

2. $222 \div 6 =$

- a) 37
- b) 58
- c) 27
- d) 15

Solution: a

3. Sally has 57 beads. If she wants to split the beads evenly between her 4 friends, how many beads will she have left over?

- a) 1
- b) 98
- c) 25
- d) 4

Solution: a

4. $8 \overline{)280}$

- a) 35
- b) 38
- c) 40
- d) 42

Solution: a

5. $230 \div 5 = 57$

- a) True
- b) False

solution: b

6. $72 \div 8 = 9$

72 is the ____.

- a) dividend
- b) remainder
- c) sum
- d) quotient

Solution: a

7. Jared and his dad went fishing. Jared caught 7 fish and his dad caught 21 fish. How many times more fish did Jared's dad catch than Jared?

- a) 3
- b) 6
- c) 14
- d) 28

Solution: a

8. When 7,819 is divided by 4, what is the remainder?

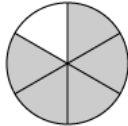
- a) 3
- b) 5
- c) 8
- d) 1,954

Solution: a

A.4. Fractions

A.4.1. 3rd Grade

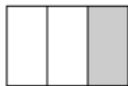
1. What fraction does this picture show?



- a) $\frac{1}{3}$
- b) $\frac{1}{2}$
- c) $\frac{5}{6}$
- d) $\frac{1}{1}$

Solution: c

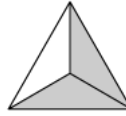
2. What fraction of the shape is shaded?



- a) $\frac{1}{2}$
- b) $\frac{1}{3}$
- c) $\frac{1}{4}$
- d) $\frac{2}{3}$

Solution: b

3. What fraction of the shape is shaded?



- a) $\frac{1}{3}$
- b) $\frac{3}{3}$
- c) $\frac{1}{2}$
- d) $\frac{2}{3}$

Solution: d

4. In a fraction, the number on the bottom is the ____.

- a) Fraction
- b) Numerator
- c) Denominator

Solution: c

5. $\frac{1}{4}$ of the pictures are dragons.



- a) True
- b) False

Solution: a

A.4.2. 4th Grade

1. Is $\frac{1}{8}$ less than, greater than, or equal to $\frac{1}{2}$?

- a) less than $1/2$
- b) greater than $1/2$
- c) equal to $1/2$

Solution: a

2. $\frac{3}{10}$ is equal to which fraction?

- a) $\frac{9}{30}$
- b) $\frac{2}{5}$
- c) $\frac{5}{10}$
- d) $\frac{3}{6}$

Solution: a

3. What is the sum in it's simplest form?

$$\frac{3}{10} + \frac{2}{10}$$

- a) $\frac{1}{2}$
- b) $\frac{2}{5}$
- c) $\frac{5}{20}$
- d) $\frac{1}{20}$

Solution: a

4. What whole number makes the equation below true?

$$\frac{1}{2} = ? \times \frac{1}{6}$$

- a) 1
- b) 5
- c) 6
- d) 7

Solution: b

5. Which is a pair of equivalent fractions?

- a) $\frac{2}{3}, \frac{4}{9}$
- b) $\frac{3}{4}, \frac{15}{20}$
- c) $\frac{6}{9}, \frac{36}{81}$
- d) $\frac{7}{8}, \frac{49}{64}$

Solution: b

6. Which fractions are equivalent?

- a) $\frac{1}{3} = \frac{6}{9}$
- b) $\frac{3}{4} = \frac{15}{20}$
- c) $\frac{5}{6} = \frac{11}{12}$
- d) $\frac{3}{5} = \frac{6}{15}$

Solution: b

Bibliography

- [1] V. V. Abeele, K. Spiel, L. Nacke, D. Johnson, and K. Gerling. "Development and Validation of the Player Experience Inventory: A Scale to Measure Player Experiences at the Level of Functional and Psychosocial Consequences." In: *International Journal of Human-Computer Studies* 135 (2020), p. 102370.
- [2] F. Aldhaban. "Exploring the adoption of Smartphone technology: Literature review." In: *2012 Proceedings of PICMET'12: Technology Management for Emerging Technologies*. IEEE. 2012, pp. 2758–2770.
- [3] T. H. Apperley. "Genre and Game Studies: Toward a Critical Approach to Video Game Genres." In: *Simulation & Gaming* 37.1 (2006), pp. 6–23.
- [4] S. Arnab, T. Lim, M. B. Carvalho, F. Bellotti, S. de Freitas, S. Louchart, N. Suttie, R. Berta, and A. de Gloria. "Mapping Learning and Game Mechanics for Serious Games Analysis." In: *British Journal of Educational Technology* 46.2 (2015), pp. 391–411.
- [5] I. Arroyo, M. Micciollo, J. Casano, E. Ottmar, T. Hulse, and M. M. Rodrigo. "Wearable Learning." In: *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*. Ed. by B. Schouten, P. Markopoulos, Z. Toups, P. Cairns, and T. Bekker. New York, NY, USA: ACM, 10152017, pp. 205–216.
- [6] J. Biburger. *Lambacher-Schweizer: Mathematik für Gymnasien, Bayern*. 1. Auflage. Stuttgart and Leipzig: Ernst Klett Verlag, 2019.
- [7] D. Bormann and T. Greitemeyer. "Immersed in virtual worlds and minds: effects of in-game storytelling on immersion, need satisfaction, and affective theory of mind." In: *Social Psychological and Personality Science* 6.6 (2015), pp. 646–652.
- [8] K. A. Cagney and D. S. Lauderdale. "Education, Wealth, and Cognitive Function in Later Life: Department of Health Studies, University of Chicago, Illinois." In: 57B (2002).
- [9] B. Cowley et al. "Toward an Understanding of Flow in Video Games." In: *Computers in Entertainment* 6.2 (2008), pp. 1–27.

- [10] E. Crawley, M. Loades, G. Feder, S. Logan, S. Redwood, and J. Macleod. "Wider Collateral Damage to Children in the UK Because of the Social Distancing Measures Designed to Reduce the Impact of COVID-19 in Adults." In: *BMJ Paediatrics Open* 4.1 (2020).
- [11] G. De Prato, C. Feijóo, J.-P. Simon, et al. "Innovations in the Video Game Industry: Changing Global Markets." In: *Communications & strategies* 94.2 (2014), pp. 17–38.
- [12] N. S. P. Development. *Brain Age*. Game [Nintendo DS]. May 2005.
- [13] S. Education. *Help Teaching*. Accessed: 2021-06-09. Feb. 2004. URL: <https://www.helpsteaching.com/>.
- [14] S. M. English. "Cognitive training with healthy older adults: investigating the effectiveness of the brain age software for nintendo DS." In: (2012).
- [15] B. Entertainment. *STARCRAFT II*. Game [Microsoft Windows, macOS]. Irvine, California, USA, July 2010.
- [16] P. Hille. *COVID: Schools Are in Lockdown and E-Learning is a Struggle*. Accessed: 2021-06-02. Jan. 2021. URL: <https://p.dw.com/p/3nacb>.
- [17] W. A. IJsselsteijn, Y. A. de Kort, and K. Poels. "The Game Experience Questionnaire." In: *Eindhoven: Technische Universiteit Eindhoven* 46.1 (2013).
- [18] S. Inc. *Prodigy*. Game [Browser, iOS, Android]. 3370 South Service Rd Suite 304, Burlington , ON, L7N 3M6, Canada, Sept. 2015.
- [19] Innersloth. *Among Us*. Game [Windows, Consoles, Mobile]. Redmond, WA, USA, June 2018.
- [20] A. A. Lapada et al. "Teachers' Covid-19 Awareness, Distance Learning Education Experiences and Perceptions towards Institutional Readiness and Challenges." In: *IJLTER* 19.6 (2020), pp. 127–144.
- [21] E. L.-C. Law, F. Brühlmann, and E. D. Mekler. "Systematic Review and Validation of the Game Experience Questionnaire (GEQ) - Implications for Citation and Reporting Practice." In: *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play*. Ed. by F. ' . Mueller and A. S. I. G. o. C.-H. Interaction. [Place of publication not identified]: ACM, 2018, pp. 257–270.
- [22] M. Lichtheim. *Ancient egyptian literature*. Univ of California Press, 2019.
- [23] H. Lin and C.-T. Sun. "Cash Trade Within the Magic Circle: Free-to-Play Game Challenges and Massively Multiplayer Online Game Player Responses." In: *DiGRA Conference*. Citeseer. 2007.
- [24] 3. LLC. *Papers, Please*. Game [Windows, OS X, Linux, iOS, PlayStation Vita]. Saitama, Japan, Aug. 2013.

Bibliography

- [25] Re-Logic. *Terraria*. Game [Windows, Consoles, Mobile]. Floyds Knobs, Indiana, United States, May 2011.
- [26] S. H. Luis von Ahn. Duolingo. Native Client [Android, iOS, Windows Phone, Windows 10 Mobile, Web Browser]. Pittsburgh, Pennsylvania, United States, Nov. 2011.
- [27] Mojang Studios. *Minecraft*. Game [Windows, OS X, Linux, Consoles, Mobile]. Stockholm, Sweden, Nov. 2011.
- [28] National Research Council, Board on Mathematical Sciences and Their Applications, and Mathematical Sciences Education Board. *Everybody Counts: A Report to the Nation on the Future of Mathematics education*. Washington, D.C.: National Academic Press, 1989.
- [29] OECD. *PISA*. Accessed: 2021-06-09. URL: <https://www.oecd.org/pisa/publications/>.
- [30] D. A. Plecher, F. Herber, C. Eichhorn, A. Pongratz, G. Tanson, and G. Klinker. "HieroQuest - A Serious Game for Learning Egyptian Hieroglyphs." In: *JOCCH* 13.4 (2020), pp. 1–20.
- [31] A. Porter, J. McMaken, J. Hwang, and R. Yang. "Common core standards: The new US intended curriculum." In: *Educational researcher* 40.3 (2011), pp. 103–116.
- [32] H.-D. Rinkens, ed. *Welt der Zahl / Bayern / [Neuausgabe 2014]: [mathematisches Unterrichtswerk für die Grundschule]*. Bayern, [Neubearb.], Dr. A. Braunschweig: Schroedel, 2014.
- [33] M. Riopel et al. "Impact of serious games on science learning achievement compared with more conventional instruction: an overview and a meta-analysis." In: *Stud. Sci. Educ.* 55.2 (2019), pp. 169–214.
- [34] K. Siu, A. Zook, and M. O. Riedl. "Collaboration Versus Competition: Design and Evaluation of Mechanics for Games With a Purpose." In: *FDG* 10 (2014), pp. 14–22.
- [35] E. Studios. *Age of Empires*. Game [Microsoft Windows, Macintosh]. Oct. 1997.
- [36] L. A. Tehrani, M. A. Majd, and M. Ghamari. "Comparison of Self-efficacy, Test Anxiety and Competitiveness between Students of Top Private Schools and Public Schools." In: *Mediterranean Journal of Social Sciences* 5.23 (2014), p. 2749.
- [37] R. Tercatin. *Coronavirus: Schools Stay Closed Until Thursday as Gov't Continues Debate*. Accessed: 2021-06-02. Feb. 2021. URL: <https://www.jpost.com/israel-news/coronavirus-after-inconclusive-meeting-govt-to-reconvene-on-schools-658227>.

Bibliography

- [38] Twilio and Team Twilio Quest. Twilio Quest. Game [Windows, Mac, Linux]. San Francisco, California, USA, Oct. 2017.
- [39] K. Zickuhr. "Tablet ownership 2013." In: *Pew Research Center* 19 (2013).