Incorporating Video Games into Education

Interactive Qualifying Project completed in partial fulfillment of the Bachelor of Science degree at Worcester Polytechnic Institute, Worcester, MA

Submitted to: Vincent J. Manzo

Ben Korza ________________________________

Benny Peake ________________________________

April 29, 2015

____

ABSTRACT

Students often struggle learning math concepts. Educational video games could help mitigate this problem by providing an immersive learning environment that complements the traditional classroom setting. The objective of this project was to introduce basic math concepts through a fun, accessible, and engaging music-oriented educational video game prototype, and ascertain some knowledge as to the efficacy of this tool as a self-directed learning aid. Additionally, the prototype game we developed includes a set of assessment tools that teachers could use to monitor students' progress. The prototype was distributed to WPI students to conduct usability testing and provide feedback via survey. Our data suggest that this educational game has potential as a viable resource for self-directed education.

ACKNOWLEDGEMENTS

We would like to thank our adviser, V.J. Manzo, for supporting and guiding our project along the way. He helped us many times and without his help our project would surely have not made the progress it has.

We would also like to thank Dianna Wentzell, Jackie Coleman, and Marcy Reed for helping us distribute our game to school teachers and for providing valuable information about school teachers curriculums.

Finally we would like to thank the WPI students who helped test our application and for taking their time to give us valuable feedback.

AUTHORSHIP

This paper was written in collaboration by Ben Korza and Benny Peake. Its contents are the responsibility of the group.

TABLE OF CONTENTS

TABLE OF FIGURES

1 INTRODUCTION

Video games have emerged in recent years as a promising way to engage students in the classroom. Increasing numbers of educational games are being made today, focusing on a variety of subjects for all age groups. A search on Amazon’s app store shows that there are more than 7000 math games alone for mobile devices. Taking into account just the mobile marketplace we can estimate that there are probably over 20,000 math games for mobile devices. When we include other subjects and computer devices it is clear that this has become quite a large field within game design. Large title games such as *Total War* or *Civilization* have integrated historical facts into the in-game text blurbs. It’s clear that the use of games as an educational tool has started to become a major market.

The major goal of this project was to develop an effective an educational video game prototype that also provided teachers with assessment tools to track students’ progress. To start this project we spent a considerable amount of time researching different studies and existing educational games. Taking this research into account we produced design guidelines that we felt would create both a good game and an education tool. We then set forth to design an educational application to be used by teachers and students alike. By the end of the project we created a prototype of this product and collected feedback on it to see which aspects succeeded and which aspects failed. This helped us expand our understanding of educational game creation.

2 BACKGROUND

There is a wide variety of literature available that investigates educational games, and there are many different educational games throughout history worth examining. This section investigates some of the various research papers, games, and tools that pertain to educational game development. First, the usage of games in the current educational environment is examined to help identify how to best integrate games into a pedagogical context. Then, software and engines are analyzed as to identify how to create useful tools and easy to use interfaces. Finally, existing games and previous academic projects are reviewed in order to determine the best way to make immersive educational applications. This total assessment is used to identify design guidelines for creating an impactful educational game and tool.

2.1 Game Integration

2.1.1 Benefits of Games

When beginning this investigation of digital media in education it was necessary to examine the educational value of games. Studies suggest that videogames have potential to increase spatial reasoning. For instance, according to Begoña Gros of the Journal of Research and Technology in Education, video games can facilitate “the ability to read images, such as pictures and diagrams,” as well as improve skills in divided visual attention. This is important since scientific and technical thinking are crucial when using many computer applications (Gros, 2004).

Additionally, there is substantial empirical evidence that suggests games can improve several other key attributes essential to a successful education including school achievement, motivation to learn, and complex thinking skills. Such can be seen with the introduction of video games into a Chilean school curriculum, which resulted in teachers reporting both positive technological transfer and increased student motivation to learn after 3 months of integration (Rosas et al., 2003). Games also inherently force attention and concentration, since without these a student could not progress through the game (Rosas et al., 2003). All of these skills are crucial to a student's ability to learn and retain knowledge.

Video games offer students familiarity with basic computer operational skills that are necessary as society moves into the digital age. According to Gros, "The suite of skills children develop by playing such games can provide them with the training wheels for computer literacy, and can help to prepare them for science and technology, where more and more activity depends on manipulating images on a screen" (2004). Taking this into consideration demonstrates that not only can video games teach students the traditional subjects, but they can also prepare them for the modern world. This in combination with their ability to improve student's spatial, motivational, and thinking skills suggests that videogames are a solid platform for pedagogical use.

2.1.2 Why Games Are Not Already Integrated

Despite the potential benefits of using games for educational purposes, as evidenced in the aforementioned studies, very few have been integrated into educational environments before. Understanding why this is the case is relevant, since if an educational game cannot be integrated into school curriculums, then it doesn't matter how good at teaching it is. This problem of integration therefore becomes as crucial a design consideration as the game design itself. It turns out that utilizing games for self-directed learning outside of the classroom may be a better alternative.

Historically efforts in educational video games have widely failed. According to Gros, this is because the gameplay often found in these educational video games (or edutainment) titles "were very repetitive, poorly designed and did not support progressive understanding" (2004). Corroborating this opinion, Jensen, the Director of the Program in Comparative Media Studies at MIT, has stated that "most existing edutainment products combine the entertainment value of a bad lecture with the educational value of a bad game" (Rosas et al., 2003). If an edutainment game fails at being entertaining then it has failed to live up to one of its defining characteristics. In cases like this the edutainment approach doesn't provide much pre-existing teaching methods cannot already accommodate.

Incorporating video games into the classroom requires that educators possess either some pre-existing technology skills or a desire to learn them, which may not always be the case. Video games are a recent innovation, and did not widely enter the public eye until the 1972 release of *Pong* ("A History of Video Games", n.d.). Many teachers today are older than this genesis, and therefore may not be convinced by the viability of video games as instructional tools. In spite of this, teachers that have been skeptical have had their opinions changed when videogames successfully increase student's positive academic traits in the classroom environment (Rosas et al., 2003). Even if a teacher is comfortable with video games, they still may not know where to begin when it comes to integrating them into the classroom setting. Until these problems of familiarity are overcome, video games may be better suited to self-directed learning activities.

Another potential reason games may be better suited to self-directed learning than classroom use is likely related to how time consuming they are. According to Gros, "The main disadvantage of the use of games in the classroom is the amount of time it takes for both the student and teacher to guide themselves within the game." She concludes, "generally, the games require many hours and, on occasion, it is difficult to establish the sequences of play that should be significant for both the students and school curriculum" (2004). In circumstances where teachers have very little free time to design curriculums around video games, it is understandable why little reformation in the integration of video games has been attempted.

School computer labs often require dozens of computers, and there are many games that require expensive high end hardware to run. Taking this into account, the use of games for education can become quite expensive (Kirriemuir, 2002). Even when games are distributed via the internet, a physical device is still needed to play them. Likewise the use of special hardware such as consoles can be impractical since it required teachers to understand and setup these devices. Thus a major problem with classroom integration is creating games that can run on low end, everyday hardware that a teacher would already be familiar with and a school has easy access too. In the home setting, computers are almost ubiquitous, so out-of-class exercises are a promising alternative.

In terms of development, there is very little financial incentive for publishers to produce educational games. This is unlikely to change until a large success in the field of educational game development is achieved (Kirriemuir, 2002). The reality in and of itself produces a paradoxical scenario: publishers aren't creating educational games because no successful educational games exists. This in turn creates an environment that makes it difficult for videogames to prove themselves as pedagogical tools.

2.1.3 Features of Educational Games

Many efforts have been attempted in the past to create educational games, but few of them have succeeded. In light of this, several studies have investigated the best ways to create successful educational games. According to the set of authors who designed the e-Adventure project, a framework for integrating games into the classroom setting, some of the most prominent qualities required to create successful educational games include (Moreno-Ger, Burgos, Martínez-Ortiz, Sierra, & Fernández-Manjón, 2008):

1. Adequate and adaptive feedback.
2. The embedding of cognitive strategies such as repetition, rehearsal, paraphrasing, outlining, cognitive mapping, and the drawing of analogies and inferences.
3. Animated graphics, which increase achievement and/or reduce task time.

However, these are not the only values that go into a good educational game. According to these same researchers, "an effective educational game design must achieve a balance between fun and educational value." They also cites some additional key features of e-learning games including "student tracking, online assessment, user feedback or community features" (Moreno-Ger, et al., 2008). These features are presumably important because they make it possible to track a student's progress and promote effective communication between students and teachers. This is crucial to achieve optimal integration of any application into an educational environment as it helps teachers introduce the tool into their curriculum in a non-obtrusive way.

It's also important to look at existing examples of games that have incorporated education into their design. A game that particularly emphasizes this is *Counting Kingdom* due to its excellent game design, and focus in mathematics. It is depicted in Figures 1 and 2. One of the things that this game seems to benefit from the most is the simple but expandable core game mechanic. The game's core mechanics centers around simple addition and subtraction. It has the player use "spells" which have a number value on them in order to kill monsters. Each monster also has its own number value on it. The player chains together adjacent monsters when using a spell in order to create a sum that is equal to the spell. By doing this the player is able to defeat those monsters. The game later expands upon these mechanics by allowing the player to combine spells and use potions to change monsters numbers.

`The key to this game’s success was the relative simplicity of this mechanic. Rather than using complex mechanics, the game uses one single, simple mechanic for its gameplay which it then expands upon to add more complexity. In turn this makes it so the learning portion of the game (which comes from the main mechanics of the game) is also simple. This allows *Counting Kingdom* to combine learning and fun in a way that is intuitive and immersive to the player.

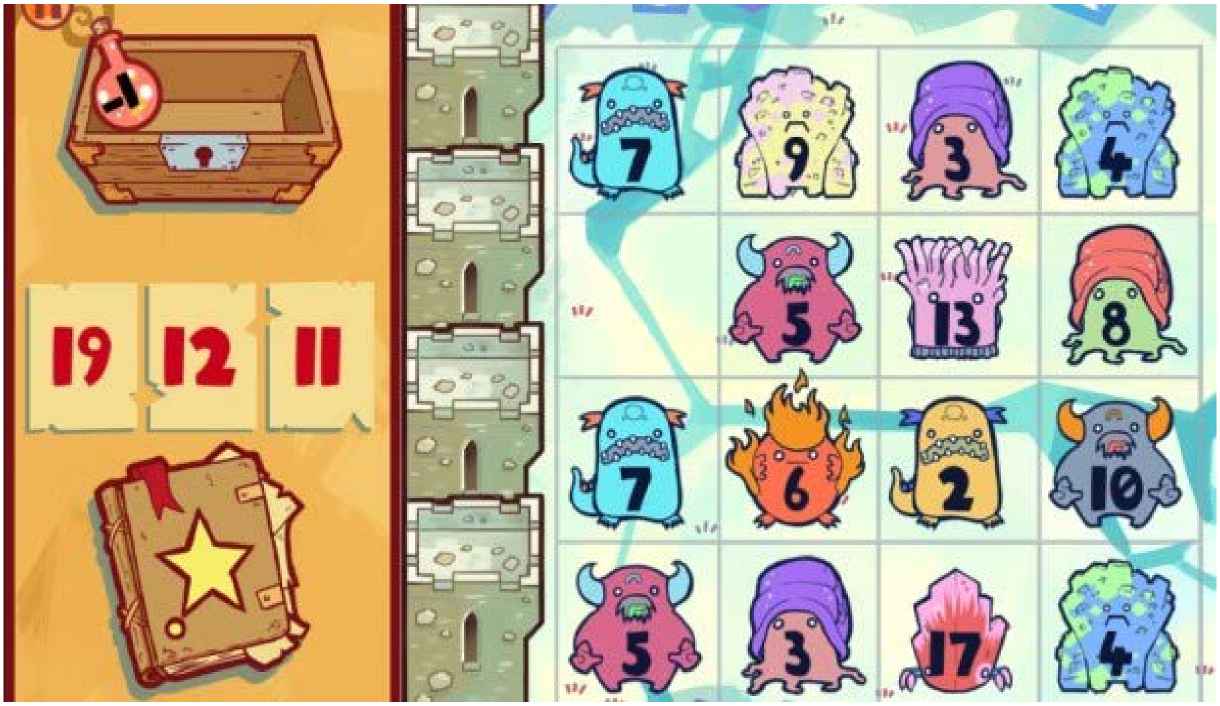


Fig 1: Counting Kingdom Gameplay Screen

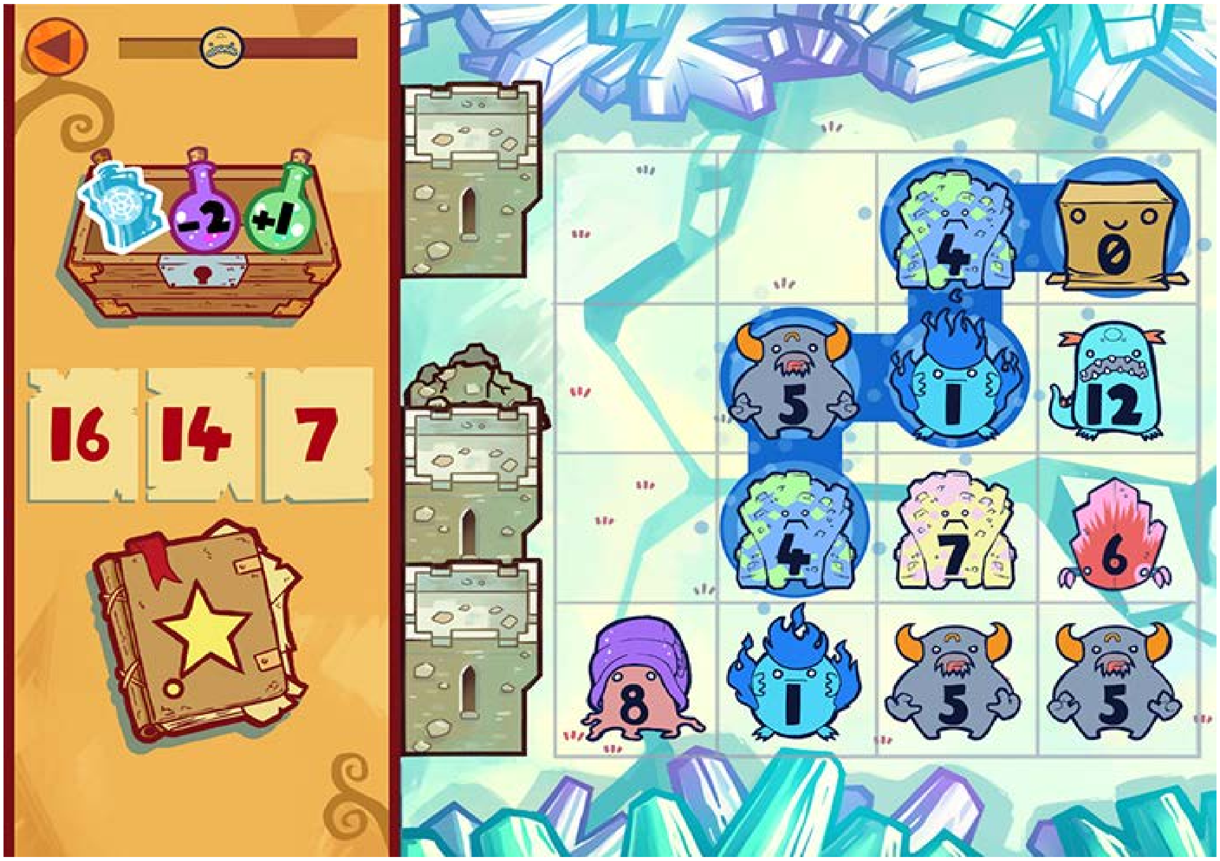


Fig 2: Combine enemies to add up to the spell number on the left

In addition to well designed games, it was also important for us to consider educational games that have failed so that we do not make the same mistakes. One game we looked at in particular was *Mario Learn to Type*, an old 3rd party Mario Game that was designed to teach students how to type. This game is shown in Figure 3: Mario Learn to Type “game” play. The game is laid out like many learn to type games, where letters would scroll by and it would show what finger to press it with. One of the main problems with this game is that it does not incorporate learning into gameplay, but rather integrates gameplay as an afterthought. If we ignore the Mario look of the game, then it is nothing more than a learn to type program. This causes the gameplay to feel rigid, uninteresting, and un-immersive. It is particularly important for an educational game to have some level of immersion, since this will allow learning to come more naturally to students. This makes it crucial to incorporate learning into the core components of an educational game, and not simply as an extra layer.

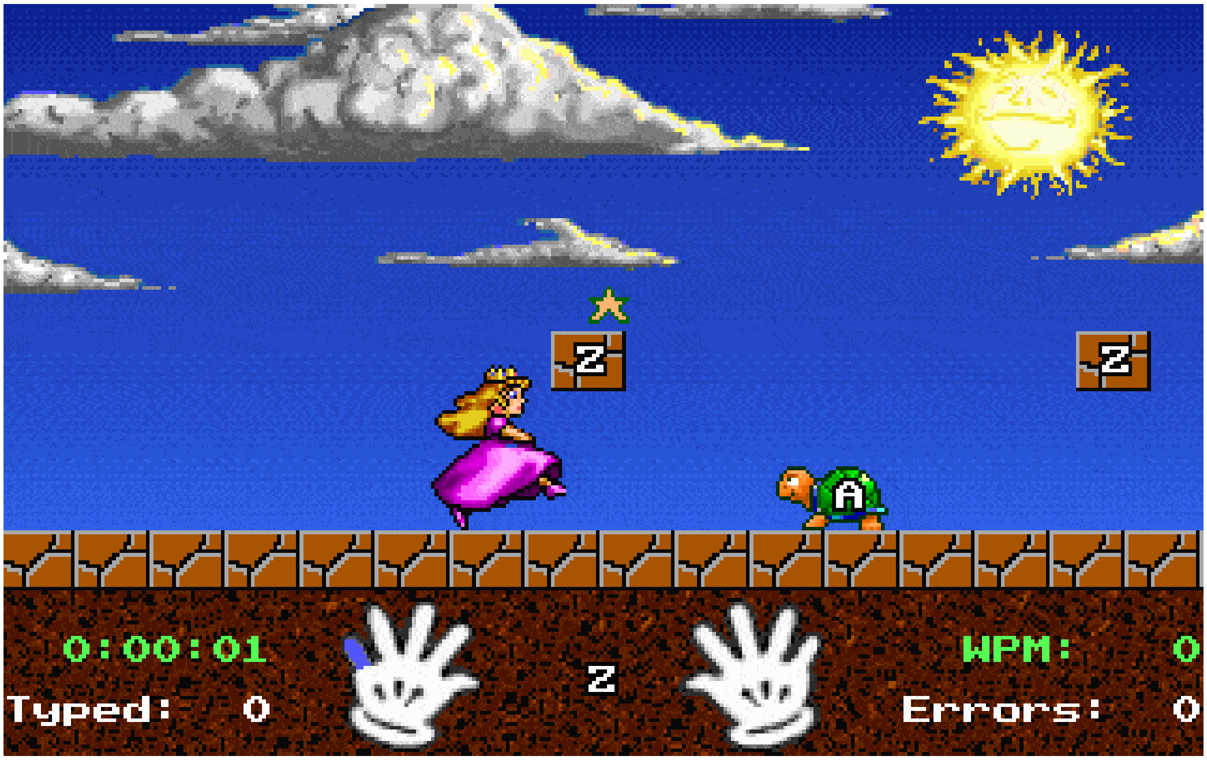


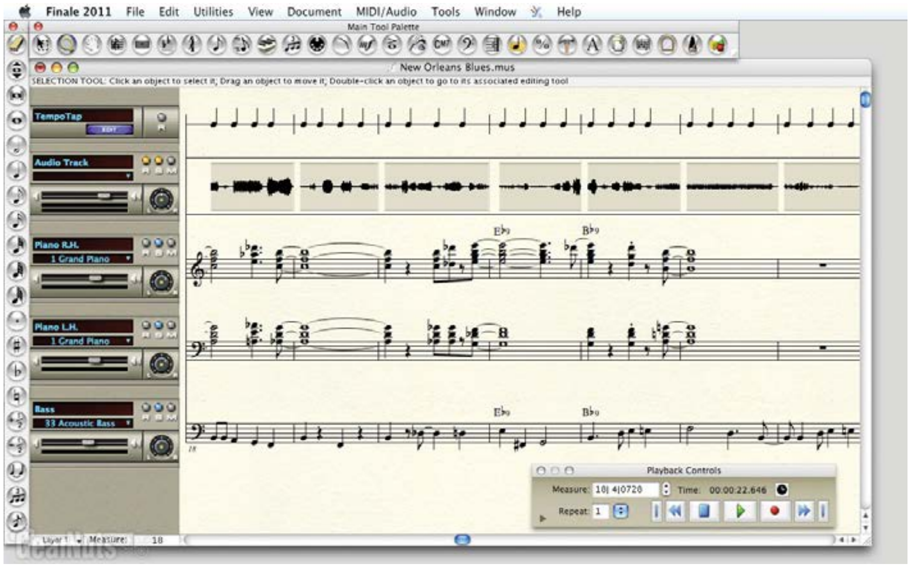
Fig 3: Mario Learn to Type "game" play

2.2 Tool Design

From the aforementioned research it was clear that games could provide benefits when used in a pedagogical context. Given this, research was now required in order to determine the best way to make an educational application. Since the previous research suggested the importance of assessment capabilities and flexible lesson development, part of this process involved examining various music applications and determining the benefits and properties of a good educational tool. Game engines were also examined, since we would need to use one of these in order to develop the prototype.

2.2.1 Tools - Composition Software:

Music composition has been greatly influenced by the presence of digital technology. Technology-based tools allow users to compose anything from simple piano arrangements to full-blown orchestral scores without the need for real-life performers. The technologies also allow their users to save their work to a digital format, share it with others, and modify it later. This helps students create work cleaner and more easily than they otherwise could using pen and paper. Additionally, these tools also help students become more technologically literate and prepare them for modern society. Two examples of such tools would be *Sibelius* and *Finale,* depicted in Figures 4 and 5 respectively. *Finale* in particular also offers classroom support by providing, “Hundreds of ready-made, educator-approved, customizable worksheets” and “hundreds of individual and classroom flash cards”(“Educator Tools”, n.d.). These tools demonstrate some of the qualities of effective student resources.



SELECTION TOOL: Click an object to select it; Drag an object to move it; Double-click an object to go to its associated editing tool

Fig 4: Finale Editor

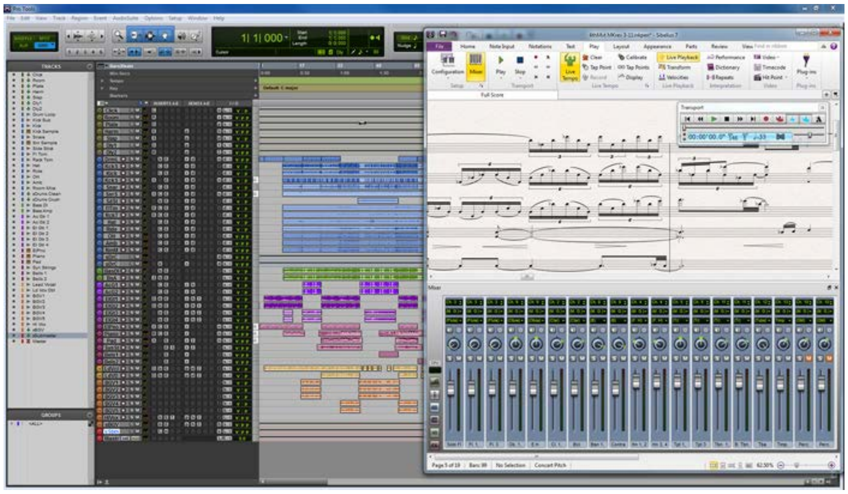


Fig 5: Sibelius Editor

2.2.2 Online Resources

Not all digital educational resources are limited to software. The internet has introduced a vast amount of resources that offer information that a user can use to research a particular subject of their interest. Some websites go so far as to offer interactive tools and games that are designed to teach a particular subject. One such example of these websites is the Pearson Music Program, which, for a fee, offers a wide variety of activities designed to teach young aspiring musicians the fundamentals of music (“Online Learning Exchange®”, n.d.). This is one among many viable instructional resources school teachers may pick from the internet. These resources make it clear that both accessibility and good game design are important for educational media to distinguish itself from the already vast amount of resources available on the internet.

2.2.3 Engines

Game engines are tools or sets of tools that are used to make games. There exist many game engines on the market varying in complexity and flexibility, and they are the most common way games are produced today.

One of the most popular engines out currently is *Unreal Engine 4*. It, along with previous versions of the engine, has been used to develop hundreds of games. One of the most beneficial things about this engine is its use of visual scripting called blueprints (referred to as kismet in previous versions). This visual scripting language provides an easy way for non-programmers to write simple scripts for gameplay without having to touch or know code. It uses a simple flow chart style to see how and when things will execute allowing for all the power of a programming language. This style can be seen in Figure 6: UE4 Kismet Editor.

While the *Unreal Engine* does provide a lot of power and creative freedom it does come at a price. The user must navigate complex menus and tools in order to access the engine’s functionality. This complexity can be quite daunting to a new user. Figure 7: Example of UE4 Complexity demonstrates how this could be the case.

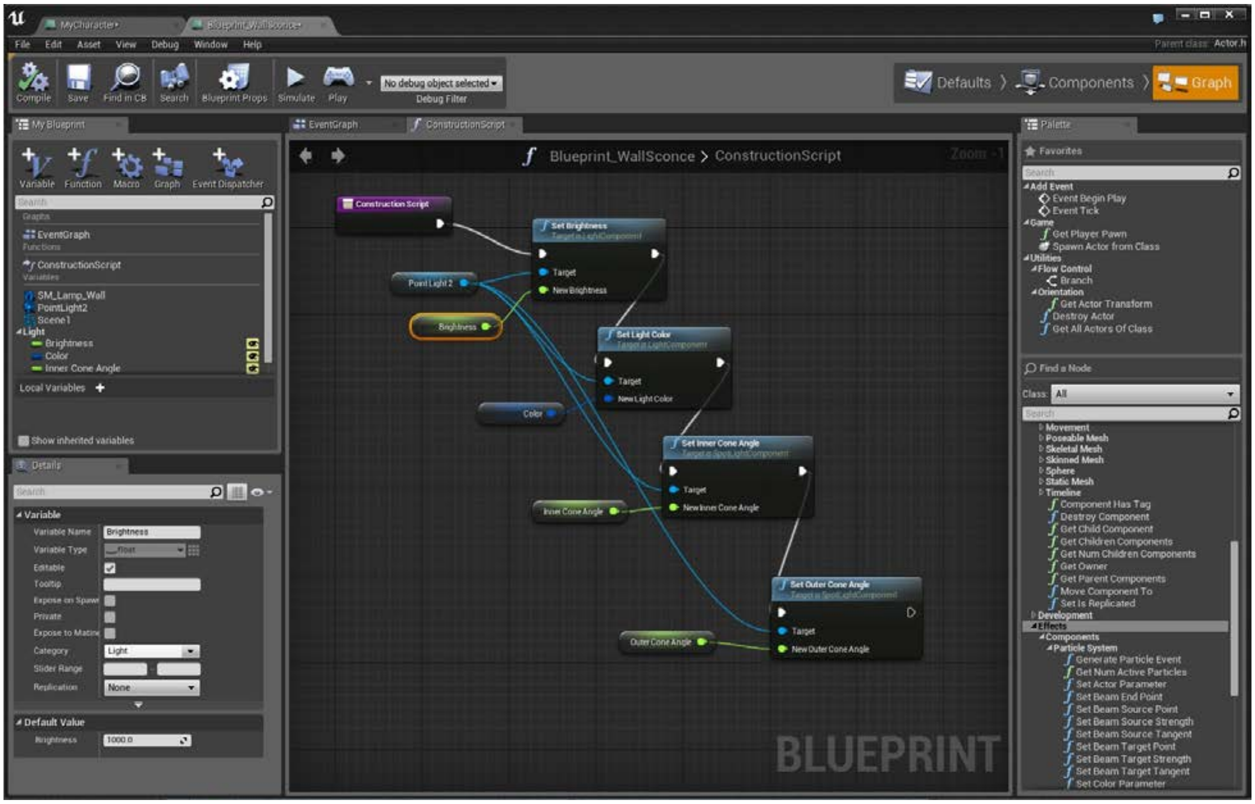


Fig 6: UE4 Kismet Editor

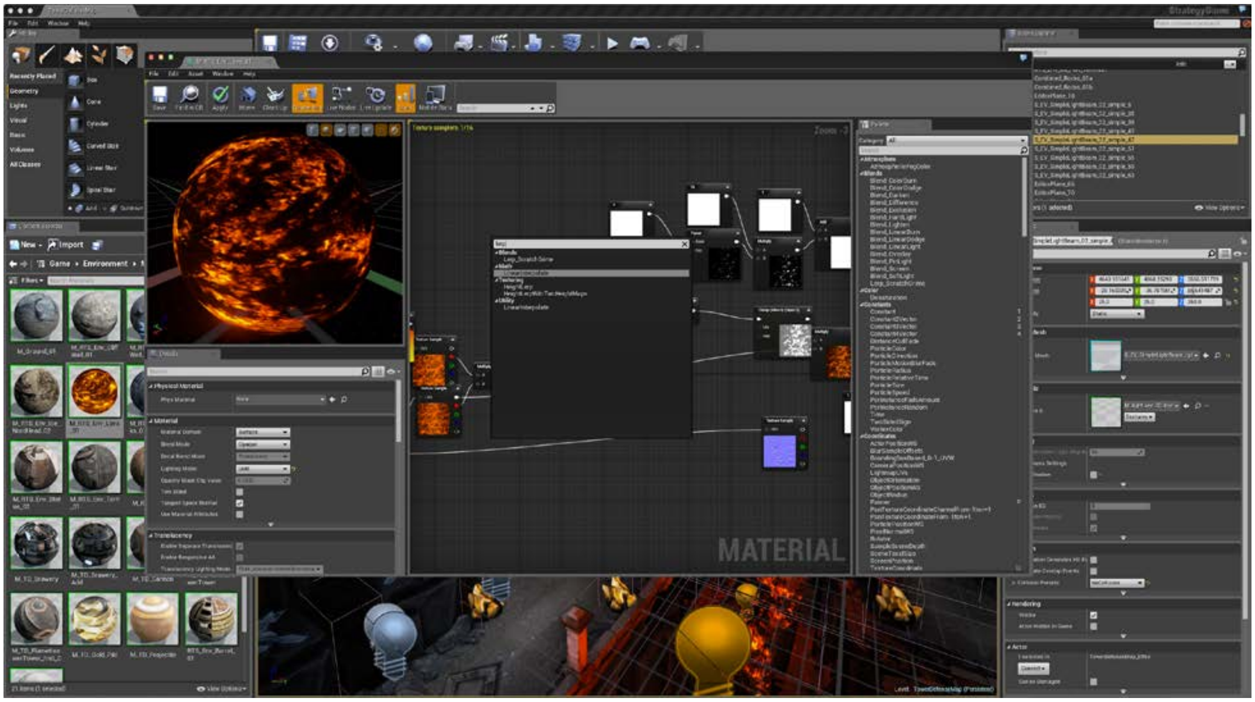
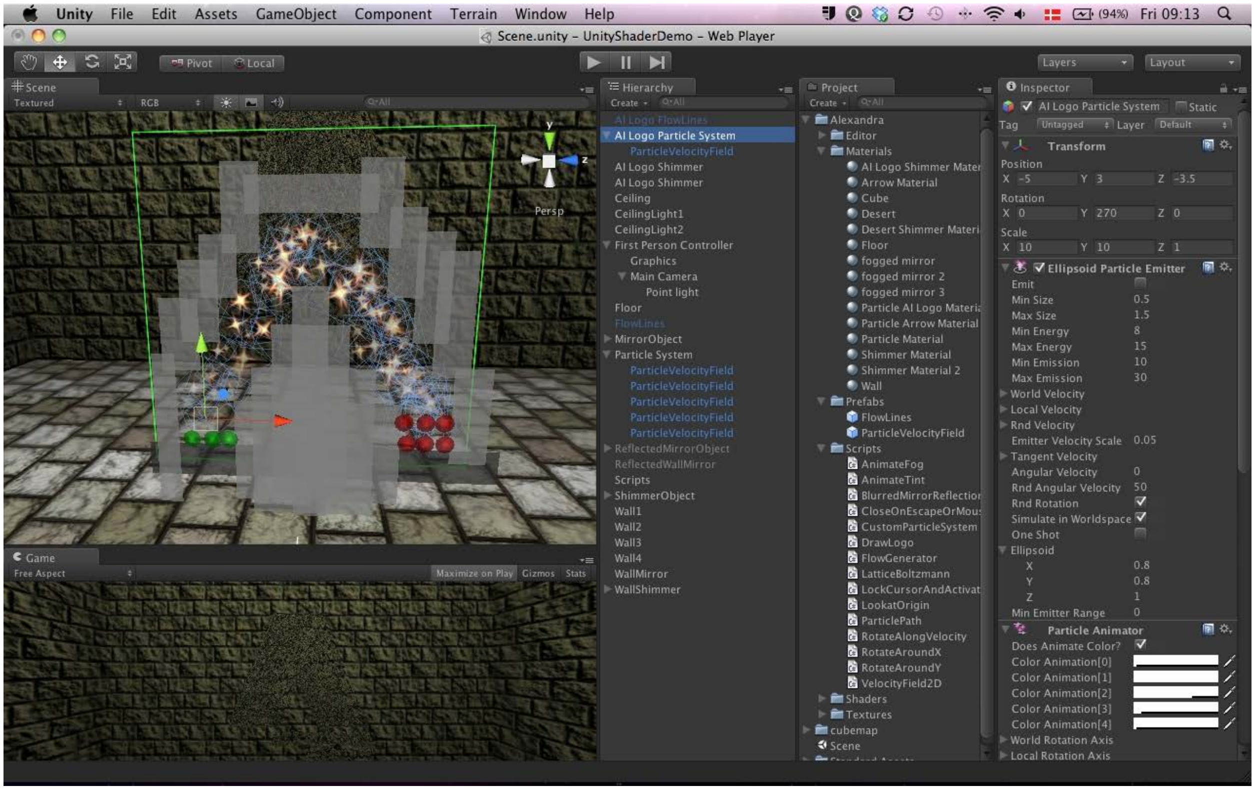


Fig 7: Example of UE4 Complexity

Another very popular engine is the *Unity* engine, shown in Figure 8: Unity Editor. One major benefit of *Unity* is its balance of simplicity and power. It has a fairly simple UI that can be learned within a single sitting, but also a component system that allows for more complex behaviors. This component system is useful because it helps to ensure new users are not intimidated by a lot of menus and terms that they may not know. The major downfall of *Unity* however is its lack of flexibility. *Unity's* core editor is missing a lot of the power that an engine like *Unreal* has, but this issue can be mitigated by using extensions or add-ons.



☑ AI Logo Particle System □ Static
Tag Untagged Layer Default

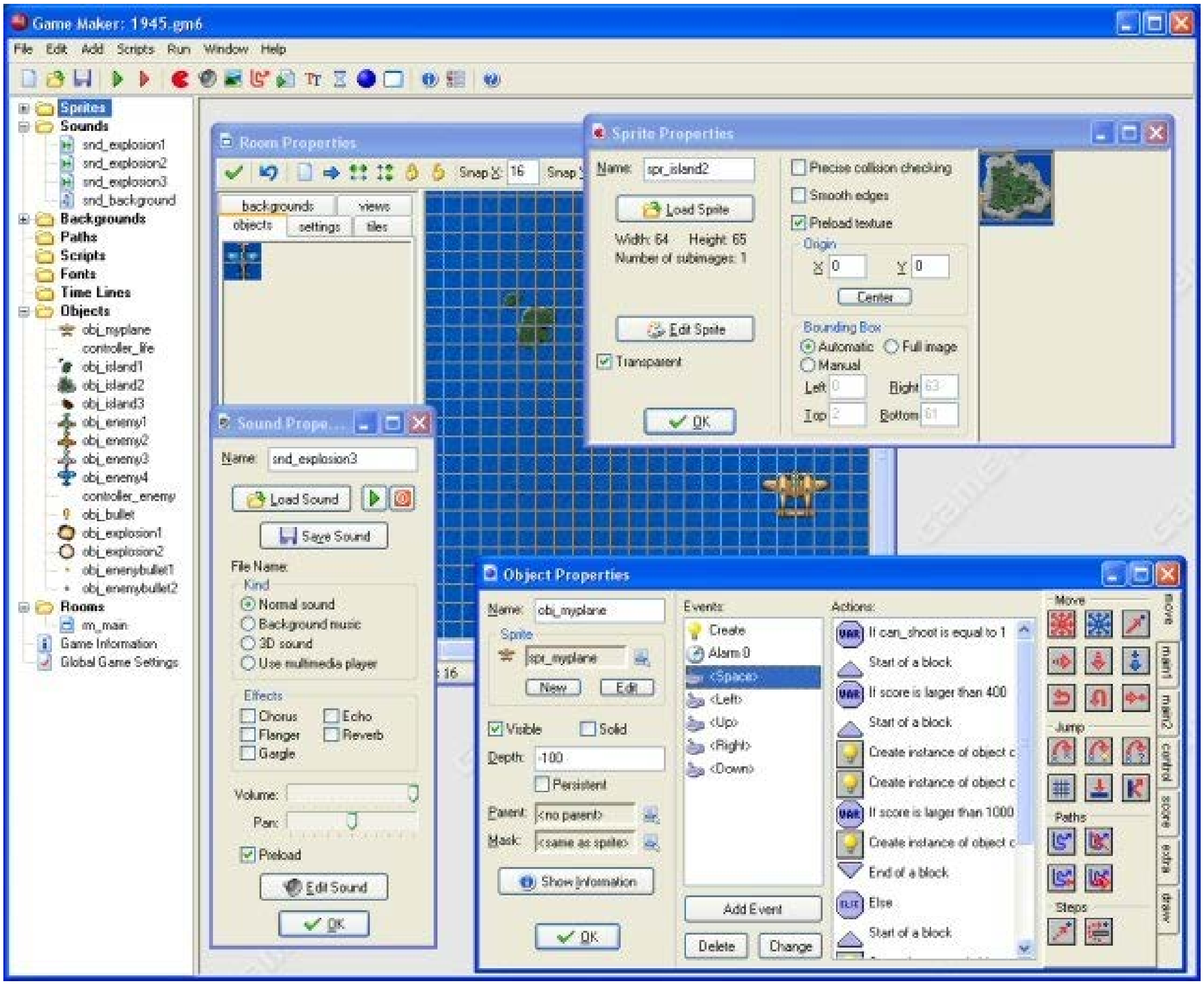
Transform
Position X -5 Y 3 Z -3.5
Rotation X 0 Y 270 Z 0
Scale X 10 Y 10 Z 1

☑ Ellipsoid Particle Emitter
Emit □
Min Size 0.5
Max Size 1.5
Min Energy 8
Max Energy 15
Min Emission 10
Max Emission 30
World Velocity
Local Velocity
Rnd Velocity
Emitter Velocity Scale 0.05
Tangent Velocity
Angular Velocity 0
Rnd Angular Velocity 50
Rnd Rotation ☑
Simulate in Worldspace ☑
One Shot □
Ellipsoid
X 0.8
Y 0.8
Z 1
Min Emitter Range 0

Particle Animator
Does Animate Color? ☑
Color Animation[0]
Color Animation[1]
Color Animation[2]
Color Animation[3]
Color Animation[4]
World Rotation Axis
Local Rotation Axis

Fig 8: Unity Editor

One last popular engine we researched is *Game Maker Studio*. *Game Maker* is a small, largely 2d sprite based engine. It has been used for several large game projects, but is mostly used as a beginner engine for many game designers. One of the reasons it is popular as a beginner engine is its streamlined interface. The interface is designed to guide the user through the process of starting at content and going to game code without the need for tutorials. Because of this a beginner can easily pick up on what they are supposed to do. Even the engine's scripting has been streamlined in this way. To do this it uses a list of blocks as commands to write scripts for objects, which can be seen Figure 9: Example of Game Maker's Visual Scripting. Additionally, the engine's menus are ordered in such a way that simple commands are listed first. This makes it likely a new user will familiarize themselves with the simple commands before moving on to the more complex ones. This streamlining does have its downsides though. It makes it hard to deviate from the basics, thus limiting the creator's flexibility, and because of this creating complex games in this engine can be a challenge.



Sprite Properties

Name: spr_island2

Width: 64 Height: 65
Number of subimages: 1

☑ Transparent

☐ Precise collision checking
☐ Smooth edges
☑ Preload texture

Origin
X 0 Y 0

Bounding Box
● Automatic ○ Full image
○ Manual
Left 0 Right 63
Top 2 Bottom 61

Sound Properties

Name: snd_explosion3

File Name:

Kind
● Normal sound
○ Background music
○ 3D sound
○ Use multimedia player

Effects
☐ Chorus ☐ Echo
☐ Flanger ☐ Reverb
☐ Gargle

Volume:
Pan:
☑ Preload

Object Properties

Name: obj_myplane

Sprite
spr_myplane

☑ Visible ☐ Solid
Depth: -100
☐ Persistent
Parent: <no parent>
Mask: <same as sprite>

Events:
- Create
- Alarm 0
- <Space>
- <Left>
- <Up>
- <Right>
- <Down>

Actions:
- If can_shoot is equal to 1
- Start of a block
- If score is larger than 400
- Start of a block
- Create instance of object c
- Create instance of object c
- If score is larger than 1000
- Create instance of object c
- End of a block
- Else
- Start of a block

Fig 9: Example of Game Maker's Visual Scripting

One thing we can conclude from looking at these different engines is that a game-making tool must have a balance of complexity and flexibility. The more complex the tool, the more likely it will be able to provide additional flexibility and freedom. The downside of this is alienating users unfamiliar with the tool. Thus in order to create a good tool a balance must be found.

2.3 Other Projects

In order to attain an increased understanding of previous attempts to integrate digital media in the classroom, we investigated two student projects. The first concerned the viability of educational videogames at teaching children with ASD about music. This study was relevant to determining how games could be entertaining to young children with disabilities as well as determining what properties exist in a good music game. The second study concerned teaching young children music composition and was similarly important for examining how to create effective educational games for children.   The study also provided information on the viability of different interaction mechanisms.

The first project, titled *Interactive Music Systems*, considers the use of digital media to provide a more engrossing educational experience (Manzo, Chupka, Burns, & LaRose, 2014). The research group investigated methods to help teach children with ASD about music, and determined the best way to go about doing so would be to develop a game. This game was tested in a school system and yielded promising results; students were enamored by the visual interface and interactive elements. From this it seems that a polished interface and interactive gameplay are crucial features for an educational game. Additionally, both the teacher and school administrator considered the approach viable for use in education. The project described in this paper is more general purpose than the one described here, but the group's research nevertheless provides a useful template.

In the second project, titled *Interactive Music Composition App for Children*, a group of students developed a game to teach young children early composition skills and cultivate their interest in music (Manzo, Goksaran, Yorulmaz, & Goksun, 2014). The game uses motion sensing controls to track users' hands and illustrates the potential of tactile feedback mechanisms as a means of interaction. This game was tested by a music professor from a public university, who posited tactile feedback mechanisms could be viable for educational use. The evaluating professor also found instructions crucial to understanding the application, which suggests the importance of providing users a guidance mechanism.

2.4 Conclusions from Background Research

From this background research we determined several guidelines for the creation of a successful educational game. First off, it's important in the creation of any tool to balance simplicity and power. Any tool that aims for power will compromise its simplicity, and vice versa. In the context of this project, the tool should be as powerful as possible while still being easy enough for the grade-school student to use.

Secondly, it's important for an application to run on a wide variety of hardware, or at least the most commonly used hardware. This maximizes the chances of students and schools already having the hardware needed to run the app. In this way an app can be made relatively inexpensive to integrate into school curriculums.

Thirdly, it's important for a tool to contain features that ease its integration into a teacher's lesson plan. These features might include tutorials, stat-tracking, or online integration. Including features like these is important as they can reduce the work-load required for the teacher to teach the game and monitor students. In this way teachers are more likely to be able to experiment with a game without taking significant amounts of time out of their pre-existing curriculums.

Finally, its important for a game-designer to incorporate both fun and education in their game design. If simply creating a fun game is prioritized with no thought to its educational value, then the game probably won't serve well as a teaching device. If the educational aspects of the game are prioritized with no regards to fun, then the game will likely struggle to maintain player interest and its purpose will be defeated entirely. Therefore, fun and education must be merged together considered in the game's design.

3 METHODOLOGY

The goal of this project was to identify whether forms of digital media could be viable for educational use. This viability was tested through the development and distribution of an educational game prototype to undergraduate university students. In order to develop this application prototype, the target audience, the platform, and the engine that would be used to build the application were determined. The application was then further developed by integrating key features as suggested by the background research. These features included a game, an assignment editor, and a stat tracker. After the development cycle was complete, a link to the application and a survey was distributed to university students. Eight of these students replied to the survey and their feedback was analyzed to determine application's viability as an educational resource and the importance of its various features.

3.1 Developing Around a Target Audience

At the beginning of development the target demographic was broad and consisted of students and teachers grades K-12 in the State of Connecticut. The game was designed to be generically appealing to any audience within this target demographic. Because of both the potentially broad age range of the student demographic and our development team's lack of artists, the first priority was building an intuitive interface with stat-tracking and an exercise editor that would be ideal for teacher use. The game would take second priority to the functionality that would be most useful to teachers, given they were the only part of the demographic that was known with certainty.

3.2 Target Platforms

The application was developed primarily for PC.   Other platforms considered included Mac OSX, Linux, Android, and IOS. Mac OSX and Linux were not a priority since Windows controls more than 90% of the desktop operating system market share (“Market Share Reports”, n.d.) as of February 2015. Android and IOS would have been a great stretch goal, but it was out of the scope of this project to undergo testing on these devices. In the end it was decided to develop for Windows since this would be the platform that most in-class computers would be running, thereby providing the easiest setup for teachers.

3.3 Engine Choice

This project was developed in the Unity Engine. This is for a number of reasons, outlined in greater detail in section 2.1.3 of the Background Research Section. Most influential to the decision was Unity’s unique ability to develop simple prototypes quickly. With its easy to use editor and relatively simple programming API, learning the engine and creating a simple prototype was a very achievable task within the small time frame allowed. Its simplicity also meant that the project could be expanded upon in the future with relative ease.

3.4 Application Development

3.4.1 The Login System

The program includes a login system as a way for the application to identify the user. This feature is necessary for the system to associate a student with their performance data. This login system is depicted in Figure 10: The Login Screen. The login system’s functionality is all handled on one screen. From this screen the user can enter a username and password and then create a profile or login. Once the user successfully creates a profile or logs in, they are brought to the application's main menu. If the user could not successfully login, bold red text displays an error message to the user indicating what went wrong.

Security was also a significant consideration when developing the login system. This is because if the data was not secure a student could just cheat the system and edit their student information manually. To solve this problem user information is encrypted using the DES standard of encryption. This process assures that students cannot manually adjust their student data without a comprehensive knowledge of this domain of cryptography.

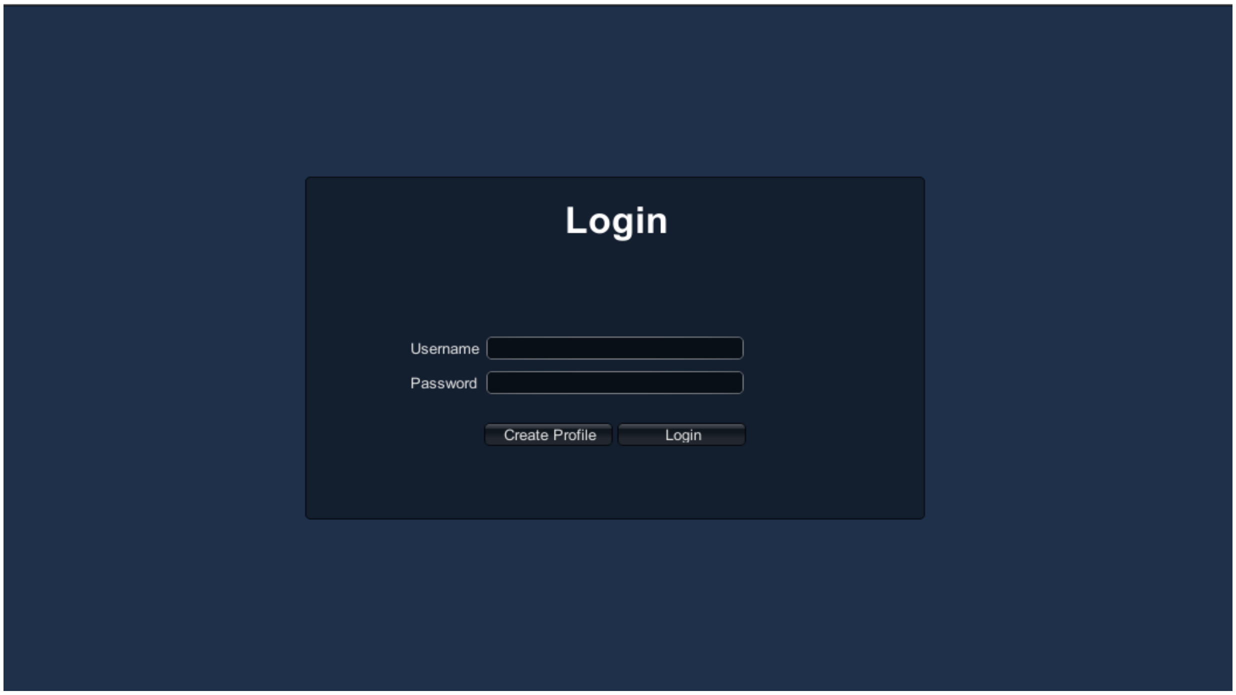


Fig 10: The Login System

3.4.2 Main Menus

The main menu system allows the user to navigate around the application's various features. It consists of two separate menus: one for the main application and one for the exercise editor. The menu system for the main application is illustrated in Figure 11: The Main Menu and consists of a list of three different buttons.   The first button brings the user to the game screen, the second to the stat-tracking screen, and the third to the achievements screen.   The menu system for the editor is similar, and contains options that allow users to load in new songs, modify exercises around existing songs, cancel back to the other menu system, and quit the application. This menu is depicted in Figure 12: The Editor Menu. Between these two menus, the user is able to find most of the application’s features.

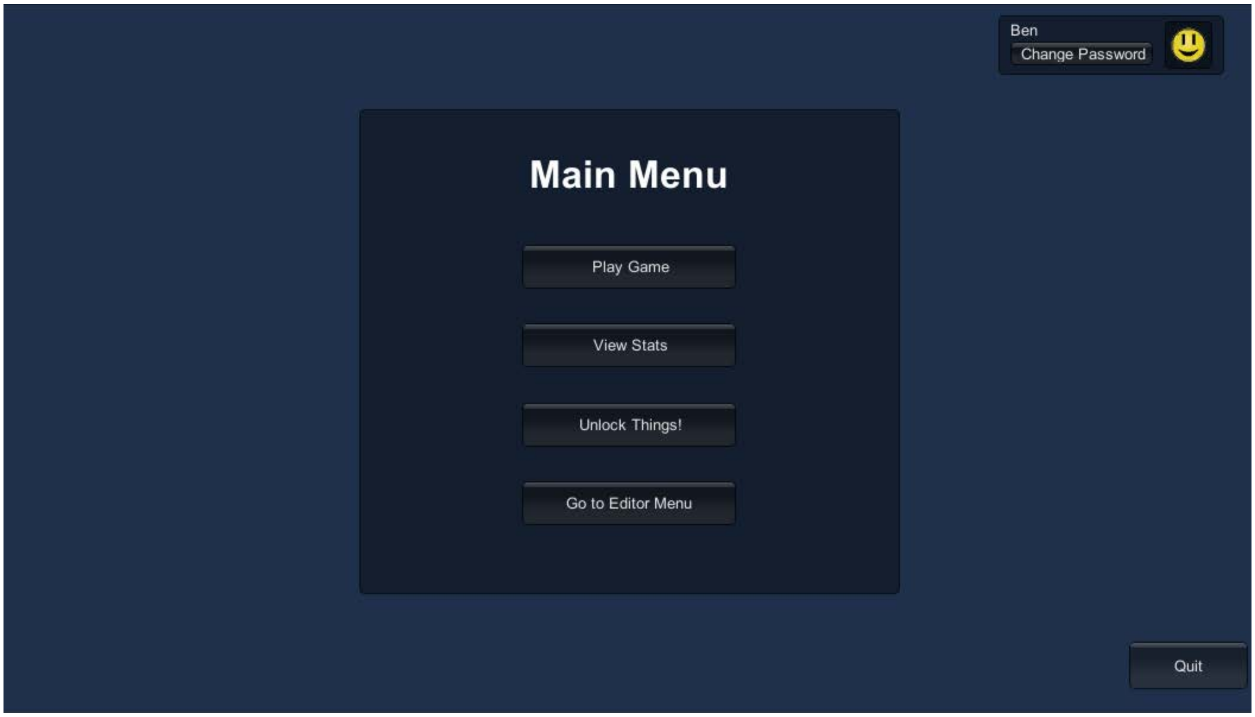


Fig 11: The Main Menu

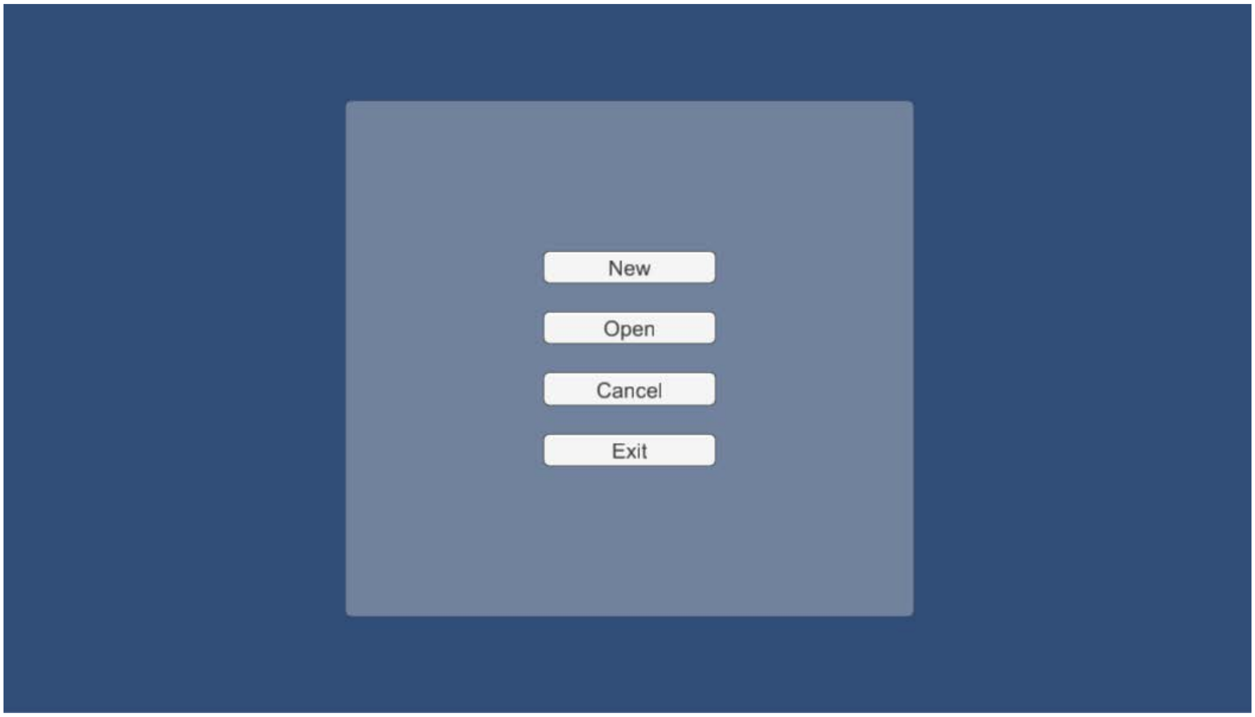


Fig 12: The Editor Menu

3.4.3 Song Registration

One thing that was crucial to the development of this application was making it so that teachers had the ability to modify the application with their own custom content. In this way the application could both better cater to a teacher's existing curriculum while improving it reusability. This was implemented by allowing teachers to create their own exercises by using song files on their computer.

In order for the user to play songs and edit exercises, the songs must be registered within the application. The user begins this process by selecting the **new** option from the editor menu. This brings them to the dialogue selector shown in Figure 13: The Dialogue Selector. This selector displays all unregistered songs contained in the **Song** folder that is distributed along with the executable. Choosing a song from this dialogue selector brings the user to the screen depicted in Figure 14: The BPM Screen. From here, the selected song plays and the user should tap to the beat of the BPM with the spacebar. This tapping is used to set the song's BPM data for its info file. Once the BPM has been determined, the song is registered and an **info.song** file is created to store the song's attributes and exercise data. Now the song can be seen in a separate dialogue selector accessed by clicking on the **open** button. Opening a song from this selector will load the song's data and transport the user to the exercise editor. They are now all set to create an exercise.

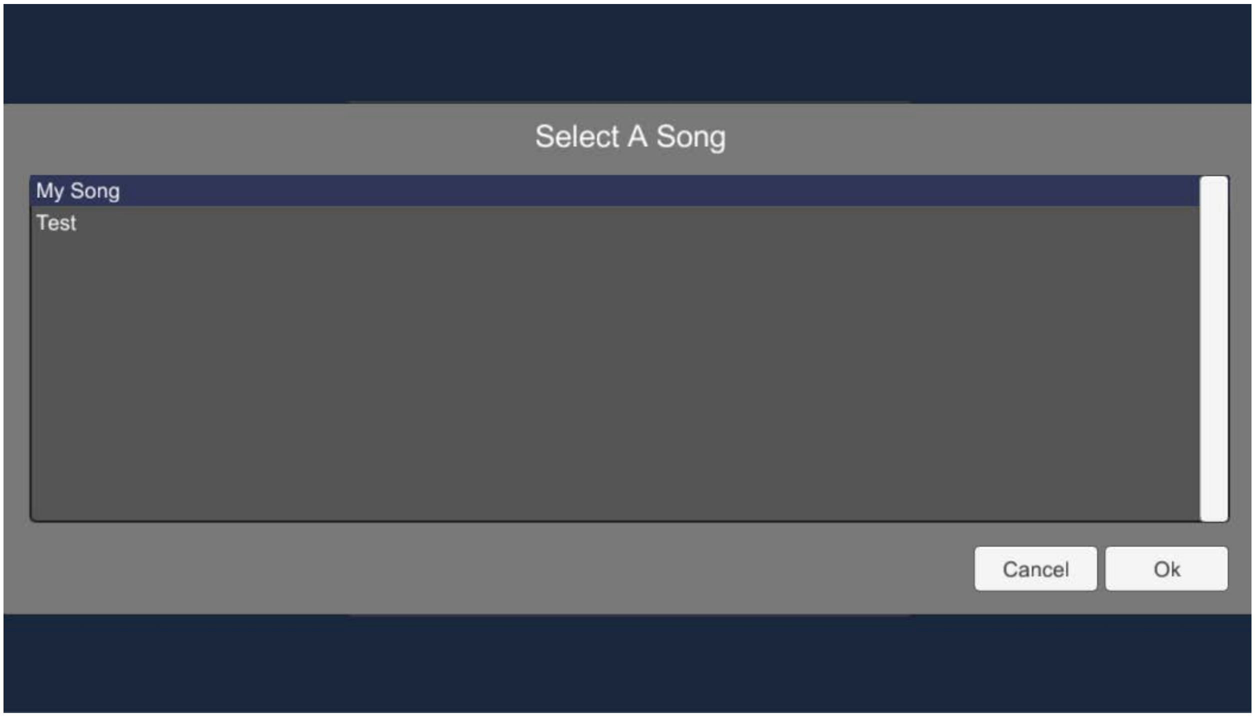


Fig 13: The Dialogue Selector

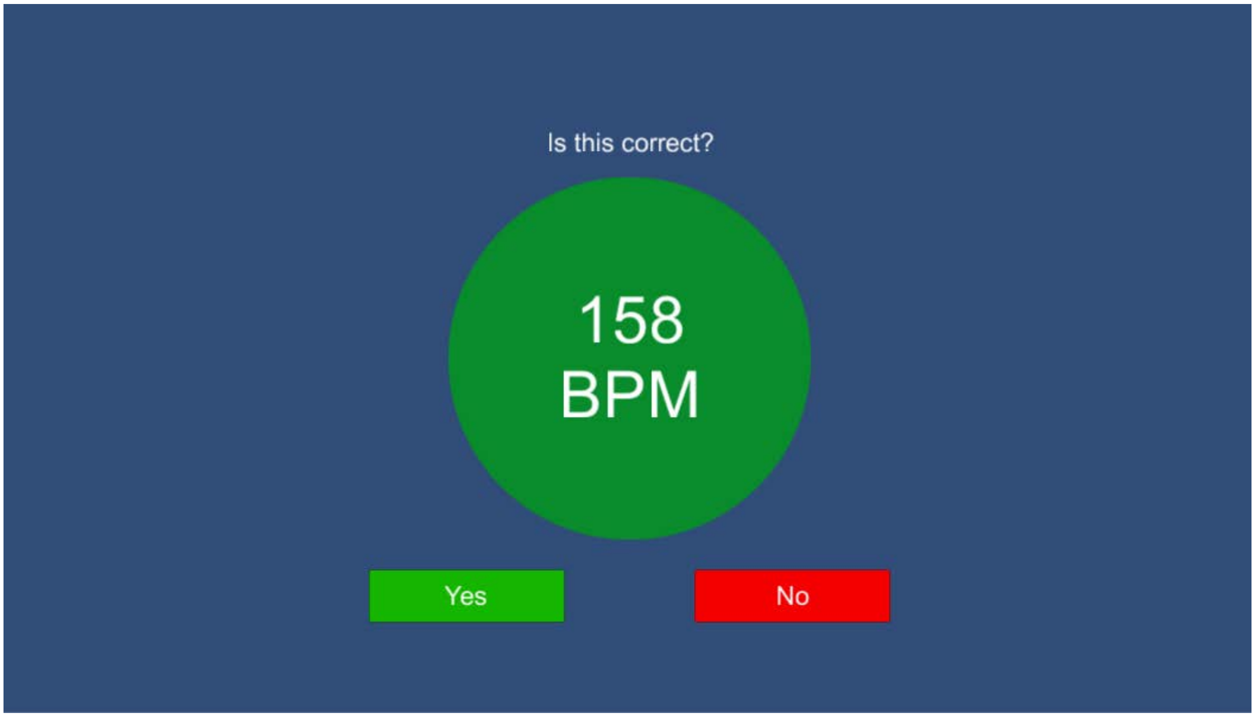


Fig 14: BPM Screen

3.4.4 The Exercise Editor

From the background research it was clear that the application needed to be easy for teachers to use while still providing a way to be molded into a teacher's curriculum. This is where the exercise editor (depicted in Figure 15: The Exercise Editor) comes in. The exercise editor allows teachers to modify the game in a familiar direct manipulation format.

From the Exercise Editor the user may select pre-made notes from the top row, and then drop them into specific slots in the editor. These slots correspond to where the notes will appear in the game as the song plays. The user may move about the song by either pressing the **arrow buttons**, or using the **scroll bar** at the bottom of the screen. They may then continue scrolling about the song and placing notes in this way until satisfied with the end result. If a user does not have the time to write the entirety of these exercises, they may use a **generate** feature that takes the current notes in the editor and loops them until the end of the song. When the user is done with creating their exercise, they can test it by pressing the **play** button. This will take the user to the game with the exercise loaded in. They can save their progress from the editor by pressing the **save** button, or they may choose to reset the exercise by pressing the **clear** button. The user can exit the Exercise Editor at any time by clicking the **close** button. This functionality provides everything a user needs to create a song's exercise.

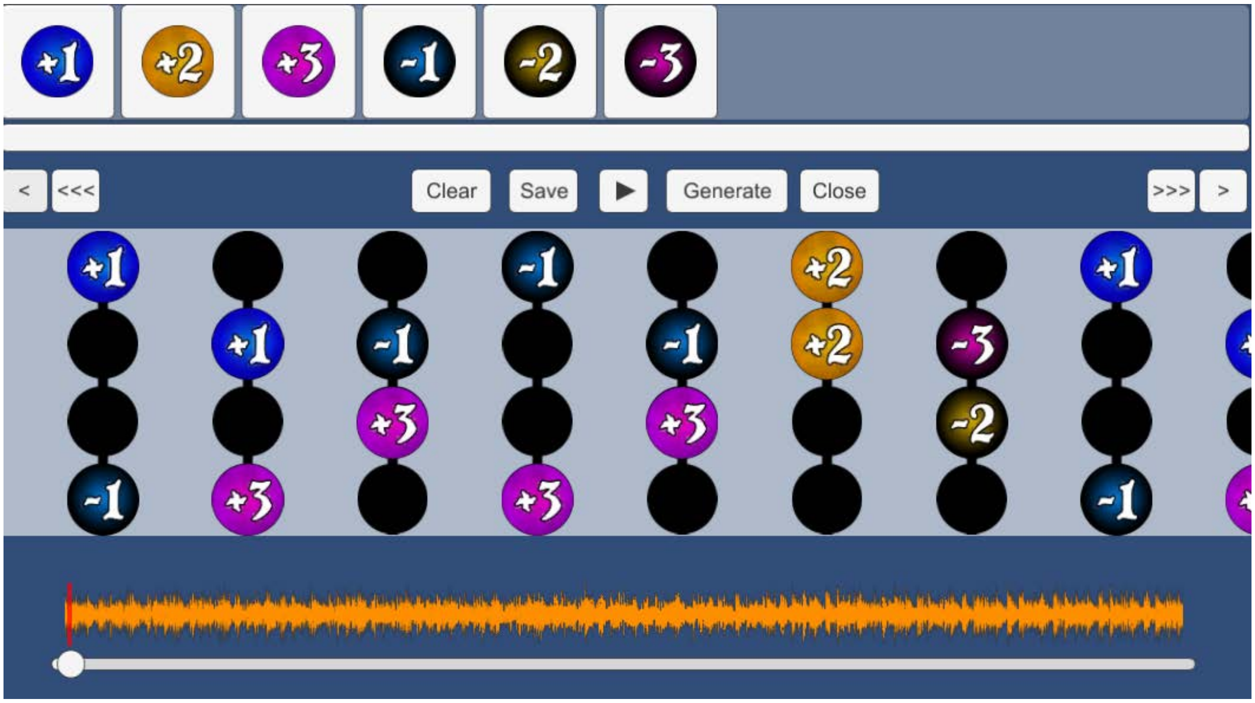


Fig 15: The Exercise Editor

3.4.5 The Game

The unifying aspect of the application is the game itself (depicted in Figure 16: The Game). It is designed as Guitar Hero-like simulation with the intention of improving a student's ability to solve basic math problems, sight-read, and make quick decisions. To do this it takes some of the basic principles of math and music and integrates them into the gameplay. By merging these basic principles with the gameplay, the background research suggests the game will be more interesting and immersive. For the original design document of this project's game, see Appendix A.

The game starts as the song the user previously selected begins playing. As this song plays, notes scroll by the track on the bottom of the screen. The player must hit these notes as they cross the red line to score points. The player may also score points by defeating enemies on the screen. In a similar manner to Counting Kingdom, the player must add together note values to achieve a sum that is equal to the number displayed on the monster. This will cause the monster to take damage and reset the sum. When the monster is defeated, the game spawns a new monster and the player receives a bonus sum of points. Monsters will occasionally launch attacks on the player that can damage them. The player may defend from these attacks by hitting special shield notes. Failing to do so will cause the player to take damage that subtracts from their health bar. The player must continue playing this way until either the monster is defeated or their health bar hits zero. If the player survives until the end of the song, it is considered successfully completed and the player's stats are updated. If the player's health reaches zero the game ends with a game over screen. From here the player will have to retry the exercise all over again. This process of completing exercises forms the core of the application and is what students would spend most of their time working on.

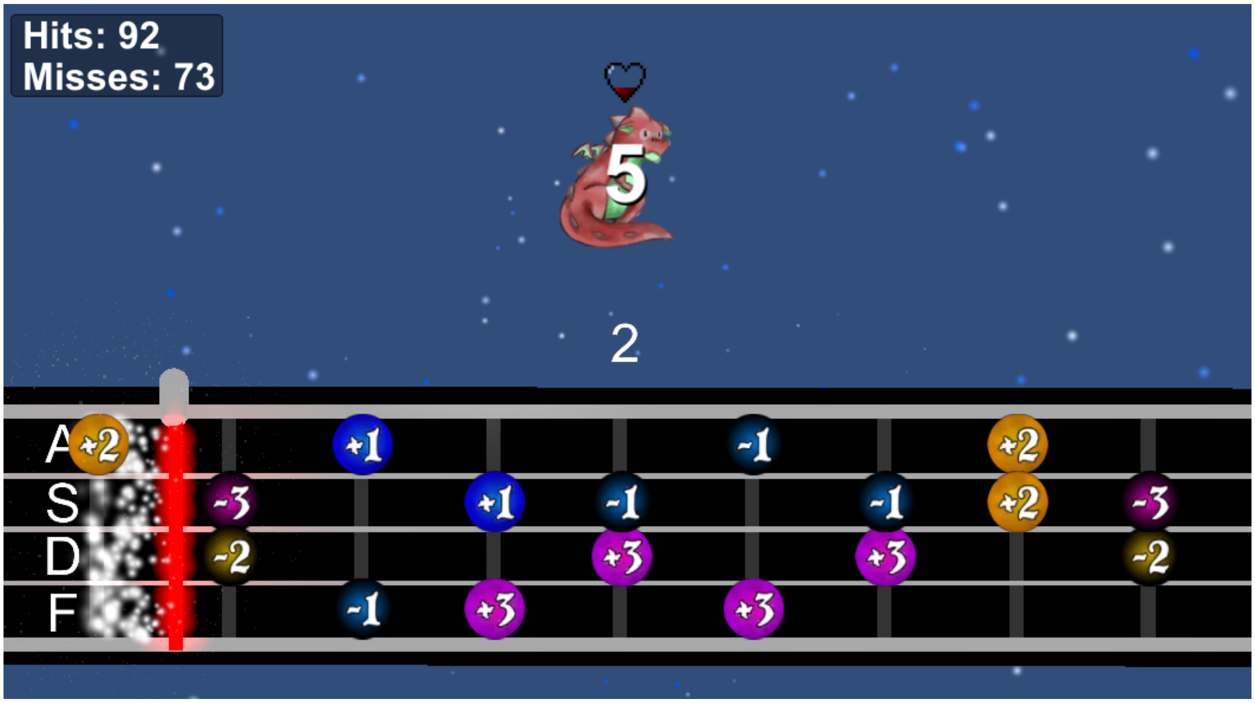
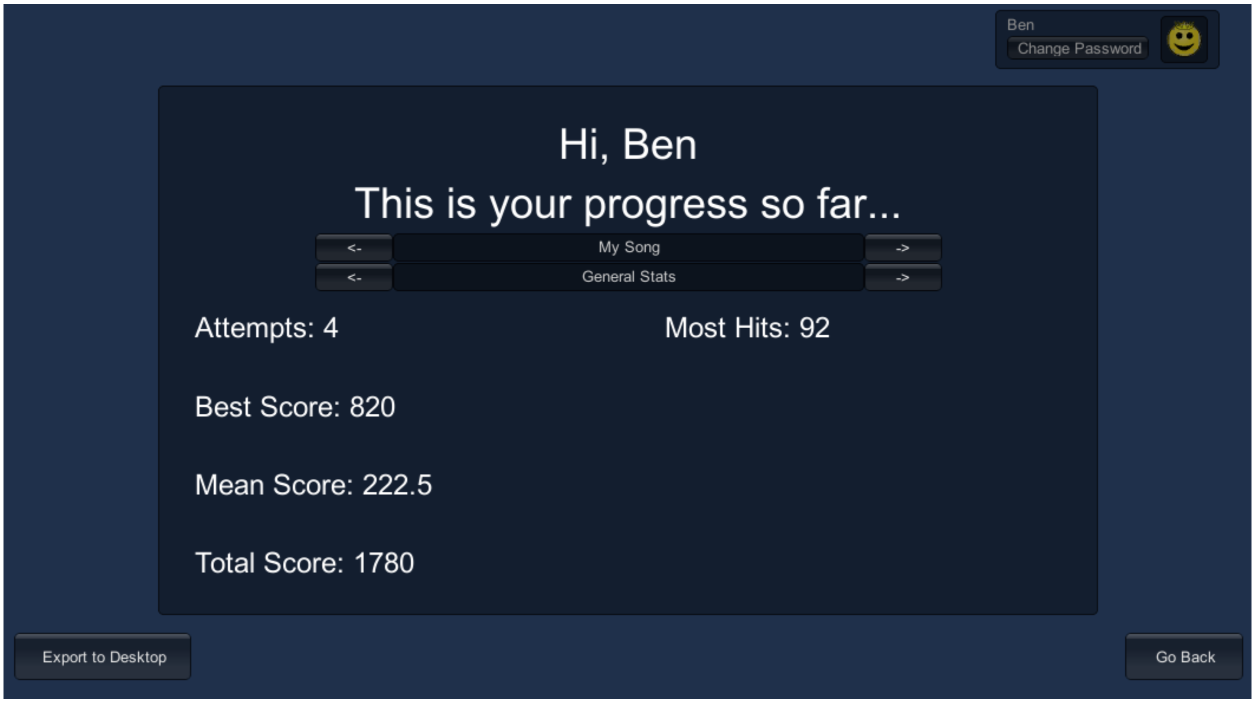


Fig 16: The Game

3.4.6 The Stats Screen

The necessity of stat-tracking is described in the e-Adventure project: "Integration of games with standards-compliant learning management systems implies… the inclusion of standard metadata to facilitate their discovery" (Moreno-Ger, et al., 2008). The user can view the application's stat-tracking by clicking on the **View Stats** option in the main menu. This will bring the user to a stats screen, where they can view all the stats the system records from their in-game performance. The stats are partitioned on a song-by-song basis, since songs can vary in terms of length and difficulty. They are also divided between two separate screens: a **General Stats Screen** and a **Graph Screen**. The General Stats Screen, displayed in Figure 17: A Song's General Stats Screen, documents all the descriptive statistics pertaining to a student's performance on a song. Some examples of these descriptive stats include a student's mean, best, and total scores. The Graph Screen displays a student's score over time in graph format. This screen is shown in Figure 18: A Song's Score / Time Graph, and is necessary to gauge a student’s performance improvement. Between these two screens, the application contains enough stats to maintain a general idea of a student’s progress.



Hi, Ben

This is your progress so far...

Attempts: 4

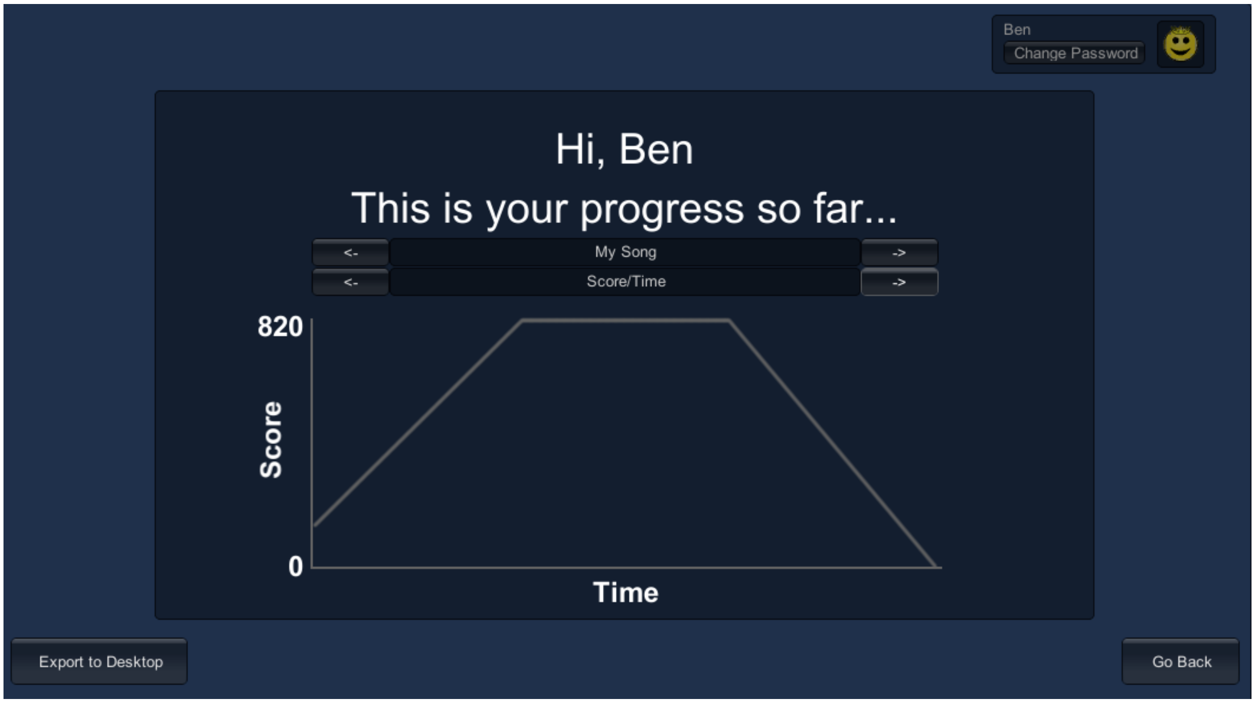
Most Hits: 92

Best Score: 820

Mean Score: 222.5

Total Score: 1780

Fig 17: A Song’s General Stats Screen



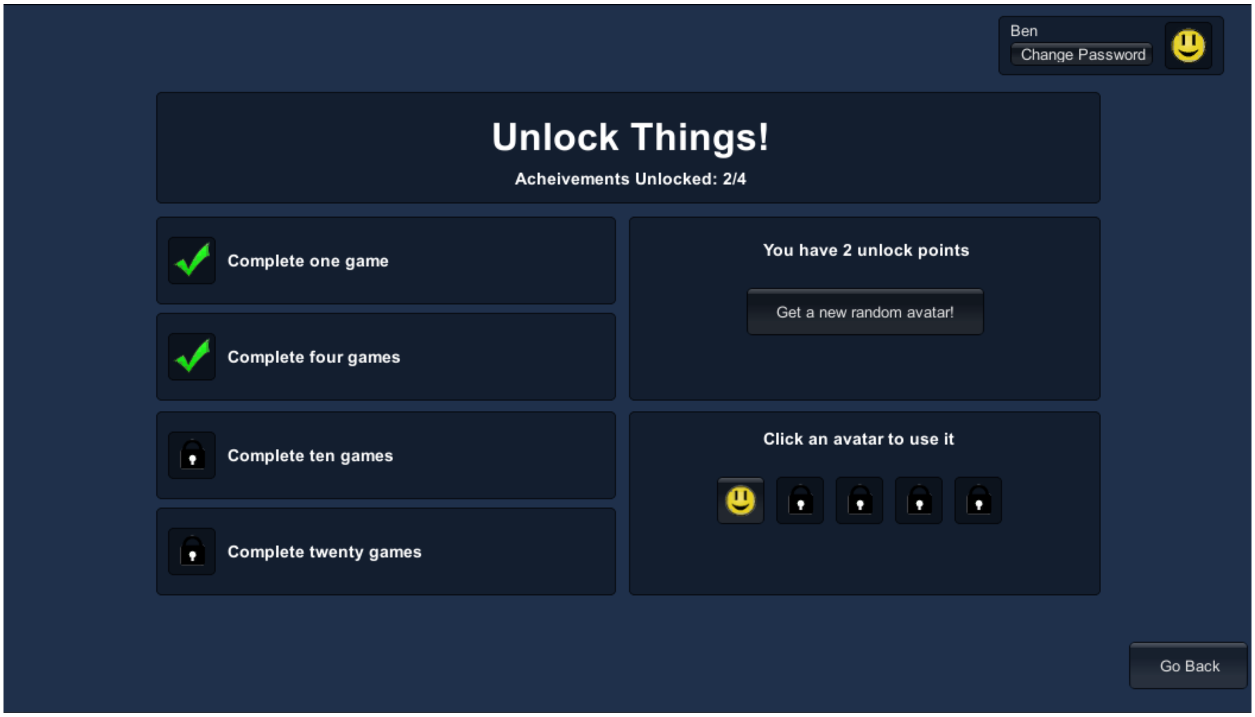
Hi, Ben

This is your progress so far...

Fig 18: A Song’s Score/Time Graph

3.4.7 The Achievement System

The achievement system provides motivation to keep students coming back. It is shown in Figure 19: The Achievement Screen, and awards unlock points to users as they complete more games. These unlock points can then be used to unlock more avatars that can be displayed in a user's profile stub. The unlocks are designed around economic knowledge which suggests users respond positively to incentives (Benabou & Tirole, 2003). They are intended to act as extrinsic motivation for the user to continue playing the game. Here a distinction should be made between extrinsic and intrinsic motivation: Whereas extrinsic motivation is increased by use of external stimulus (e.g. grades or peer competition), intrinsic motivation is increased by bolstering one's desire to do something. In contrast to the economical perspective, psychological research suggests (Benabou & Tirole, 2003) that extrinsic motivation may sometimes be counter-productive to learning objectives. The same research suggests that intrinsic motivation may return better results. This means unlocks that more directly influence the gameplay, like in-game consumables or abilities, might serve as better motivators. This project could not integrate such unlocks into the system for scope reasons. All the same, the system at least provides a means of extrinsic motivation to students.



Unlock Things!

Acheivements Unlocked: 2/4

✓ Complete one game

✓ Complete four games

🔒 Complete ten games

🔒 Complete twenty games

You have 2 unlock points

Click an avatar to use it

Figure 19: The Achievement Screen

3.5 Experimental Procedure

As part of an initial pilot test, we sought to obtain feedback from teachers and other practitioners in the field. After identifying potential participants, a link to the previously described application was distributed to school teachers in the State of Connecticut via email. In addition to this link, a twenty minute tutorial was provided as well as a google survey. After a week of open testing, the survey only got a single response.

A follow-up test was prepared to provide more complete data. In this second test, undergraduates from WPI University were solicited in person and by email. The research participants were told to test the game and did so without external guidance in order to simulate a self-directed learning environment. They were then told to fill out a new survey, which can be viewed in Appendix C. This revised survey included a mix of quantitative and qualitative questions that were designed to both assess the application and attain opinions on the importance of certain features in educational software. The twenty minute long instructional video used in the previous test was replaced with a brief in-game tutorial. While the demo was open reminder emails were sent out on a weekly basis to those who volunteered. After three weeks with the demo open, the survey was closed and the results were collated. There were a total of eight responses.

4 DATA AND ANALYSIS

4.1 Summary

After development finished a vital part of our project was seeing objectively how well the application did. The application and surveys were sent out to a variety of WPI students, eight of which responded. The survey asked students questions concerning the effectiveness of the application as a teaching device and what features they thought were important in an application like this. The results of the survey (with abridged questions in each title) are graphed out below. To view the survey in full, including the unabridged questions, see Appendix C. The survey's qualitative data can be viewed in Appendix D.

4.2 Quantitative Data

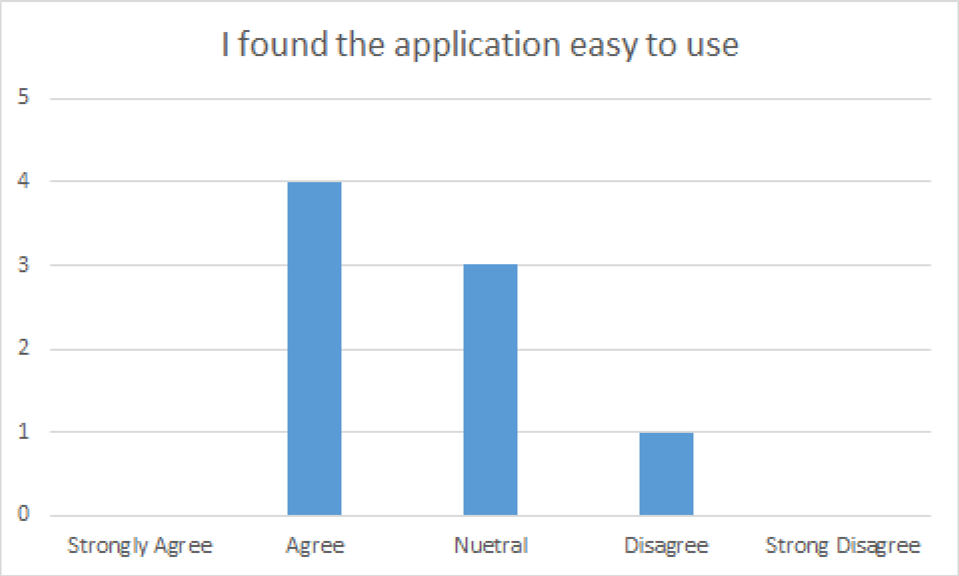


Figure 20: Question 1 Graph

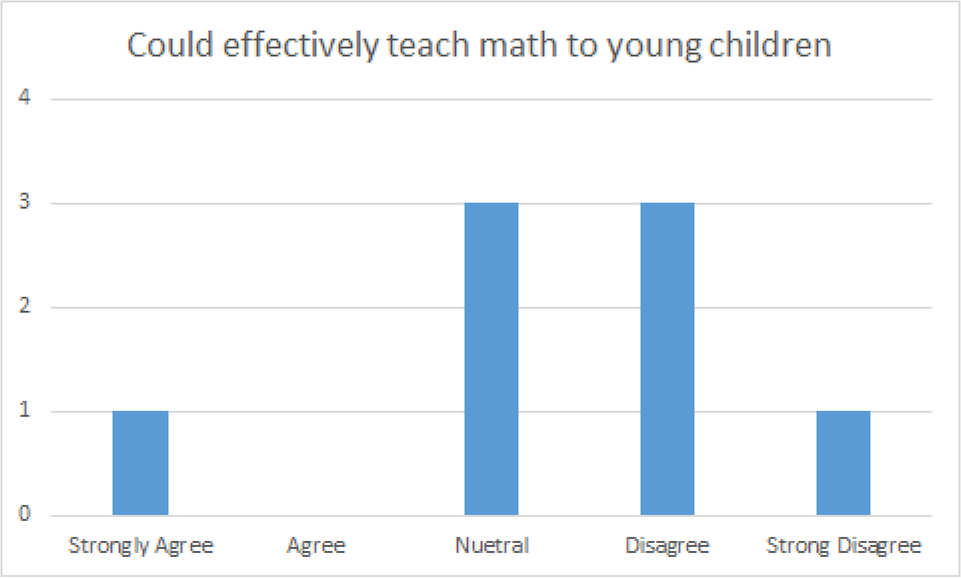


Figure 21: Question 2 Graph

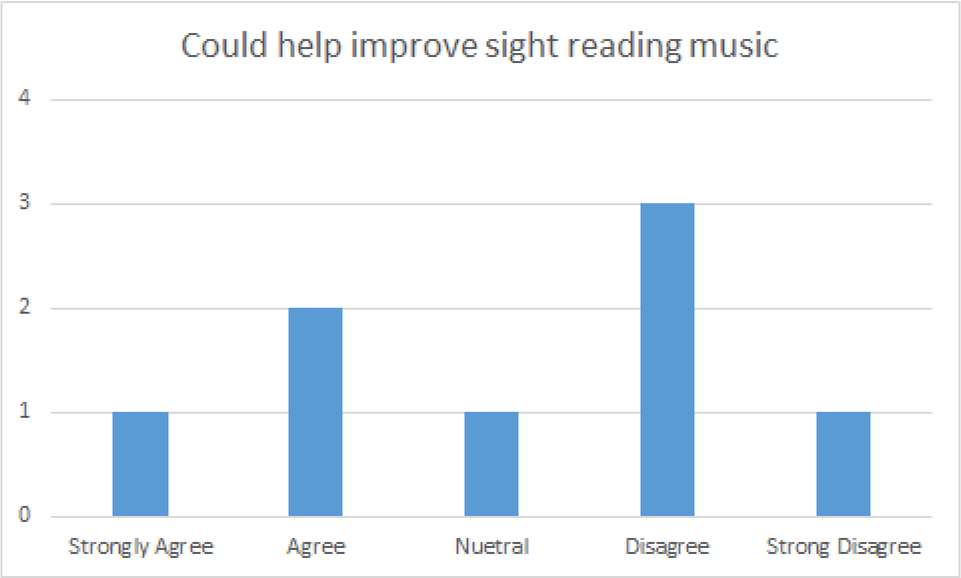


Figure 22: Question 3 Graph

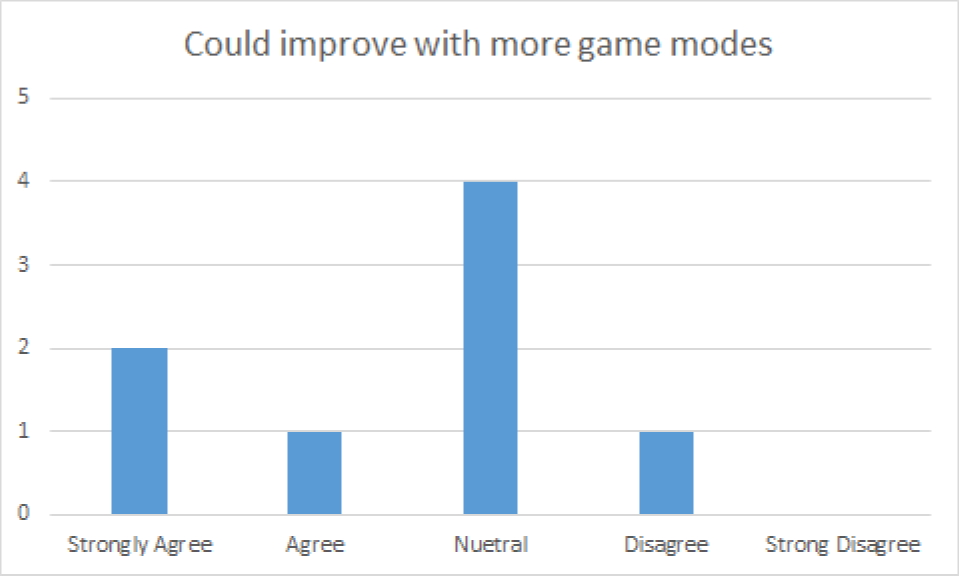


Figure 23: Question 4 Graph

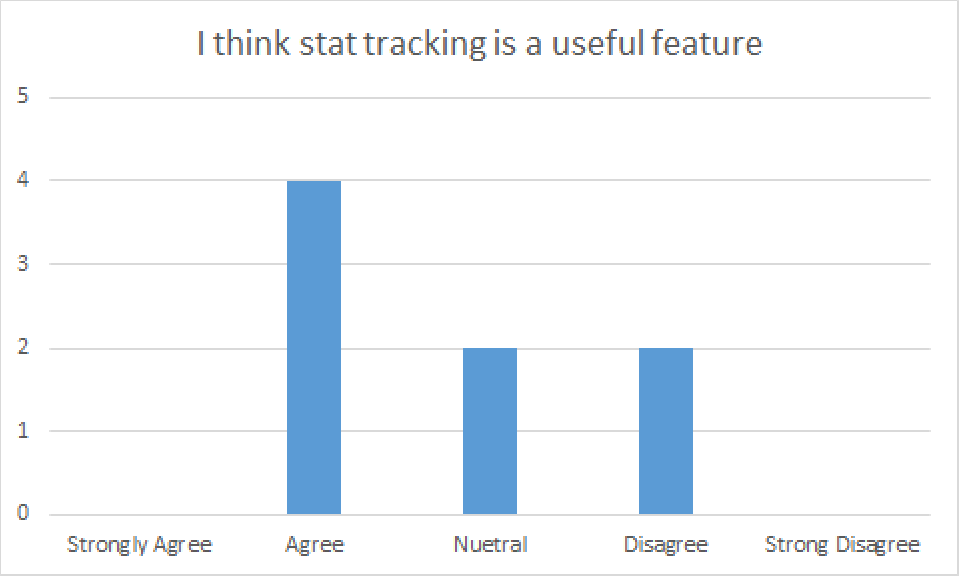


Figure 24: Question 5 Graph

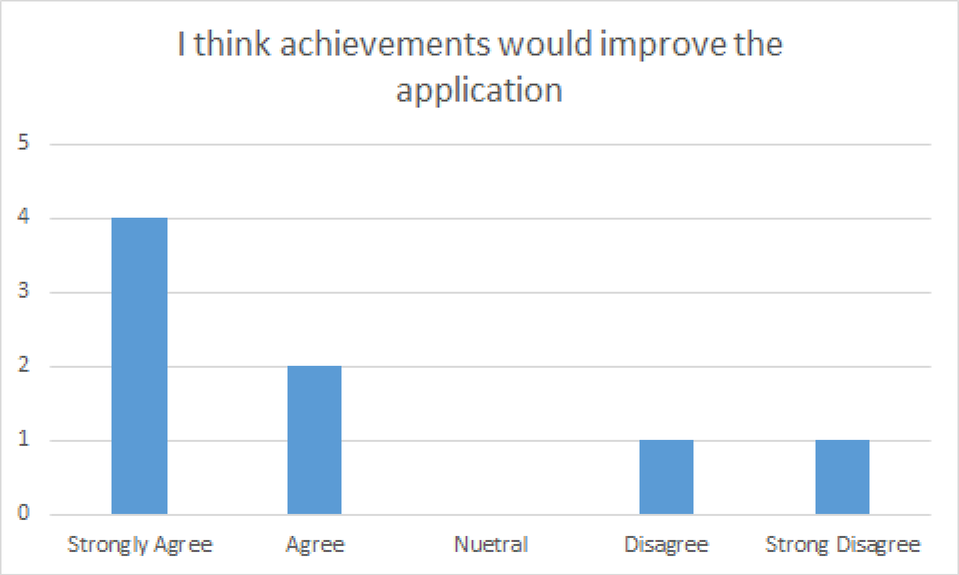


Figure 25: Question 6 Graph

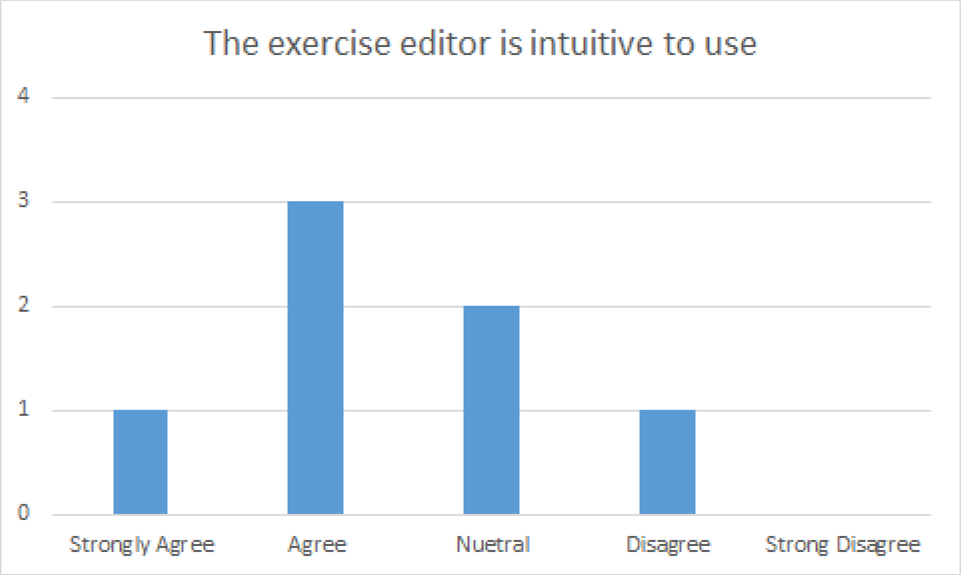


Figure 26: Question 7 Graph

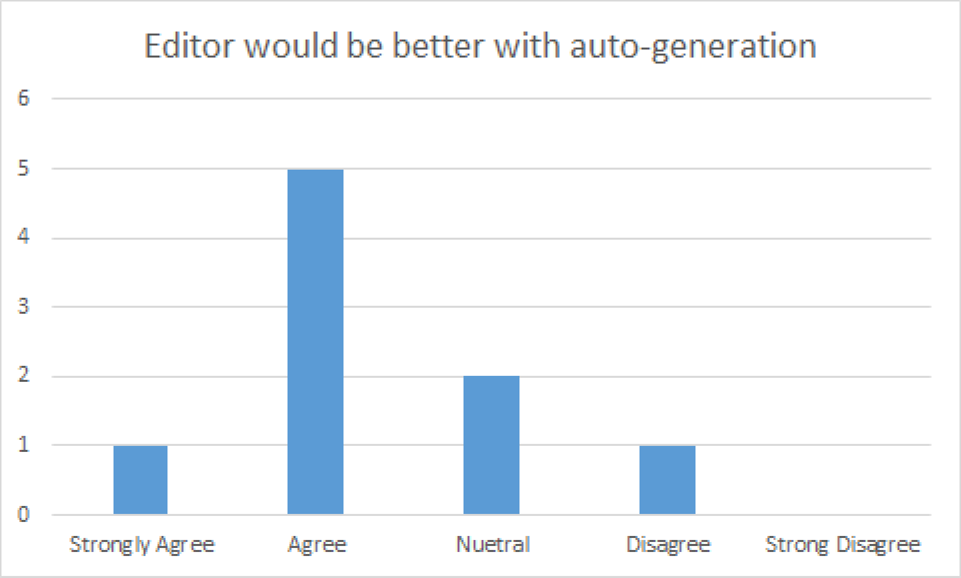


Figure 27: Question 8 Graph

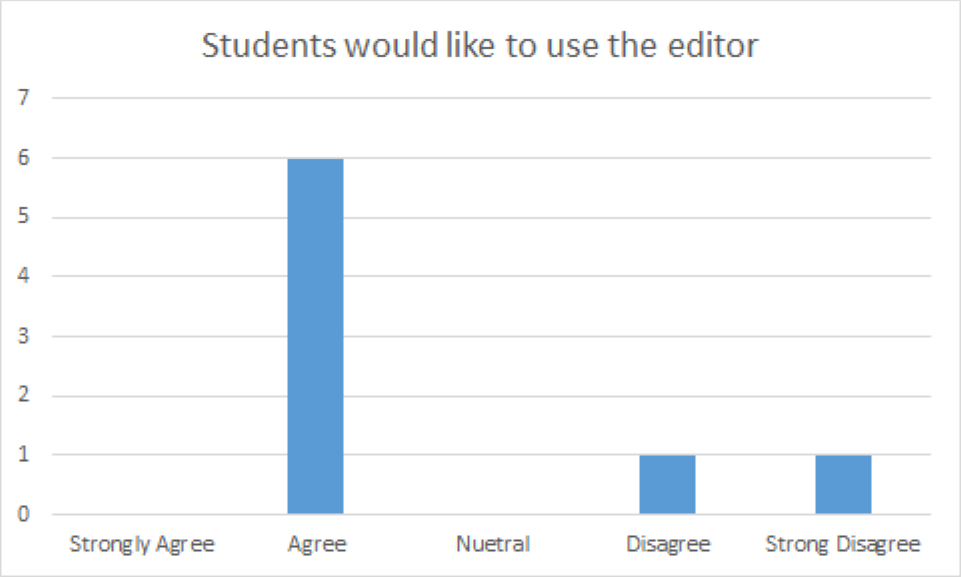


Figure 28: Question 9 Graph

5 CONCLUSION

5.1 Data Interpretation

There were a total of eight responses received during the length of student testing. Although this is nowhere near enough to make a statistically sound argument, it is still enough to identify several significant patterns. The analysis of these patterns reveal a potential path for the future of this project and any similar projects.

The first major point of discussion is extrapolated from looking at the first few questions. These questions focus mainly around the current state of the application and its ability to educate. The average score of these questions tended to lean around the neutral to disagree area. A possible cause of this is the fact that the game is currently in a very primitive stage. This is particularly supported by the free response section. Many testers brought up the fact that the game felt unfinished, was a good starting point, or just in general needed work. Likewise for many of the later questions that focused on the future progression of the project we got predominantly good responses. This suggests the application is on the right track, and that an expanded version or an application that follows a similar path could have very positive feedback. Another possible reason for the negative responses pertains to the project's background research. When looking at games such as *Counting Kingdom*, one of the major causes of success was its simple base mechanic with layers of complexity put on top. Being in a primitive state, our application did not have these layers of complexity but was rather a proof of concept for only the base mechanic. By expanding the game's base mechanics to include more layers of complexity, the research suggests the game could perform much better as an educational tool.

The survey also had several questions that focused around the expandability of the game. These questions got very positive responses, especially with the editor. In general it seemed that testers desired for the game to have more expandability. However when taking into account some of the other questions, many testers showed that they would prefer automation in a lot of areas, such as song generation and student score tracking. This would suggest that while testers like the idea of expanding the game as much as they can, they also want it to be both as easy to use as possible, and to have the option to skip large time sinks.

On a final note about the feedback, people’s majors along were collected along with their responses to see if there were any biases between fields. The majority of our responses ended up being IMGD majors, which could have added a bias to the results. Since these people understand and study video games, this could have very likely changed how the product was viewed as compared to a survey of teachers.

Given that the survey data collected was in no way statistically accurate enough to make any true claims, the amount of consistency the application’s design has with the background research would suggest that it has begun to follow the correct path. For applications of similar merit this too might act as a good starting point.

5.2 Future Development

Throughout this project we experienced a mix of successes and failures. The sum of these contributed to the final outcome of the project. By identifying what we did right and wrong we hope to provide a better guideline for any future teams working on this project or a project of similar merit.

Of the many factors that impacted the project’s outcome, there were several that had a very positive effect. One such factor was the use of a repository. By using a repository, code could easily be worked on separately and merged together when needed. It also provided the ability to revert to previous builds when the project broke. A second positive factor was the use of *Unity3d*. By using *Unity3d* the prototype was able to be developed quickly and with general ease. The flexibility that *Unity3d* provided us was also very useful since it allowed us to rapidly implement features. One final factor that played a positive role was the use of weekly meetings. This helped to keep the project on track and maintain a clear goal for the project at all times, even when things got hectic.

There were also some factors that had negative effects on the project as a whole. These are examples of things we would go back and change if possible. As mentioned previously the use of a repository was a very strong point of our project. However, we did not have a very structured approach to using the repository before the project began. Having a clear design of assigned tasks for commits and merging would not only have helped organize the repository, but also have helped organize tasks and collaboration between people. It would have also been beneficial to have a slightly larger team on the project. This would have hopefully allowed people to constantly be working on all features that needed to be implemented. If we had a dedicated artist it also would have invariably made the application look better. A final major change we would have made to the project structure was an increase in development time. The ratio of research to development felt skewed throughout the project. An increase in development time at the expense of research would most likely have allowed a much more polished and functional product.

In terms of future development, we plan to put this project on a GIT repo so that interested parties can access it later. Refining the game by adding more robust mechanics and better art is one possible way the project could be expanded. The project could also be improved by adding additional features. These might include networked file transfer or the ability to interface with more games and traditional exams. Any of these changes would benefit the project and help create a more complete application.

REFERENCES

1. A History of Video Games : Innovation Timeline | EduTube Educational Videos. (n.d.). Retrieved May 10, 2015, from http://edutube.org/en/history-video-games-innovation-timeline

2. Benabou, R., & Tirole, J. (2003). Intrinsic And Extrinsic Motivation. Review of Economic Studies, 70(3), 489-520. doi:10.1111/1467-937X.00253

3. Educator Tools - Finale. (n.d.). Retrieved April 28, 2015, from http://www.finalemusic.com/products/finale/features/educator-tools/

4. Gros, B. (2007) Digital Games in Education, Journal of Research on Technology in Education, 40(1), 23-38, doi: 10.1080/15391523.2007.10782494

5. Kirriemuir, J. (2002). The relevance of video games and gaming consoles to the higher and further education learning experience. *Techwatch report TSW*, *2*.

6. Manzo, V., Chupka, Z., Burns, B., & LaRose, D. (2014). *INTERACTIVE MUSIC SYSTEMS* (Interactive Qualifying Project, Worcester Polytechnic Institute).

7. Manzo, V., Goksaran, O., Yorulmaz, K., & Goksun, D. (2014). *Interactive Music Composition App for Children* (Interactive Qualifying Project, WORCESTER POLYTECHNIC INSTITUTE).

8. Market Share Reports. (n.d.). Retrieved April 28, 2015, from http://www.netmarketshare.com/

9. Moreno-Ger, P., Burgos, D., Martínez-Ortiz, I., Sierra, J., & Fernández-Manjón, B. (2008). Educational game design for online education. Computers in Human Behavior, 24(6), 2530-2540. doi:10.1016/j.chb.2008.03.012

10. Online Learning Exchange® Interactive Music powered by Silver Burdett™ with Alfred. (n.d.). Retrieved April 28, 2015, from http://www.pearsonschool.com/index.cfm?locator=PS1dIe

11. Rosas, R., Nussbaum, M., Rodriguez, P., Salinas, M., Cumsille, P., Marianov, V., Correa, M., Flores, P., Grau, V., Lagos, F., Lopez, X., & Lopez, V. (2003). Beyond Nintendo: Design And Assessment Of Educational Video Games For First And Second Grade Students. Computers & Education, 40(1), 71-94. doi:10.1016/S0360-1315(02)00099-4

12. Shaffer, D. W., Halverson, R., Squire, K. R., & Gee, J. P. (2005). Video Games and the Future of Learning. WCER Working Paper No. 2005-4. *Wisconsin Center for Education Research (NJ1).*

APPENDICES

A. Game Design Document

Beat Battle

A turned based rhythm game

Target Audience

This game is mostly intended for elementary and middle school students.

Pitch

Beat Battle is a turned based rhythm game that will help children better understand math and sight reading in music in a fun way. The game works similarly to *Rock Band* gameplay mixed with a *Final Fantasy* theme.

Gameplay

The game will mostly focus on the combat section sense that is where the educational content will be. Additional portions of the game will vary depending on the gameplay mode which are specified below.

Combat

Combat is similar to a *Guitar Hero* style gameplay. When combat starts a song will start playing. The player has a vertical bar where notes come down to the beat of the song. The difference however is that each one of these notes has a symbol on it. Notes can have one of two symbol types:

Operation:

An operation symbol when pressed on a beat will be applied to a counter that the player has. So for example if the operation is “+2” then two is added to the counter.

Special:

Special symbols will have a special effect on the combat in some way. For example the block symbol will block an enemy's attack

In order for the player to deal damage, the player uses the operation notes to increase his counter (or possibly decrease). When an enemy's health becomes a multiple of the players counter then the player deals that much damage and the counter is returned to zero.

Game Modes

While combat is the same through all game modes (mostly) variances in how the combat is applied changes depending on the game mode.

Challenge Mode:

In challenge mode the player will play through a series of songs with increasing difficulty until losing where a final score will be given to them.

Dungeon Mode:

The player will go through a series of pre-defined battles until reaching the end of the dungeon.

Teacher Tools

In addition to the game, there will be additional teacher tools that allow teachers to modify and create activities for their students. These tools are listed below.

Task Creation:

Teachers can create tasks by giving students a “task” file. This file contains two major pieces of information. FIrst it specifies the game mode. This specifies what type of game mode the task should be played as. The second key part of information is a collection of battles which will be used by the game mode. Tasks will generate a result file when a student finishes them which can then be given back to the teacher so they can see how they did.

Song Creation:

Teachers will be able to input their own songs as well as use pre-existing songs that come with the game. The editor will provide an easy way for teachers to pick a song file and then either randomly generate notes, or edit the notes themselves.

Ranking:

The teachers version of the game will contain the collective classes data, which can then be displayed in various ways to show ranking such as by day, task, or overall.

Art:

Art assets will be gotten mostly from open source resources and from the Unity Asset Store to save time

B: Email to Solicit Students

Hello everyone,
We are an IQP team investigating the influence of digital media on education. We have built a prototype of an application designed to teach young students basic math, sight-reading, and decision making skills. The application also comes with features our research suggests would be important to teachers, such as stat tracking and an exercise editor. We would like to know your opinions on how important or effective certain aspects of this application are. To do so we are asking you to try out our application and then fill out a brief survey on your opinions. This survey is entirely voluntary and we assure you that you're results will be kept anonymous. The links to the application and survey are listed below. Please contact us if you have any questions.

Application Link (Working Demo):
http://users.wpi.edu/~bpeake/Games/BeatBattle/BeatBattle.zip

Survey Link:
https://docs.google.com/forms/d/1kyiZEvZ050LOi_fzDYVOP0RiSbw-ZTOXXREKqXTQTm4/viewform

Thanks for your time,
Ben Korza, bnkorza@wpi.edu
Benny Peake, bpeake@wpi.edu

C: Student Survey Questions

On the next page is a copy of the student survey that we sent out to WPI undergraduates.

Application Survey

Before taking this survey, please test our application and be prepared to provide feedback.

If you have any questions, feel free to email us at bnkorza@wpi.edu or bpeake@wpi.edu.

Please fill out the information.

1. **What is your major?**

Please select the statements that most accurately reflect your opinions.

2. **I found the application easy to use**
Mark only one oval.

○ Strongly Agree
○ Agree
○ Nuetral
○ Disagree
○ Strongly Disagree

3. **The game presented could be an effective way to teach young students basic math and decision making skills**
Mark only one oval.

○ Strongly Agree
○ Agree
○ Nuetral
○ Disagree
○ Strongly Disagree

4. **The game presented could help improve a student's ability to sight-read music**
Mark only one oval.

○ Strongly Agree
○ Agree
○ Nuetral
○ Disagree
○ Strongly Disagree

5. **This application could be improved if it was expanded to include more games and formats**

More formats could include traditional test and survey formats.
Mark only one oval.

○ Strongly Agree
○ Agree
○ Nuetral
○ Disagree
○ Strongly Disagree

6. **I think stat tracking is useful in an application like this**

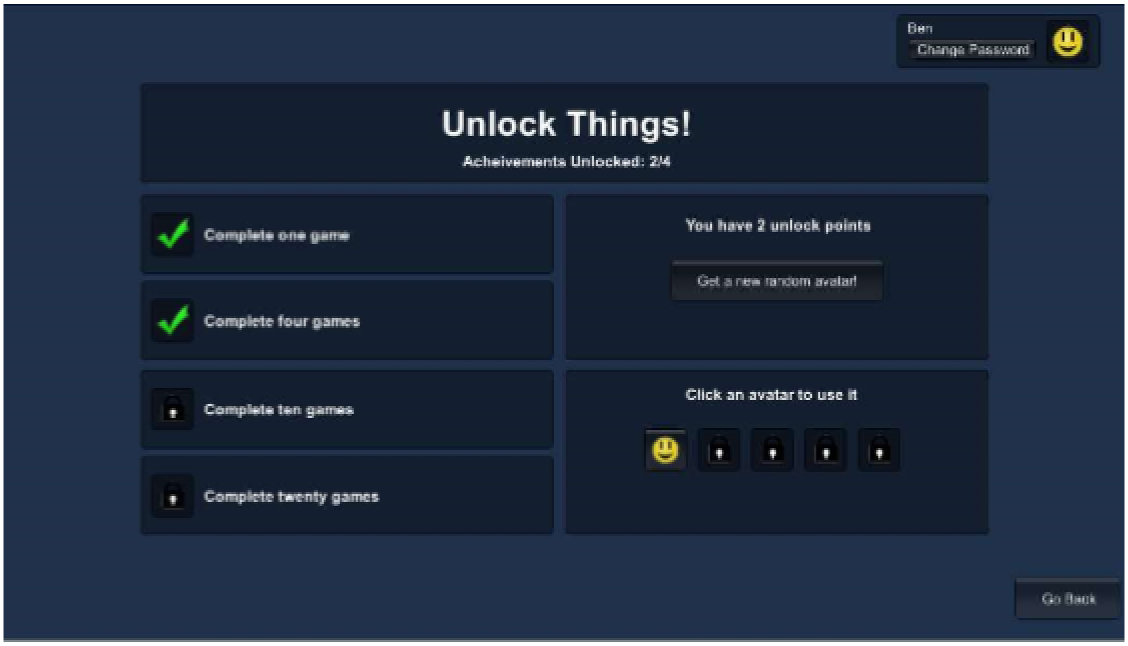
Mark only one oval.

○ Strongly Agree
○ Agree
○ Nuetral
○ Disagree
○ Strongly Disagree

7. **I think an achievement system (see below) is useful in an application like this**

Mark only one oval.

○ Strongly Agree
○ Agree
○ Nuetral
○ Disagree
○ Strongly Disagree



Ben

Unlock Things!

Acheivements Unlocked: 2/4

✓ Complete one game

✓ Complete four games

Complete ten games

Complete twenty games

You have 2 unlock points

Click an avatar to use it

8. **The exercise editor (see below) is intuitive to use**

Mark only one oval.

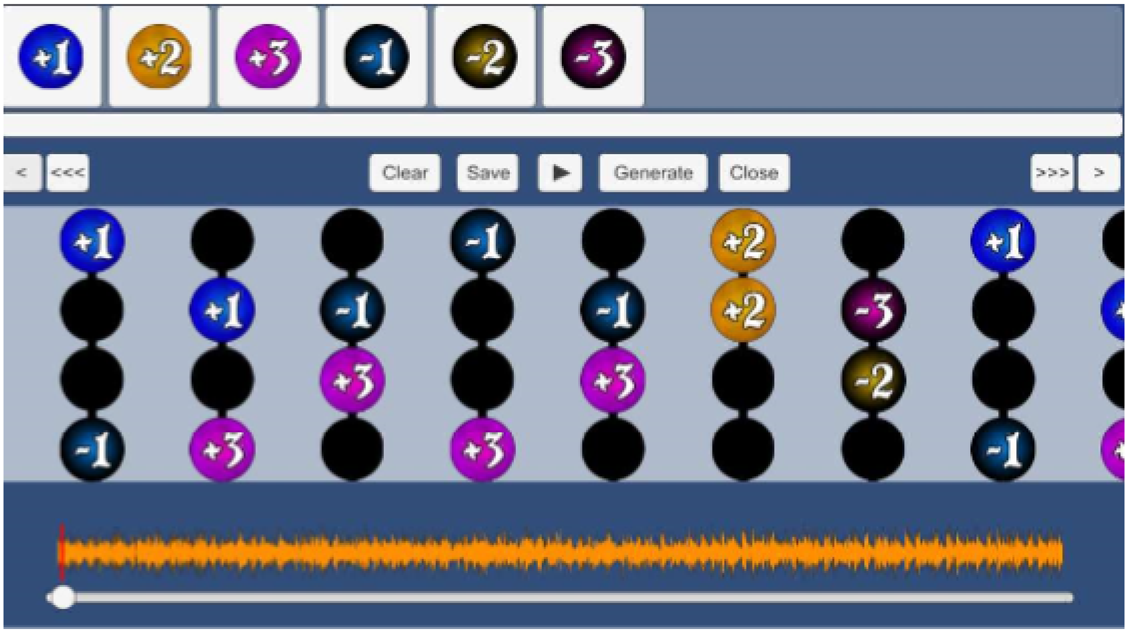
○ Strongly Agree

○ Agree

○ Nuetral

○ Disagree

○ Strongly Disagree



9. **The exercise editor would be more practical if it could auto-generate tracks**

Mark only one oval.

○ Strongly Agree

○ Agree

○ Nuetral

○ Disagree

○ Strongly Disagree

10. **I think students would like it if the exercise editor was accessible to them**

Mark only one oval.

○ Strongly Agree

○ Agree

○ Nuetral

○ Disagree

○ Strongly Disagree

11. **What do you feel is the strongest aspect of the application?**

12. **What do you feel is the weakest aspect of the application and how do you think it could be improved?**

13. **Do you have any other feedback on the application?**

D: Qualitative Survey Responses

Q1: What do you feel is the strongest aspect of the application?

Respondent 1: N/A

Respondent 2: The quality of graphics and the easiness to use it sets it apart from other applications.

Respondent 3: nice gui, not too complex for children

Respondent 4: It's a good start

Respondent 5: The UI is clear-cut and intuitive to use. Also, having an editor extends the utility of any game indefinitely.

Respondent 6: Not really too much.

There isn't an onscreen instruction for attacking enemies, and games like Guitar Hero, DDR, and Rock Band already have already done this system successfully.

Respondent 7: ...

Respondent 8: UI and gameplay is good.

Q2: What do you feel is the weakest aspect of the application?

Respondent 1: N/A

Respondent 2: The log in menu throws me off as it does not really fit the structure of the applications. The log in menu gives more of a sense of a mature game where I find the actual application is very young and easy to use.

Respondent 3: rules were not clear, how do i defeat the dragon? +3 is the best right? what does shield do?

Respondent 4: I'm not sure what the point of the game was

Respondent 5: The application doesn't really reflect how sight-reading music works (unless we're talking ukulele tabs or something else with only four strings, and even then).

The progression also doesn't really seem to be synchronized with the music very strongly, making the button presses feel arbitrary rather than timed to a beat.

Respondent 6: see Above

Respondent 7: The game mechanics seem to have nothing to do with music. At first glance, the game looks like a Guitar Hero knock-off, but then I realized that the placement of the "notes" (or whatever the circles are) are completely unrelated to the music. And then I realized that the bars don't even move at the tempo of the song!!!! You could literally play any song in the background and it wouldn't make a difference because the "game" has nothing to do with the music. The music is nothing more than a backtrack.

Respondent 8: Graphics. Would be nice if there were more animations to provide user feedback.

Q3: Do you have any other feedback on the application?

Respondent 1: Add an option to quit the game instead of having to close the window every time its needed to restart. Also, I had 0 health and the game kept on going, that should get fixed to make it realistic. It is hard to use ASDF when the game is vertical, can you flip the game to make it vertical not horizontal?

Respondent 2: Make a tutorial for the first few rounds. That could really help develop some skills before they even begin.

Respondent 3: 1) when i start it up , my mouse becomes unresponsive, it is unsettling to me as the game engine starts up.

2) i couldn't find a way to quit out of anything, so i had to keep hitting x

3) user profiles were not saved.

4) progress bar on the song, because it felt like it was going on forever

Respondent 4: Needs work

Respondent 5: The application doesn't seem to allow for two inputs at once, which seems very counter-intuitive for a music genre game.

Respondent 6: N/A

Respondent 7: - The mob killing mechanic is confusing and doesn't make any sense

- In the demo, the series of circles repeated every like 10 seconds, making for an extremely boring experience

- I can't play multiple notes at once

- I have no idea where in the song I am while using the exercise editor (a waveform as a navigator? really?)

- Sorry if this is coming across as really negative, but I fail to see how this game could teach music in any way.

Respondent 8: Repetition of the notes. The music is a bit quiet compared to the sound effects.