

Clef Miner

Learn Staffs the most amazing way!

Himangshu Ranjan Borah
Department of Computer Science
NC State University
hborah@ncsu.edu

Arnab Saha
Department of Computer Science
NC State University
asaha@ncsu.edu

Abstract

Musical notations are used to visually represent the aurally perceived music through symbols like *Clefs* and *Notes*. However learning musical notations is not a trivial task and many users find it challenging to identify them. *Clef Miner* is a game based learning approach to help users learn about the musical notations. It is a multi-level 2D platformer game where the user has to find the hidden musical notes in the arena to win the game. Finally, we conducted some user tests which showed that a significant number of users learned the musical notations after playing the game.

Keywords

Game Based Learning, Intelligent Tutoring Systems, Clefs, Staffs, Musical Notations, Unity, Education Games, Notes.

1. Introduction:

Constructivism, or learning by doing, is a classic approach to educational instruction that has generated renewed interest in the digital era [1]. Constructivist practices such as apprenticeships have a long history, but in contemporary classrooms, they and other hands-on programs often take a back seat to teachers telling students what they need to know rather than facilitating each person's natural curiosity and learning style.

One approach to digital learning is to harness the broad appeal of video-games for educational purposes. While research on the cognitive and behavioral impacts of violent video games have shown mixed outcomes, some nonviolent games have shown promise. Certain video games

have been shown to improve brain functions while others have the potential of reversing cognitive loss associated with aging. These “serious games” require players to make decisions to drive its progress, and they can range from the simple to the sophisticated.

When education or training feels dull, we are not being engaged and motivated. In other words, we're not really learning. [2] *Learning* doesn't mean rote memorization, it means acquiring the skills and thought processes needed to respond appropriately under pressure, in a variety of situations.

We don't need more time in the classroom to learn how to think and perform in the face of real-world challenges. We need effective, interactive experiences that motivate and actively engage us in the learning process. This is where game-based learning comes in. As it turns out, for many years, video-game designers have been producing and refining highly motivating learning environments for their players to enjoy.

Good game-based learning applications can draw us into virtual environments that look and feel familiar and relevant. Within an effective game-based learning environment, we work toward a goal, choosing actions and experiencing the consequences of those actions along the way. We make mistakes in a risk-free setting, and through experimentation, we actively learn and practice the right way to do things. This keeps us highly engaged in practicing behaviors and thought processes that we can easily transfer from the simulated environment to real life. Research supports the effectiveness of game-based learning in virtual environments, for example, according to a meta-analysis of flight simulator training effectiveness, simulators combined with aircraft training consistently produced training improvements compared to aircraft-only training .

In contrast, traditional, passive training approaches drill us on certain narrow procedures and then evaluate us on our memory of what we were told. Even when we successfully retain the lesson's facts and procedures, our behavior in true-to-life situations remains untested. In addition, even the most comprehensive training program cannot cover procedures for every complex eventuality that we will encounter no matter how thick the binder is. In game-based environments, we learn not only the facts but also the important, underlying hows and whys. This understanding of deeper, more abstract principles prepares us to perform consistently and effectively, even in new and unexpected situations.

In this paper, we discuss Clef Miner which is a game based learning environment which helps you to learn the basics of Musical Notation Theory. As beginners, children and adult music students bring different mental abilities to the task of understanding music notation. Adults understand the concepts of note and rhythm faster than children do. On the other hand, it's hard for adults to grasp that notation is ultimately about sound and physical action where children have an advantage. [3] Clef Miner helps the user in identifying the musical notations in a limited amount of time so it is helpful for both adults and children.

This paper is structured as follows. Section 2 presents some of the related works in this field. Section 3 presents the learning concepts in musical notations related to the game. Section 4 provides the gameplay mechanics with screenshots of the game, and section 5 presents the

different components of Intelligent Tutoring Systems with respect to Clef Miner. In section 6 and 7, we discuss the target audience of the game and the user tests conducted on a few users. Finally, section 8, 9, 10 and 11 presents the softwares, future work, conclusion and future work respectively.

2. Related Work:

Among all the rapidly expanding facets of Computer Science, Artificial Intelligence or AI is probably the hottest area of research and development in the era we are living in. Building Intelligent Systems for Education has been in practice for quite a long time now(since the late 1960s), though it has not reached its extreme yet. More and more genius minds from AI fraternity are shifting towards this emerging field. *Kurt VanLehn's(KVL)* framework in *The behavior of tutoring systems* is one of the most appropriate starting points for the *Intelligent Tutoring Systems(ITS)*. Building Games as an ITS is one of the major areas of research in this domain. It has been widely used for different tutoring domains like military training, medical training etc.

However, we saw that teaching Musical Notations through Games is a relatively unexplored domain. There are indeed a few Games available which thrives to achieve this goal, but none of them has captured the market, as well as some other games for some other domains have done. *Staff Wars 1*, *Staff Wars2*, *Whack A Note*, *Word Warrior*[13] are some among the very few Games that we found in this subject matter. Among all the games that we tried, we found *Whack A Note* to be the most engaging and effective in terms of the tutoring functionalities. But none of the Games are dynamic enough in nature and are very limited in terms of expandability of the domain. In our project , we provide a dynamic platform on which almost all the major aspects of the music domain can be planned to be tutored in gradually evolving levels. To our best knowledge, our Game is the first one of its kind and we found it pretty effective as corroborated by user tests.

3. Knowledge Domain and Learning Objectives

The game is centered around the basic concepts of the Musical Notation theory. Like we write normal languages like English, French etc. using symbols called alphabets, the same way we can express Music in a written form too. There is a whole different world of musical notations which are used to express music. The staffs are the fundamental latticework of music notation used to express music in written form which is one of the very basic things anybody has to learn whenever he or she is starting off with any kind of musical education. Frequently we see that learners find it very difficult in the beginning to grab the concepts of even basic notations and often jumble up different notations and their audio feedbacks. Before going to start off with reading and writing complex scales and full songs in these symbolic notations, it is very important to grab the basic symbol to sound mapping at the very beginning of music education. Here we focus on teaching the rudimentary staff notations with their corresponding notes'

audio feedback and the relevant basic concepts a student needs to get themselves started with any kind of formal musical notation education in the most fun way.

Here we give a basic introduction to the different aspects of the initial lessons that Clef Miner attempts to teach the students. Unlike other languages like English where we need only a set of alphabets to start expressing it in written form, music essentially needs two kinds of symbols even in its most basic form. The first one represents the pitch or frequency range in which a particular note falls and the other for the notes themselves. The first kind of symbols are called *Clefs* and indeed drives the title of the Game. The second kinds are called *Notes* and are represented in terms of its location on a set of 5 lines called the Staff or the Stave. Essentially, we have a very limited no. of Notes in music and those notes keep repeating in equal intervals with gradually increasing frequencies. Before going to the Notes, we show below the meaning of the Clefs and how it maps to the different ranges of the notes in a piano keyboard in the below figure.

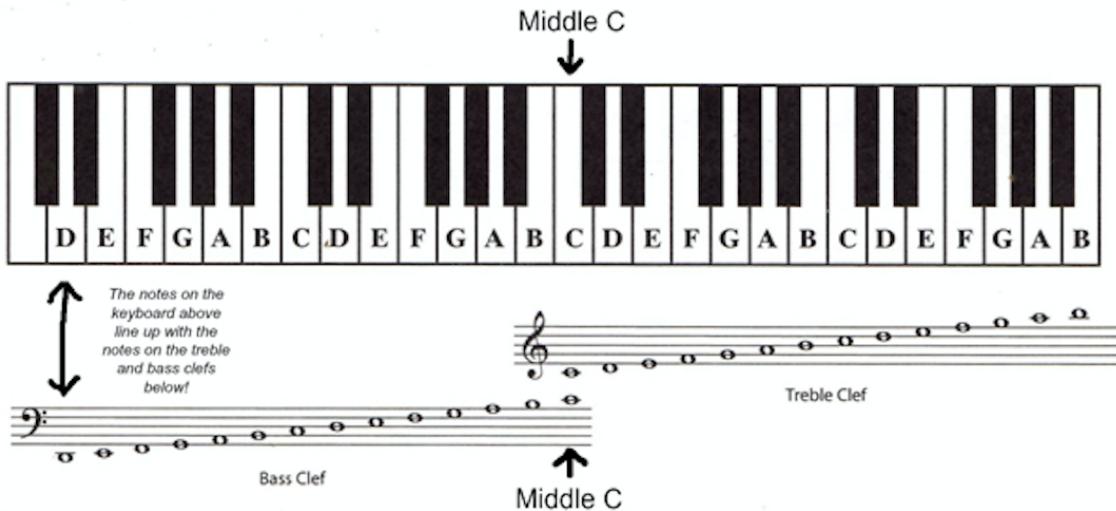


Figure 1 : A rough mapping of the piano keys to Clefs.

So we can see how the Clefs represent the different regions of the keyboard and is able to distinguish between high, medium and low notes. Normally, we use three clefs namely C-Clef or Alto Clef, G-Clef or Treble Clef and F-Clef or Bass Clef which are shown below.



Figure 2 : C-Clef, F-Clef and G-Clef

After the Clefs, there comes the Notes. We use 5 lines called the Staff or Stave as mentioned above to express written music. At the beginning of the lines, a Clef is written down and the Notes are denoted by an empty or a filled dot on one of these lines. The line on which the dot is written defines the note along with the Clef, which denotes its pitch. So basically, using these two kinds of symbols, a player who is reading the notation knows which key to play in the keyboard! The first step to learning while taking any formal music writing class is to identify the combination of Clefs and Notes and their corresponding audio feedbacks.

What we have described here is a pretty basic form of written music which probably is very insufficient to express music for most of the cases. Of Course, the whole music theory contains a whole bunch of other features and symbols of gradually increasing complexity which we haven't covered here. But the aim of this section is not to introduce the reader to the full domain, it's just an introductory section to give a flavor of what kind of subject matter will we be dealing with.

4. Gameplay Mechanics

Clef Miner has a comical backdrop where the central character, *Hazel* was becoming one of the best singer and musician of all time but the other musicians grew jealous of her. So they conspired against her and magically stole her musical abilities. They hid them in crates and shipped them to the 2D world which is the arena of the game. Now she has to fight back against her enemies and collect all the musical notations in an attempt to gain back her singing and musical abilities. The various features of the game are described in the next subsections.

Architecture:

Unlike OOP languages, it is difficult to implement any pattern using Unity but as two programmers were working on the project, a standard way had to be followed to avoid confusion and spaghetti code. We implemented our code in *Model-View-Controller* architecture. We organized our code into the groups as follows:

- **Data:** Model classes
- **Logic/Workflow:** Controller classes
- **Rendering/Interface/Detection:** View classes

Gameplay:

We start off with the moves and animations that were built into the central character of the story, Hazel. The player has 5 animations namely idle, run, jump, shoot and knife attack. All the animations are being used except the knife attack which is currently not being used and we hope to use it in near future for next levels. In the main menu page of the game, an item called "Controls" is there which shows the basic key events for different attack and movements and

also describes the scenarios where the player dies. In one level, the player gets 3 lives and dies in the following three scenarios,

- When the Player touches an enemy.
- When the Player falls off the platformer.
- When the player touches the bottom of the moving ladders.

Typically, the player moves around the arena using the direction keys, jumping with the space bar and shooting with shift keys. It has to touch the friends to get the points. The arena is quite big as compared to the camera view and the camera follows the player all the time with a Camera Follow mechanism. There's another camera view on the right top corner of the main screen which covers a lot more of the arena and we name it as a the *minimap*. This *minimap* helps the players find the hidden treasure with ease.

The left top corner of the main view contains two things, one is the score board and the other is the review screen. The score board contains 4 items as described below,

- **True Hit Score :** A metric to measure the correctness of the identifications.
- **False Hit Score :** A metric to measure the wrong identifications.
- **Life Left :** No of lives left
- **No. of Mines :** It shows the no. of mines that the player still needs to find in the arena to goto the next level.

Scoring:

The scoring mechanism used here tries to capture the effective amount of correct identification of the projected learning material. Whenever the player identifies something correctly, the *true hit score* goes up. Whenever he misidentifies something, he gains *false positive scores* and also loses some amount of true positive score. Moreover, whenever he fails to identify something within a timeout period, then also he gains a false positive score. We use these two scores to define something called delta, which is the difference between the true positive score and the false positive score and is used in many places to asses the player's achievements with respect to the learning goals. The second part of the top left screen is the review screen, which is there to help the player with a short review of the current level's tutoring materials in case his delta is going positive. For the first level, the review screen is always on. But in the second level and onwards, the review screen only pops up as and when the delta stays positive. Whenever it goes below zero, it's switched off automatically. This provides a really useful guidance to the player and only when it is needed.

The Two Mines:

We designed a kind of GameObject called the *Note Mines* which encapsulates the Notes and the Clefs as described in the learning domain overview. The player has to shoot the Note Mines to open up the actual notes inside it. There are two kinds of Notes as well as note mines. One kind is friends and the others are opponents. The friend Note Mines encapsulate Friend Notes and

the opponent Note Mines hides enemy Notes. In the first level , a friend Note starts to move away from the player and an enemy note starts to move towards the player. The player has to shoot the enemy note before it touches and grab the friend note before it goes away. There's a timer of 10 secs on every note after which it explodes itself. In the second level, due to the complex nature of representation of the learning components, we found that it was very difficult to identify the Notes if they keep moving, so we had to make them stationary. Even for the similar reasons, we had to remove the rotation of the clefs in the first level. There are 4 scenarios which are mainly handled with respect to the Notes. They are briefly described below.

- When a player shoots a friend note, he loses true hit scores, gains false hit score.
- When the player grabs a friend note, he gains true hit scores.
- When the player touches an enemy note, he loses a life, gains false hit score and loses true hit score.
- When the player kills an enemy note, he gains true hit score.

Also, whenever the player uncovers a Note Mine, the Note Mine count in the scoreboard is decreased by one.

Active and Passive Learning:

Clef Miner has a concept of friends and enemies. The actions are pretty simple, shoot an enemy and grab a friend. However, this simple model has helped the tutoring module up to a great extent with some thing that we call "Active Learning" and "Passive Learning". In a particular level, one of the Notes is your friend and the others are enemies. While playing, we have seen that the users are pretty much concentrated on the friend Note's identity and pays less attention to the enemy. As told by one of our users from the survey, "I don't care about the enemies, I just see if it's not the friend !". And it turned out that, at the end of the level, the players are able to recognize the friend note with 100% confidence and enemies with errors. Whenever we teach a notation by making it a friend, we call it as "Active Learning" and when we introduce the note as an enemy, we call it "Passive Learning". As the names suggest, the active learning is much more effective than the passive learning. However, passive learning acts as a good base to build up the identity of the notes in the unconscious mind of the player and when that enemies become friend notes, it just reinforces the learning.

Feedbacks:

Feedbacks are one of the major component of any Intelligent Tutoring System. They provide the actual tutoring functionalities which the software intends to provide. Here in our Game, we introduce mainly two kinds of feedback mechanism namely, immediate feedbacks and overall feedbacks. The immediate feedbacks are given using some non-blocking popups inside the game view and keeps a conversation going with the player. For example, whenever, you shoot a correct note, it says "yay ! Correctly Shot a G Note !" or whenever you touch a wrong note, it says "Oops ! That was a D note, not your friend !" etc. We give immediate auditory feedbacks even if the player makes a mistake, and this is very helpful for the passive learning that we explained above. Also it keeps track of the delta as mentioned above and changes the feedbacks

according to that. For example, when the review screen appears it says, "Please review the Notes above !". These popups are also used to guide the player at certain points where we think it's necessary to guide the player to find something out in the arena like the Bonus Boxes. We found from our user tests that the immediate feedbacks were excellent in terms of guiding the player and they found it very engaging. For the overall feedbacks, we used some space from the Game Over / Game Completed screen. Again based on the delta value when the game ends, we give an overall feedback message to the player saying how he did in the game. For example when you score a delta of more than 1000, the message "Genius in Action! Hatz off to you Sir!" pops up in the Game Over Screen. This message acts as a great encouragement to the player and they can't help moving to the next levels.

Tutorials:

The Game Has a tutorial mode where it introduces the player to the basic moves and the actions to be taken on the Notes depending on whether they are friends or enemies in the Game Arena. By pressing the "TUTORIAL" button on the main menu, we can enter the tutorial mode. It first gives a basic introduction of the subject matter and then goes into the play mode. As the player starts playing, we direct the player using blocking pop ups in the game-view and instruct the player what to do next. If the player makes mistakes, then we tell them and restart the tutorial level to try again. After the tutorial is over, we show the users a review screen of the friend and the enemies in the main arena and also give a chance to play the audio feedbacks of the Notes(Only for the levels with Notes). Then the user enters the main arena.

Levels and Audio Feedbacks:

Currently we have 2 levels implemented. The first one introduces us to the Clefs and the Second one introduces to Notes. The audio feedback of the Notes are also played for the Notes in the second level whenever a note is touched or shot. Also for the levels with Notes, we play a background which is consistent with the scale of the Note to keep the environment grooved for that particular note. The Game has a static class called SoundManager which manages the different playbacks of the various audio in separate channels.

Apart from all the tutoring functionalities, we also introduced some entertainment components in the game like Bonus Boxes, which has to be shot in order to activate a lift which starts moving and helps the player to get to the upper portion of the arena. Also, the friends and enemies' generation is controlled dynamically by a class called FriendEnemyManager which helps designing the objects in the arena dynamically. We have kept this flexible design so that going forward we can implement procedural content generation to generate levels automatically which will leverage the similarities between different musical concepts. The different concepts explained here can be very easily mapped to the snapshots of the Game that we provide towards the end of the report.

5. Intelligent Tutoring Functionalities

Clef Miner has all the four major components of an Intelligent Tutoring System: the domain model, the student model, the tutoring model, and the tutor-student interface model.

Domain Model:

The domain model as the name suggests contains the set of skills, knowledge, and strategies of the topic being tutored. It normally contains the ideal expert knowledge and may also contain the bugs, mal-rules, and misconceptions that students periodically exhibit. In Clef Miner, we have represented the Clefs and Notes as domain elements. We have also added several hints and feedbacks for the misconceptions that students show during the game. We plan to add the scales in the domain model in the future.

Student Model:

The student model consists of the cognitive, affective, motivational, and other psychological states that are inferred from performance data during the course of learning. Typically, these states are summary information about the student that will subsequently be used for pedagogical decision making.[4] Clef Miner keeps track of the user's cognitive model by keeping track of the correct and incorrect hits made by the user during the gameplay. Other than that it keeps track of the number of lives left where the user starts with 3 lives and loses a life when he collects incorrect clefs or notes or falls below the arena. Although it provides feedback, it follows Programmed student model. The student model is designed as a linear programming model where the student is expected to learn the Clefs in the first level and Notes in the second level.

Pedagogical Model:

The pedagogical model takes the domain and student models as input and selects tutoring strategies, steps, and actions on what the tutor should do next in the exchange with the student to move the student state to more optimal states in the domain. Although Clef Miner doesn't have a pedagogical agent, it consists of a sequential pedagogical model where the user faces a sequence of friend and enemy clefs or notations which are randomized in the arena.

Tutor-Student Interface Model:

The tutor-student interface model interprets the student's contributions through various input media (speech, typing, clicking) and produces output in different media (text, diagrams, animations, agents). In Clef Miner, other than the gameplay, we have added three forms of feedback - Instructions, Hints, and Non-blocking pop-ups.

- **Instructions:** The instructions are shown in the beginning of each level which explains the user about the contents of the level. In the first level, the user gets two instructions stating about the definition of clefs and a brief introduction about how they are used in musical notations. In the second level, there is one instruction interface to explain the notes. Instructions are always attached with images to help the user understand the domain better.
- **Hints:** The hints are shown in the tutorial level which freezes the gameplay so that the user can read the contents. Sometimes they are associated with an image to help with the visual understanding. Hints have lesser number of lines than Instructions, usually in the range of 2 to 3 lines. The user can either select Okay or press Enter to disable the Hint. Hints help with the understanding of the game-play.
- **Non-blocking pop-ups:** The Non-blocking pop-ups are some in-gameplay feedbacks which usually is a single line of feedback explaining the current state of the user. The primary difference between the non-blocking pop-ups and hints is that unlike Hints, it does not freeze the game. The pop-ups last for 2 secs and animates out.

6. Target Audience

The target audience domain of our game can be novices who has started to learn music or can be amateurs who have trouble in learning the notations spread across different age groups. Basically, anybody who is starting off with any kind of formal education in music will find it very useful. Even people who don't belong to any of these categories may find it interesting to play the game as one can learn some basic idea of written music in a fun way.

7. Evaluation

User Test:

In order to test our game and to find out if our game helped the users learn about the musical notations, we conducted a user test on a sample of 5 users. As we were still in the development phase for Level 2, we managed to test Level 1 on the users and collected the results. Initially, we presented a Pre-survey where we asked them if they knew about the musical notations. If they selected no they were simply asked to play the game. In this case, we have assumed that the users have no knowledge about the domain. If they selected yes, they were asked 5 multiple choice questions about the clefs. The first two were about the general knowledge regarding clefs and the last three were to identify the 3 common clefs. If the users managed to answer all the questions correctly he or she was considered an expert, thereby they did not play the game.

After the pre-survey, they were asked to play the game. First, the tutorial level was selected and the users were motivated to read all the tutorials properly and not help them out. The reason for this was to check if our game has enough tutorial and feedbacks to help the users figure out all

Clef Miner Pre-Survey

Part II

Indicates the knowledge of the user

What is true about Clefs? (multiple options can be correct)

- Clefs have specific sounds
- Indicates the pitch of the written notes
- Denotes the frequency ranges of the notes

How many clefs is normally used?

- 3
- 4
- 5

the problems on their own. Finally, we presented the user with the post survey which has the same questions as the pre-survey. This helped us to find out if the users can identify all the clefs and can answer all the questions correctly.

Figure 3 : Snapshot from the survey

Identify the clef A

- C-clef
- F-clef
- G-clef

Clef A



Figure 4 : Snapshot from the survey

User Experience:

During the development phase, we followed the agile model; we got a lot of important feedback from the users and used their opinions in designing our game and learning component. In the initial stages of development, we didn't have the tutorial levels and the game started directly. There were some static pages or hints which told the users what to do. The users found the static hints confusing and many users ended up shooting both the friend and enemy clefs. With the help of their suggestions, we decided to add the extra two levels later which reduced the confusion. During the user tests, most users told us that the feedback was short and helpful. Initially, the users had trouble finding the different crates so we added a mini-map camera to show more of the map. The users found it helpful to find the notes.

Although Level 2 was in the development phase during the user-tests, we asked the users to play a prototype. They enjoyed the game and one of them suggested to show all the notes if they are shooting the wrong notes which we later included in the game. Overall most of the users told us they enjoyed playing the game and more importantly learned about the clefs and the notes.

Results:

The tests were conducted in the graduate commons section of James B. Hunt Library. None of the users had knowledge about musical notations so they played the game directly. The users spent an average of 8 mins on the game and less than a minute on the post-survey. 4 out of 5 users answered all the questions correctly in the post-survey. One of the users made mistakes while identifying the two enemy clefs. In all the results pointed out on the fact that the users, in fact, learned about the clefs by playing the game.

8. Softwares and Programming Languages

The Game was primarily developed in Unity with C# Scripting for everything. Apart from Unity, we also had to use few other third party softwares and tools for resource generation and modeling. We list them below.

- Photoshop CS5 Extended
- Garage Band
- Audacity
- POD HD 300 Edit

9. Future Work:

In the limited timeline available for us, we could implement only two levels of the Game and going forward we plan to construct more of them. After the notes, we can start putting them

together and teach players about the concept of the basic scales. In terms of gameplay, we can implement something like you have to collect the notes in a sequence in order to achieve something. We can introduce new enemies in the arena and try to map the concepts of the enemy attacks to the actual behavior of different elements of the music theory. Currently, we have only one Note or Clef as a friend and rest as enemies, but going forward we have to make every Note or Clef friend once for the benefits of active learning as described above. The scoring system that we used is also quite naive, we can use more sophisticated scoring systems going forward. This will help us to do more advanced student modeling and the feedbacks can be refined using that. We can detect the mistaken pair of Notes and give Note Specific feedbacks. We also believe that doing more user tests will fine tune the game more and more.

10. Conclusion:

This report describes the motivations and implementation level details of an intelligent tutoring game called Clef Miner which primarily aims to teach basic musical notations to the beginners. The game has been implemented as a 2D platformer where the central character Hazel, keeps moving around the open 2D Arena to find out hidden Clefs and Notes which can be unlocked from mines by shooting them. The basic gameplay rules of the game are pretty simple where we introduce one kind of Notes as friend notes and the others as enemy notes in one particular level. The player has to shoot the enemies and grab the friend notes in order to gain score. There has been lots of tutoring functionalities embedded into the game like active and passive learning, blocking and non blocking popups, immediate and overall feedbacks which we have explained in the report in detail. Currently the game has been implemented in Unity using C\# programming language and 2 levels have been completed. From the user surveys it was seen that the overall learning gain from the game was significant and approximately 4 out of 5 students were able to recognize the intended Notations after playing the game.

11. Acknowledgement:

We would like to express our sincere thanks and gratitude to Dr. James C. Lester and Dr. Bradford W. Mott for their precious guidance during the Spring 2016 semester in the course Game Based Learning Environments. Their constant guidance and encouragement, and their immense dedication to work, motivated us to put our best efforts towards the course project and the entire course. We would also like to thank our friends for being the source of continuous feedback throughout the development cycle of the Game and the students who took part in the user test and post-survey of Clef Miner and helped us with their valuable feedbacks.

References:

Papers and Articles:

1. <http://journalistsresource.org/studies/society/education/outcomes-of-game-based-learning-research-roundup>
2. <http://www.newmedia.org/game-based-learning--what-it-is-why-it-works-and-where-its-going.html>
3. <http://www.musicalfossils.com/>
4. Pavlik et al. A Review of Student Models Used in Intelligent Tutoring Systems.
5. <https://methodbehindthemusic.com/theory/notation/#staff>
6. <http://www.musicnotes.com/blog/2014/04/11/howtoreadsheetmusic/>
7. <https://en.wikipedia.org/wiki/Clef>
8. https://en.wikipedia.org/wiki/List_of_musical_symbols
9. Vanlehn, K. (2006). The behavior of tutoring systems. International journal of artificial intelligence in education, 16(3), 227-265.
10. <http://www.newmedia.org/game-based-learning--what-it-is-why-it-works-and-where-its-going.html>
11. Building Intelligent Interactive Tutors, Beverly Park Woolf. Morgan Kaufmann: San Francisco, 2008.
12. Computer Games for Learning: An Evidence-Based Approach, Richard Mayer. MIT Press: Boston, 2014.
13. <http://midnightmusic.com.au/2014/08/30-fun-ways-to-teach-the-notes-of-the-staff-using-technology/>

Resource Repositories:

1. <https://www.assetstore.unity3d.com/en/>

2. <http://opengameart.org/>
3. <http://www.freesound.org>
4. <http://www.gameart2d.com/>
5. <https://www.youtube.com/watch?v=SCRupyidkYA>
6. <https://www.youtube.com/channel/UCyBsvsU7uiurMiBZIYXvnyg>
7. <https://unity3d.com/learn/tutorials>

GitHub Link to the project's source code

<https://github.com/asaha1/ClefMiner>

Snapshots From the Game:



Figure 5 : Main menu

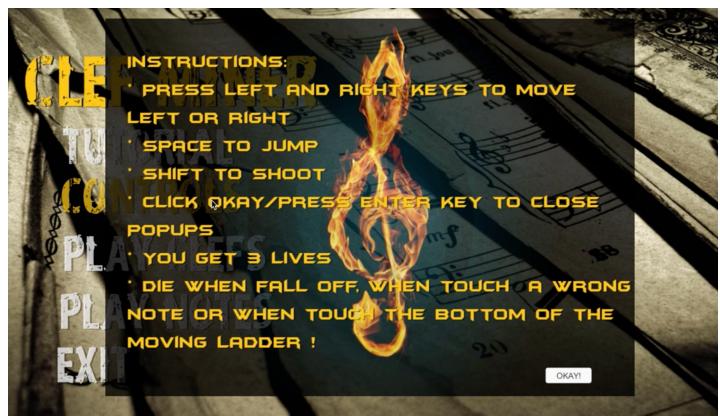


Figure 6 : Controls

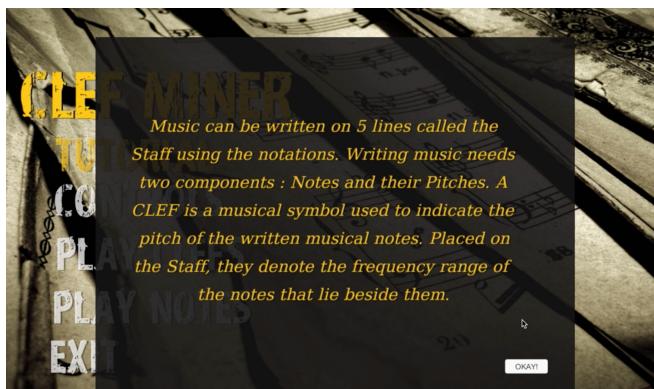


Figure 7: Tutorial Introduction

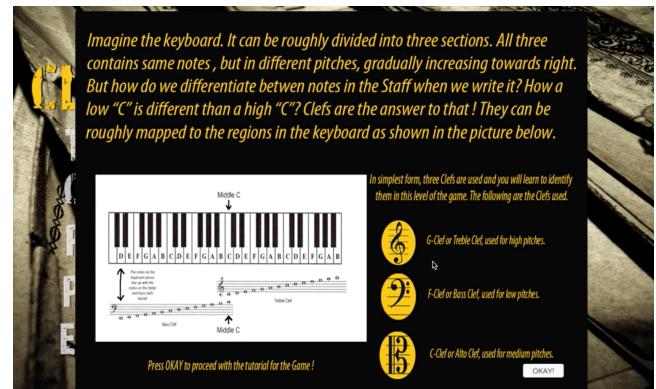


Figure 8 : Tutorial Description



Figure 9: Tutorial phase 1



Figure 10: Intro to friend clef



Figure 11: Pops open mine

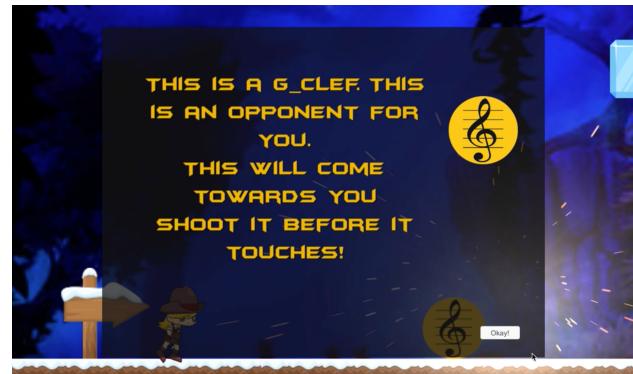


Figure 12: Enemy Clef



Figure 13: Correct Shot

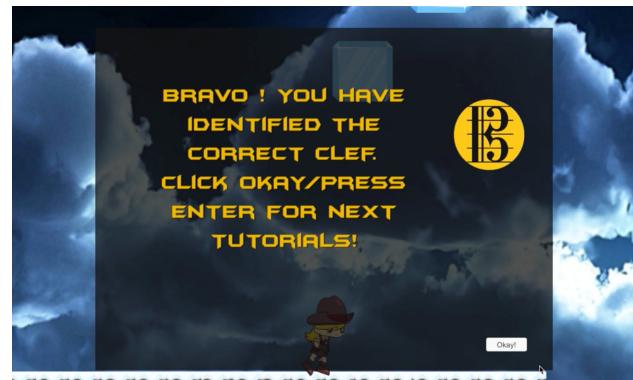


Figure 14: Tutorial phase 2



Figure 15: Killing a wrong Note



Figure 16: Tutorial Review Level 2

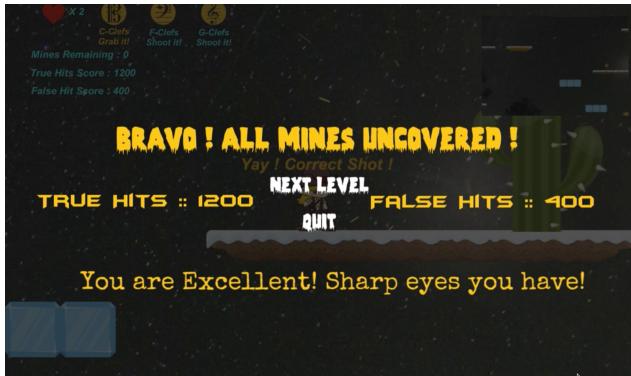


Figure 17: Game Won Screen



Figure 18: Tutorial Level 2

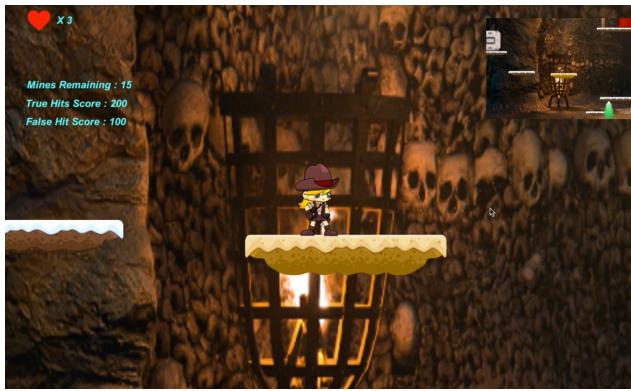


Figure 19: Hazel on Ladder



Figure 20: The Bonus Box