Objectively Analysing Musical Improvisations for use in Games

<u>Abstract</u>

This paper analyzes the viability of an improvisational musical game.

Research is conducted into rhythm games and their history with special attention paid to rhythm games that attempted to emulate the musical experience. Currently used Improvisational assessments carried out by humans are researched to help shape the artefact's algorithm and play structure. Computational musical analysis methods are researched to aid in the creation of an algorithm that would be able to judge musical improvisations.

To test this viability a research artefact was created in the form of an Unreal Engine 5 vertical slice of a musical game that would allow players to improvise within it and then have that improvisation judged by an algorithm. This artefact is somewhat unfit for testing this viability as it does not include many analytical methods researched and so it is unequipped to judge musical improvisations.

The research artefact was tested by a small group of testers, all of whom had different levels of musical skill. After testing the artefact this group was given a questionnaire, made up of both qualitative and quantitative questions, that attempted to grasp their opinion and/or enjoyment of the artefact as well as features they would like to see added to improve it.

From this small questionnaire, there is evidence to suggest that there is enjoyment that can be derived from an improvisational game which in turn shows viability in the idea. The responses to the questionnaire also suggest that those with more musical skill are likely to enjoy the game more than those without musical skill. No concrete conclusions can be drawn from results with such a small sample size.

Introduction

Videogames often base their core gameplay loops on abstractions of real-life tasks and actions. One example of this abstraction taking place is in rhythm games. Whilst rhythm

games encompass some elements of the real-life experience of playing music, they are often limited in scope and depth, focusing only on rhythm. This exclusive focus creates a more accessible genre at the sacrifice of musical expression and exploration; however, the genre has experimented with and recontextualized how rhythm can be tested.

Much of this experimentation has been focused on the abstraction of keeping rhythm, such as by having you time punching flowerpots in *Rhythm Heaven Fever* (2011). Other experimentation has been undertaken by introducing other non-musical elements such as in *Osu!(2007)* which has players aim and keep rhythm at the same time. Other games have gone in the other direction, attempting to present that you are playing music. *Guitar Hero*(2005) and *Rock Band(2007)* have users playing the game using controllers shaped like instruments and if the player doesn't correctly perform what the game tells them to do the music sounds worse and a virtual crowd will boo. These games present themselves as an imitation of a musical performance more so than contemporaries in the genre, but due to the limited focus on musical elements as well as the lack of player agency, a simulacrum of musical performance is created.

This paper will attempt to investigate the efficacy of creating musical games that attempt to test more than just the player's sense of rhythm and better simulate the full experience of playing music by creating a game that allows a player to improvise over a backing track and then judge that improvisation by giving it a score.

To create this game a system for judging musical improvisation must be created and to do this research is conducted into the current assessments for musical improvisation. Whether they use more objective measurements such as rubrics or rely on human musical intuition and judgement to appraise improvisation. Also, of interest to the research artefact is what the assessments give to players to improvise over and do they separate into multiple tasks which test individual skills of improvisation or are they simply playing over one backing track and that performance is judged?

This paper researches how computers read and understand music to aid in digitizing the improvisational assessments for use in the game. Research into how computers read music consists of an exploration into the MIDI file format, how it works and how to create one. Research into computational musical analysis i, the study of how computers can analyze music, is conducted also. Special attention will be given to the analysis of pitch, harmony, rhythm and improvisational techniques such as repetition and variation of sequences because they are highly important to improvisations. Special attention must be paid to the time complexity of these analysis techniques due to the real time nature of games, if an analysis takes a day to produce it is less useful for application in a game.

Aims and objectives

The ultimate aim of the paper is to analyze the viability of an improvisational musical game as an alternative to traditional rhythm games, to complete these three smaller objectives must be completed.

The first objective is conducting research into rhythm games, their history and how they do or do not test the musical capabilities of the players. Specific focus should be put on guitar hero and rock band as these games present themselves as a musical experience compared to games featuring more abstracted musical systems such as *Osu!(2007*) Or *Rhythm Heaven Fever(2011)*.

Research into Improvisation assessments is important, how the assessments are conducted and how they are being marked, what is being looked for by the assessors. This research will be conducted to better inform what the algorithm should look for and how it marks.

Research into Computational musicology is being conducted, how computers process music and turn it into something they can read as well as how computers can understand chords or structure in order to have the algorithm recognize and judge the player's input

The Second objective is to create a research artefact in which the algorithm can be tested by the player. The player will need to be able to play some kind of virtual instrument where the algorithm will be able to listen to the player. The game should also offer some kind of assistance to help players improvise.

The third objective is to have both talented musicians and not test the game and answer an anonymized survey about their experience relating to their enjoyment of the game and its perceived musical correctness as well as what features they felt were lacking or should be added, in order to gauge whether or not this is a viable game.

Literature Review

The State of Rhythm Games

Rhythm games are researched due the artefact being built upon the foundation of the rhythm game.

Guitar Hero and Rock Band inhabit a strange middle ground between rhythm game and musical performance which is of great interest to the algorithm. Other rhythm games were heavily abstracted from playing a real instrument in front of a crowd, Parappa the Rapper instructs players to press buttons on a controller repeating a rhythm presented in a visual style befitting of a children's cartoon. Guitar Hero instructs players to play on a guitar shaped controller over recognizable rock songs where a virtual crowd cheers or boos depending on how well you play the song. Roland Barthes posits in a collection of essays that there are two kinds of music: "the music one listens to" and "the music one plays" he thought highly of the latter writing "the body controls, conducts, coordinates having itself to transcribe what it reads, making sound and meaning" (Barthes, 1977). Kiri Miller in her book titled "Playing along" states that Guitar Hero and Rock Band blur the lines between listening and playing, referring to it as "Schizophonic Performance" (Miller, 2012) using R.Murray Schafer's definition of" Schizophonia" the splitting of an original sound and its electroacoustic reproduction. She justifies the use of the phrase because the feelings R. Murray Schafer had about the reproduction of sound mirror the feelings of ambivalence and paranoia around much of Guitar Hero and Rock Band's reception, she even goes as far to say "These games threaten the sanctity of those distinctions. They dramatize schizophonia, endorsing the idea of a split between the live and the recorded and inviting people to play at mending that split."

Much of these feelings of paranoia and fear came from media critics and rock musicians, many of their arguments point out the perceived lack of authenticity and that possible future musicians play these games and feel satisfied never having picked up a real instrument. and even goes on to suggest that had these games released earlier great influential rock bands would never have formed. Players and developers of *Rock Band(2007)* and *Guitar Hero(2005)* defended against these accusations by claiming that it is simply a game and does not act as a cheap copy of really playing music. Greg LoPiccolo, Vice President of product development for Harmonix Music Systems repeats this sentiment exactly.

However, despite his statement some players describe having musical experiences with them. From a survey Kiri Miller conducted from 2007 to 2010 about *Guitar Hero(2005)* and *Rock Band(2007)*'s gameplay contexts, the game's impact on players' musical tastes, and comparisons to other musical experiences. One respondent said "I feel like I'm jumping into the artist in their time and playing along and maybe even feeling what it was to be that creative individual in their time" (Miller, 2012). This experience of immersing someone in a world they could never get to in real life is what games do best. Another respondent named Heather who describes herself as unmusical loves Rock Band and felt whilst playing that she understood the beat better. Her account and many others describing how playing these games gave them a taste of what its like to be a musician leads supports the supposition of the defenders of *Rock Band(2007)* and *Guitar Hero(2005)* that these games act as introductions to playing music. The company CEO of Harmonix states that "our

mission was to show nonmusicians how it feels when you finally get to the other side. And hopefully, to inspire them to start making music the old-fashioned way."

How are Musical Improvisations Assessed?

Musical Improvisational assessments are researched in an attempt to find the structure that could be used for the artefact. Also of interest how they are marked and how much human judgement is used compared to standardized rubrics.

Test of Ability to Improvise (TAI)

McPhearson with the help of music educators, researchers, composers and jazz improvisers designed a set of tasks that would attempt to accurately assess a young musicians' improvisational abilities. Five tasks were developed and iterated on, four of which focused on giving students restraints to improvise within.

The first task, Improvisation of a Closing Phrase, gives students two short opening phrases and asks them to continue a given phrase "in a way that would provide a well-balanced melody comprising two phrases". This type of task is present in much music teaching such as in jazz instruction and organ/keyboard instruction. This type of task is also used in official musical evaluations such as the exams provided by ABRSM(Associated Board of the Royal Schools of Music). Tasks like this were present in some of the first creativity tests, such as Vaughan's (1971) Musical Creativity Test. In this test musically untrained children were asked to improvise an "answering" rhythm and closing phrase using tom tom and bells. The task is designed to test the student's ability to intuit the shape and flow of an opening phrase and how to create a complimentary closing phrase.

The second type of task on the TAI is Improvisation on a Rhythmic Pattern. Said task has been used widely to cultivate improvisational ability (Baker, 1969)Students are given a rhythmic pattern that they must follow, the improvisation portion consists of choosing which notes to play over the given rhythm, so creating their own melody. As this task doesn't test their rhythmic improvisation but instead their harmonic improvisation the interest in this approach likes with the students' ability to choose pitches that fulfill stylistic requirements.

The third task, Improvisation on a Motif, is similar to the first task, but gives students an opening phrase and asks the students to generate a balanced melody of at least eight bars in length based upon it. Commonly used to develop improvisational skills, the exercise is also present in early creativity assessments such (Vaughan, 1971)'s Measures of Musical Divergent Production. The task attempts to test the students' ability to improvise a melody as a homage in shape and rhythmic feel to the given opening phrase. This could be done

using a variety of musical techniques such as repetition, inversion, transposition, fragmentation or elaboration of the given phrase. The instructions given to the student before attempting the task alert the student to the previously mentioned techniques.

The fourth type of task given is Improvisation to an Accompaniment, this asks students to improvise over an accompaniment and were also asked to complement the style of the accompanying passage. Improvising over an Accompaniment is the foundation of most jazz, popular music improvisation and traditional forms of improvisation. Classroom/instrumental music instruction constantly uses this improvisation to teach students, due to its widespread use early creativity measures such as (Vaughan, 1971) included this as a task.

The final task was created to assess the student's ability to improvise in a "freely conceived" style. Its inclusion was in part due to previous work by Gorder W. D(1980) and Flohr, (1979) as well as discussions Webster had with expert musicians and music educators. In the final task the student is asked to provide their own "freely conceived" response, without any kind of accompaniment or opening phrase to base their improvisation on. The instruction given is to "perform an extended improvisation in any style or mood that you choose. You are free to play anything you like so let your musical imagination roam free" (McPherson G. E., 2019)

McPherson developed the scoring procedure with the help of existing literature and discussions with academics, music educators and expert improvisers. At the time of development, no improvisation assessments were designed to examine high school instrumentalists' ability to improvise and so a new criterion had to be devised. The difference is needed because previous tests were made and administered to young untrained children who are given open-ended tasks instead of the more rigid tasks devised for high school students. Another key difference between the testing of untrained children and skilled high school students is the testing of Instrumental fluency, one of the four main criteria that the TAI quantify. Instrumental fluency is defined by McPherson G. E. (2019) as the ability to execute musical ideas clearly and accurately and is demonstrated through their ability to perform in a spontaneous manner, such as by moving easily from one musical idea to another. Musical syntax is the second of the four main criteria and refers to the ability to organize musical material by adapting to the prevailing style and complementing set criteria (McPherson G. E., 2019). Proficiency is shown by demonstrating rhythmic feel, melodic sense, tonal organization and providing a response that makes sense musically. Creativity is the third of the four main criteria and is made up of two parts Musical flexibility, how the improvisor can manipulate pitch rhythm etc., whilst building upon the given phrase or accompaniment, and Musical Originality, how unique or

unusual the response is. Musical Quality the fourth main criteria is described as "Overall Musical Appeal" (McPherson G. E., 2019) and measures the improviser's ability to perform fluently creatively conceived material that complements existing musical criteria

The criteria with which it measures the aspects of an improvisation; small snippets meant for influencing human judgement rather than a machine's but will still work as a guide to how the algorithm should judge improvisations. The breakdown of how inexperienced players and experienced players responded to the tasks and ultimately succeeded or failed at that task will be another influence when creating the algorithm. The tasks laid out follow a game-like structure with the tasks slowly building in complexity and length similar to video game levels, implementing these tasks could create a way to ease players into improvising.

CAT

The Creative Assessment Technique differs to the TAI due its lack of rubrics and mark schemes, Amabile did not base it on any ideas of objective creative thinking but simply uses experts' knowledge and opinion to assess.

To apply the CAT, judges rate the creativity of given creative products on a six-point scale, they determine their given score by using their own subjective ideas and expertise in their own fields to determine the products value. Using judges' opinion on their own was a innovative idea, other creativity assessment techniques at the time were focused on creating models that would break down creativity into several different attributes and rubrics to score those attributes (McPherson G. , 1993), (Torrance, 1974) (Webster, 1994). These more rigid assessment techniques also needed to be broken down differently for different fields. Amabile and other researchers have tested the reliability of the CAT and have found that it's highly reliable across a wide variety of creative fields and across a large age range of participants.

The CAT has been used to test musical creativity with the same level of consistency and reliability (Amabile, 1996) however the majority of these assessments have been focused on testing music composition skill rather than musical improvisation. Which makes it a worse candidate for inspiration for the artefact. Two studies have been conducted using the CAT. Eisenberg and Thompson(2003) asked adult musicians to listen to a minute long excerpt of music and then asked to improvise for a few minutes on the basis of their impression of that music. Beaty et al.(2013) gave musicians a piece of music they had never previously performed and after a practice trial of one minute they played the melody with a band once and then improvised over two complete iterations of the song, this improvisation was then judged using the CAT.

The TAI and CAT have been compared directly in the study "A quantitative analysis of two improvisation assessment instruments". Performances of middle school clarinet players playing both their instrument and an iPad were given all the tasks on the TAI (Healy, 2016). These recordings were then reassessed using both the CAT and the TAI. Six judges were chosen all of which had at least 5 years of experience teaching middle school-level band students, these judges were then split into two groups of 3 where one which would assess the recordings using the TAI and the other would use the CAT. The inter-rater reliability which means how little each judges score differs from the other was calculated for both assessment methods and it was found that the TAI had higher inter-rater reliability than the CAT with the TAI achieving.79 compared to CAT's .71, this suggests that the TAI is slightly more reliable because the judges scores were closer together, The study posits that this might be due to the TAI's rubric giving anchors to the judges of which to base their score off. The means of total scores for each assessment was also calculated, TAI had significantly higher means of total scores than the CAT, TAI's mean being 29.91 and CAT's mean being 20.06, this implies that the CAT may be more difficult for students to score higher than on the TAI.

TAI is a much better source of inspiration for this project than the CAT due to its solid researched task list and rubric based marking. McPherson developed the TAI in 1993 and has been conducting studies with it and refining it over multiple decades, whereas uses of CAT for musical improvisation have only appeared in two studies. This is relevant because McPherson has analyzed the results of the TAI over decades, generating many insights about how it assesses people, the improvisations that are generated by a good player and a bad player and the difference in improvisations from different genders. These insights are incredibly valuable when attempting to digitize these assessments because the algorithm can be designed judge on the differences set out by this paper. The rubric of the TAI also helps to explain the thinking behind what makes a good improvisation. The CAT leaves all these considerations and judgements to the brains of the judges and whilst that does make an effective assessment technique the thoughts that produce the scores remains obfuscated by their brains and unless the algorithm will attempt to emulate an expert's mind the TAI will be more closely followed.

Discrimination Present in Improvisation Assessments

The TAI was developed to test specifically jazz improvisational ability in students who have been taught how to read musical notation and perform in a classical style. McPherson G. E., (2019) admits in the introduction of his paper that "the development of appropriate ways to assess students' abilities to improvise needs to begin with a thorough understanding of the context and the educational needs of the students." For example, four of the five task types present in the TAI would be useless to students of different musical styles such as Classical Indian or Gamelan music which use completely different scales and notations to the ones used on the test. The final task however does allow for students of any musical origin to understand it and attempt.

CAT does not face this problem because it is much more flexible than the TAI. Where the rubric and the different aspects that are being assessed by the TAI would have to be retooled entirely for music from different parts of the globe. The CAT would only require experts in the musical style that wanted to be assessed to conduct an assessment, due to its lack of rubrics and full faith in people and their subjectivity.

Care must be taken with any observed variance between submissions from different demographics. For example on the Tai, evaluators met eight months after assessing students to discuss the taped performances. When listening over they showed an ability to tell whether they were listening to a male or female student. The difference between a recording performed by a male or female student was even more apparent with experienced musicians. The difference found was that "female students tended to play expressive improvisations that were often slower and had more space and rests" (McPherson G. E., 2019) whilst "Male students … tended to provide more outgoing, faster and busier improvisations than females."

When creating this evaluation algorithm, attention must be paid to make sure to minimize discriminatory judging as well as whenever necessary admitting to the bias present in the algorithm.

All these assessment techniques still heavily rely on human beings giving their opinion on improvisations, some of these assessments are able to act as guidelines to what the algorithm should be judging. However the assessments rely on human judgement to recognize an improvisation's merit and give no help to how a computer could do the same.

How can Computers Assess Music?

Before computers can assess music it first must be converted into a form they can understand. Whilst research has been done into turning raw audio into a computable data format it is quite difficult (Cihan Isikhan, 2008). Therefore, it is much easier to assess a musical performance if it is already recorded into a computable format. One of the most popular formats for this is MIDI as it does not record waveforms but instead stores a list of events of different categories such as when a key is pressed and released. This has advantages and disadvantages; a Midi file is significantly smaller in size compared to an mp3 or wav file of comparable length however it doesn't store anything about the sound and simply stores pitch, rhythm and velocity. To play midi you require something that plays sounds according to a midi file

Algorithms have been developed that analyze midi files for various parameters one such example being Zheng Jiang's Automatic Analysis of Music in Standard MIDI files .when given a midi file of a song will attempt to discern what part of the midi file is the melody of the song, what part of the midi file is the bass of the song, the chords played in the song and the structure of the song (Jian, 2019). It carries out these processes for use by researchers training music generation AI. Normally they will use prepared midi files with this information worked out by a human, this presents a significant time investment and is compounded by the large datasets needed for machine learning algorithms. Which isn't what the artefact is trying to create

Determining which part of a midi file is the melody requires analysis of many variables, seven were considered but testing found that only five were needed. These being note density, pitch mean, pitch standard deviation, Inter-Onset Interval Mean and Inter-Onset Interval standard deviation. Note density is the sum of all note durations divided by the total length of the music, this is measured because the melody is often the part that is played the most in a file. Pitch mean is the mean of all the pitches of a single channel, pitch standard deviation is the standard deviation of all the pitches, this is measured due to the assumption that the melody is normally being played with higher pitches than the bass or the chords. There is an algorithm (Alexandra L. Uitdenbogerd, 1999) that chooses the melody based on whichever channel has the highest average pitch. Inter-Onset Interval is the time that elapses between the onsets of two consecutive notes (Murphy, n.d.). Both the mean IOI and the IOI standard deviation are used in the final algorithm. For each channel each bar these five values are calculated and then put into a Bayesian Probability model to evaluate the chance that a given channel holds the melody.



The goal of the melody analysis is not useful to the algorithm; the melody will be provided by the player so we know what part is and is not the melody. However, the values used to calculate the channel of the melody could be used by the algorithm to judge improvisations. In the TAI (McPherson G. E., 2019) responses to the tasks given were marked and the reasoning for them was broken down to show what responses good and bad players gave. For example, "Low-scoring responses also displayed a limited repertoire of rhythmic devices". The repertoire of rhythmic devices could be measured by the standard deviation of the Inter Onset Interval, a small standard deviation could show a limited use of rhythmic devices, if a response was all of the same rhythm, then the standard deviation would be zero. Another characteristic of a poor response was a lack of range, players would "restrict their performance to the middle register", this could easily be measured by pitch standard deviation. Identifying bass line uses a simpler algorithm than that of melody; it only measures two things, pitch mean and argmax selection are used to select the bass part. The bass line algorithm uses similar logic to the algorithm skyline which was deemed too simple for identifying the melody (Alexandra L. Uitdenbogerd, 1999). Where Skyline determines melody based on the highest average pitch, the algorithm used to determine bass finds the channel with the lowest average pitch. This simple approach was able to accurately identify bass from a midi channel, 22 songs were tested 12 out of the 22 got 100% accuracy, 9 out of the 22 were between 90% and 100% and only 1 song was below 80%. This simplistic approach is also significantly faster with a linear time complexity as only 2 values have to be calculated each bar. If the algorithm were to accommodate using midi files for the backing over which the player would improvise, identifying the bass from that could help analysis between how the player improvises with relation to the bass line such as following the rhythm of the bass line.

Chord analysis is much more complex to do objectively because it involves human judgement and subjective decision making (Jian, 2019). Since music has non-chord tones sprinkled throughout that only suggest chords and harmonic function. The analysis of chords includes three main values for analysis, the starting and ending points of each chord, providing the root note of the chord and determining the mode of the chord. Jiang's algorithm indicates all three of these by displaying the starting and ending points of chords with the time and duration all measured in beats. The root and the mode is shown by the chord type. When determining the chord type there's only five types of chords the algorithm places a chord into; major, minor, augmented, diminished and suspended fourth. The algorithm does not make any attempt to save information to the chords about added notes such as sevenths or ninths. Jiang explains that this is because chord types are often ambiguous, he gives the example that a c13 chord may or may not contain a fifth or ninth or eleventh. Jiang believes it is just as meaningful to label the chord type as C or C7 and then provide a set of pitches used. For a jazz improvisational tool that attempts to grade jazz music where seventh and ninth chords are abundant, not including that information could be detrimental because those added notes create different harmonies which change how the melody sounds, and the scales used by jazz musicians. If the algorithm were to lack that information it could lead to incorrect scoring due to the algorithm believing the melody is being played over a triad and not a seventh, a ninth or an eleventh.

The Chord analysis of Jiang's algorithm is based on Temperley Algorithm, which uses five rules to determine a chord. First the Pitch Variance Rule: Try to label nearby pitches so that they are close together on the line of fifths. Compatibility Rule: In choosing roots for chord

spans prefer certain Tonal Pitch Class root relationships over others (Temperley, 1997). Prefer them in the following order: one, five, three, flat three, flat seventh, ornamental. Strong-Beat Rule: Prefer chord spans that start on strong beats of the meter. Harmonic Variance Rule: Prefer roots that are close to the roots of nearby segments on the line of fifths. Ornamental Dissonance Rule: An event is an ornamental dissonance if it does not have a chord-tone relationship to the chosen root. Prefer ornamental dissonances that are closely followed by an event a step or half-step away in pitch height.

Temperly's (1997) algorithm does not calculate type information, the output is only a letter such as "D" which indicates the "D chord" with no information about whether its major or minor. Jiang adds to Temperly's algorithm to determine the chord type as well as the start and end points of the chord. This is done in 3 steps, from Temperly's output the timestamp of each chord is given and the starting time and finishing time is calculated. From that the notes being played at that time can be taken and turned into a pitch class set. From the chord root from Temperly's algorithm and the pitch class set the chord type can be determined. For example, if the third and fifth exist the chord type will be assigned as major.

This method of chord analysis success rate varied drastically between songs even in the same genre. For example when tested songs named *SanFransisco* and *lovelyrose* were analyzed by the algorithm it was over 90% correct. But when the song *lovestory* was analyzed the algorithm was less than 60% correct. On average the algorithm was over 80% correct when testing songs from the pop genre. The algorithm was not tested on songs from other genres such as jazz which based upon analysis of the rules used could lead to inaccuracy when analyzing jazz.

It has to be noted that this chord analysis is for use by datasets to rapidly expand the chord labeling on songs so that AI models have more data they can be trained on. For these models whilst accuracy is definitely valued it doesn't need to be at 100% accuracy. Whereas for chord analysis to be useful for marking a given improvisation, the chords have to be completely accurate for the algorithm to give a proper analysis of it. If the player believes they're in the correct key and musically they are but the chord analysis of the algorithm says they're wrong they may begin to doubt themselves, and play with the wrong information that the algorithm making them a worse musician. The speed at which the algorithm can generate these chords is useful for the algorithm, any midi file that the player would want could be used for analysis. But the accuracy of the algorithm being as low as 60% percent on some songs in the test is not a tradeoff that can be made for the previously stated reasons.

In music, melodies have structures comprised of repetitions, transposition and reuse of melodic materials. Understanding and analyzing how musicians reuse and retool previous melodic ideas is brought up as a way of assessing students in the TAI (McPherson G. E., 2019). Computers have never truly understood these techniques. A recent paper has proposed an algorithm that could understand them. Carnovalini(2021)iterates upon existing melodic analysis tools such as Schenkerian analysis(2016) or GTTM(2013) which formats a melody into a tree representation. Carnovalini's algorithm produces trees but instead of them being the end goal they are a means to analyze the internal repetition structure.

Schenkeerian trees are created for each user chosen sized segment of the piece and then they are compared against each other, these differences are then stored in a difference tree. These trees allow computers to see repetitions or similar patterns by comparing the difference tree of two segments.

Methodology

To test the efficacy of a musical game which utilizes an algorithm to judge a player's improvisation, research artefact was created in the game engine Unreal Engine 5. This engine was chosen because of its excellent audio programming functionality including the basic building of synthesizers as well as community made plugins for MIDI playing. Other engines such as Unity need many plugins to achieve what Unreal can do without plugins.

The player is able to interact with the game through the keyboard and mouse. The mouse is only used for navigation through menus and not used in the core loop of the game, this was chosen due to its similarity with other games which would not confuse the player. The keyboard controls all inputs in the core loop of the game, there are fourteen keys on the keyboard that interact with the game. '9' when pressed raises the octave of the pitches set to the piano keys by one and '0' can be pressed to do the same but lowers the octave by one. These were placed far away from the home row because they are meant to be pressed occasionally . The twelve other keys are mapped to piano keys, seven keys on the home row of the keyboard from a to j correspond to all the white keys on a piano for one octave and the five keys on the top row. This corresponds to all the black keys on a piano form w to u excluding r. This layout was chosen because it mimics the layout of a piano by having all the white keys next to each other and maintaining the vertical displacement between white key and black by having them be on separate rows of the keyboard. Another reason this layout was chosen was because of its similarity to other applications that play music with the keyboard. An example of this can be seen used in music making applications, for







An array labeled "Note array" is used to store twelve midi values that correspond to the midi value of the piano keys. An array was chosen so that all the midi values stored could be changed quickly by the octave logic as well as being easily accessible for calculations required by the Key Change function.

When '9' or '0' is pressed the midi values stored in the "Note array" are decreased or increased respectively by one octave. This is achieved by first checking whether the octave that the player has requested to change to is above or below the limits which is octave one and seven. This is achieved by first checking whether the octave that the player has requested to change to is above or below the limits which is octave one and seven. Whilst few notes exist beyond these octaves, they are not commonly used so a limit is in place to stop players from reaching these pitches. Once the check has been passed, the Current Octave variable is subtracted or added by one to mark the change and then a for loop is used to go through the "Note array" and for every value stored adds or subtracts twelve from it to facilitate a change in octave.



Figure 4 shows of a section of a blueprint which changes the octave being played

The" Key Change" function is used to determine the notes that are held within the given scale in the given key, then notes in a given scale are marked either correct or incorrect. Two variables are passed to the "Key Change" function, the Base note which is the midi value of the note that the key is in. For example if the C major key was the key to change to twenty four could be passed through, the value for C1 in midi. The other is the scale stored in the form of an array holding the values of the scale as the difference in semitones from the root of the scale. For example, a major scale is stored as zero, two, four, five, seven, nine and eleven. This format was chosen because it is the output of the "Scale to Note Array" present in Unreal Engine's Meta Sounds. Whilst this node is not used in the artefact its format proved useful for calculating the midi values of a scale given any starting value.

The "Correct Note array" array is the end product of the function, it is an array of Booleans which stores whether a note is correct or incorrect, represented by true and false respectively. The function first sets the "Correct Note array" to hold only false values. If this wasn't done the key change function would not overwrite the previous key's information and instead repeatedly set values to be correct making the "Correct Note array" unrepresentative of the given scale in the given key.

The function then enters into a "For Each Loop" node which sends out a number of triggers for each value in an array, the array passed into this node is the scale array which needs to be traversed through to calculate the correct and incorrect notes. For each trigger sent out for this node another "For Each Loop" is started which instead uses the notes array as input. For each cycle of the Loop the value in the notes array at a given index of the loop is stored temporarily in a variable named "temp". This variable is then checked against the value stored at the current index of the earlier loop which iterates through the scale array, added to the base note which is multiplied by the current octave so that it can be checked against the note array which uses current octave multiplication. If these values are equal to each other The Correct Note array has the value stored at the same index as the index of note being checked to true. If the Temp variable is less than the scale array added to the modified base note, it has twelve added to it and is then checked using the previous



Figure 5 shows of a part of a blueprint file which determines which notes are correct and incorrect.

method.

This is all done to pre-calculate if a key pressed on the keyboard is in the correct key or not. It would be a much less complex implementation if when the key is pressed the note is checked. This would significantly increase the number of processes that take place when a key is pressed. With this method in place a keypress only checks one value in an array to see if its correct or not.

The Key Change Function is called every time a new bar is played in the backing track. This can be easily detected because the backing track is being played using a midi file which stores its rhythmic data. Unreal Engine 5 doesn't natively support playback of a midi file so an experimental plugin titled "Harmonix" is used. Featured in this plugin are the meta sounds nodes to play a midi file as well as the "MIDI Clock Subdivision Trigger" which sends out a trigger after a variety of musical note lengths. In the research artefact the node sends out a trigger every bar. This was chosen because having the key change more

frequently than that could confuse those with less musical experience. This trigger then needs to be sent to the blueprint file which stores the key change function. This functionality was only recently introduced experimentally in march of 2023 (Yen, 2023). When the trigger is sent out by the bar counter into the blueprint file an event is called which increases the blueprint's variable bar number by one, so that the file can keep track of the current bar. This recently increased variable is then checked against the "Lead Sheet" array, and if the bar number is equal to the array's length the midi file is stopped and the player is sent to the end screen. The "Lead Sheet" array is a list of numbers that correspond to the base notes of the key at a given bar. The midi file is an eleven bar blues(a 12 bar blues without the last bar) beginning on G so the lead sheet consists of 4 values of G (midi value 31), then 2 of C(midi value 24), then 2 of G, then 1 of D(midi value 26), then 1 of C, then 1 of G. When a new bar is triggered the value of the "Lead sheet" array at the index of the bar number is passed as the "Base Note" of the Key change function, changing the



Figure 6 shows part of a blueprint file which counts the bar numbers and changes the key accordingly.

key whenever the key of the backing track changes.

In the code, when a key is pressed it calls an "On key Press" function and when it is released it calls an "On key Release" function, each different keyboard key has its own number associated with it called its index.



Figure 7 Shows keyboard events triggering the "On Key Press" and "On Key Release" functions

The On Key Press function is responsible for determining the pitch played, it does this by getting a midi note value from the notes array stored at the index that was passed with the function. The code then passes the retrieved value to the meta sounds file entitled korgm10, this value is then passed to the oscillators in the meta sounds file which converts it from the midi value to a frequency. After the frequency has been set the envelope is activated, which begins to play the sounds produced by the oscillators. Then the correct note check happens which checks the given index with the correct note array. This can be done because the correct note array and the notes array are the same length. If the value stored at the index is false then the incorrect note counter goes up by one and if the value stored at the index is true then the correct note counter goes up by one. After this the color of the piano key that corresponds to the keyboard key pressed is darkened slightly indicating that it has been pressed.



Figure 8 shows the "On Key Pressed" function

The On Key Release function is responsible for sending the "Trigger Release" trigger to the meta sounds file "korgm10". This triggers the envelopes trigger release which begins to stop the note playing by turning down its volume. After this the function then brightens the darkened key. Indicating that the key has been released



Testing will be conducted on this artefact using surveys answered by testers to gauge their opinion on the artefact and in turn the viability of a musical game. These surveys will feature both quantitative and qualitative questions. They will ask specifically about the tester's improvisational ability, their enjoyment of the game, any features they would like to see added and whether they experienced any bugs or glitches whilst testing.

The tests will be one on one and will begin with the basics of the artefact and its functions being explained to the tester. Then the tester will play through the artefact multiple times, and at any time during this they will be able to ask questions about the artefact. After this the tester will answer the survey and submit their response anonymously through Microsoft forms.

Five people have agreed to test the artefact and answer the survey. These five people have been chosen due to their range of improvisational abilities. Which will help inform whether or not the musical game has appeal to both musicians and non-musicians. Due to the need for live guidance whilst testers play the artefact more people could not be selected.

First Iteration

Results and findings

The First Iteration of the build had one tester, who answered a survey with both qualitive and quantitative responses. The first question the sole tester was asked was to 'describe their improvisational ability where one was wholly incompetent and ten was a virtuosic performer', put a two. This contextualizes the answers to the rest of the questionnaire by showing they are being answered by someone who considers themselves unfamiliar with musical improvisation. The questionnaire featured questions that were not applicable to the iteration and so some questions were excluded by the tester. On the third question about the tester's opinion on the score of the pitch calculation the tester answered three out of ten and described that they were able to increase their score drastically at the beginning by repeatedly pressing the keys in quick succession that they described as "didn't sound good at all". Also they described that after time had passed playing the prototype no matter what they pressed they never got a correct note, even when it seemed correct to them. The tester's answer to the fourth question "did you feel the score given by the algorithm on your improvisational techniques such as repetition was correct one to ten" was a three and they elaborated by saying they found no penalty for reputation The tester's answer to the fifth question "Did you utilize the info screen yes or no?" was yes. The tester's answer to the sixth question "Did you use any options that would make the improvisation easier yes or no?" was no. The tester's answer to the seventh question "Would you like to see more options to make improvisation easier yes or no?" was yes, and expanded by saying they would like more feedback about when the keys were pressed as well as information about the current octave that the piano is in. To the eighth question of the survey "Rate your enjoyment of the game from one to ten" the tester responded an eight and went on to say that "it's a fun experience to try and improve at, given I have no real musical ability". To the ninth question which asked, "Are there any features you would like to see added that would improve the game?" the tester reiterated their answer to question seven bringing up the unresponsiveness of the keys and how they should indicate that they were being pressed. They also suggested that the keys that were incorrect shouldn't activate when pressed by the player.

Changes were made in response to the sole tester's feedback. A system for darkening the keys when pressed was developed. A readout of the current octave that the keyboard is playing in was implemented. The problems described by the player with incorrect pitches calculation have been fixed as well as the issue with the player being unable to score any points later in the test.

Discussion and analysis

The sole tester described themselves as not particularly adept at improvisation, and so their input is very valuable as it represents the perspective of a non-musician. They indicated that they felt that the algorithm for calculating whether a pitch was correct was unsuited to the goals of the artefact. This is a problem that someone with little musical experience thinks that the algorithm does not perform its function and needs to be fixed if the artefact is to work correctly. The second problem the tester described was that there was no penalty for repetition. This iteration allowed for a player to press correct notes repeatedly with no regard for rhythm or generating new musical ideas and give them the same score as if they did. If this artefact is trying to be a musical game, musicality must be rewarded. The tester used the information on the screen such as the correct note counter, showing that it could possibly have use to a wider player base. Overall, the tester rated their enjoyment of the game an eight out of ten, meaning that even with these major flaws in the first iteration that fun can be found even by the unmusical shows that with fixes done to the problem that this artefact could be made even more fun.

Four changes were made because of the tester's feedback. The first and most simple change to implement was the addition of a current octave readout to the UI. All this required was an addition of a text box to the UI and a bind for the value within the text box to the current octave variable in the blueprint, which would write the value of the current octave to that of the text box thereby showing it to the player. The second change implemented because of the tester's feedback was darkening the piano keys on screen when pressed. This was done to aid responsiveness by giving visual feedback to the player when a key was pressed instead of just audio feedback. This functionality is also present in other keyboard based piano input systems such as the previously mentioned Logic and Garageband DAWs, where a key is slightly darkened when pressed. The implementation of this feature helped the artefact become more in line with other systems like it.



The other changes were to solve the problems that the tester described with the artefact. The first problem described was that in the beginning of the game the player could increase their score drastically by hitting two keys repeatedly that didn't sound correct. This was because the midi file's opening bars had a different key to the lead sheet and so keys that sounded wrong were perceived as correct by the game. This was fixed by changing the lead sheet values to better reflect the song's key throughout its runtime. The second problem faced by the tester was that later into the test of the game, no matter what they pressed they could not play a correct note. This was because the bar number which is used to access a lead sheet value from the array was too high leading to out of bound errors when the array was trying to be accessed. This led to the key change function writing the correct note array as all false meaning that whatever the tester pressed none of it was correct by the game's logic.

Second Iteration

Results and Findings

The second survey, conducted on the second iteration, had five testers which was significantly more than the first. These testers varied greatly in musical skill. The first question of the survey "How would you describe your improvisational ability from a scale of one to ten (one being wholly incompetent and ten being a virtuosic performer) had an average rating of four and eight tenths.

4.80 Average Rating 1 2 3 4 5 6 7 8 9 10
Level 10
Level 9 1
Level 8
Level 7 Level 7
Level 6
Level 5
Level 4 1
Level 3
Level 2 2
Level 1
Figure 11 shows the results of the first question

Testers answered the second question "Did you feel the algorithm could correctly tell if a pitch played was correct or incorrect one to ten." With an average rating of nine and two tenths.



This average can be broken down into two groups: one group is made up of the respondents who put themselves as higher than five to the first question and the other group is made up of the respondents who put themselves as lower or equal to five. The average rating from the musicians was nine and the average rating from the non-musicians was nine and one third. The fourth question was an optional one asking the testers' reasoning for their answer to the third question. All testers gave small explanations of their reasoning. Four out of five testers explained that it could tell the pitches correctly. One tester put "I added a few a accidentals which I felt went with the jazz vibe, and I think they were marked as wrong. A bit inflexible maybe." The fourth question asked the testers "Did you utilize the info on screen yes or no?". The testers unanimously answered yes, all of them utilized the info one way or another.



The fifth question asked the testers "Did you use the correct note overlay". Four of the testers used the correct note overlay whilst one did not. The one tester who answered no, rated their musical ability a seven out of ten.



The sixth question asked the testers who answered yes to the previous question whether they found it useful. Two testers answered this question with just a yes whilst the two others, both of which answered two to the question on musical skill, described having trouble understanding the information the overlay provided. One tester put "it was certainly useful, however as someone with no musical experience I could not intuit the logic behind why certain notes were highlighted or not." The other answered "I was a little confused going into it because I thought the ones which were coloured differently were the correct notes so I just spammed f and t. But yes once I knew how it worked."

The seventh question asked, "Would you like to see more options to make playing easier yes or no?" Four testers answered yes and one answered no. The one who answered no put a nine for the first question.



The eighth question asked those who answered yes to the previous question, what they would like to see added to make playing easier. Different responses came from the four testers, those which rated their improvisational ability at a two, echoed similar suggestions of wanted more explanation given to the player. One of these two also suggested a reworking of the analysis methods stating "a "correct" point right now is just pressing notes that aren't incorrect, it would be good to reward good improv vs spamming "correct keys".". Another tester who rated their improvisational ability at a four answered with "Metronome and/or tuning noted to nearest beat/fraction of beat". Another tester who rated their improvisational ability at a seven answered "more octaves visible / easier to switch between the octaves. Would be nice to see the letters of the notes on the scale".

The ninth question "Rate your enjoyment of the game from one to ten" was met with a wide array of values, but the average remained positive with an average of six and six tenths. The difference between average ratings of the musically skilled group and not musically skilled group was quite large, the musically skilled group averaged a rating of eight whilst the not musically skilled group averaged a rating of five and two thirds. However, a small sample size for both groups could mean that this value could change with more testers.



The tenth question asked testers to explain their reasoning for their answer to the previous question. The first tester, who put a low score for their improvisational ability, stated "Good concept but right now I'm not sure what to do beyond press correct keys. As someone who doesn't improvise music I'm not sure where to go from there." The second tester, who put a high score for their improvisational ability, stated "It was fun, it would do well from more sustain on the notes." The third tester , who put a middling score for their improvisational ability, stated "Listening to the tunes was nice." The fourth tester, who put a high score for their improvise in". The fifth tester, who put a low score for their improvisational ability, stated "It was fun. It felt like a safe space to improvise in". The fifth tester, who put a low score for their improvisational ability, stated "I scored highly despite believing that my improvisation was extremely rudimentary, there was not a great deal of depth in the musical assessment"

The eleventh question asked testers if there were any features that they would like to see added that would improve the game. All of them had features they would like to see implemented. The first tester requested a myriad of small features. They would like if the overlay was on or off was easier to see on the menu possibly using some highlighting on the selected option, the option to quit a playthrough with the music once started and a tutorial section to accommodate non-musical people. The second tester would like to see two octaves playable at once. The third tester requested an indicator for if the overlay is on or off in the menu. The fourth tester repeated their answer to the eighth question and suggested a feature for the player to manually switch keys. The fifth tester "would like to see some other method of assessment implemented. Perhaps, some manner of rhythm assessment would improve the player experience."

The twelfth question asked testers if they encountered any bugs in their testing. No testers reported any bugs.

Discussion and analysis

The most impactful difference between the last survey and this one is the number of testers, there are now five, this provides many more experiences with the artefact to draw insight from. It is not a large enough sample size to gauge public opinion on musical games as a concept, but it is enough to gather feedback for the purpose of iteration and addition of new features.

The testers' musical skill varies greatly. Their responses to the question "How would you describe your improvisational ability from a scale of 1 to 10(1 being wholly incompetent and 10 being a virtuosic performer)?" averaged four and eight tenths. This variety is significantly improved from the first test which featured only one person, who answered the same question with a two. This spread of musical skill allows the survey to hear one demographic not previously heard, that of the skilled musician. It is important to gather skilled musicians' opinion on the artefact as they are an audience likely to be interested in a musical. Another reason to gather their feedback is that they can more confidently critique the algorithms behind the artefact. For example the question asking the testers how correct they think algorithm was when regarding pitch, musicians and non-musicians averaged very similar numbers with their responses. But the language used when answering the follow-up question which asked them to explain their answer to the previous question varied between musician and non-musician. The tester who answered a nine to the first question put a ten to the second question and explained it by saying "It could tell perfectly". One tester who answered a two to the first question also put ten to the second question and explained it by saying "To my understanding, every note that I played was assessed correctly." This difference in language used shows the lack of confidence on the part of the non-musician.

All testers, no matter their musical skill, answered that they utilized the information on the screen. This shows the information show on screen by the artefact is currently useful to players.

To question five, four answered that they did use the correct note overlay whilst one did not. The one tester who answered no answered seven to the first question, this could suggest a possible connection between musical skill and lack of need for a correct note overlay but with such a small sample size of testers this is hard to say definitively.

Question six asked those four who had answered yes to the previous question whether or not they found the correct note overlay useful. The testers who answered a score higher than three to the first question simply answered with a yes. Those who answered lower than three to the first question described having issues with intuiting the meaning behind the correct note overlay. The fifth tester answered, "It was certainly useful, however as someone with no musical experience I could not intuit the logic behind why certain notes were highlighted or not" and the first tester answered "I was a little confused going into it because I thought the ones which were coloured differently were the correct notes so i just spammed f and t. But yes once I knew how it worked". These answers show that the two testers struggled with understanding what the correct note overlay meant. This may show that the correct note overlay requires an amount of musical knowledge to understand. This could be because the idea of correct and incorrect notes may be not understood by nonmusicians, this is backed up by the first tester's experience with believing the notes coloured differently the correct ones. Another reason the correct note overlay may require musical knowledge to intuit is that it is over a piano, utilizing a musical instrument for a user interface may confuse non musicians due to their unfamiliarity with a musical instrument.

Question seven asked the testers if they would like to see more options to make playing easier yes or no, four answered yes and one answered no. The one tester who answered no, answered the first question with a nine. This may show some link that with enough improvisational skill no more options other than the correct note overlay are needed. However there is not a large enough sample size to prove this.

Question eight asked the testers who answered yes to the previous question, what features would you like to see be added. One tester suggested that more octaves visible on the keyboard and to make it easier to switch between the octaves. These suggested changes could be implemented in a way that would bring the implementation of keyboard input to piano more inline with other similar implementations like the Musical Typing software used in apple's logic and garage band digital audio work stations. As seen in figure three.

Using this design would allow more notes to be played without use of shifting octaves and help to alleviate the tester's problem by having more keys on the keyboard that play a pitch. This design also features octave shifting but the buttons to do that are significantly closer to where the keys to play the piano are. The artefact has the octave shift buttons on the '9' and '0' keys, much further away than the z and x keys this program uses. Both testers who answered two to the question about musical skill, both suggest that more guidance and explanation is required for the system. This shows that currently the artefact does not explain enough about how it works to those who don't already have some musical knowledge and more needs to be done if people who lack musical skill are to have fun with the artefact.

The ninth question asks testers to rate their enjoyment of the game from one to ten. The average rating is six and six tenths. However, when broken down between by general musical skill, the non-musicians ratings averaged five and two thirds whilst the musicians' ratings averaged eight. This gap in average ratings could be due to the previously discussed complaints of lack of guidance for non-musical people. Part of this gap may also be that the non-musical group did not enjoy simply playing music within the artefact. The first tester answered the tenth question, which asks testers to explain their answer to the ninth question, with "Good concept but right now I'm not sure what to do beyond press correct keys. As someone who doesnt improvise music im not sure where to go from there." This contrasts with the fourth tester's answer "It was fun. It felt like a safe space to improvise in." These contrasting experiences with the artefact provide evidence that the nonmusically skilled testers require extrinsic motivation to have fun with the game whilst the musically skilled testers enjoy playing music enough that they are intrinsically motivated to play the game. The average enjoyment is down from the first iteration even though more features have been added and bugs removed. This Difference could however be due to the small sample size of both tests and if both tests were done with more people a different result would result would be reached.

The twelfth question asks if the testers encountered any bugs. All of them answered no. This is sizeable improvement from the first iteration where the tester described two bugs relating to the algorithm algorithm's accuracy at different points in time.

From the qualitative responses in the questionnaire, four features have been identified that have been suggested by Two or more testers. Two testers have suggested that the main menu should indicate whether or not the correct note overlay will be on when playing. Two testers have suggested that more notes be accessible when playing. Two testers in multiple responses have indicated that there is a lack on analysis methods and more should be introduced. Two testers in multiple responses have indicated that there needs to be more information to introduce non-musical players to the concept of the game.

Conclusion

The primary discovery of the paper is that there is a possibility that a musical game could be enjoyable to the public. This is suggested by the responses on the questionnaire about the player's enjoyment with the game. However this metric is not evidenced due to the lack of statistical significance. The paper has also suggested that musical games are more enjoyable for the musically skilled and efforts are needed to onboard players with less musical skill.

Another discovery is the distinct lack of musical games created. Most games that have musical elements rely solely on rhythm for use in gameplay. Even when controllers that are designed in the image of musical instruments are bundled with the video game, it still only tests the player's rhythm.

Aims and Objectives

The aim of this paper is to analyze and investigate the viability of the research artefact that tested a player's overall musical skill rather than just measuring a single aspect such as rhythm. The viability of a musical game was investigated through research into current improvisational assessments, rhythm games and computational musical analysis. Also, through the creation of an artefact which resembled a musical game and then investigated further by testing this game by having people play it and answer a questionnaire about their experience with it.

The literature review researches the three areas outlined in the objective in varying levels of detail. Rhythm games and their history is researched in the literature review but of greater focus to the paper is the psychological effects of certain rhythm games which attempt to emulate the feeling of playing music. In this research there is shown a distinct lack of games which test musical skills. Improvisational assessments are successfully researched in the literature review to aid in the creation of the algorithm that powers the musical game. Two assessment techniques are analyzed in depth and then compared against each other to attempt to find an assessment technique viable for use in a real time scenario. TAI was found to be better suited than the CAT than the other due to its use of marking rubrics and so was chosen as a basis for the musical game. Computational musical analysis is researched to find how computers analyze music so that they could then be retooled for use in the game. This research found a lack of fully fledged analysis tools for improvisation but there were tools for different uses of musical analysis. Those tools used variables and

processes that could be helpful in creating an algorithm for judging musical improvisations.

An artefact which tests musical skill in a way other than rhythm was created. A research artefact was created where players could improvise utilizing the computer keyboard over a jazz backing track made up of a double bass and a piano. After two loops of the backing track the game would end and how many correct notes and incorrect notes were played by the player would be displayed on screen.

This research artefact was then tested by five people of varying musical skill levels. They were given a questionnaire made up of both qualitative and quantitative questions about their experience with the game as well as features they would like to see in the game. This survey was successful in gauging the opinion and possible enjoyment of a musical game as well as getting suggestions and feedback which could be used to improve it.

Limitations

The primary limitation of this paper is the artefact and its lack of analytical capabilities outside of pitch. Research into other methods of musical analysis methods for rhythm and structure techniques such as repetition was conducted but never implemented. These missing features make it harder to test the viability of a musical game because the artefact only tests one feature and not all the skills required to improvise music. A lot of development time was spent on the UI features and game Input of the game instead of the evaluation algorithm. No midi file creation was developed which led to a lack of analysis methods being produced because most of the analysis methods identified in the literature review analyzed midi files. This project should've had a smaller scope, an artefact should've been considered which only developed the algorithm to analyze midi files and then tasked players with using some other method with which to generate midi files such as recording a midi file using a DAW and then exporting that file for analysis.

Another limitation of the paper is the small sample size of testers. Whilst five was a large enough sample size to gather suggestions for features that need to be added to improve player experience, it is not large enough to determine the public's overall opinion on musical games and in turn their viability. A larger sample size is required to gauge overall opinion. A limit on the number of tests that could take place was due to the need for live guidance from the testers due to the unpolished nature of the artefact and its lack of instruction. If instruction could be provided by the game and the need for live guidance removed the artefact and questionnaire could be sent out and answered by thousands of people which would better grasp public opinion on a musical game.

Recommendation 500 words

Further research could investigate more computational musical analysis techniques. Late into research a large book entitled Computational Music Analysis(2016) was found .A cursory glance through the book's contents shows that the book covers multiple analysis methods in detail such as Chord analysis, Pitch Class sets and Musical Form analysis. One issue however is that none of these are marked as improvisation specifically and so further research must be done on their suitability for real time applications.

Another field that would be further researched would be AI music generation. Within many ai music generation systems, there exists internal evaluation structures which determine what note should follow another, this system could be retooled for use in game by using it to instead evaluate a player's improvisation. Suitability for real time applications in relation to processing time and computer power needed would require research before AI could be considered for use in a future artefact.

Further research could be done into game engines and audio middleware applications with regard to their compatibility with the midi file. Whilst research into Unity and Unreal Engine's compatibility with the midi file format was done , none was done into Godot's compatibility and no research at all went into audio middleware applications such as Wwise or Fmod. Within these applications could be tools that could create midi files from player input. This is useful because the analysis methods covered in the literature review conduct their analysis predominantly on midi files which the artefact could not create due to the lack of functionality.

Bibliography

- Alexandra L. Uitdenbogerd, J. Z. (1999). Melodic matching techniques for large music databases. *ACM International Conference on Multimedia*. Orlando: MULTIMEDIA.
- Almeida, L., Prieto, D., Ferrando, M., Oliveira, E. P., & Ferrándiz, C. (2008). Torrance Test of Creative Thinking: The question of its construct validity. *Thinking Skills and Creativity 3*, 53-58.
- Amabile, T. M. (1996). *Creativity In Context: Update to the social psychology of creativy.* Boulder: Westview.

Baker, D. (1969). Jazz improvisation. Chicago: Maher Publications.

Barthes, R. (1977). Musica Practica. In Image Music Text (p. 149). New York: Hill & Wang.

- Beaty, R. E., Smeekens, B. A., Silvia, P. J., Hodges, D. A., & Kane, M. J. (2013). A first look at the role of domain-general cognitive and creative abilities in jazz improvisation. *Psychomusicology*, 262-268.
- Cihan Isikhan, G. O. (2008). A Survey of Melody Extraction techniques for Music Information Retrieval. *Interdisciplinary Musicology*. Thessaloniki: C. Tsougras, R. Parncutt.
- Eisenberg, J., & Thompson, F. W. (2003). A matter of taste: Evaluating improvised music. *Creativity Research Journal*, 287-296.
- Filippo, C., Antonio, R., Nicholas, H., Steven T., H., & Geraint, A. W. (2021). A New Corpus for Computational Music Research and A Novel Method for Musical Structure Analysis. *Audio Mostly* (pp. 264-267). New York: Association for Computing Machinery.
- Flohr, J. (1979). Musical Improvisation behavior of young children. Urbana-Champaign, Illinois, United States of America: University of Illinois.
- Gorder, W. D. (1980). Divergent production abilities as constructs of musical creativity. *Journal of Research in Music Education*, v28 n1 p34-42 .
- Healy, D. (2016). The Influence of Technical Proficiency On Clarinet and iPad Improvisation Achievement . Retrieved from ProQust Dissertations & Theses Global: https://www.proquest.com/pqdtglobal/docview/1794655880/abstract/B773334551 8F49FAPQ/1
- Hickey, M., Healy, D., & Schmidt, C. (2021). A quantitative analysis of two improvisation assessment instruments. *Psychology of Music*, 1-12.
- Jian, Z. (2019). *Automatic Analysis of Music in Standard MIDI File3s*. Pittsburgh: Carnegie Mellon University.
- Marsden, A., Hirata, K., & Tojo, S. (2013). Towards computable procedures for deriving tree structures in music: Context dependency in GTTM and Schenkerian theory. *Sound and Music Computing Conference* (pp. 360-367). Stockholm: KTH Royal Institute of Technology.
- Masatoshi, H. K., & Staoshi, T. (2016). Implementing methods for analysing music based on lerdahl and jackendoff's generative theory of tonal music. In D. Meredith, *Computational Music Analysis* (pp. 221-249). New York: Springer.

- McPherson, G. (1993). Evaluating improvisational ability of high school instrumentalists. Bulletin of the Council for Research in Music Education (pp. 11-20). Champaign: University of Illinois Press.
- McPherson, G. E. (2019). Developing Tests of Music Performance Improvisation. In T. S. (ed.), *The Oxford Handbook of Assessment Policy and Practice in Music Education, Volume 1* (p.). Oxford: Oxford University Press.
- Meredith, D. (2016). Computational Music Analysis. New York: Springer.
- Miller, K. (2012). How Musical Is Gutiar Hero. In *Playing Along: Digital Games, YouTube, and Virtual Performance* (pp. 111- 158). Oxford, New York: Oxford University Press.
- Temperley, D. (1997). An algorithm for harmonic analysis. *Music Perception: An Interdisciplinary Journal*, 31-68.
- Torrance, E. P. (1974). Torrance Tests of Creative Thinking. Lexington: Ginn and Company.
- Vaughan, M. M. (1971). *Music as model and metaphor in the cultivation and measure-ment of creative behavior in children*. University of Georgia, Athens, GA: M.M. Vaughan, 1971.
- Webster, P. R. (1994). *Measure of Creative Thinking in Music: (MCTM-II) : Administrative Guidelines*. Chicago: Northwestern University.
- Yen, G. (2023, March). Unreal Engine Public Roadmanp. Retrieved from MetaSound Output Watching(Experimental: https://portal.productboard.com/epicgames/1-unrealengine-public-roadmap/c/1022-metasound-output-watching-experimental