Adaptive Music in Narrative Videogames based on individual relationships

Miguel André Galinha Camões Cunha

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Rui Filipe Fernandes Prada

Examination Committee

Chairperson: Prof. Alberto Manuel Rodrigues da Silva
Supervisor: Prof. Rui Filipe Fernandes Prada
Member of the Committee: Prof. Carlos António Roque Martinho

January 2021

Acknowledgments

I would like to acknowledge my dissertation supervisor Prof. Rui Prada for his insight, support and sharing of knowledge that has made this Thesis possible. Also, I appreciate and would like to thank Prof. Carlos Martinho as well for the advises and helpful information given to help this Thesis succeed. Both of them have been an inspiration to my creativity and hard work, the wish to be able to create exciting and innovative games is now well lit in my mind, thanks to you.

A great shout of appreciation to the team at Conservatório Regional de Setúbal, Prof. António Lacerda and fellow colleague Tiago Ribeiro, for their own hard work, our joint sessions and their cheerful nature throughout the entire research. I thank you both very much, and hope that our collaboration does not end here, as I can see we can accomplish great things together.

To the companies I've been working in since the beginning of my Masters degree course, I thank you for all the support and help, giving me solutions to organize my time between work and the course. Without your understanding and help, I do not think I could've gotten this far. Thank you.

To all my colleagues in the course, thank you. Particularly, I would like to give a big thank you to Carolina, Lourenço and Susana, for being a great inspiration, motivating me to do my best and being great friends. This college experience would not be the same if it weren't for you three.

To my parents and family, thank you for all the support and believing in me. Through my many times of fear and joy, you were always there to cheer me on, no matter what, my entire life. And the little ones, my nephews, who have been a great inspiration to me and have pumped me into creating awesome games.

Last but not least, to all my friends that helped me grow as a person and were always there for me during the good and bad times in my life and this hard year of 2020. In one last remark, I would like to give a special thank you to Mariana, who from the beginning was the biggest cheerleader and motivator, dealt with a lot of my frustrations and hardships along the way and never stopped believing in me.

To each and every one of you – Thank you.

Abstract

Soundtrack that is dynamic in response to certain triggers or events is known as Adaptive Soundtrack, and this technique holds immense study and production interest as it enriches the gaming experience and makes games more appealing. Usually, the methods used trigger the adaptation with some altered state of the moment in the game.

This study is dedicated to the development of an algorithm for adapting a narrative style game's soundtrack based on game-long built relationships, by having the characters on scene or opening dialogues about other characters. With this, we intended to understand if such a technique can improve the gaming experience. The changes in the soundtrack should help the player progress through the game by making him aware of, not only the initial but also the evolution throughout the game, of the relationships' states.

For this, we developed a plugin in a game engine and created a small game like scenario for participants to play. The player's objective is to figure out the answer to two questions, for which the player must understand the state of the relationships between the characters through the alterations in the background music.

Though we were able to evidence some influence of the technique on the player's gaming experience, players seemed to still struggle to understand the state of the relationships through only the adaptations in the soundtrack. This could be due to the implementation done in this research not being the most effective, not invalidating this technique as worthy of more exploration.

Keywords

Adaptive soundtrack; Music in video games; Narrative games; Characters' social relationships

Resumo

Música que é dinâmica em resposta a certos parâmetros ou eventos é conhecida como Música Adaptativa, e esta técnica tem grande interesse de estudo e produção devido ao quanto enriquece a experiência dos jogos e os torna mais apelativos.

Este estudo dedica-se ao desenvolvimento de um algoritmo para criar adaptações na música, em jogos de estilo narrativo, baseadas no estado das relações entre diferentes personagens ao serem mencionados em diálogo com outros personagens. As alterações na música deverão ajudar o jogador a progredir ao dar-lhe noção, não só do estado inicial como da evolução das relações sociais entre os personagens.

O nosso objectivo é entender como a experiência de jogo pode ser influenciada com esta técnica e se as alterações na música ajudam o jogador a compreender o estado das relações e como estas são afetadas.

Assim, desenvolvemos um plugin de motor de jogo que implementa o algoritmo e criámos um cenário jogável. O objectivo no jogo é descobrir uma palavra-chave construída pelas respostas a duas questões, e para averiguar estas respostas, o jogador terá de compreender o estado das relações entre diferentes personagens com base nas alterações na música.

Embora tenhamos conseguido evidenciar alguma influência desta técnica na experiência dos jogadores, estes parecem ainda ter tido de se esforçar muito para entender o estado das relações apenas através da alterações da música. Poderá ter a ver com a implementação feita neste estudo não ser a mais eficaz, não invalidando que esta técnica é digna de maior exploração futura.

Palavras Chave

Música adaptativa; Música de videojogos; Jogos de narrativa; Relações sociais entre personagens

Contents

List of Figures

List of Tables

Listings

Acronyms

AIFF	Audio Interchange File Format
ANOVA	Analysis of Variance
API	Application Programming Interface
CRS	Conservatório Regional de Setúbal
GUESS	Game User Experience Satisfaction Scale
PC	Personal Computer
RPG	Role Playing Games
UI	User Interface
VAME	Verbal Attribute Magnitude Estimation

1

Introduction

Adaptive Soundtrack is a technique that makes soundtrack dynamic in response to certain triggers or events, only truly achievable in the Video game Industry [3]. Many video game companies have implemented some form of Adaptive Soundtrack in their games in order to captivate or impress the player. Usually, the methods used will trigger the adaptation with some altered state of the moment, such as the character's position [4], the character's health going below a certain percentage or the number of enemies increasing in a battle type of game.

Although it is now fairly common to see games that implement adaptive soundtrack, there's hardly a game that triggers the adaptation based on historical states or variables that are modified throughout the game's play run. Of course, not all games are able to easily employ these states as inputs for the adaptive soundtrack due to their own nature, especially arcade style games with quick playthroughs or irrelevancy of information between levels. But considering Role Playing Games (RPG), for example, that are usually rich in narrative content, the relationship between the different characters in the game can evolve with said narratives, bringing a more dynamic sense to the emotions in the dialogues. Using these relationship state changes as an input to an adaptive soundtrack algorithm can greatly impact the player's game experience.

Not only that, but adaptive soundtrack could play a bigger role in this scenario, making the player aware of the state of those variables at any given time.

Seeing as how adaptive soundtrack is becoming such an important and immersive feature in games, studying it and ways to use it further could open paths to creating even more beautiful and exciting games. During a previous semester, as part of Game Development courses, we developed a detective RPG game called "A Detective Story"[1]. During development and story design, we felt that, for investigation purposes, the interrogated character's emotional state and how he feels about the interrogator could be an important factor (though this mechanic is not yet included).

To make it more interesting, we thought about using adaptive soundtrack to describe these states, which could actually make the player understand if he's going in the right course in terms of the different characters' feelings with each other.

With this in mind, we intend to develop this feature in the game and have adaptive music that changes according to the relationship state between characters we interact with.

[1] A Detective Story - https://www.adetectivestory.com

2

Background

Music and Sounds are very important forms of art in all media. They give emphasis to certain points of interest, amplify intensity of different scenes or situations and can actually be used to influence various emotions on the audience, such as happiness or fear[1] [5] [6]. Even silent films used to be accompanied by music, to highlight the occurrences or events in the scene. Musical scores may also be interpreted as stories, thus bringing to the surface emotions on the audience.

Video games also adopt very intensively the art of music, retaining the same benefits and properties discussed before [7]. Game soundtracks commonly become so iconic that many people can actually identify the game itself easily only by hearing said tracks. Everyone knows the musical tune from, for example, the Super Mario Bros.[2] series or The Legend of Zelda[3] series.

The soundtrack in video games can have all kinds of purposes, apart from being one of the fundamental creativity game facets[4] [8]. It can be used to motivate the player for an upcoming event, give a notion of the ambience to the player (e.g. cheerful music in lively towns) or alert/hint to some particular object[5] or objective. It becomes a feature that gameplay mechanics can use, so important that many games become too difficult or completely different in the absence of sound.

There are also certain types of bonds between musical properties and game scenario or ambience. For example, for situations with high tension like exploring an eerie cave or battling a very difficult enemy, some melodies introduce dissonant tones or harsh timbres to go along with the scenario. When meeting a comical character or reaching a happy status like winning a battle or race, the pitch of the melody or pace is usually a bit higher, in sort of a rejoicing feeling.

The way the games implement reproducing the soundtrack can vary between mainly two choices: static and adaptive.

Static soundtrack, as the name implies, is reproduced in a non-variable way. This means that the trigger that sets the soundtrack to play at a certain game time frame is always the same between playthroughs[6]. A fine example of this technique is the transitions between locations or situations in RPG games. Entering a new town, going to the world map, entering a battle, all these scenarios will have their own soundtrack melody, set to play by the transition from one scenario to the other.

Adaptive soundtrack is reproduced differently based on what the player is doing or the situation in the game he's in. It could be due to something like a health or a timer parameter, where, while the melody might remain the same, the execution of the melody could become more aggressive or intense, for example, the lower the character's health or the closer to the end the timer is. By execution, we mean

[1] The Sound of Fear
https://www.youtube.com/watch?v=HQQmFocLDng
[2] Super Mario Bros. theme
https://www.youtube.com/watch?v=NTa6Xbzfq1U
[3] The Legend of Zelda theme
https://www.youtube.com/watch?v=cGufy1PAeTU
[4] The six creativity facets identified in games being: visuals, audio, narrative, ludus, level architecture and gameplay
[5] Tomb Raider - Secret found sound
https://www.youtube.com/watch?v=d8eskuLLkk4
[6] Playthrough is the act of playing a game from beginning to the end

the type or number of instruments being played, the pace of the music or the volume of the notes.

Adapting soundtrack by changing the musical properties' values and maintaining the melody is called vertical mixed adaptive soundtrack. Examples of games that became notorious making use of a vertical mixing are Banjo Kazooie by Rare, Mario Kart 8 by Nintendo and Metal Gear Rising: Revengeance by Platinum Games. Banjo Kazooie makes alterations to the melody based on the situation or environment, like approaching the area where a boss is or going underwater. Mario Kart 8 adapts the racing track's melody based on various different elements, such as being in first place, running the final lap, being affected by items, and many more. Metal Gear Rising's best example of adaptive soundtrack is the boss battles, where the song plays along in an instrumental version and as soon as you reach a certain point in the battle (usually when you're about to win), the voice track starts playing.

Another kind of adaptive soundtrack is the horizontal mixing, where the melody can be different but the change of the variable determines also the time translation of the destination melody. A game series that implements this technique in a simple but effective way is the Metal Gear Solid franchise by Konami, where switching through different states of alert alternates the soundtrack being played. As an example, Metal Gear Solid 2: Sons of Liberty uses an horizontal mixing[7], where for the different states (Alert or Evasion), different tunes play with minor similarities between them, such as a background beat or the pacing. Every time you enter a different state, just a small fade-out/fade-in effect occurs to change the tune, but the current position in the track time remains.

There are many advantages of using adaptive soundtrack, with the main being that it can create a more immersive experience (static soundtrack tends to become repetitive with play time) and it can also be a marker to a game variable state, leading the player to be aware of that state without any visual indication required. In the case of the Metal Gear Solid series, the game could simply not have the alert state visual indicator and the player could understand if it's character was being seen or had been lost by the enemy simply from hearing the music.

One of this work's objectives is to use adaptive soundtrack, taking the advantages above into account, in a narrative style video game. Many Narrative style video games fit into the RPG (Role Playing Game) genre and usually follow a long complex plot throughout the course of the game. The main characteristic of these types of games is, as the name implies, the dialogues and interactions with other characters. These gameplay mechanics not only provide the player with lore and context of the game's world, but can also be used to solve puzzles and progress through the story. There are games that can actually have this as the main mechanic. Many Investigation style type of RPGs use dialogue as a main mechanic. To progress through the story, the player must interact with characters in a dialogue fashion to understand the plot and gather clues. An example of such a game series is the Ace Attorney games by Capcom, which are visual novels depicting different murder cases where the player, an attorney-at-law, must

[7] Metal Gear Solid 2 Adaptive Music
https://www.youtube.com/watch?v=wPISB1-d7bw

establish dialogues and collect evidence to prove his client innocent in court. By talking to different people and choosing dialogue options, the player can get closer to the truth and progress through the story.

But the way these dialogue options appear are usually very static, with simple triggers to enable them. The player will not be presented the choice to ask where the character knows someone from before selecting the option to ask if the character knows that someone in the first place. Adding another layer of complexity, we could use what is called a relationship state algotrithm, that influences the dialog choices available or the flow of the game based on actions performed along the game that affect certain characters. These relationship states are usually based on some kind of social relation variable or even a combination of such variables, such as love/hate, trust, tension, cooperativeness, competitiveness or dominance. Some RPG games implement a relationship mechanism that maintains throughout the course of the story a state of the relationship with the different characters. Final Fantasy VII by Square Enix makes use of a relationship state algorithm to, for example, determine who will attend a cable car ride with the protagonist based on a valence type of variable. This variable is altered by multiple factors such as choosing dialogue options that affect the relationship, or having the character in the main party most of the time. Another example of a game using a relationship state algorithm is any of the Mass Effect games by Bioware, in which each character holds a romance variable that can altered with dialog choices. This will determine which character the protagonist is more likely to develop a romantic relationship with.

Though being two different concepts, a connection between relationships or social interactions and feelings or emotions can be established to a certain degree. For example, when you love someone you may feel happy when talking to them or when you hate someone you might feel angry at them. Of course, this is never linear, considering for example lack of reciprocity (e.g. the person does not love you back), or the occurrence of an event (e.g. though you love them, they have hurt you and you feel sad). Due to the complexity of the subject at matter and the difficulty to quantify these concepts in way that can perfectly reflect real life sociological studies, we will not delve so deep in the scope of this work, though it does pave the way for possible future research.

3

Related Work

Contents

In this section, we will be showing research done that can contribute to our work. Most importantly to understand if this study has great importance, we researched other works on the adaptive soundtrack's influence on the player's game experience and on the behavioural responses influenced by music. This can help us to understand what properties to alter in the soundtrack based on what relationship parameters to cause a greater impact in the experience.

3.1 Music Matters: An empirical study on the effects of adaptive music on experienced and perceived player affect

The research of Plut et al. [9] (2019) tries to assert the effectiveness of adaptive music on an experience and on a perception level from the player's side. They stand to believe that adaptive soundtrack can have a relevant impact on the player's experience of a game, although most games still implement linear soundtrack mainly due to cost and risk[1] [8] involved with developing dynamic soundtrack, or possibly due to lack of study that defends it's effectiveness. Therefore, more studies on the subject of adaptive soundtrack on video games are imperative.

Their hypothesis is that adaptive music based on, and directly proportional to, a tension curve impacts the player's experienced tension, contributing to a richer and more immersive gaming experience.

One important fact that is stated in the background of their study is that though it is agreed there is an effective impact from the music to the listener, the way music and emotion relate is complicated. In our work, having this in mind can help us understand the emphasis needed on the relationship between music and emotion, mainly when selecting the types of changes to be done to music based on how the relationship with the character is.

There is also a reference to how a variable state can take many forms apart from visual, such as musical. This can be used in our work to understand if the player is aware or can be aware in what state is the relationship with a character.

In this study, the tool for the assertion was a game developed for the purpose called Galactic Escape. The game consisted in controlling a ship in outer space going between spots where challenges had to be completed. These challenges also impaired different possible difficulty choices to give a feeling of control over the tension generated. There is a time limit, after which an enemy ship starts chasing the player's ship, slowly taking the same route but without needing to stop for challenges. As the ships' distance gets smaller, the music intensifies in terms of tension, creating a sort of growing panic feeling in the player. It's a concept used in many games to give the player an extra conflict so he can keep up the pace, examples being Bubble Bobble (Taito, 1988) or The Flash (Probe, 1993).

[1] The risk referred here is that, in a musical context, there is a possibility that the soundtrack becomes so dynamic and out of the music composer's hands, that it starts becoming something not intended that can break the player's experience if the adaptation is not carefully implemented

To map the gameplay tension to music tension and create the adaptations to the music, Plut et al. [9] used FMOD Studio[2]. A single parameter, tension, was used. While the tension remains null, in terms of rhythm, the neutral music does not present strong accents. As the tension rises, dissonant tones, harsh timbres and crowded harmonies are added, as well as rhythm going out of balance.

In terms of methodology, Plut et al. [9] had the players clear the game in four different conditions: no music, linear music, adaptive music inverse to tension and adaptive music matching tension. To evaluate the work, after each gameplay of a different condition, the players would be presented with a small questionnaire. From these questions, one group seems interesting to our own work: Perceived emotion questions. We want to know if the player understands the state of the relationship, and perceiving emotions from the music is a fundamental step in that path.

For their work, the results showed a surprising fact that the higher impact occurred when introducing adaptive music rather than when matching the game tension, suggesting that the adaptivity of the music is more important of a factor than the emotional correlation with the music. That does not mean that the congruency of the soundtrack with the emotions is not relevant, it only means that for this study's purpose the impact of adding adaptive music to a game that had no music or linear music was much greater. In our case, we don't need to establish emotional correlations with the player's affection, but we want there to be a better understanding of the situation based on the music being played. [9]

3.2 Immersion in the Virtual Environment: The Effect of a Musical Score on the Video Gaming Experience

Though this study is somewhat older (2004), it stands as a good example of how important music is in the gaming experience and has become, suggesting game musical scores be specifically created for the purpose. The objective of this work by Lipscomb et al. [10] is to investigate the impact of musical soundtrack on the video gaming experience.

From articles researched about the effect of music in a racing video game [11] [12], Lipscomb et al. [10] suggested that in games that require control, concentration and good motor skills, the presence of music can actually have a negative effect on the experience, or more specifically, on the scoring. Though this fact is interesting, as it is one of the very few negative examples of music implementation in games, it is not relevant enough to our work. The type of game we're focusing on allows the player to calmly select choices that progress the game.

With this study, Lipscomb et al. [10] show interest in questions of which two are relevant to our work, mostly the second:

[2]FMOD Studio (Firelight Technologies)
`https://www.fmod.com/studio`

1. Does music influence the perception a player has on a game?
2. How does the music impact the gaming experience?

To execute the experiment, the game The Lord of the Rings: The Two Towers (Electronic Arts, 2002) was used as the instrument of experimentation. The game has a very rich soundtrack that was designed specifically for it. The experiment contained three condition variants: playing the game with no music, playing the game with music and listening only to the music, without actually playing. This last one was more to understand the level of perception of the environment and situation a person can have through sound only. The participants (both genders, high school students and college students) were divided into groups to be subjected each to a different condition. The people in the first two condition groups played three different segments of the game, each with different levels of tension and mood. The people in the last group were subjected to music played during those segments.

In terms of results, Lipscomb et al. [10] used Verbal Attribute Magnitude Estimation (VAME) scales with value categories evaluative (good, cold, exciting, pleasant), potency (gentle, high, intense, loud, powerful) and a single scale for the ”active” term as the domains for repeated measures Analysis of Variance (ANOVA). From these measures, apart from understanding that the presence of music deeply impacts the experience, in the different sets of participants, the responses were so different between the genders and schools that, in that way, is inconclusive. Still, it is important to retain that between the various conditions, the experience results seem to be significantly different.

Lipscomb et al. [10] suggest a relationship between music and game, specially if the music is developed specifically for that game, to be highly complicated. And though it is not certain if the study results are impacted by the measuring tool (The Lord of the Rings: The Two Towers), further investigation should be performed on this subject. [10]

3.3 barelyMusician: An Adaptive Music Engine For Interactive Systems

In 2014, Gungormusler et al. [3] understood that as interesting as adaptive soundtrack may be, it is not entirely dynamic when, as implemented by most games out there, it adapts offline constructed music and not real-time generated music. Transitions and variations can be predicted. Therefore, the work proposed was to build an adaptive music engine that generates melody with respect to the current environment mood in real-time.

The engine was called barelyMusician, and it features a graphical user interface to alter parameters as a Unity3D component, to be integrated with games built in Unity3D.

In a preliminary stage of research, Gungormusler et al. [3] refer to the use of audio elements as one of the most important aspects to affect immersion and quality of experience, as we strongly agree from our own research. And truly dynamic music can only be offered by videogames, unlike movies for example. Game development is focusing more and more on audio technologies and innovations, especially on adaptive soundtrack experiences.

Gungormusler et al. [3] take it a step further stating that although adaptive soundtrack can truly be immersive and dynamic, in most cases it operates over a very static element, offline music generation. To reach pure dynamism, it is suggested that real-time music generation is a good path. Though this is agreeable, with the lack of full control from the developer over the music generated, undesired/non-calculated scores may be constructed, that could easily break the experience. It is an interesting method, though caution is required.

One of the most intriguing aspects of this research to our own work is the parameter correlations done in the engine. Gungormusler et al. [3] explain in detail how the change in parameter values affects the music played. For the engine developed, the parameters to be used as triggers for the audio adaptation are energy and stress, that combined can mimic emotional states. The correlation of the parameter values with the musical attributes is described as followed:

- Tempo - faster as energy rises
- Music Scale - scale differ with stress. For example, an harmonic minor scale is used when stress is high
- Articulation - average note length shortens with higher energy
- Articulation variation - alteration of note length in certain pattern is directly proportional to energy
- Loudness - volume of a note rises along with energy
- Loudness variation - alterations of volume increase with both energy and stress risings
- Pitch height - notes are higher pitched when energy and stress increase
- Harmonic curve - direction of melody and relation between note pitches. Proportional to energy and stress
- Timbre properties - tone of the instrument voice. Brightness and tension of timbre are directly proportional to both energy and stress

As how these two parameters describe the environment mood, Table 3.1 is presented in the work by Gungormusler et al. [3], as Table 4.2 from Chapter 4.2.6. Between these moods, the parameters and musical attribute changes, we can analyse what relationship state properties are interesting to our algorithm and how the soundtrack could adapt to their alterations.

Table 3.1: High-level abstractions and the corresponding mood values

Mood	Energy	Stress
Exciting	1.0	0.0
Happy	0.5	0.0
Tender	0.0	0.0
Neutral	0.5	0.5
Depressed	0.0	1.0
Sad	0.25	0.75
Angry	1.0	1.0

3.4 Influencing the Behavioral Responses of Players in an Interactive Narrative Game through Music and Arousal Congruency

Stepping a bit out of the adaptive soundtrack theme, there is also an interesting study on how music could possibly influence players to choose different options on a narrative game dialog. Though our objective is not to influence the player, understanding how that occurs could be substantially relevant to our own work, as one of the objectives is to assert if the player is able to understand how the relationship with the character is by means of soundtrack.

Logan et al. [13] present the hypothesis that, in a social behaviour narrative game, high-arousal music may influence the player into choosing avoidance options and low-arousal may promote social behaviour exhibition. The importance of this work becomes greater as, like we've seen before, the majority of studies on effects of music in video games mainly focus on player performance [14].

Taking into account one of the researched articles for the study, Logan et al. [13] adopted Murray's definition of agency[3] [15], and one of the article's conclusions was that players reported a higher sense of agency if the choices they made had significant impact and result in different situations, rather then having no effect on situational content. This result is interesting as it reveals a possible point of interest for our work objective: if the player is more engaged noticing the changes to the mood during dialogues, and possibly understand the current disposition through the soundtrack. Also from another study [16] on arousal congruency in influencing consumers in-store behaviours, a good definition is made for how as music tempo gets higher, so will the environment's arousal. This connection demonstrates a type of modification that can be performed to the soundtrack based on a relationship state property, as for example tension rising between two peoples usually increases the arousal to extremities where at least one of them wishes to flee or end an argument. Nonetheless, in another study [17] about player choice and roles, even when players are not given an explicit role, they tend to make choices that go inline with the type of role they are familiar with or are more fond of. This means that caution must be taken in the

[3]"Player agency is the satisfying power to take meaningful action and see the results of our decisions and choices", as defined by Jane Murray in "Hamlet on the holodeck"

evaluation method as the player can be actually selecting dialogue choices based on the type of person they are rather than from understanding the relationship state through the music.

Logan et al. [13] selected 8 tracks of different genres and levels of arousal from a creative commons music archive database and developed a text-based narrative style game, where the player is presented with 8 scenarios in total. After a short play of one of the music samples, each scenario presents 4 different option actions the player can make to respond to the current scenario situation, 2 of avoidance and 2 of social behaviour. Both the hypothesis that high-arousal music can influence the player to select avoidance options and that low-arousal music can influence the player to select social behaviour actions were put to the test with this experiment, taking into account as well the hypothesis that the music actually has no influence in those departments.

By means of a Wilcoxon signed-rank test, Logan et al. [13] came to the intriguing conclusion that though high-arousal music may have an impact in making the player prefer avoidance choices, the same cannot be concluded from the low-arousal music hypothesis. Several possibilities for this inconclusive phenomenon are presented such as how a player's narrative role had an impact on the choices done, as proven in a researched study. Logan et al. [13] believe that most participants made choices related to how they would act in real-life, on the presentation of such scenarios. Another proposed possibility is that the arousal level may not be sufficient to determine the influence the music will have. For example, in that manner, some high-arousal tracks seemed to be far more effective than the other high-arousal tracks. Thus, it is difficult to understand if it is actually the music's arousal level the determinant factor for the influence the music will have on the behavioural response.

Despite this uncertain conclusion, given the fact that high-arousal had a great percentage of success in its hypothesis, it is safe to assume that music can influence the player's choices. Once again, though this is not what we're aiming at, it can still prove useful in terms of musical production and soundtrack adaptation measures. [13]

4

Development Tools

Contents

4.1 Game Engine

The original lines of this work were to evaluate the possibility of implementing this relationship state based adaptive music in the game developed for the course of Game Design Methodologies "A Detective Story". But for this study to be carefully controlled and asserted, we must create experiments in a smaller and more easily controlled environment. If the results of this research prove to be interesting in the least, we might as well implement this technique in the full game.

Being the game developed in RPG Maker, we figured it would be advantageous to implement this work in the same Game Engine[1].

In this section, we will describe both the game engine and the audio engine used, respectively RPG Maker MV and FMOD Studio.

4.2 RPG Maker MV

The RPG Maker Game Engine[2] is a Javascript based engine with great popularity and community contributions. Though it is not Open Source, it is highly customizable, attracting developers that like to have specific implementations and custom plugins.

Almost effortlessly, one can develop an RPG game in this engine, without needing a great knowledge of code programming. The User Interface (UI) offers many features such as:

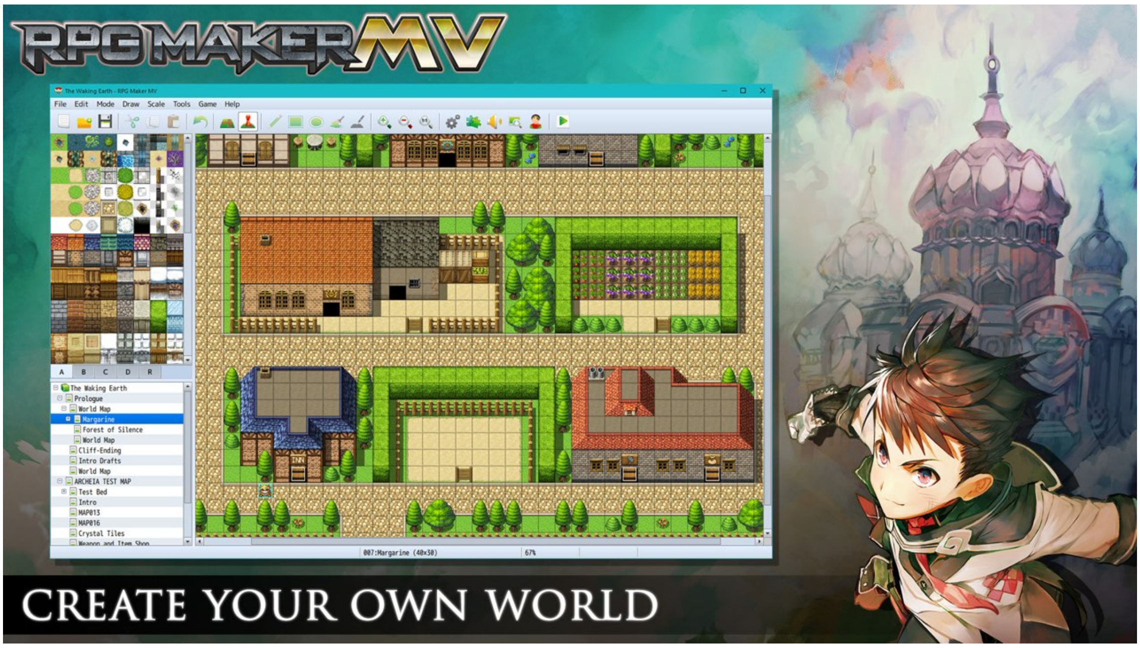
- Visual Map design (Fig. 4.2)
- Map location based events
- Easy Game Database editing
- Character generation
- Different Battle Systems
- Javascript Plugin Management (Fig. 4.2)

Although it is mainly designed for RPG Battle games, the Javascript customization plugin system allows for almost anything to be built in this engine. With that in mind and the fact that these RPG games usually rely on big narratives, it was chosen for the "A Detective Story" game.

Another upside of using RPG Maker MV for this research is that it's easy and quite quick to create a test map where to conduct the experiments with a more controlled environment. The event based game

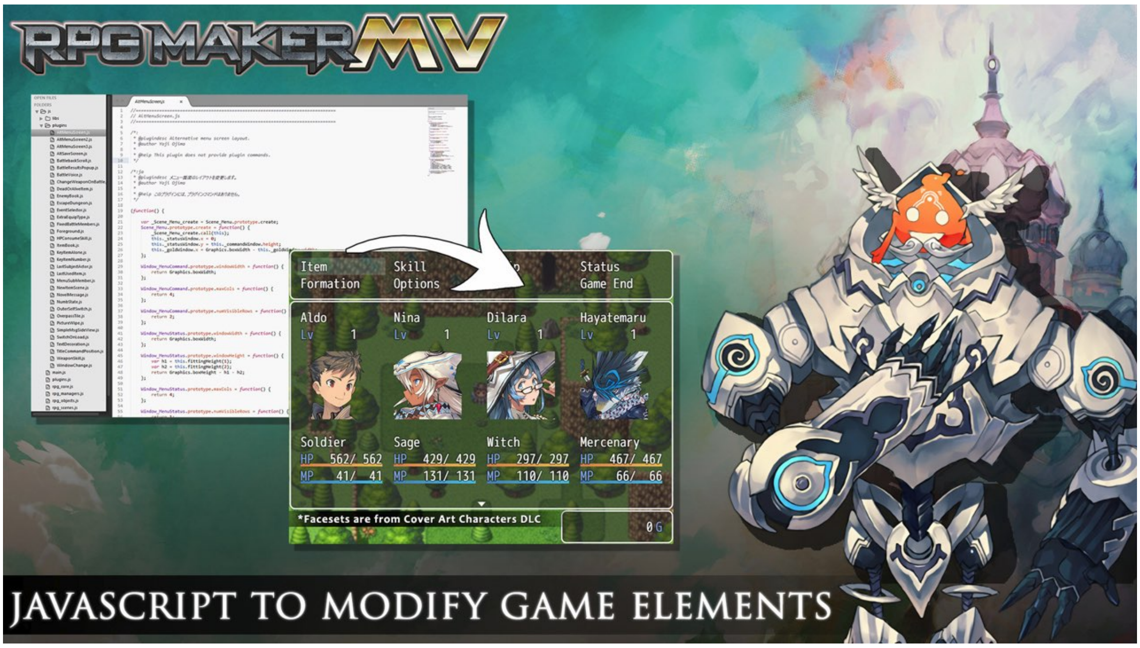
[1] Though RPG Maker was decided as a Game Engine for this research, most of the Adaptive Soundtrack capable audio engines have support for many other Game Engines, such as Unity and Unreal Engine

[2] RPG Maker website - https://www.rpgmakerweb.com/



Source - https://www.rpgmakerweb.com/products/rpg-maker-mv
Source - https://www.rpgmakerweb.com/products/rpg-maker-mv

Figure 4.1: RPG Maker MV Map Design
Source - https://www.rpgmakerweb.com/products/rpg-maker-mv



Source - https://www.rpgmakerweb.com/products/rpg-maker-mv
Source - https://www.rpgmakerweb.com/products/rpg-maker-mv

Figure 4.2: RPG Maker MV Javascript
Source - https://www.rpgmakerweb.com/products/rpg-maker-mv

building interface makes it easier to just focus on the parameters and conditions for testing, instead of having to worry too much with code programming.

4.3 Audio Engine

As far as Audio Engines go, there are many options to choose from. But for this research, an Adaptive Soundtrack capable Audio Engine is required. For this purpose, there are mainly two audio engines that are the popular choices:

- Firelight's FMOD Studio[3]
- Audiokinetic's Wwise[4]

There are no major differences in terms of features and functionality between these two engines, but after some experimentation, we found it easier for us to work with FMOD Studio.

4.4 FMOD Studio

FMOD Studio is an end-to-end solution for adding sound and music to any game[5]. It features two main components:

- Workstation authoring tools
- End-platform audio engine

4.4.1 Authoring tools

In the Authoring tools, we can arrange our sound banks and tracks, as well as parametrize them with different variables that are fed from the audio engine. Essentially, this component acts as a sound staging and editor, taking the form of many standard sound editors in terms of UI and functionality, such as Adobe Audition or Audacity. To any audio engineer, FMOD Studio's Authoring tools UI should feel familiar (Fig. 4.3).

Track automations (using the parameters) can be discrete or continuous, so different values can make many different modifications to the music. The modulations are also numerous, from volume to pitch shifting. They will perform different alterations based on the parameter values.

It supports many of the known formats for sound files.

[3]FMOD Studio webpage - https://www.fmod.com/
[4]Wwise webpage - https://www.audiokinetic.com/products/wwise/
[5]Statement taken from the website - https://www.fmod.com/studio

Figure 4.3: FMOD Studio's Authoring Tools

4.4.2 End-platform audio engine

After creating and building the sound banks, they can be included in our game and used by FMOD Studio's audio engine. This audio engine component is what runs on the game platform side, reads and processes the sound banks and alters the parameter values based on events or actions from the game.

Natively, FMOD Studio supports, without much change to the game structure, Unity Engine and Unreal Engine, with some 3rd party plugins for other engines. The engine increases a bit the overhead of the game running, but taking into account the benefits it brings, it is actually not that significant of an increase.

Through the use of a well documented Application Programming Interface (API), the game can communicate with the audio engine to exchange information (mainly, send values for the parameters to the audio engine).

4.5 Integration of the Engines

Despite FMOD Studio having some 3rd party integration on different engines, a plugin for the RPG Maker Game Engine could not be found, though there appear to be in place some requests in the community for such a plugin.

It does, however, have an implementation for Javascript. So taking advantage of the high customization of RPG Maker using Javascript, we decided to build a plugin that integrated FMOD Studio's API into the game engine.

We used the Javascript code in the FMOD Studio API's example to integrate as a module in the RPG Maker Javascript code loading. The code is a bit heavy, which was quite a challenge to optimize. This means that, although there shouldn't much of a problem running a game with this integration on a standard Personal Computer (PC) or Web Browser, the same cannot be said if the platform is for example mobile. In fact, we had to request that people played the experiments in this research on a PC Web browser, and not on a mobile phone. Further improvement of performance and platform compatibility may still be implemented, but for the purposes of this research, we did not find it feasible, as the module works fine in a PC execution and the changes required would consume too much time.

The pseudo code for the plugin is presented in Listing 4.1 and the code implementation is in Appendix A.1. The plugin's responsibility is to load the FMOD Studio Module and the sound banks, and if the map's background music is configured as an FMOD Studio sound bank, start playing upon entering the map. The plugin holds the instance of the FMOD Module as reference so it can call its routines, such as setting up the module for music playback, load the sound banks and control the sound playback (start, stop, set volume, etc.).

Listing 4.1: FMOD Studio API plugin pseudo code

```
scene_start() {
    if(scene_bgm is FMODSoundbank)
        fmod_module_start_playing();
}

scene_update() {
    fmod_module_update();
}

scene_change(new_scene) {
    if(new_scene is MapScene) {
        fmod_module_stop_playing();
    }
}

main() {
```

```
        fmod_module_setup();
        fmod_module_start();
    }
```

Alongside this API integration, we developed another plugin that serves as a controller for the Relationship parameter values to be passed to the Audio Engine. With a simple call to the Script Command[6] "relationship <parameter_values>", the plugin sends the values to the audio engine, and the modulations created in the sound bank will modify the music playing. In the future, this plugin could be used not only to feed the FMOD module with the relationship parameter changes, but any other module or system that could make use these same parameters. Suggested improvement would be to move the code that sends the values to the audio engine into the FMOD Studio API plugin mentioned above, and create a module registration list that, for each module registered for relationship parameter change, gets called on value update. For now, and for the purposes of this research, we did not find the change worth diving into, but we leave here the improvement suggestion. The pseudo code for this plugin is presented in Listing 4.2 and the code implementation is in Appendix A.2. The overall integration architecture is represented in Figure 4.4.

Listing 4.2: FMOD Studio Relationship plugin pseudo code

```
def FMODParameters;

updateParameters(parameters) {
    fmod_module_update_parameters(parameters);
}
```

[6]One of RPG Maker's event actions available

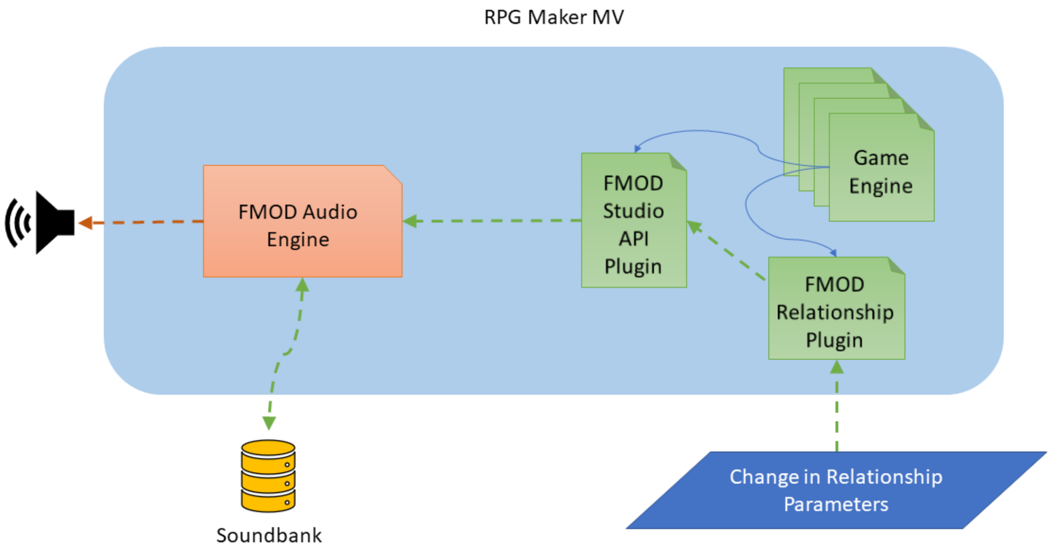


Figure 4.4: Integration of the FMOD Audio Engine with RPG Maker MV

5

Musical Content

Contents

Even though an immense library of public domain and unused musical content is available on the internet, for this work to be effective, acquiring original soundtrack for it may prove to be of extreme importance. Therefore, we decided to try and establish a partnership with a philharmonic group or a conservatory of music. Fortunately, we knew a contact in Setúbal's Conservatory of Music[1], professor António Laertes. We scheduled an appointment to share our vision and needs and try to establish such a partnership. After a very positive response due to there being a student in CRS that was looking for such a project to use for his own research, the partnership was set. This collaboration did also contribute to a second purpose which is to have CRS help us understand the connection between musical properties and relationship emotions and what kind of modulations are best to represent the changes in these relationship states.

5.1 First draft

We were now set in terms of musical content acquisition. The defined line of work for the CRS team would be to compose one song to be used in the final experiment. Two members would constitute the Musical Composition team:

- Prof. António Laertes
- Student Tiago Ribeiro

In the preliminary meetings with CRS, some conditions were established in order for the work to go as smoothly as possible.

We showcased the game "A Detective Story" in order to both demonstrate the artistic style of the game for which the composition was intended (Hi-Res 2D RPG) and give context on the genre of the game, which would be imperative for the composition (Suspense/Mystery Investigation).

The team then proceeded to create some small tracks with different instruments to understand if they were going on the right direction. This protocol also makes developers from the game design and programming side like us how important it is to commute with the different artistic teams, as in the end the entirety of the game will be a mixture of all these features. So a nice blending of all departments is essential.

In total, three Tutti [18] were developed and were assigned the letters A[2], B[3] and C[4]. After thorough inspection, our remarks on each of the tracks were:

[1] Henceforth referred to as Conservatório Regional de Setúbal (CRS)
[2] https://soundcloud.com/user-901650047/a-a-ventura/s-ZgSmLzbp9YF
[3] https://soundcloud.com/user-901650047/b-a-detective-story/s-ZLghifWHK3k
[4] https://soundcloud.com/user-901650047/c-quazyum/s-4TqQTMmHvZi

A Motivational theme, great for a battle or interrogation scene. But still, it contains many elements that can be present in a more neutral form. Very interesting tribal elements. Tension and mystery tone present.

B Tragic theme. Contains some neutrality which is great for modulation. Good to use in complement or as an introduction. Piano and wind elements well set.

C Comical/tragic tone. Good example to characterize the original game.

In the end, we all agreed to have a mixture of both A and B, considering A as the main loop theme and B elements as intro or bridge between sections.

5.2 Final composition

Taking everything defined previously into account, the CRS team created the final composition[5]. It features an introductory drama/suspense, great for a dark realization, followed by the main section that gives the feeling of a brave/assertive motivation. It is our opinion that the entire piece is very well suited for a video game soundtrack.

The CRS team handed us each of the instrument tracks individually in the Audio Interchange File Format (AIFF) format, which retains maximum quality. This way, we were able to modulate each track on FMOD Studio based on the music model to be used.

[5] https://soundcloud.com/user-901650047/a-detective-story/s-KB906ctkzxX

6

Relationship state to Music Concept Model

Contents

In order for the results to be accurate with the objective of this research, a model of how musical properties and texture are modulated based on an affective and emotional relationship state must be established.

6.1 Emotions in relationships model

Finding enough previous research that can sustain such a model proved to be a very difficult task, as relationship states are a very ambiguous and complex matter, let alone finding material to correlate them with musical modulations. Therefore, some simplifications had to be taken into account with the relationship states, as referred in Chapter 1. We decided to use a simplified version of an emotions in relationships model.

As a proof of concept that is this research, we are going to use two emotions representing the relationships between characters: love and hate. This two-dimension model should be enough, for now, to give a glimpse of how we can use relationship states as a feed to interesting changes in the game, such as the music. In contrast, using the alterations in the sound as a feedback can prove to be a new way of giving the player some idea of the state of things and allow it to take the desired counter actions.

Love can be viewed as a positive extension of being given to the other, affectionate about the other, whilst Hate can be seen as negative extension of pushing the other away, be disgusted or unhappy with the other [19]. Of course, and as stated before, relationships are much more complex than these simple associations, but for the purposes of this research, we do not believe it feasible to deal with more complex emotions and symbols for relationships. Proven worthy of further investigation the work done here, we suggest future research more specific to this kind of association model. For now, this should be a good starting point to check how this model can be translated to musical property and tone modulations.

6.2 Emotions to Music Model

Having defined the emotions to be represented by modulations of the musical properties, we are in a good position to define how they will affect said properties. After some research, we've decided to use the information described in Chapters 3.4 and 7.7 of "How Music Really Works" by Wayne Chase [1]. Chase et al. describe the emotions associated with different types musical property values as Tables 6.1, 6.2, 6.3.

Pitch Characteristics	Associated Emotions
Low pitch	Fear, seriousness, generally negative emotional valence, also majesty, vigour, dignity, solemnity, tenderness
Low pitch, monotonic	Anger, boredom, sometimes fear
Low pitch, especially octave leap downwards	Sadness, melancholy
High pitch	Generally positive emotional valence, happiness, grace, surprise, triumph, serenity, dreaminess
High, rising melody, especially octave leap upwards	Happiness, excitement
Wandering, unfocused	Sadness

Table 6.1: Emotional Effects of Pitch - Table 5 in Chapter 3.4 of "How Music Really Works" [1]

Loudness Characteristics	Associated Emotions
Soft (quiet)	Generally negative emotional valence—sadness, melancholy; but also tenderness, peacefulness
Soft, not varying much	Tenderness
Moderate, not varying much	Happiness, pleasantness
Loud	Joy, excitement, happiness, triumph, generally positive emotional valence
Very loud, to distortion levels	Anger
Wide changes, soft to loud, especially if quick	Fear

Table 6.2: Emotional Effects of Loudness - Table 6 in Chapter 3.4 of "How Music Really Works" [1]

Tone Color Characteristics	Associated Emotions
Simple tone color, few overtones (e.g., flute)	Pleasantness, peace, boredom
Complex tone color, many overtones (e.g., over-driven electric guitar)	Power, anger, fear
Bright tone color, crisp, fast tone attack and decay in performance	Generally positive emotional valence, happiness
Dull tone color, slow attack and decay in performance	Generally negative emotional valence, sadness, tenderness
Violin sounds	Sadness, fear, anger
Drum sounds	Anger
Sharp, abrupt tone attacks	Anger

Table 6.3: Emotional Effects of Tone Color - Table 7 in Chapter 3.4 of "How Music Really Works" [1]

Cross-checking these associations, we designed a simple music model that roughly translates them to modulation in the soundtrack. A good starting base would be to use two relationship states: Love and Hate. This is because, though they are in fact complex relationship state, they are very intrinsically connected to emotions. Taking this into account, the model can be defined as:

- Love
  1. The pitch will rise as the parameter value gets higher.
  2. The volume gets softer as the parameter value gets higher.
  3. The predominant instrument types for this parameter's higher value are wind and strings (though of the latter, only piano or harp instruments should be considered).
- Hate
  1. The pitch will fall as the parameter value gets higher.
  2. The volume get louder as the parameter value gets higher.
  3. The predominant instrument type for this parameter's higher values are percussion (mainly drums), brass and strings (mainly guitar and violins)

6.3 Music model experiment

In order to assert the effectiveness of the model we conjured, we decided to create a preliminary experiment.

A simple sandbox like environment would allow players to talk to a character in the game[1] about another character, and their relationship's nature would influence the modulations in the background music. The player would then state its assumptions, but to be sure that no others factors would drive the player towards the answers, the content of the conversation with the character would have to be as neutral and non-informative as possible.

The setting is based on a small house. Inside there are three people: a bartender and two suspects on a case. The idea is to talk to the bartender about each of the suspects and understand whom he is affectionate about and whom he loathes. So, opening a dialog with the bartender, the player can choose to ask about Priscilla (suspect 1) or about Sophia (suspect 2). This choice instantly sends different parameter values for Love and Hate to the audio engine in order to perform the implemented modulations.

To set its answers, the player would be given a selection of emotions to pick two that it felt best suited for the relationship between the characters. The emotional attributions and relationships available to the

[1] Though defined as a game, one could hardly call it that, as the experiment has little challenge to overcome, with no regards to success or failure. We call it game due to it's nature and purpose.

Figure 6.1: Music Model experiment's map

players are:

- Love
- Caring
- Friendship
- Hate
- Disgust
- Neutral
- Curious

These emotions can be ranked in terms of being defining and assertive of the relationship status. So when Love, Friendship or Caring are selected, we can assume the player clearly understands their relationship as being positive, whilst selecting Hate or Disgusted transmits a negative relationship in the player's answer. The Neutral and Curious can be seen as very blend and ambiguous emotions that cannot be easily (or at all) measured as positive or negative. We expect that these will be selected in a case of doubt or uncertainty.

In this experiment, we're not yet using the soundtrack from CRS, as first we'd like to just test the music model modulations, and make the experience different in case players from this experiment end up playing the final experiment as well. Instead, a multi-track song from the Mixing Secrets Cambridge Music Technology website[2]. The song is called ”Alex the Adventurer” by Jeffrey Hayat.

[2] https://www.cambridge-mt.com/ms/mtk/

Letter	Control	Talk about Priscilla	Talk about Sophia
A	Adaptive	Love = 1 & Hate = 0	Love = 0 & Hate = 1
B	Adaptive	Love = 0 & Hate = 1	Love = 1 & Hate = 0
A	Non-Adaptive	No modification	No modification
B	Non-Adaptive	No modification	No modification

Table 6.4: Different versions and action consequences

6.4 Experiment website

The experiment was published online[3] and advertised for people to participate and play. The web page is form based, starting with demographic purpose queries (gender, age interval, gaming habits and likeability of detective or investigation type games), followed by a sound check and the game experiment. The emotion responses are available in-game to be later extracted by the form. It also includes a control question, to check if the player payed attention to the experiment (number of suspects).

In order to further clarify the experiment's purpose results, two versions of the game are available, only switching the emotions of the bartender between the two suspects (Sophia or Priscilla). In game A, the bartender is affectionate towards Priscilla, and hateful towards Sophia. Game B has these feelings reversed. Whether a player will play version A or B is decided based on a random number generator outputting 0 or 1 (B or A, respectively).

Moreover, for each of the games A and B, a non-adaptive version is also available, that introduces no modifications to the soundtrack when talking to the bartender. The purpose behind this version is to validate the effectiveness of the experiment. The player will play this version if the previous participant played the adaptive soundtrack version, and vice versa. The expected tendency is to have more responses with assertive emotions on the adaptive soundtrack version and more responses with blend/ambiguous emotions on the non-adaptive soundtrack version. Particularly speaking, we expect to see more answers of Love, Caring and Friendship for Priscilla and more answers of Hate and Disgust for Sophia in the adaptive version A, vice-versa in the adaptive version B and more answers of Neutral and Curious in the non-adaptive version A and B.

All in all, there are a total of 4 games[4], described in Table 6.4, the server is able to pick from to present to the player.

6.5 Experiment results

A total of 28 people participated in this experiment. In this section, we will be analysing the results, focusing on the effectiveness of the music model by looking at the emotions reported.

[3] Experiment's website - https://research1.adetectivestory.com

[4] Adaptive Soundtrack A game; Adaptive Soundtrack B game; Non-adaptive Soundtrack A game; Non-adaptive Soundtrack B game

In terms of demographics, the results are presented in Figure 6.2. Summarizing, roughly 2/3 of the participants are Male and 1/3 are Female, 2/3 of the participants are aged between 19 and 30 years old and 1/3 are aged between 31 and 50 years old. Half of the participants claim to be daily game players, while 6 state playing once or twice a month, 3 state playing once or twice a month, and the other 4 state playing once or twice a year. This means we have a good quantity of avid experienced players. Lastly, all participants claim to like detective/investigation games (including board games, escape rooms, etc.), meaning that as long as the experiment upholds its nature, the players should find the experience positive (though this music model experiment does not hold much of a game like nature).

As for the results of the experiment, represented in Figures 6.3 and 6.4, we can see that the difference between the adaptive and non adaptive soundtrack versions of the game is somewhat as we would expect. There is a tendency to assertively define the relationships as loving or hateful in the adaptive soundtrack version, and a doubtful or ambiguous definition of the relationships in the non-adaptive version. Though it seems to be a small contribution, it does offer good results. The choice of the song, and how many modulations we did could be an influence in this result. Still, we find the adaptive soundtrack as feedback a success from this experiment.

The effectiveness of the music model can be asserted by looking at the differences between the adaptive soundtrack versions A and B where:

A Bartender loves Priscilla and hates Sophia

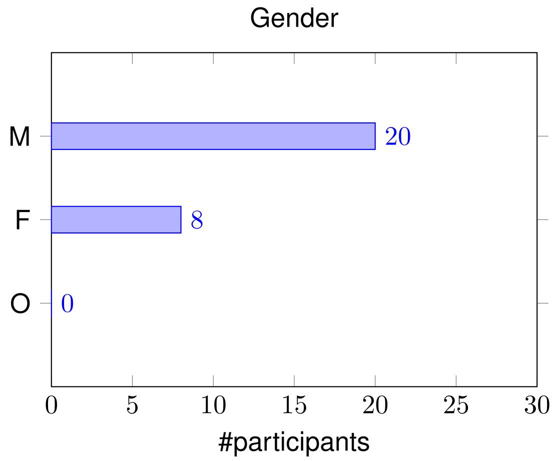
B Bartender loves Sophia and hates Priscilla

We can see that the players from game B were almost all able to state clearly that the Bartender is affectionate towards Sophia and hateful towards Priscilla, though the inverse is a little harder to see in game A. This does mean that although there is a positive change towards the purpose of this research, it can still be a bit unclear, since the only changing fact between versions A and B is switching the emotions for the characters. No different modulations to the soundtrack are done whatsoever in terms of the parameter values, only the suspect relationships are swapped. Again, the song in place can have a role in this issue.

6.6 Outcome

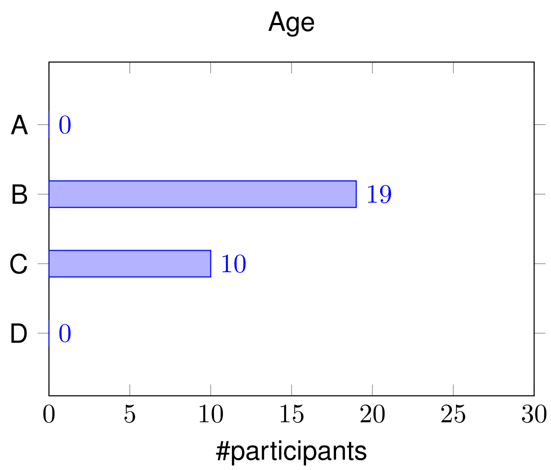
It is fair to say that we can, with what we've seen here, the music model seems to hold up to what we're striving for. We admit the results could be more promising, but we'll have to consider the different factors in play:

- The song chosen as background

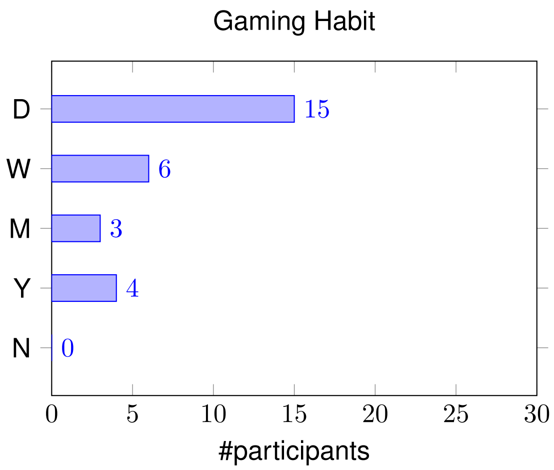


k,

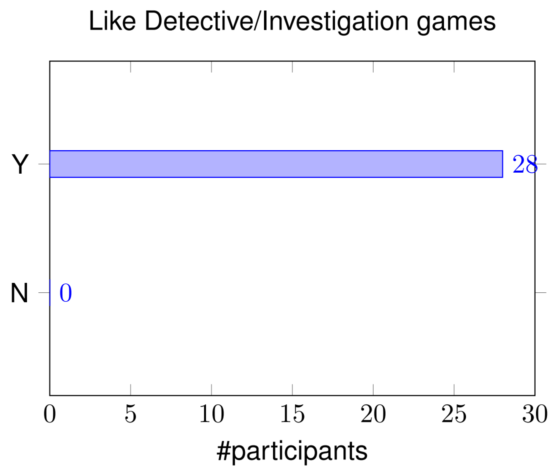
(a) M: Male
F: Female
O: Other



(b) A: 0-18
B: 19-30
C: 31-50
D: 51 or more



(c) D: Daily
W: Once or twice a week
M: Once or twice a month
Y: Once or twice a year
N: Never



(d) Y: Yes
N: No

Figure 6.2: Music Model Experiment Demographics

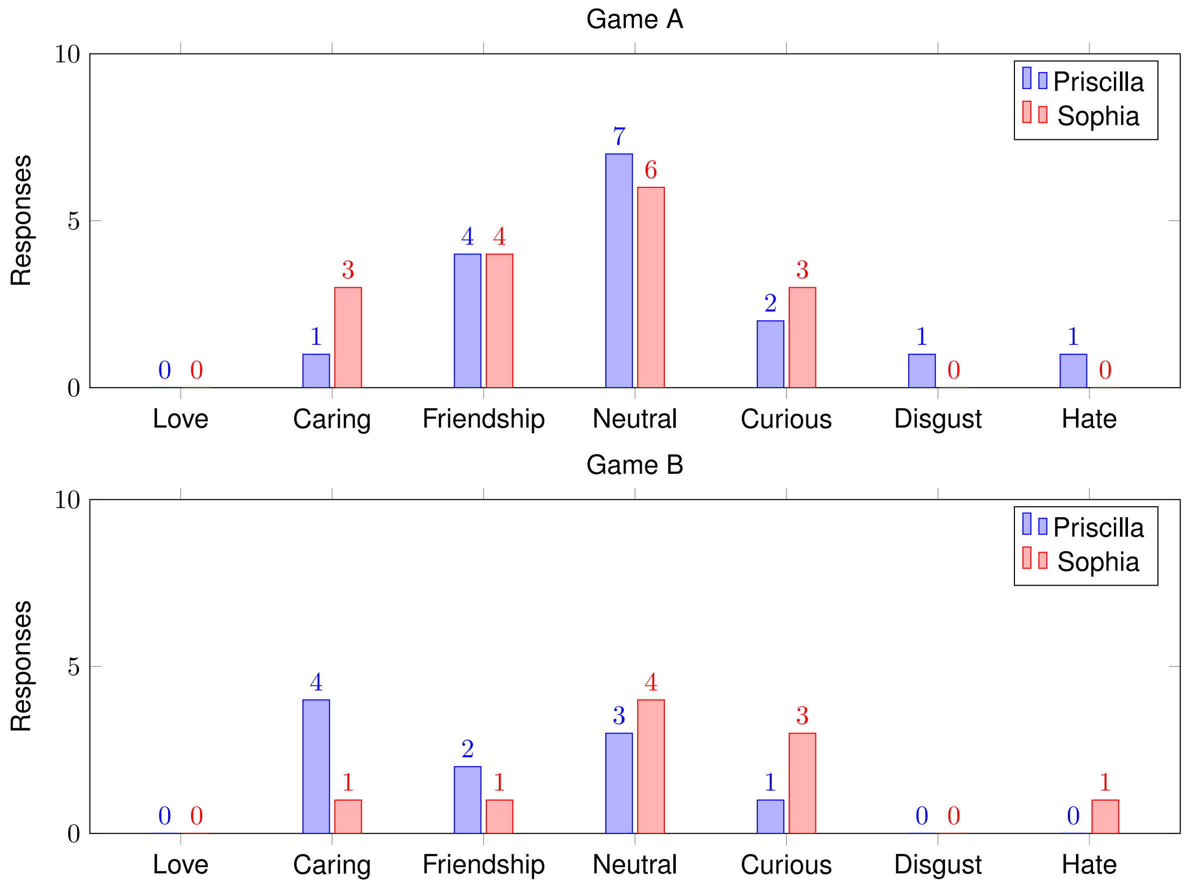


Figure 6.3: Non-adaptive soundtrack version results

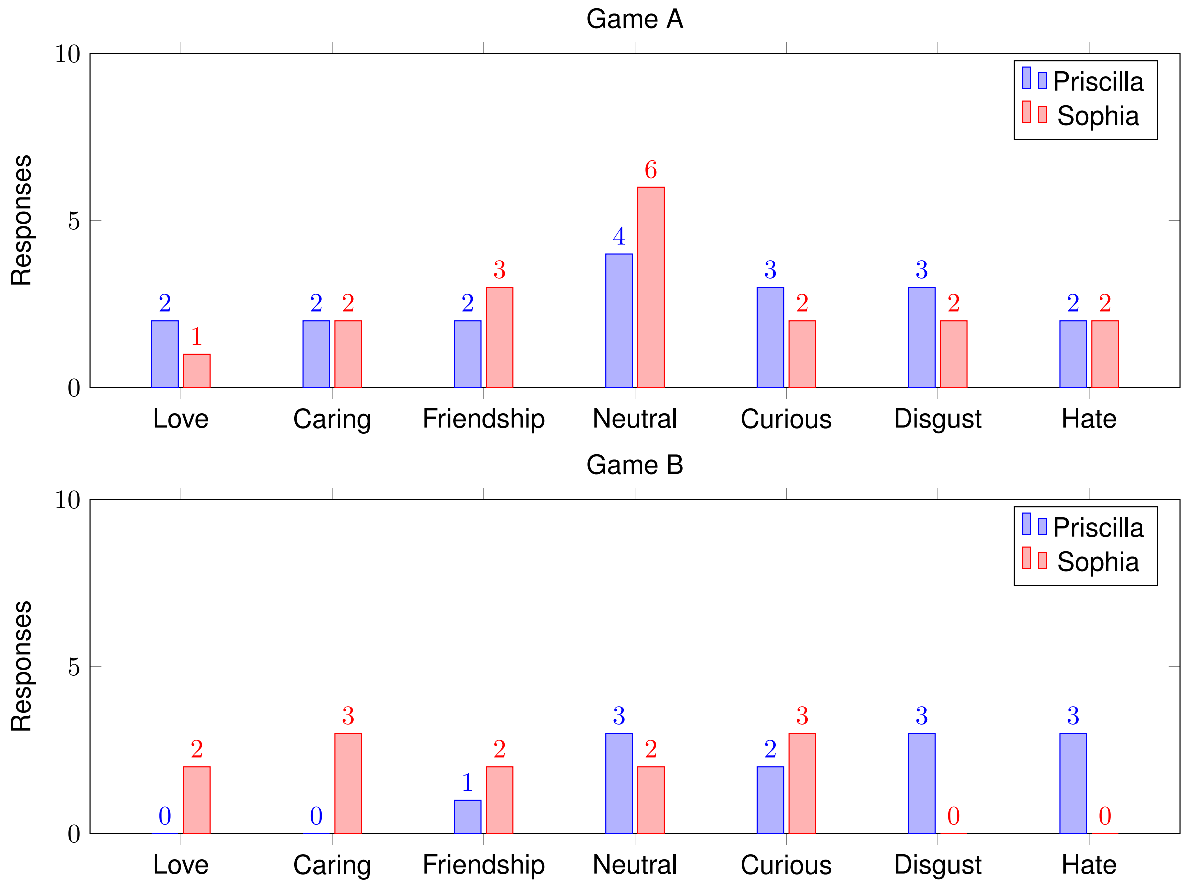


Figure 6.4: Adaptive soundtrack version results

- The modulations performed
- The abstraction level of spoken information in the game
- The experience and focus of the players

All in all, we believe that this music model should be acceptable for our final research experiment.

7

Experiment and Evaluation Methodology

Contents

With the music model proven effective and the final composition from CRS in hands, we developed the research experiment.

7.1 Symphony of Mystery

The experiment consists of a simple mini-game written in RPG Maker MV called ”Symphony of Mystery” in which the player must figure out a code to unlock a door. The character is trapped inside a room by a colleague who will only open the door if the player can answer two questions correctly. In the room, the player will find three holograms, each representing a suspect in one of the department's investigation cases. For each of these suspects, the personal info available is:

- Name
- Age
- Profession

They are all related in some way, and though they've given the department little to no information about themselves, they were willing to give information on each of the other suspects. This information being:

- Where does X person live
- How are they affiliated with X person
- What are X person's hobbies

7.1.1 Player Goal

The goal here is to understand how each of the suspects feels about the other suspects in terms of affection or hatefulness by hearing the changes in the melody each time the character asks about a particular suspect. The idea here is to try to understand how the characters feel about each other by listening to the modulations performed on the background music. Each time the player selects a question to suspect X about suspect Y, one of two things can happen[1]:

- If suspect X is affectionate towards suspect Y, the parameter love goes up by 0.1 and the parameter hate goes down by 0.1
- If suspect X is hateful towards suspect Y, the parameter love goes down by 0.1 and the parameter hate goes up by 0.1

[1] Take into account that both love and hate are parameters ranged from 0 to 1, and that in the beginning the parameters are both at 0.5

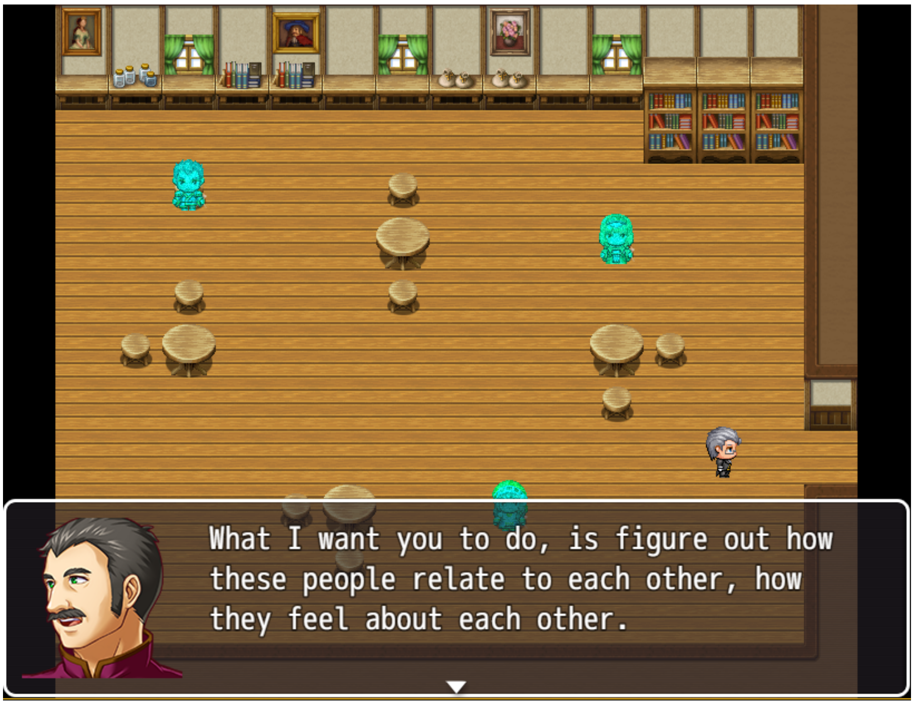


Figure 7.1: Symphony of Mystery's game map

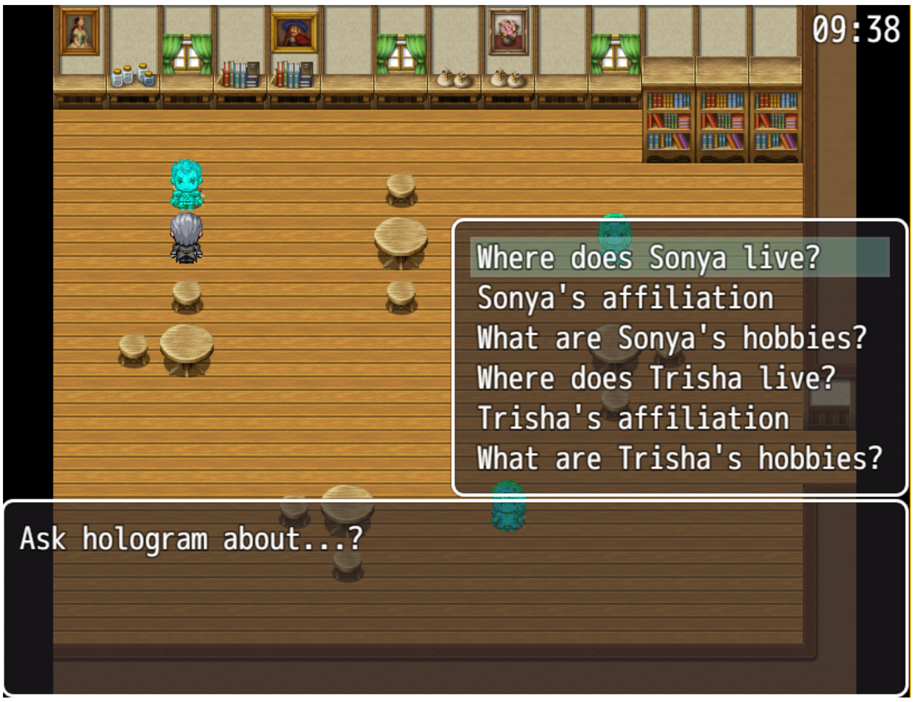


Figure 7.2: Symphony of Mystery's queries example

This difference with how the music model experiment was developed had one objective: to not make sudden jumps in terms of music modulation and possibly wreck the game experience for the player. But one issue that we found with this approach is that since the increments are performed based on the previous question, and do not reset when talking about another character for instance, a player may not reach the maximum parameter value modulations when asking the questions. The player has three chances to correctly answer the colleague's questions, and a time limit of 10 minutes to figure out the solution. The questions of the puzzle are:

1 What is the age difference of the two lovebirds?

2 In what towns do the people that hate each other live?

Answering correctly these two questions, will have the colleague open the door and the player wins. If the player fails the code after three tries or the time runs out without giving an answer, it's game over.

To get to the answers needed, one must pay attention to the modulations performed int the music as a consequence of asking a question to the suspect about another suspect. As an example of an interaction in version A:

- Player talks with Sonya's hologram (which the initial info shows she is 39 years old) and asks any question about Trisha.
  - The parameter love goes up and the parameter hate goes down
  - The change is instantly perceptible in the music
- Player talks with Trisha's hologram (which the initial info shows she is 37 years old) and asks any question about Sonya.
  - The parameter love goes up and the parameter hate goes down
  - The change is instantly perceptible in the music
- Player may now assume that Sonya and Trisha are in love with each other and their age difference is 2
- Player talks with Trisha's hologram and asks where David lives.
  - Trisha's hologram responds "Nibelheim"
  - The parameter love goes down and the parameter hate goes up
  - The change is instantly perceptible in the music
- Player talks with David's hologram and asks where Trisha live.
  - David's hologram responds "Midgar"

- The parameter love goes down and the parameter hate goes up
- The change is instantly perceptible in the music

- Player may now assume that David and Trisha hate each other so the answer's towns are Nibelheim and Midgar

7.2 Experiment website

In the same way as the music model experiment, "Symphony of Mystery" was published online[2] and advertised for people to participate and play. The web page is form based, starting with demographic purpose queries (gender, age interval, gaming habits and likeability of detective or investigation type games), followed by a sound check and the game experiment. The data extracted from the game for response analysis is:

- Answers in the 3 tries
- Time left
- Whether the player won

Similar to the previous experiment, two versions are available to play in the website, game A and game B. The difference between these versions are the levels of affection and hatefulness between the characters. Whether a player will play version A or B is decided based on a random number generator outputting 0 or 1 (B or A, respectively). With the suspects being David, Sonya and Trisha, their relationships and feelings in each of the versions of the game are described in Tables 7.1 and 7.2.

By looking at the table, and knowing the questions asked to the player, the correct answer for each of the game versions is:

A
- What is the age difference of the two lovebirds? 2
- In what towns do the people that hate each other live? Nibelheim and Midgar

B
- What is the age difference of the two lovebirds? 5
- In what towns do the people that hate each other live? Nibelheim and Kalm

[2]Experiment's website - https://research2.adetectivestory.com

Suspect	Relationships
David	Is in love with Sonya
	Hates Trisha
Sonya	Is in love with Trisha
	Hates David
Trisha	Is in love with Sonya
	Hates David

Table 7.1: Relationships in game A

Suspect	Relationships
David	Is in love with Trisha
	Hates Sonya
Sonya	Is in love with Trisha
	Hates David
Trisha	Is in love with David
	Hates Sonya

Table 7.2: Relationships in game B

7.3 Evaluation Methodology

In the end of the website wizard, the participant is presented with a link to a Google Form[3] to answer a few questions about its experience. These questions are a short extraction of the Game User Experience Satisfaction Scale (GUESS), developed by Phan et al. [2]. Each response is in a range of 1 to 5, with 1 meaning the player does not agree at all with the statement and 5 meaning the player agrees without a doubt with the statement. The questions are more focused to our needs in terms of user experience evaluation, and the effects of the audio on the game experience. The form questions available to the user can be seen in Table 7.3.

Regarding the questions selected to be on the Google Form, here are our remarks as to what purpose each section has for our evaluation:

- Enjoyment - Assert the effectiveness of the technique studied with this research in terms of gaming experience improvement
- Audio Aesthetics - Understand if the song that was composed for this experiment is adequate to the theme and contributes to the experience
- Playability - Assert that the way the experiment was built was successful in providing the player with the necessary feedback and control to play and reach the goal
- Personal Gratification and Play Engrossment - Understand if this sort of experiment could be the foundation to a future implementation of a game thus validating this technique

[3] https://docs.google.com/forms/d/e/1FAIpQLSc1cHXYebetJiKoNZxdfIIiBKBIvobbdwlJcHE8kqSu-pn0Lw/viewform

Questionnaire section	Question
Enjoyment	I think the game was fun
Enjoyment	I enjoyed playing this game
Enjoyment	I felt bored while playing the game
Enjoyment	I would likely recommend this game to others
Enjoyment	If this becomes a real game, I want to play it
Playability	I was able to understand the objective
Audio Aesthetics	I felt the game's audio enhanced my gaming experience
Audio Aesthetics	I think the game's audio fit the mood or style of the game
Personal Gratification	I was in suspense about whether I would succeed in the game
Personal Gratification	I felt successful when I overcame the obstacle in the game
Playability	I felt very confident while playing the game
Playbility	I was very frustrated while playing the game
Playbility	I felt the game provided me the necessary information to accomplish the goal
Play Engrossment	I lost track of time while playing the game

Table 7.3: GUESS questions used in the questionnaire [2]

To further help us in our evaluation, we believe the data extracted mentioned earlier can be quite useful:

- Tries - The answers given and how many each player tried can help us understand if the player was just throwing responses at random and how accurate the responses are with the expected outcomes
- Time left - How many seconds were left to the player can tell us if it's responses could've been under stress or the time was not enough for the experience of the player (too little time left) which we have to take into consideration when analysing the results.
- Success - Whether the player cracked the code or not can help us understand further the effectiveness of the technique, as a complement to the questionnaire's answers.

8

Results

Contents

Game	Try 1	Try 2	Try 3	Time Left (seconds)	Success
A	3,MK	3,NM	3,NK	496	No
A	2,MK	2,NM		410	Yes
A	3,NK	2,NM		498	Yes
A	3,MK	5,NM	2,MK	414	No
A	5,MK	3,MK	5,MK	231	No
A	2,NK	3,NM	2,NK	292	No
A	3,NM	3,MK	5,MK	240	No
A	5,MK	3,MK	3,NM	261	No
A	3,NK	3,MK	3,NM	436	No
A	3,MK	5,MK	3,MK	304	No
B	5,MK	5,MK	5,NK	433	Yes
B	5,MK	3,MK	5,NK	218	Yes
B	3,NK	3,MK	3,NK	444	No
B	2,NK	3,NK	5,MK	439	No
B	5,NK			493	Yes
B	3,NM	5,NK		483	Yes
B	5,MK	5,NK		53	Yes
B	3,NM	3,MK	2,MK	308	No
B	3,MK	3,NK	3,NM	390	No

Table 8.1: Research data extraction

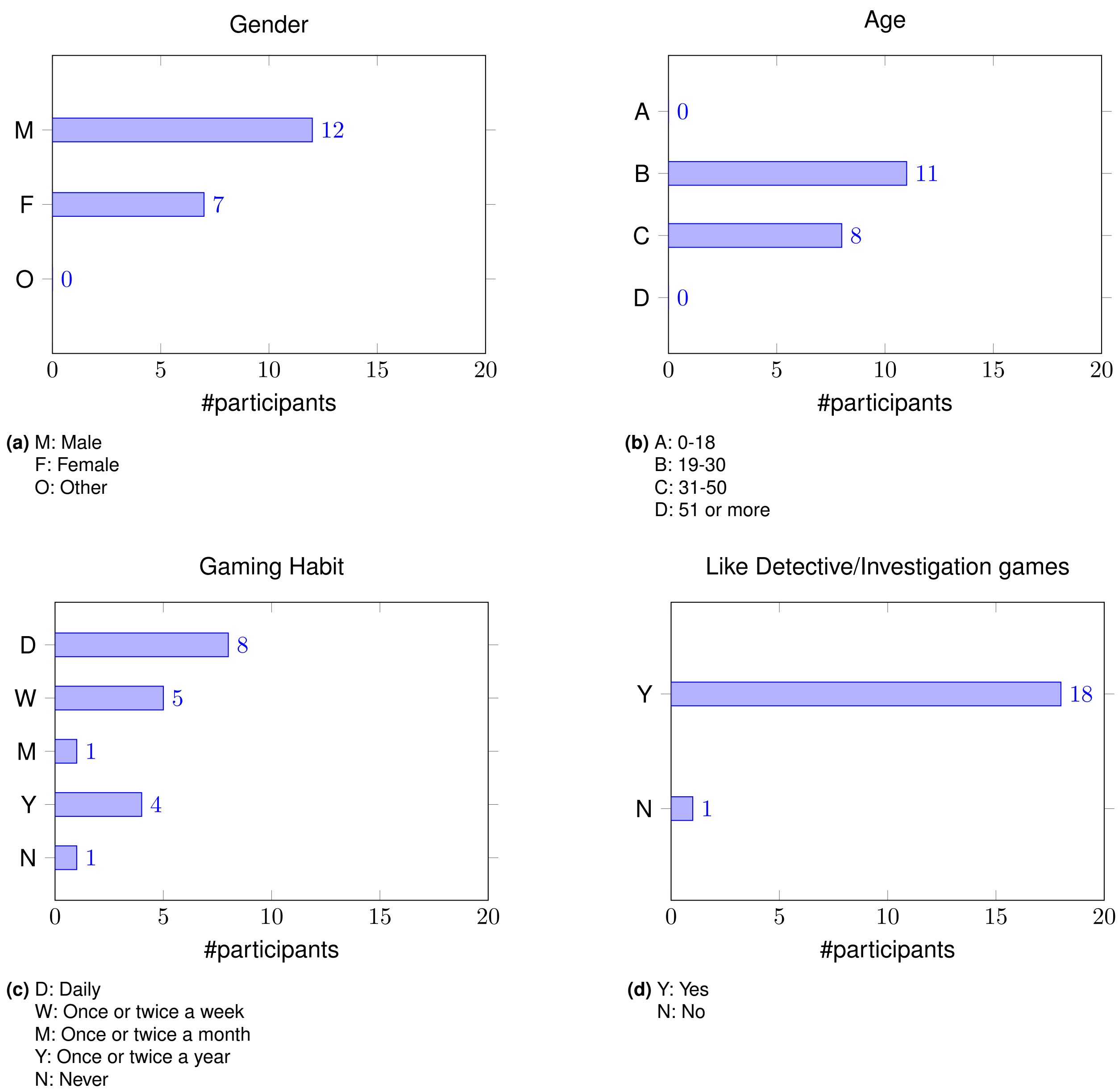
8.1 Experiment results

Up to the date of this dissertation, a total of 19 people participated in the experiment. In this section, we will be analysing the results, on the effectiveness of the technique studied with this research and the player's game experience.

In terms of demographics, the results are presented in Figure 8.1. Summarizing, roughly 2/3 of the participants are Male and 1/3 are Female, roughly half of the participants are aged between 19 and 30 years old and with the rest aged between 31 and 50 years old. 8 participants claim to be daily game players, while 5 state playing once or twice a month, 1 state playing once or twice a month, 4 state playing once or twice a year and there is even a single participant that claims to never (could be rarely as well) play games. Again, we see a good quantity of avid experienced players (about half the participants. Lastly, all participants claim to like detective/investigation games (including board games, escape rooms, etc.) with the exception of one participant who claims it does not. Still, as long as the experiment upholds its nature, the players should find the experience positive.

The data extracted from the game experiment, each version, is in Table 8.1 and plotted in Figures 8.2 and 8.3.

As for the results from the GUESS questionnaire, they are represented in Appendix B.1.



(a) M: Male
F: Female
O: Other

(b) A: 0-18
B: 19-30
C: 31-50
D: 51 or more

(c) D: Daily
W: Once or twice a week
M: Once or twice a month
Y: Once or twice a year
N: Never

(d) Y: Yes
N: No

Figure 8.1: Research Experiment Demographics

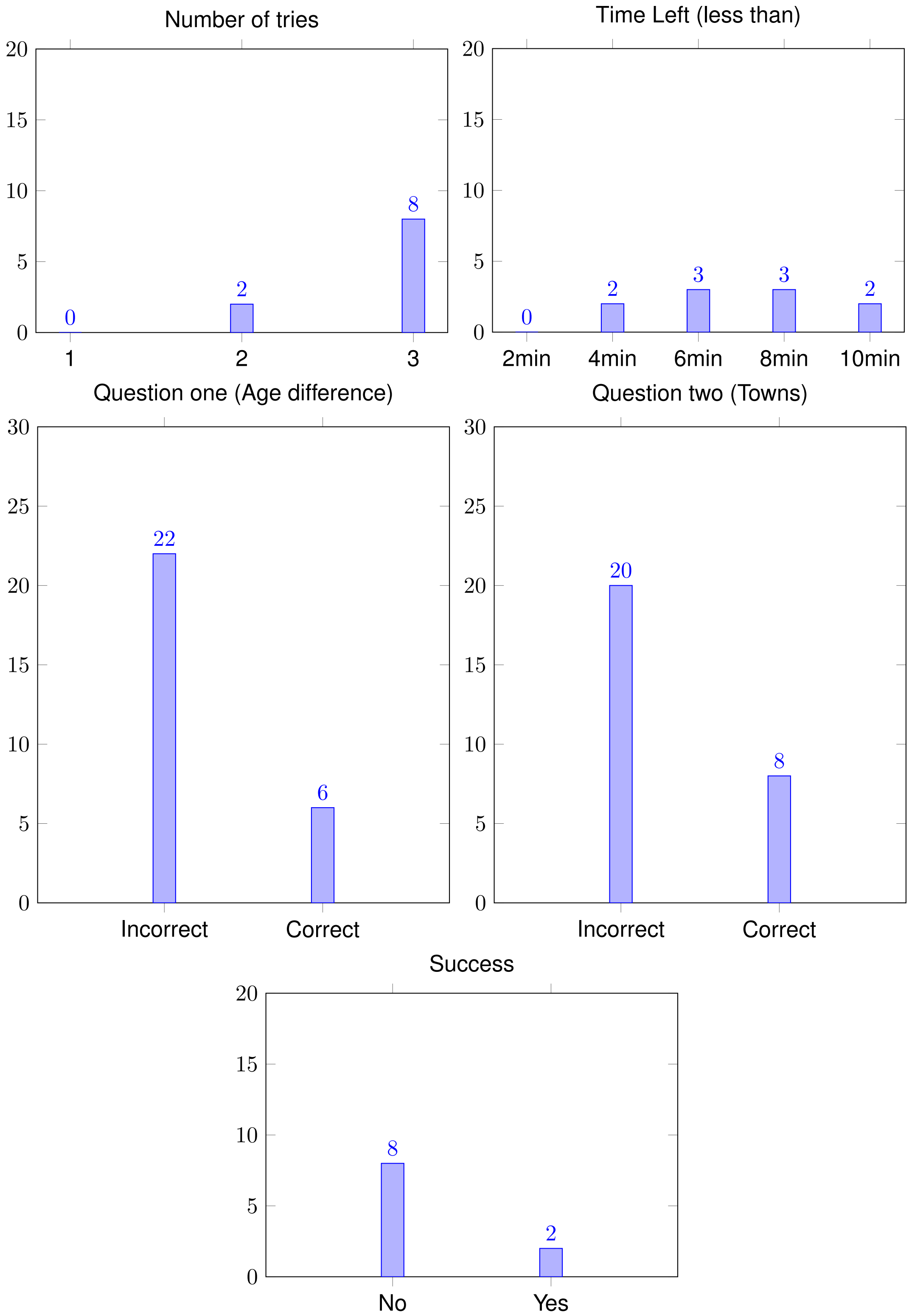


Figure 8.2: Research Game A Data Extraction

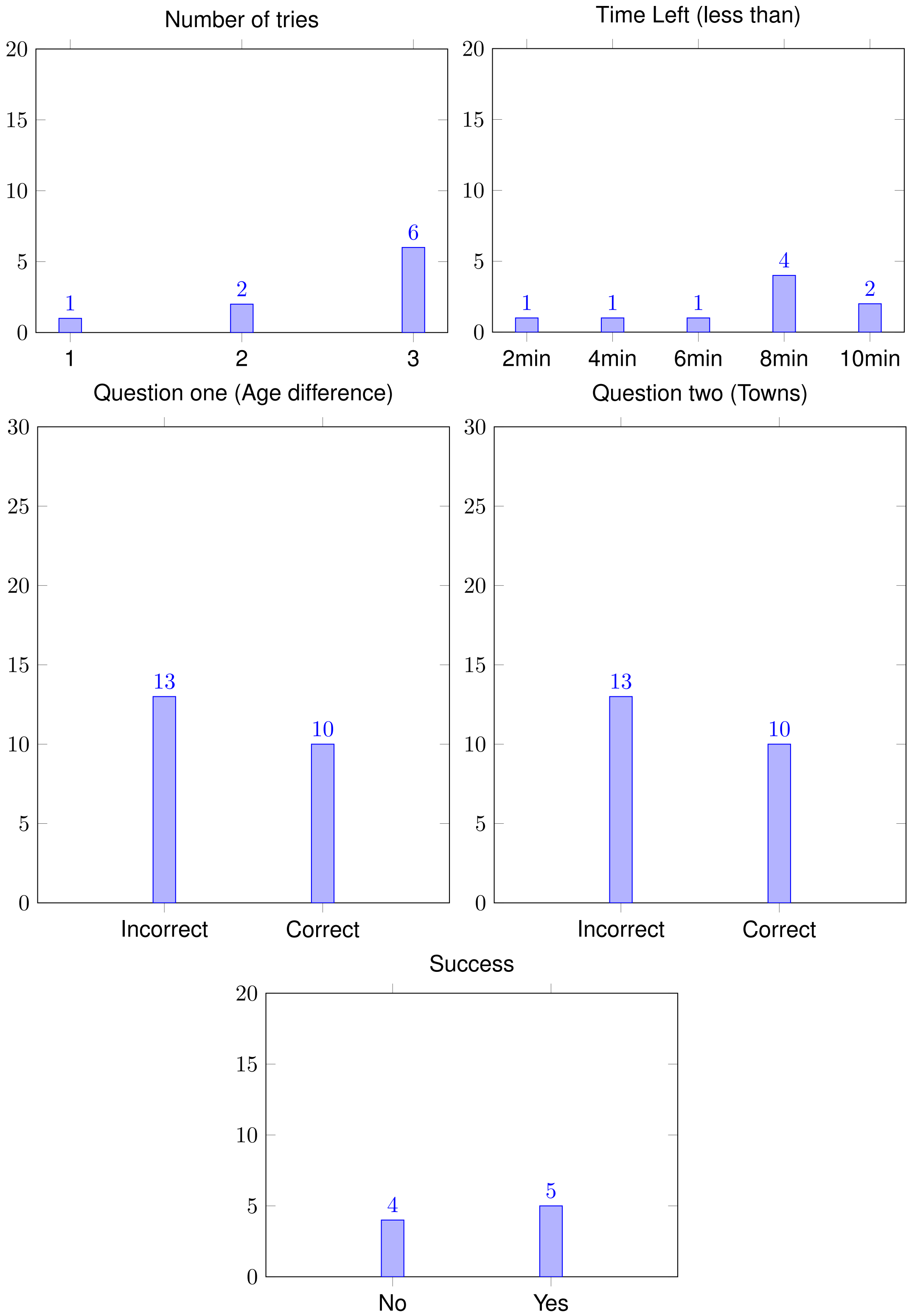


Figure 8.3: Research Game B Data Extraction

8.2 Results analysis

Let's look at the number of tries and time left first, since between both versions of the game, these results seem to be balanced.

8.2.1 Number of tries

About 3/4 of the participants had to give the answers in 3 tries and almost all the rest took 2 tries, with a single participant answering correctly on the first try. Taking into account that all participants finished the experiment, 2 tries or a single try means that these participants cracked the code with success. There is one caveat to this experiment which is that the player is only aware of the questions to answer by trying at least once. This means that, unless the player exhausts searching each and every piece of information, and records it well either in a support tool (such as a notepad) or even in their own mind, it is fairly common to have at least 2 tries from the player. This can be seen not only as a metric for the randomness in the responses, but as a way to understand if players are taking the gaming experience seriously.

Furthermore, of the 14 players that used up all 3 tries, only 2 finished the game successfully cracking the code. We may assume from this that it was not easy to understand the relationship state from the modulation in the music. The fact that we do not reset the parameter values between questions of different suspects, as stated in 7 could have an influence in this phenomenon, meaning that the system used to test the technique was not exactly the best. It does not necessarily invalidate the technique.

8.2.2 Time left

In general, the participants took between 2 and 4 minutes to finish the game experiment. With the success/failure results in mind, one possibility is that the participants did not take much time trying due to not being able to understand the relationship state was being represented by the music modulations. The quantity of information in the game (not necessarily fully useful) is somewhat big, which means that either the participant takes notes and does not feel the need to keep looking for more info or that maybe the participant did not collect the info necessary to answer the questions correctly. In actuality, from the GUESS questionnaire's question 14 (see Appendix B.1), the feeling that the game provided the necessary information to accomplish the goal is well divided between participants that think so and participant that don't.

8.2.3 Question answers and success

In terms of the question answers in game A, we can see that about 2/3 of the tries were incorrect for both questions. Also, only 2 players in 10 were able to crack the code. This means that it was very hard for the participants to reach the goal, consequently understand the relationship state through the alterations in the music.

For game B, the ratio is about 50/50, but for question answers and success cases. So it seems that this version was interpreted as easier than version A.

This difference does not seem very logical in terms of technique implementation since, as with the case of the music model experiment, these versions differ only on the relationships between the characters. The modulations applied to the different values of love and hate remain the same. So the increments and decrements of the parameter values in each version produces the exact same output. The only other factors we believe could be an influence are the particular relationships themselves between the suspects (player's own fondness/likeability of the characters) or, once more, the difficulty in maintaining the information as abstract as possible. Maybe one of the queries the player has available to ask in every hologram could unconsciously influence, such as the affiliation of the suspects (For example, Sonya says she is friends with David, though in both versions we should be able to assert hatred from Sonya towards David through the modulations).

9

Conclusion

There have been, and are in place, many studies on Video game music and Adaptive Soundtrack. Though there do not seem to be many studies on how relationship states can be represented in music properties, there is some content out there on how emotions relate to music properties and flavors. This research attempts something unseen before, which is to use Adaptive Soundtrack and have the modulations give feedback to the player on relationship states. There's a very gray area bridge that we were not able to get much information on, which is a correlation between relationship states and emotions, let alone relationship states and musical properties.

In the future, it should be interesting, not only in the gaming area, but in the musical and psychology areas as well, to further research on this topic. A great next step would be to do some joint research with someone from the psychology area to get a grander grasp of the issue at hand.

As for the results of this research, we can see that there is some influence in using this technique to provide the player with subliminal information on the relationship states between characters. However, due to various factors, such as the psychology and sociology predicament stated above, or the lack of experience on such themes by the authors of this research, the technique may not have been implemented in the most effective way.

Nonetheless, having seen that there is at least some influence, we believe that it would fruitful to have future contributions between the areas to develop an effective implementation of this new input technique in the gaming area to build even more interesting and artistic games.

Bibliography

[1] W. Chase, *How Music Really Works!: The Essential Handbook for Songwriters, Performers, and Music Students*. Roedy Black Pub., 2006.

[2] M. H. Phan, J. R. Keebler, and B. S. Chaparro, "The development and validation of the game user experience satisfaction scale (guess)," *Human factors*, vol. 58, no. 8, pp. 1217–1247, 2016.

[3] A. Gungormusler, N. Paterson-Paulberg, and M. Haahr, "barelymusician: An adaptive music engine for video games," in *Audio engineering society conference: 56th international conference: audio for games*. Audio Engineering Society, 2015.

[4] P. Lopes, A. Liapis, and G. N. Yannakakis, "Targeting horror via level and soundscape generation," in *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2015.

[5] X. Fu, *Horror movie aesthetics: How color, time, space and sound elicit fear in an audience*. Northeastern University, 2016.

[6] T. Garner, M. Grimshaw, and D. A. Nabi, "A preliminary experiment to assess the fear value of preselected sound parameters in a survival horror game," in *Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound*. ACM, 2010, p. 10.

[7] T. A. Garner, "from sinewaves to physiologically-adaptive soundscapes: The evolving relationship between sound and emotion in video games," in *Emotion in Games*. Springer, 2016, pp. 197–214.

[8] P. Lopes, A. Liapis, and G. N. Yannakakis, "Sonancia: Sonification of procedurally generated game levels," in *Proceedings of the 1st computational creativity and games workshop*, 2015.

[9] C. Plut and P. Pasquier, "Music matters: An empirical study on the effects of adaptive music on experienced and perceived player affect," in *2019 IEEE Conference on Games (CoG)*. IEEE, 2019, pp. 1–8.

[10] S. Lipscomb and S. Zehnder, "Immersion in the virtual environment: The effect of a musical score on the video gaming experience," *Journal of physiological anthropology and applied human science*, vol. 23, pp. 337–43, 12 2004.

[11] M. Yamada, "The effect of music on the performance and impression in a video racing game," *Journal of Music Perception and Cognition*, vol. 7, pp. 65–76, 2001.

[12] ——, "The effect of music on the performance and impression in a racing video game," in *Proc of the 7th International Conference on Music Perception & Cognition. Causal Productions, Adelaide, 2002*, 2002.

[13] L. Parmeter and M. Fendt, "Influencing the behavioral responses of players in an interactive narrative game through music and arousal congruency," in *2019 IEEE Conference on Games (CoG)*. IEEE, 2019, pp. 1–4.

[14] S.-L. Tan, J. Baxa, and M. Spackman, "Effects of built-in audio versus unrelated background music on performance in an adventure role-playing game," *IJGCMS*, vol. 2, pp. 1–23, 07 2010.

[15] J. H. Murray and J. H. Murray, *Hamlet on the holodeck: The future of narrative in cyberspace*. MIT press, 2017.

[16] J. Wirtz, A. S. Mattila, and R. L. Tan, "The role of arousal congruency in influencing consumers' satisfaction evaluations and in-store behaviors," *International Journal of Service Industry Management*, vol. 18, no. 1, pp. 6–24, 2007.

[17] I. X. Domínguez, R. E. Cardona-Rivera, J. K. Vance, and D. L. Roberts, "The mimesis effect: The effect of roles on player choice in interactive narrative role-playing games," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 3438–3449.

[18] Wikipedia contributors, "Tutti — Wikipedia, the free encyclopedia," 2020, [Online; accessed 26-December-2020]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Tutti&oldid=992828621

[19] J. De Rivera and C. Grinkis, "Emotions as social relationships," *Motivation and emotion*, vol. 10, no. 4, pp. 351–369, 1986.

A

Code of Project

Following is the code for the FMOD Studio API plugin written for RPG Maker MV.

Listing A.1: FMOD plugin for RPG Maker MV

```
//=============================================================================
// FMODStudioAPI.js
//=============================================================================

/*:
 * @plugindesc Integrates FMOD Studio HTML API with RPG Maker
 * @author Miguel Cunha
 */
const FMODStudioAPI = function() {

    var FMOD = {}; // FMOD global object which must be declared to enable 'main'
                   // and 'preRun' and then call the constructor function.
    FMOD['preRun'] = prerun; // Will be called before FMOD runs, but after the
                             // Emscripten runtime has initialized
    FMOD['onRuntimeInitialized'] = main; // Called when the Emscripten runtime
                                         // has initialized
    FMOD['TOTAL_MEMORY'] = 64 * 1024 * 1024; // FMOD Heap defaults to 16mb which
                                             // is enough for this demo, but set
                                             // it differently here for
                                             // demonstration (64mb)
    FMODModule(FMOD); // Calling the constructor function with our object

    var gLoveID, gHateID, gDominanceID;

    //=========================================================================
    // Example code
    //=========================================================================

    var gSystem; // Global 'System' object which has the Studio API functions.
    var gSystemLowLevel; // Global 'SystemLowLevel' object which has the Low
                         // Level API functions.
```

```
var gEventInstance; // Global Event Instance for the footstep event.
var gSurfaceInstance = 0;
var gAudioResumed = false; // Boolean to avoid resetting FMOD on IOS/Chrome
                           // every time screen is touched.
var gMasterChannels;

// Simple error checking function for all FMOD return values.
function CHECK_RESULT(result) {
    if (result != FMOD.OK) {
        var msg = "Error!!! '" + FMOD.ErrorString(result) + "'";

        alert(msg);

        throw msg;
    }
}

// Will be called before FMOD runs, but after the Emscripten runtime has
// initialized
// Call FMOD file preloading functions here to mount local files.  Otherwise
// load custom data from memory or use own file system.
function prerun() {
    var fileUrl = "soundbanks/";
    var fileName;
    var folderName = "/";
    var canRead = true;
    var canWrite = false;

    fileName = [
        "Master.bank",
        "Master.strings.bank"
    ];

    for (var count = 0; count < fileName.length; count++) {
        FMOD.FS_createPreloadedFile(folderName, fileName[count], fileUrl +
        fileName[count], canRead, canWrite);
    }
}

function main() {
    if (gSystem) {
        return;
    }
    // A temporary empty object to hold our system
    var outval = {};
    var result;

    console.log("Creating FMOD System object\n");

    // Create the system and check the result
    result = FMOD.Studio_System_Create(outval);
    CHECK_RESULT(result);

    console.log("grabbing system object from temporary and storing it\n");

    // Take out our System object
    gSystem = outval.val;

    result = gSystem.getCoreSystem(outval);
    CHECK_RESULT(result);

    gSystemLowLevel = outval.val;

    // Optional.  Setting DSP Buffer size can affect latency and stability.
    // Processing is currently done in the main thread so anything lower
    // than 2048 samples can cause stuttering on some devices.
    console.log("set DSP Buffer size.\n");
    result = gSystemLowLevel.setDSPBufferSize(2048, 2);
    CHECK_RESULT(result);

    // Optional.  Set sample rate of mixer to be the same as the OS output
    // rate.
    // This can save CPU time and latency by avoiding the automatic
    // insertion of a resampler at the output stage.
    // console.log("Set mixer sample rate");
    // result =
    //    gSystemLowLevel.getDriverInfo(0, null, null, outval, null, null);
    // CHECK_RESULT(result);
    // result =
    //    gSystemLowLevel.setSoftwareFormat(outval.val,
    //                                      FMOD.SPEAKERMODE_DEFAULT, 0)
    // CHECK_RESULT(result);

    console.log("initialize FMOD\n");

    // 1024 virtual channels
```

```
        result = gSystem.initialize(1024, FMOD.STUDIO_INIT_NORMAL,
                                    FMOD.INIT_NORMAL, null);
        CHECK_RESULT(result);

        gSystemLowLevel.getMasterChannelGroup(gMasterChannels);

        // Starting up your typical JavaScript application loop
        console.log("initialize Application\n");

        initApplication();

        // Set up iOS/Chrome workaround.  Webaudio is not allowed to start
        // unless screen is touched or button is clicked.
        function resumeAudio() {
            if (!gAudioResumed) {
                console.log("Resetting audio driver based on user input.");

                result = gSystemLowLevel.mixerSuspend();
                CHECK_RESULT(result);
                result = gSystemLowLevel.mixerResume();
                CHECK_RESULT(result);

                gAudioResumed = true;
            }

        }

        var iOS = /iPad|iPhone|iPod/.test(navigator.userAgent) &&
                  !window.MSStream;
        if (iOS) {
            window.addEventListener('touchend', resumeAudio, false);
        } else {
            document.addEventListener('click', resumeAudio);
        }

        // Set the framerate to 50 frames per second, or 20ms.
        console.log("Start game loop\n");

        return FMOD.OK;
    }

    // Helper function to load a bank by name.
    function loadBank(name) {
        var bankhandle = {};
        CHECK_RESULT(gSystem.loadBankFile("/" + name,
                     FMOD.STUDIO_LOAD_BANK_NORMAL, bankhandle));
    }

    // Called from main, does some application setup.  In our case we will load
    // some sounds.
    function initApplication() {
        console.log("Loading events\n");

        loadBank("Master.bank");
        loadBank("Master.strings.bank");

    }

    const sleep = (milliseconds) => {
        return new Promise(resolve => setTimeout(resolve, milliseconds))
    }

    const checkSystem = async () => {
        var systemUp = false;
        while (!systemUp) {
            systemUp = gSystem;
            await sleep(200);
        }
    }

    var _Game_Map_prototype_autoplay = Game_Map.prototype.autoplay;
    Game_Map.prototype.autoplay = async function() {
        var fmod = $dataMap.meta.FMOD;
        if ($dataMap.autoplayBgm) {
            if ($gamePlayer.isInVehicle()) {
                $gameSystem.saveWalkingBgm2();
            } else {
                if ($dataMap.bgm && $dataMap.bgm.name != "") {
                    AudioManager.playBgm($dataMap.bgm);
                } else if (fmod) {

                    await checkSystem();

```

```
                var eventDescription = {};
                CHECK_RESULT(gSystem.getEvent("event:/" + fmod,
                             eventDescription));

                if ($dataMap.fmod && $dataMap.fmod.parameters) {
                    for (var i in $dataMap.fmod.parameters) {
                        var param = $dataMap.fmod.parameters[i];
                        // Find the parameter once and then set by ID
                        // Or we can just find by name every time but by ID
                        // is more efficient if we are setting lots of
                        // parameters
                        var paramDesc = {};
                        CHECK_RESULT(eventDescription.
                                     val.getParameterDescriptionByName(
                                                                param.name,
                                                                paramDesc));

                        param.id = paramDesc.id;
                    }
                }

                var eventInstance = {};
                CHECK_RESULT(
                        eventDescription.val.createInstance(eventInstance));
                gEventInstance = eventInstance.val;

                if ($dataMap.fmod && $dataMap.fmod.parameters) {
                    for (var i in $dataMap.fmod.parameters) {
                        var param = $dataMap.fmod.parameters[i];

                        // Make the event audible to start with
                        CHECK_RESULT(
                            gEventInstance.setParameterByID(param.id,
                                                            param.value,
                                                            false));
                    }
                }

                $dataMap.fmod.eventInstance = gEventInstance;
                $dataMap.fmod.checkResult = CHECK_RESULT;
            }
        }
    }
    if ($dataMap.autoplayBgs) {
        AudioManager.playBgs($dataMap.bgs);
    }
};

var _Scene_Map_prototype_start = Scene_Map.prototype.start;
Scene_Map.prototype.start = async function() {
    _Scene_Map_prototype_start.call(this);
    if ($dataMap && $dataMap.meta.FMOD) {

        await checkSystem();

        var state = {};
        CHECK_RESULT(gEventInstance.getPlaybackState(state));
        if (gSystem && gEventInstance &&
            state.val != FMOD.STUDIO_PLAYBACK_PLAYING) {
            CHECK_RESULT(gEventInstance.start());
        }
    }
};

var _Scene_Base_prototype_update = Scene_Base.prototype.update;
Scene_Base.prototype.update = function() {
    if (gSystem && gEventInstance && $dataMap) {

        // Update FMOD
        result = gSystem.update();
        CHECK_RESULT(result);
    }
    _Scene_Base_prototype_update.call(this);
};

/*SceneManager.goto = function(scene) {


}*/

SceneManager.goto = function(sceneClass) {
    if (sceneClass) {
```

```
            this._nextScene = new sceneClass();
        }
        if (this._scene) {
            if ($dataMap && $dataMap.meta.FMOD &&
                 (this._nextScene instanceof Scene_Map ||
                  this._nextScene instanceof Scene_Gameover) &&
                this._scene instanceof Scene_Map) {
                if (gSystem && gEventInstance) {
                    CHECK_RESULT(
                        gEventInstance.stop(FMOD.STUDIO_STOP_IMMEDIATE));
                }
            }
            this._scene.stop();
        }
    };


};

FMODStudioAPI();
```

Following is the code for the FMOD API Relationship plugin written for RPG Maker MV.

Listing A.2: FMOD API Relationship plugin for RPG Maker MV

```
//=============================================================================
// FMODStudioAPI_Relationship.js
//=============================================================================

/*:
 * @plugindesc Makes use of a Relationship state variable for FMOD
 * @author Miguel Cunha
 */

(function() {

    var eventInstance;
    var checkResult;
    var parameters;

    setupFMOD = function() {
        if($dataMap.meta.FMOD) {

            $dataMap.fmod = {
                parameters: [
                    {
                        name: "Love",
                        id: 0,
                        value: 0.5
                    },
                    {
                        name: "Hate",
                        id: 0,
                        value: 0.5
                    }/*,
                    {
                        name: "Dominance",
                        id: 0,
                        value: 0.5
                    }*/
                ]
            }
        }
    };

    var _Game_Map_prototype_autoplay = Game_Map.prototype.autoplay;
    Game_Map.prototype.autoplay = function() {
        setupFMOD();
        _Game_Map_prototype_autoplay.call(this);
    };

    /*var _Scene_Map_prototype_start = Scene_Map.prototype.start;
    Scene_Map.prototype.start = function() {
        _Scene_Map_prototype_start.call(this);
    };*/

    var _Game_Interpreter_prototype_pluginCommand =
                                Game_Interpreter.prototype.pluginCommand;
    Game_Interpreter.prototype.pluginCommand = function(command, args) {
        if(!eventInstance && $dataMap.fmod && $dataMap.fmod.eventInstance) {
            eventInstance = $dataMap.fmod.eventInstance;
```

```
                checkResult = $dataMap.fmod.checkResult;
                parameters = $dataMap.fmod.parameters;
            }
            if(command == 'relationship' && eventInstance) {
                for(var i in parameters) {
                    if(args[i][0] == '$') {
                        args[i] = eval(args[i]) / 100;
                    }
                    var result =
                        eventInstance.
                            setParameterByID(parameters[i].id,
                                             Math.max(0,
                                                      Math.min(
                                                           parseFloat(args[i]),
                                                           1.0)), false);
                    checkResult(result);
                }
            }
            _Game_Interpreter_prototype_pluginCommand.call(this, command, args);
        };

})();
```

B

GUESS Questionnaire Results

Number	Question	Median	Average
1	I think the game was fun	4	3.58
2	I enjoyed playing this game	4	3.79
3	I felt bored while playing the game	2	2.32
4	I would likely recommend this game to others	3	3.37
5	If this becomes a real game, I want to play it	4	3.37
6	I was able to understand the objective	4	3.89
7	I enjoyed the music in the game	4	3.47
8	I felt the game’s audio enhanced my gaming experience	4	3.47
9	I think the game’s audio fit the mood or style of the game	4	3.53
10	I was in suspense about whether I would succeed in the game	4	4.21
11	I felt successful when I overcame the obstacle in the game	3	3.21
12	I felt very confident while playing the game	3	2.95
13	I was very frustrated while playing the game	3	2.53
14	I felt the game provided me the necessary information to accomplish the goal	4	3.47
15	I lost track of time while playing the game	2	2.58

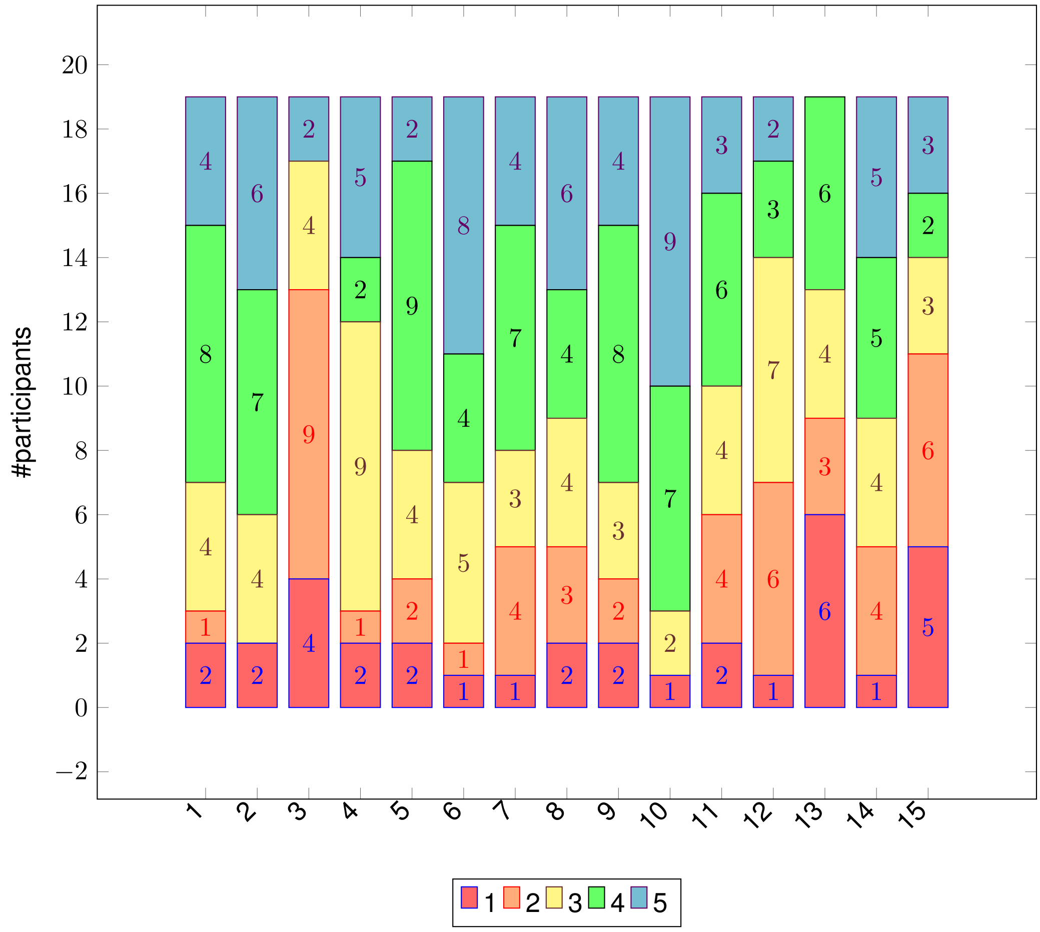


Figure B.1: GUESS Questionnaire results
With 1 as "Not at all" and 5 as "Without a doubt"