Adaptive Music in Narrative Videogames based on individual relationships

Miguel Cunha
Instituto Superior Técnico
Lisboa, Portugal
miguel.cunha@tecnico.ulisboa.pt

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.

INTRODUCTION

Adaptive Soundtrack is a technique that makes soundtrack dynamic in response to certain triggers or events, only truly achievable in the Videogame Industry[1]. Many videogame 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[2], 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 playrun. 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 RPGs (Role-Playing Games), 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.

BACKGROUND

Videogames nowadays adopt very intensively the art of music, with benefits and properties discussed in many studies[3]. 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.*[1] series or *The Legend of Zelda*[2] series.

The soundtrack in videogames can have all kinds of purposes, apart from being one of the fundamental creativity game facets[3][4]. 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[4] 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.

[1]Super Mario Bros. theme
https://www.youtube.com/watch?v=NTa6Xbzfq1U
[2]The Legend of Zelda theme
https://www.youtube.com/watch?v=cGufy1PAeTU
[3]The six creativity facets identified in games being: visuals, audio, narrative, ludus, level architecture and gameplay
[4]Tomb Raider - Secret found sound
https://www.youtube.com/watch?v=d8eskuLLkk4

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, changing melodies with the passing of the game only, like with 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 the type or number of instruments being played, the pace of the music or the volume of the notes.

Some examples of games that became notorious for making use of this kind of technique are Banjo Kazooie by Rare, Mario Kart 8 by Nintendo and Metal Gear Rising: Revengeance by Platinum Games. Environment changes (Banjo Kazooie), reaching first place or running the final lap (Mario Kart 8), emphasizing critical moments in battles (Metal Gear Rising), these are just some of the most usages of adaptive soundtrack.

Apart from changes in the same melody (also known as vertical mixing), time translation between melodies is another subtype of adaptive soundtrack, called horizontal mixing. 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[5], where for the different states (Alert or Evasion), different tunes play with minor similarities between them, such as a background beat or the pacing.

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.

This work's objectives is to use adaptive soundtrack, taking the advantages above into account, in a narrative style videogame. Many Narrative style videogames 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.

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.

Development Tools

RPG Maker MV

The RPG Maker Game Engine[6] 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 UI offers many features such as:

- Visual Map design
- Map location based events
- Easy Game Database editing

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

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

- Character generation
- Different Battle Systems
- Javascript Plugin Management

An 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 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.

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[7]
- Audiokinetic's Wwise[8]

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.

FMOD Studio

FMOD Studio is an end-to-end solution for adding sound and music to any game[9]. It features two main components:

- Workstation authoring tools
- End-platform audio engine

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. 1).

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.

[7]FMOD Studio webpage - https://www.fmod.com/
[8]Wwise webpage - https://www.audiokinetic.com/products/wwise/
[9]Statement taken from the website - https://www.fmod.com/studio

Figure 1. FMOD Studio's Authoring Tools

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 API, the game can communicate with the audio engine to exchange information (mainly, send values for the parameters to the audio engine).

Integration of the Engines

Despite FMOD Studio having some 3rd party integrations 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 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 improve-

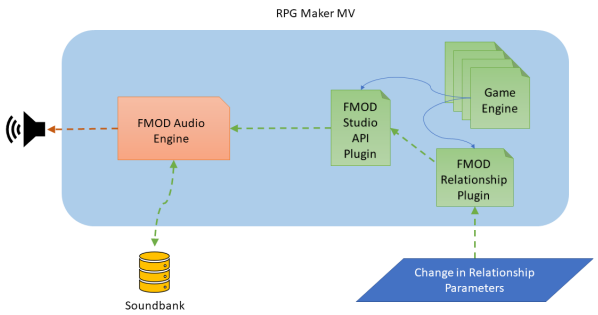

Figure 2. Integration of the FMOD Audio Engine with RPG Maker MV

ment 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 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 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[10] "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. The pseudo code for this plugin is presented in Listing 2. The overall integration architecture is represented in Figure 2.

Listing 2. FMOD Studio Relationship plugin pseudo code

```
def FMODParameters;

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

MUSICAL CONTENT

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[11], professor António Laertes. We scheduled an appointment to share our vision and needs and establish such a partnership.

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

The team created some small tracks with different instruments to understand if they were going on the right direction.

In total, three Tutti[5] were developed and were assigned the letters A[12], B[13] and C[14]. After thorough inspection, our remarks on each of the tracks were:

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.

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

[11] Henceforth referred to as Conservatório Regional de Setúbal or CRS

[12] https://soundcloud.com/user-901650047/a-a-ventura/s-ZgSmLzbp9YF

[13] https://soundcloud.com/user-901650047/b-a-detective-story/s-ZLghifWHK3k

[14] https://soundcloud.com/user-901650047/c-quazyum/s-4TqQTMmHvZi

Final composition

Taking everything defined previously into account, the CRS team created the final composition[15]. 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 videogame soundtrack.

The CRS team handed us each of the instrument tracks individually in the 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.

RELATIONSHIP STATE TO MUSIC CONCEPT MODEL

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.

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 the Introduction. 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[6]. 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.

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[7]. 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)

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[16] 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.

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

[16] 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 3. Music Model experiment's map

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 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, a multi-track song from the Mixing Secrets Cambridge Music Technology website[17]. The song is called "Alex the Adventurer" by Jeffrey Hayat.

Experiment website

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 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.

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

Letter	Talk about Priscilla	Talk about Sophia
A	Love = 1 & Hate = 0	Love = 0 & Hate = 1
B	Love = 0 & Hate = 1	Love = 1 & Hate = 0

Table 1. Adaptive Soundtrack versions and action consequences

Letter	Talk about Priscilla	Talk about Sophia
A	No modification	No modification
B	No modification	No modification

Table 2. Non-adaptive Soundtrack versions and action consequences

All in all, there are a total of 4 games[18], described in Tables 1 and 2, the server is able to pick from to present to the player.

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.

Demographics

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).

Game Results

As for the results of the experiment, represented in Figures 4 and 5, 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

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

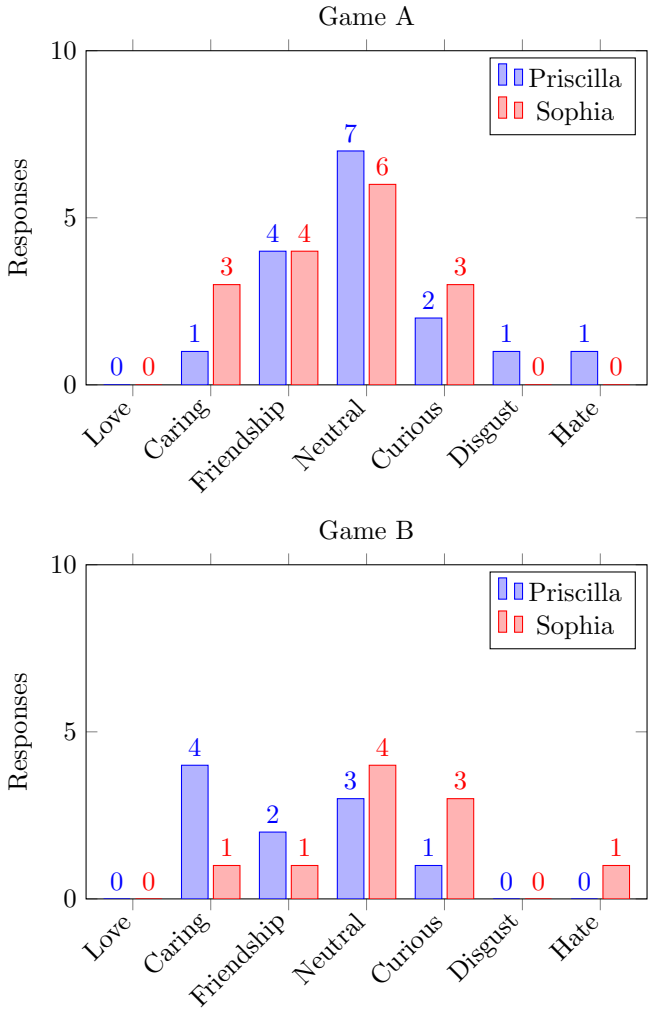

Figure 4. Non-adaptive soundtrack version results

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.

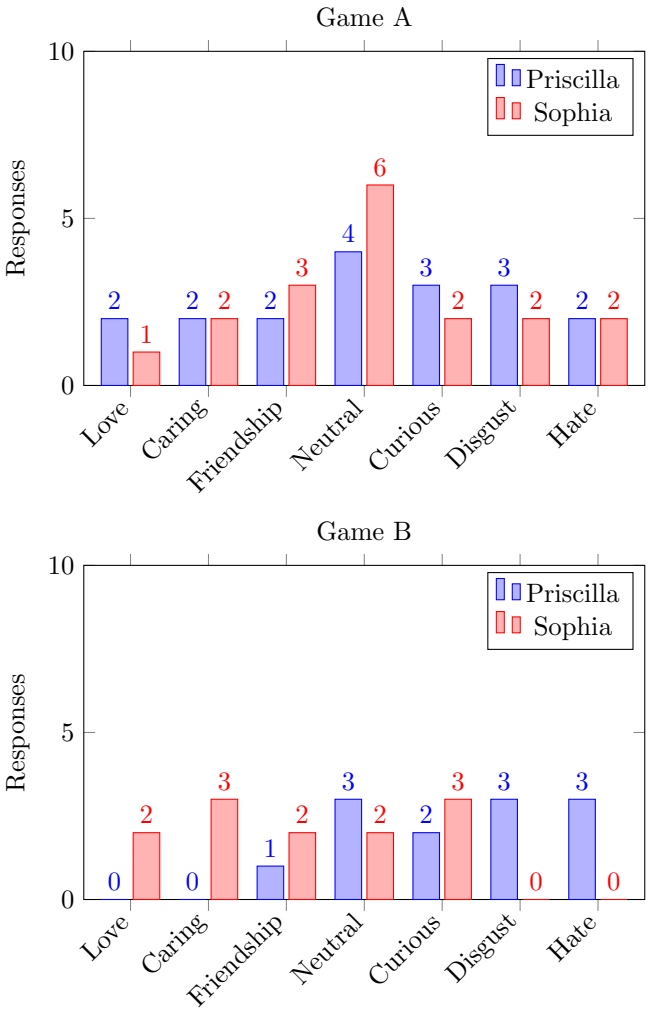

Figure 5. Adaptive soundtrack version results

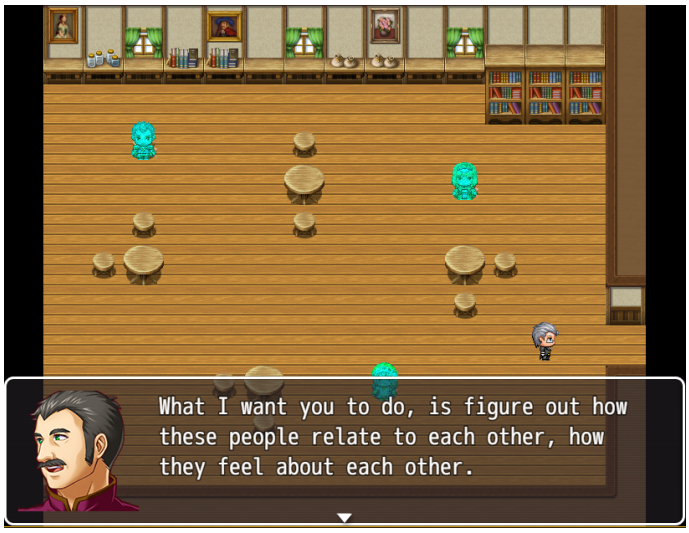

Figure 6. Symphony of Mystery's game map

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
- 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.

EXPERIMENT AND EVALUATION METHODOLOGY

Symphony of Mystery

The experiment consists of a simple minigame 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 their name, age and 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

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[19]:

- 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

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.

Experiment website

"Symphony of Mystery" was published online[20] and advertised for people to participate and play. 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 3 and 4.

[19]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

[20]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 3. 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 4. Relationships in game B

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*

Evaluation Methodology

In the end of the website wizard, the participant is presented with a link to a Google Form[21] 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.[8]. 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. 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.

[21] shorturl.at/rBGKS

RESULTS

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, 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.

Results analysis

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 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.

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.

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).

The GUESS questionnaire results are represented in Figure 7 in Annex A.

CONCLUSION

There have been, and are in place, many studies on Videogame 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

REFERENCES

1. 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.
2. 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.
3. 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.
4. 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.
5. 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
6. J. De Rivera and C. Grinkis, "Emotions as social relationships," *Motivation and emotion*, vol. 10, no. 4, pp. 351–369, 1986.
7. W. Chase, *How Music Really Works!: The Essential Handbook for Songwriters, Performers, and Music Students*. Roedy Black Pub., 2006.
8. 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.

ANNEX A - GUESS RESULTS

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

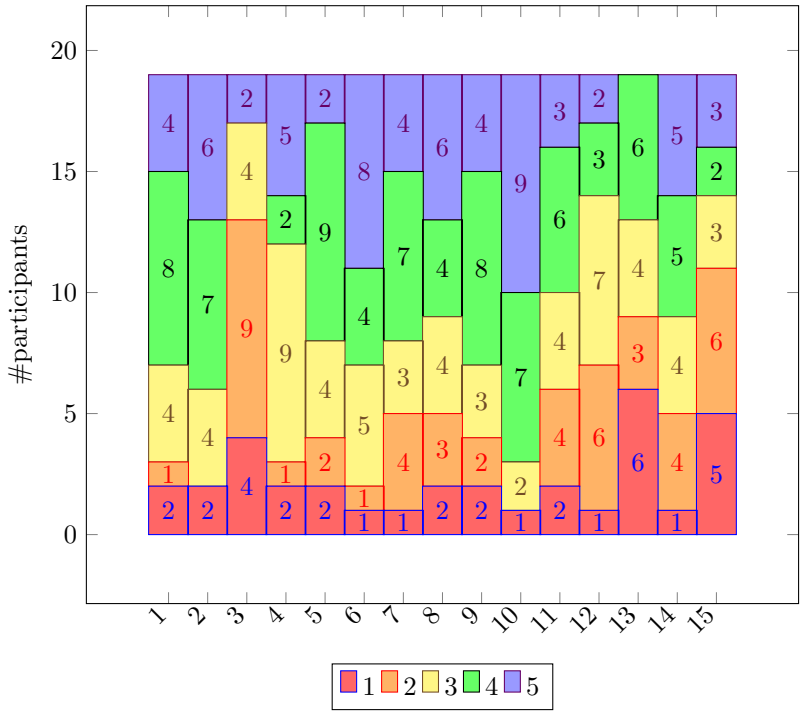

Figure 7. GUESS Questionnaire results
With 1 as "Not at all" and 5 as "Without a doubt"