

# NONLINEAR MUSIC DESIGN IN NARRATIVE GAMES

Danae Dekker  
Music & Technology  
Hogeschool voor de Kunsten Utrecht

*Supportive Narrative*  
*8 June 2020*

## **ABSTRACT**

The research in this article explores in what ways nonlinear music design is used in narrative video games, i.e. games where narrative storytelling plays a major role in their gameplay. The article covers this music design from two perspectives: first, the function of the music in relation to the story and setting, and second, how the implementation of this design conveys that role. In a critical review common techniques for creating nonlinear music are discussed. Moreover, these techniques are used to describe the music design in several existing narrative games in relation to the role of the music. Afterwards a design strategy is created based on the findings of this critical review. The author discusses how they use this strategy in the music design for three of their own narrative projects. The goal of this exploration is to get a better understanding of the techniques used in creating a music design for a narrative game, and then apply these in current and upcoming projects.

## CONTENTS

1. Introduction .....	4
2. Critical review: music design in existing games .....	5
2.1. Common interactive music techniques.....	5
2.2. Rayman 2: The Great Escape .....	6
2.3. The Elder Scrolls V: Skyrim .....	8
2.4. Life Is Strange.....	9
2.5. Celeste.....	11
2.6. Conclusion.....	14
3. Application: music design in my own projects .....	15
3.1. Implementation methods.....	15
3.2. Pien en de Vissensteen .....	16
3.3. Invisible Wings.....	18
3.4. Ocarime: The Game .....	21
4. Conclusion and reflection .....	23
References.....	24

## 1. INTRODUCTION

Apart from writing music and developing software, I enjoy storytelling and building often fantastical worlds to accompany these stories. Within the field of music and technology my specialty lies in composition and technical audio design for interactive media, in which I combine the things I enjoy doing and try to turn them into my profession. I envision that music can strengthen and add to the immersion in a story and world. This is the main reason why I love to work on music and audio integration for narrative games, i.e. a video game where narrative storytelling plays a major role in its gameplay.

In this supportive narrative, I review the music design of several existing narrative games. Afterwards, I describe and reflect on the music design in three of my own projects that are part of my graduation portfolio. In the discussion of these items I look into music design from two perspectives. First, I go over the role that music takes on in a video game in relation to the underlying narrative. Secondly, I review which adaptive music techniques are used to create a music system that conveys this role.

Discussing composition techniques that are used in the music does not fall under the scope of this article, but they can nevertheless be relevant for decision making in the music design. In my critical reviews I will also not go into detail about which methods are used to implement these techniques in code, as these tend to be dependent on the requirements of the projects and the available tools at the time. I will however take these implementation methods into account when discussing my own projects, because they are part of my own music design process as well.

From the findings of the critical review and the reflections on my own projects I expect to gain more insight in what roles music can fulfill in narrative games and how adaptive systems convey this role. This new found knowledge can in turn be used to adapt my own music design process for future projects.

In the next chapter I critically review the role of the music in existing narrative games and how the implementation of the music design facilitates that. Afterwards, in chapter 3, I review the music design of my own projects, using the findings from the critical review. Lastly, chapter 4 gives a conclusion of this article.

## 2. CRITICAL REVIEW: MUSIC DESIGN IN EXISTING GAMES

In this critical review I describe the role of music in existing narrative games, and how this is implemented using the techniques explained by Michael Sweet in his article *Top 6 Adaptive Music Techniques in Games – Pros and Cons*. In the context of the critical review, I focus on narrative games which feature "traditional" gameplay—or at least use gameplay elements to immerse the player into the game's story. This gameplay can be linear, where the player works through the story in a fixed order, as well as nonlinear, where the player has choice in which order they unfold the narrative. This is opposed to games such as visual novels, which do not feature interactive gameplay apart from selecting choices and clicking through the dialogue.

The four games that I have reviewed are *Rayman 2: The Great Escape*, *The Elder Scrolls V: Skyrim*, *Life Is Strange* and *Celeste*. Apart from the fact that these are all games I enjoy playing, these particular games are chosen to feature a broad spectrum of roles that music can take on in narrative games:

- *Celeste* features music that is tightly coupled to the narrative events;
- *Life Is Strange* features music that is not tightly coupled to the narrative, but mostly reflects the characters and their identities;
- *Rayman 2: The Great Escape* and *The Elder Scrolls V: Skyrim* feature music that define the setting and mood of the story.

In video games in general there are other roles that the music can take on. As an example, some levels from *Rayman Legends* feature music that is one to one coupled to the gameplay and guides the player through the level. In *Guitar Hero* and *Rhythm Tengoku* the gameplay is even centered around the music, instead of vice versa, which are therefore dubbed "music games". Because these musical roles are generally not used in narrative games in the context of this research, these roles are not discussed further in the critical review.

First I give a brief summary of Sweet's adaptive music techniques, and in the sections that follow the four mentioned games are reviewed in terms of music design.

### 2.1. COMMON INTERACTIVE MUSIC TECHNIQUES

According to Michael Sweet (Sweet, 2016; Bonkerk, 2019) there are six common methods of creating interactivity in music that consist of pre-composed material, which can be divided in two categories: **vertical remixing**, which uses multiple layers that can be toggled on or off independently depending on the interactions, and **horizontal resequencing**, which uses different musical cues that are triggered one after another. There are five different techniques to implement the horizontal transitions between musical cues, which are:

- crossfading two musical cues, where one cue fades out and the other fades in at the same time;

- phrase branching, where the start of the next cue waits until the musical phrase in the current cue is finished;
- musical demarcation branching, where the start of the next cue occurs only at demarcation points such as a beat or measure;
- bridge transition, where a separate (short) musical cue is used as a transition between the current and next cue for a more seamless transition;
- stinger-based sequencing, where the cues are composed as short stingers that are played back based on game events and have generally no underlying rhythmic connection.

These terms are from now on used to describe the implementation of musical cues in the reviewed games and in my own projects.

## 2.2. RAYMAN 2: THE GREAT ESCAPE

*Rayman 2: The Great Escape* (Ubi Pictures, 1999) is a 3D platforming video game published by Ubi Soft. The game features the titular limbless hero Rayman as the protagonist. The story of *Rayman 2* takes place in the Glade of Dreams, a magical world that was created by the god Polokus. Shortly prior to the events of the game the Glade is invaded by an army of Robo-Pirates, with the intent to destroy the world and enslave its inhabitants. During the game Rayman must get rid of the invading Pirates and find four ancient masks to wake up Polokus to aid Rayman in his journey.



The reason this particular game was reviewed is because its story and vast world are great inspirations for my own fictional works and stories, especially the concepts of magic used in the world. From time to time I replay *Rayman 2* for its world building and setting.

### 2.2.1. MUSIC DESIGN

Most of the soundtrack of *Rayman 2*, which is composed by Eric Chevalier, seems to be designed to accompany the world and setting of the game instead of the story or characterization. In this way acts more like a film score than other platforming game music from that time, such as *Super Mario 64*.

According to reviews of *Rayman 2* "[the music] focuses a lot more on creating an atmosphere" (Calcom, 2018) and "[the game has] one of the most cinematic and kick-ass soundtrack for something as innocent [...] as Rayman" (Caddicarus, 2018).

Almost every level features its own distinctive music. This means that the soundtrack that is playing is in first place dependent of Rayman's location in the world, i.e. the current level. Usually the soundtrack for a level is composed of different coherent cues, that are triggered based on events in the level design. These transitions are usually limited to new bars or cadences in the music, an example of the horizontal resequencing using demarcation branching. Since *Rayman 2* is a platformer with plenty of action, this delay is not inferior to the playing experience.

Apart from soundtracks that are tied to the level design, the game also contains music that is purely played based on the events that occur in the level, which is accomplished by using horizontal resequencing using crossfades. The best example is the collection of cues that only play while Rayman is fighting a Robo-Pirate. When such a foe appears in the level, the level music fades out quickly and is replaced by the Robo-Pirate music, until the foe is defeated.

#### 2.2.2. EXAMPLE: THE SANCTUARY OF WATER AND ICE

A good example of the systematic use of music is the soundtrack of the level *The Sanctuary of Water and Ice*. As the name suggests, this level contains a sanctuary where one of the masks of Polokus is located. The sanctuary is located on a tropical island characterized by its cliffs. Rayman must find two magical spheres of different colors in the level and put them on bases of the same color, in order to open a strange door that forms the entrance to the temple.

The soundtrack of the first part of this level mimics a tropical style that fits the aesthetic of the environment. The music is composed of three distinguishing parts. The level opens with no music, until a Robo-Pirate appears out of the ground. During the battle with the foe, the battle variant of the soundtrack plays until they are defeated. From then on the first variant of the soundtrack is triggered using a crossfade to introduce the player to the setting. Later in the level Rayman stumbles on another Robo-Pirate, where the battle soundtrack is played as well. It is worth mentioning that in *The Sanctuary of Water and Ice* the Robo-Pirate music is incorporated in the level music itself, in contrast to most other levels, where one of the special Robo-Pirate cues is used.

As soon as Rayman finds one of the two magical spheres, the soundtrack transitions with the use of demarcation branching to a more progressive chord scheme. Through these chords the cue emphasizes the importance of those spheres to open the door to the sanctuary. This variant plays until the sanctuary is entered, and thus this part of the level is finished.

### 2.2.3. DISCUSSION

In my opinion, the music system of *Rayman 2: The Great Escape* is a great example of a system that uses musical cues to deliver the mood and setting of the story. A point of improvement would be to make the transitions between musical cues smoother and more integrated with the level design. However, the used techniques, i.e. horizontal resequencing using crossfades and phrase branching, don't fall short to the gaming experience, especially for a game from an era where cinematic game music was still in its infancy.

## 2.3. THE ELDER SCROLLS V: SKYRIM

*The Elder Scrolls V: Skyrim* (Bethesda Game Studios, 2011) is a 3D action role-playing game published by Bethesda Softworks. The story of *Skyrim* continues upon the lore of previous entries in the series and takes place in the eponymous province of the continent Tamriel, which is engaged in a civil war. The player takes the role of the Dragonborn, giving them ancient magical powers by absorbing the souls of dragons, on their quest to defeat Alduin, a dragon who is prophesied to destroy the world. Due to the open-world design of *Skyrim* the player can travel anywhere in the world at any time and even ignore or postpone the main storyline indefinitely.



I reviewed *Skyrim* because it has a vast and open world in which narrative events can occur randomly, which could lead to interesting music implementations. Moreover *The Elder Scrolls* series features lots and lots of worldbuilding and lore, which is an inspiration for my own projects.

### 2.3.1. MUSIC DESIGN

The soundtrack of *Skyrim* is composed by Jeremy Soule, who worked on *The Elder Scrolls IV: Oblivion*, the previous entry in the series, as well. The soundtrack is in the first place composed to set an atmosphere for the setting, and thus denotes few to no narrative elements. There are a few cues that are only used in certain parts of the world where the player only travels to once in the main quest, but in the overall music design these are neglectable.



In an interview (Noclip, 2018) Mark Lampert, the audio director and sound designer for the game, states that together with Jeremy Soule he worked out a system for the music, which they called "palette music". First Soule composed tracks as he normally would, based on the environments of the game, such as forests, mountains and different types of dungeons. Afterwards he and Lampert split the tracks in separate stems, for example only the separated woodwinds or the drums of a track. The stems are then set up in a simple system that plays the cues back randomly with little breaks in between, but still based on the environment the player is facing. Lampert states: "[This is done] so you getting the color and kind of the flavor and hopefully the vibe of some of those exploration music cues." This music system is an example of horizontal resequencing with a bridge transition, but instead it uses the absence of music as transitional cue.

Apart from the "narrative music", as Soule and Lampert call the environmental music, the system also contains various cues that are used as "little breathers in between." Lampert: "It's the kind of music you might have in a scene in a movie where there's heavy dialogue and I don't want it to get in the way and get all dramatic."

Aside from this system Lampert denotes that there are few scripted events beyond the combat music, which is usually triggered shortly before the player engages in a fight using horizontal resequencing with crossfades. He does state however that the system for the combat music has become more sophisticated since the release of *Oblivion*: "We now can kind of gage what you're in combat with [and] your relative combat abilities, and play nothing [...] or play something that's a little more subdued."

### 2.3.2. DISCUSSION

The approach of creating a music system to set an atmosphere is quite logical in an open-world game like *Skyrim*, because the player has a lot of choice which quests they embark on and the narrative design is heavily nonlinear. Designing a system that reacts to the narrative events would require a more complex system and may more musical cues, which may not be ideal when composers and system designers are tied to time and budget constraints.

Although the soundtrack of *The Elder Scrolls V: Skyrim* is not tied to the narrative, in my opinion it does contribute to the immersion in the world and its lore, because of its atmospheric nature.

## 2.4. LIFE IS STRANGE

*Life Is Strange* (Dontnod Entertainment, 2015b) is a 3D graphic adventure game published by Square Enix. The game revolves around Max Caulfield, an 18-year old photography student who discovers that she has the ability to rewind time at any moment. She discovers this ability when she witnesses the death of her childhood friend Chloe Price, and reverses time to save her from being killed. This action however sets a huge chain of events and

alternate realities in motion, which inevitably lead to the destruction of their hometown, Arcadia Bay.



Throughout the story, the actions of the player will adjust the narrative as it unfolds, and reshape it when travelling back in time, making heavy use of the butterfly effect. The story and character arcs are the main focus of the game. Dialogue trees, fetch quests and making changes in the environment serve as the puzzle solving mechanics.

I chose to review *Life is Strange* because of its heavy focus on the narrative, as opposed to more traditional adventure games that rely more on puzzle solving. Moreover I envision my own future projects to be graphic adventure games as well in the same spirit as this game, for which it is a huge source of inspiration in terms of narrative and design.

#### 2.4.1. MUSIC DESIGN

The music used in *Life is Strange* is a mix of several licensed tracks and an original soundtrack composed by Jonathan Morali. Jean-Maxime Moris, the creative director of *Life is Strange*, states in an interview with DualShockers (Taveras, 2014) that the music is a huge part of the experience and atmosphere of the game. To tie into the setting of the narrative, the team chose for a soundtrack that is heavily influenced by modern indie folk, which is also the reason to use many licensed tracks. Moris: "It's all part of that nostalgic Pacific northwest autumnal mood that we're trying to create."

The game uses music either diegetically (part of the game world itself and thus hearable by the characters) or non-diegetically. A vast part of the music—in particular the licensed tracks—is implemented in a diegetic way, which can be heard by interacting with objects in the environment, such as MP3 players, hi-fi installations or musical instruments. Some of these interactions are even mandatory to further unfold the narrative. Raoul Barbet, one of the game directors, mentions that in this way the music is also used to describe the personality of the characters, by using their personal musical taste (Dontnod Entertainment, 2015a). Max's MP3 player for example plays a different music genre than Chloe's hi-fi.

Non-diegetic use of the soundtrack includes cues that are used to build up tension during interactive scenes, as part of time skips in the narrative, or transitions between scenes.

Sometimes a diegetic piece of music even evolves into a non-diegetic cue used in a transition.

In terms of nonlinearity the music featured in *Life Is Strange* is predominantly linear, since it isn't coupled to narrative events. This implies that there are few actual transitions between musical cues, as many parts of the game have no accompanying music. An example of a diegetic transition is swapping CDs in Chloe's hi-fi, which could be seen as a form of horizontal resequencing.

#### 2.4.2. DISCUSSION

*Life Is Strange* is a game that heavily focuses on storytelling and conveying that through dialogue. I think the developers did a great job on balancing out the amount and function of music with the story and gameplay. Since the narrative takes on such a big role in the game, adding too much prominent music to the gameplay would only interrupt and take away attention from the dialogue in my opinion. Adding ambience music in the background could be an option, but I think that this doesn't fit well in the serious and semi-realistic setting the developers want to convey in this game, especially when diegetic music takes such a prominent role.

For me the music design of *Life Is Strange* is a very good example of how music can be balanced with storytelling and not getting in the way of the latter. Since this game is an inspiration for my own vision in terms of narrative design, I'd like to embrace the things I learned from this review.

#### 2.5. CELESTE



*Celeste* (Matt Makes Games, 2018) is a single-player 2D platformer published by Matt Makes Games. The game features over 250 rooms with platforming gameplay, divided in eight chapters. The player controls a girl named Madeline as she climbs up the titular Mount Celeste. During the events of the game Madeline is confronted with her mental issues by means of a dark reflection of herself, who breaks out of a mirror and disrupts her

journey on multiple occasions. In her journey and self-healing process she is often accompanied by Theo, a fellow traveler.

The narrative of the game focuses on Madeline's mental insecurities, such as anxiety and depression. During the game the player learns about Madeline's story of self-discovery and acceptance. This is also one of the reasons why I chose this particular game to review, since I try to convey a similar story in my own project *Invisible Wings*. Moreover the soundtrack has a similar style of combining genres and instruments that I use in my own compositions as well. It is also noteworthy—and very helpful for this critical review—that the audio implementation of *Celeste* by Power Up Audio is available as an FMOD Studio project for educational and modification purposes.

### 2.5.1. MUSIC DESIGN

*Celeste* features a soundtrack written by Lena Raine. The music is clearly non-diegetic, leaving room for diegetic sound effects. According to Raine, there is intentionally little connection between the level design and soundtrack directly. In an interview with Gamasutra (Couture, 2018) she states: "So many challenging platformers really play along with the difficulty of their games to amp up the intensity, which I feel, is a very literal way of scoring a difficult game. [...] But for *Celeste*, everything about the game is a reflection of its characters. The story isn't just some fluff that loosely contextualizes why you're jumping a lot. *Celeste's* challenge and its gameplay ARE the story. So, in scoring the game, my primary thought was in how Madeline is feeling."

Every world, or "chapter" as the game calls them, features its own distinctive soundtrack, that is composed of one or more musical cues. Which cue is playing is based in the first place on the narrative events that occur in the chapter. In the majority of cases however the narrative events are closely coupled to the level design and thus Madeline's current location, but the narrative events are still leading.

Other types of soundtracks are those that are used to denote a character. This music is at least used in two different ways in the game. For example rooms where Theo has set up camp feature his distinctive soundtrack that is different from the soundtrack of that chapter. Another example is a theme for the character Mr. Oshiro in the third chapter, that fades in when he is present in the room where Madeline is located.

The interactivity of the soundtrack uses two predominant techniques. A musical cue of a chapter may contain one or more layers, which are triggered by using vertical layering based on the narrative events. In chapters that feature two or more cues, the transition between these cues is generally accomplished by using horizontal resequencing using crossfades.

### 2.5.2. EXAMPLE: CELESTIAL RESORT

Two notable examples of *Celeste's* music system are found in the third chapter of the game: *Celestial Resort*. In this level Madeline enters an presumably abandoned hotel at the side of the mountain, which is inhabited by a ghost concierge named Mr. Oshiro. While he

persuades Madeline to stay a night in hotel, she really wants to continue her journey. She does help him clean a part of the resort however, showing how she sacrifices her own self-care to care for someone else. When the two reach the Presidential Suite, Madeline's reflection appears and creates an escape route for Madeline, however insulting Mr. Oshiro in the process. This leads to a boss fight, after which Madeline reaches the back exit of the hotel and leaving Mr. Oshiro behind, who is going to "close the hotel for repairs".

The first part of the chapter features rooms where Madeline has to find keys to a series of doors in order to follow Mr. Oshiro to the Presidential Suite. The progress in the level is reflected in the played soundtrack: after each door is unlocked, a new vertical layer is triggered in the cue. Apart from that there exists a separate layer that is only toggled on when Mr. Oshiro is present in the room.

After the doors are unlocked, Madeline reaches the second part of the chapter. This part features a new cue, but remains in the spirit of the whole chapter. In this part Madeleine encounters a room full of laundry that hasn't been cleaned in ages. She has to find three switches in the accompanying rooms, that clean the mess automatically upon impact. When one of the three switches is activated, a new and more intense arrangement of the cue is played. In the implementation this is done using horizontal resequencing, by fading the previous arrangement out, playing a transition sweep and starting the next arrangement. The transition to a next arrangement always occurs at the beginning of a new bar, thus using musical demarcation branching. Figure 1 gives an overview of this implementation in FMOD.

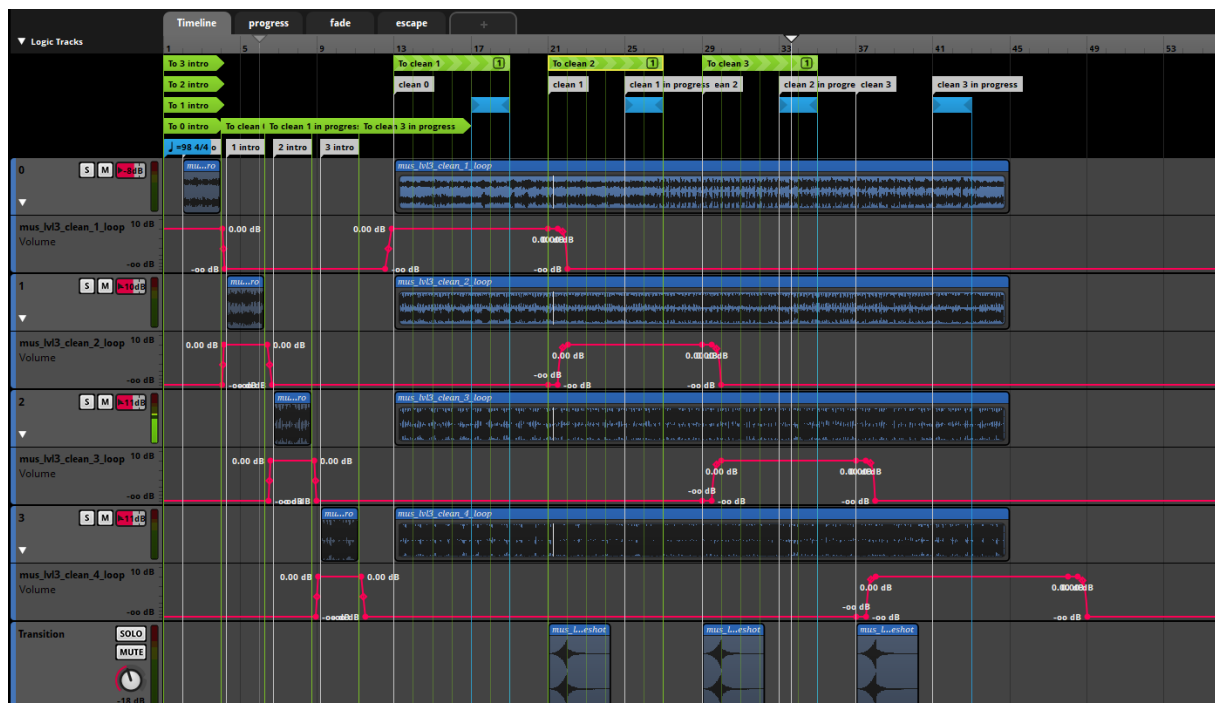


Figure 1: The FMOD event for the second part of Celestial Resort, showing various loop regions and markers that indicate transitions.

According to Raine the music used in this second part is composed in a way to reflect Mr. Oshiro's rising mental state, because he is not able to clean the mess he made himself. The new cues that are played when Madeline presses a button are directly linked to this (Game Score Fanfare, 2018).

### 2.5.3. DISCUSSION

I find *Celeste* to be a great example of a game where gameplay, narrative, and music are tightly coupled with each other. This leads to an engaging music system with interesting usage of adaptive music and transitions. For a narrative platformer this is in my opinion a good approach to handle the music design.

## 2.6. CONCLUSION

Reviewing the aforementioned narrative games have given a lot of insight in the music design techniques used in these games. The main conclusion is that music design in narrative games is to a great extent dependent on the intended function of the music. This function can in turn be dependent of the gameplay or conveyance of the narrative.

*Rayman 2: The Great Escape* and *The Elder Scrolls V: Skyrim* feature a similar music design, because the music is intended to set the mood for the story. In *Skyrim* the implementation of the music is however done differently, because the story is heavily dependent on the actions of the player. In *Rayman 2* the story is quite set in stone in terms of linearity, which gives more room to work with music.

In *Life Is Strange* the main focus is the conveyance of the story through dialogue, so the music is designed to not get in the way of that, but instead is either used diegetically or during scenes with few to none dialogue. *Celeste* on the other hand uses the music as a part of conveying the narrative. Because this game is primarily a platformer with an underlying story, instead of having the story as its main focus, music can be used more extensively in conjunction with the gameplay.

### 3. APPLICATION: MUSIC DESIGN IN MY OWN PROJECTS

Based on the knowledge and findings I have gained through my critical reviews, I have formed a design strategy for creating a music design in narrative games: first defining the function of the music based on the gameplay and narrative and afterwards designing a music system that takes into account this function.

In my graduation year I worked on several (narrative) games as a technical audio designer, which I did as a team member of Ocarime, a multidisciplinary team of composers, sound designers and software developers. This gave me excellent opportunities to apply this strategy in those games. In this chapter I will describe and reflect on the music design and implementation of three of those projects, which are *Pien en de Vissensteen*, *Invisible Wings* and *Ocarime: The Game*.

For most of the projects I was involved with the conceptualization as well, either as part of the development team or by initiating the project myself. Because of this, I had quite some creative freedom in deciding on the function of the music in the games. This helped me a lot to shape the music designs in a way that fits the games.

Because implementing my music system designs is also a big part of my professional portfolio, I will discuss this briefly in these projects as well, as opposed to the game reviews. First I will give a quick overview of common implementation methods I use, after which I will discuss the aforementioned projects.

#### 3.1. IMPLEMENTATION METHODS

The implementation of adaptive music can be commonly accomplished in two ways: either by directly coding it in the game engine, or by using audio middleware. On the one hand, an audio designer can implement the system directly into the game engine. This can be done either using the already existing sound engine, or by writing a new engine tailored to the needs of the project, e.g. if performance is critical in the game.

I personally prefer to use this method for music systems that are simple to implement, to avoid the overhead of audio middleware, or on the other hand for systems that require a tightly coupling to the game. An example of this is *Exodus Burned* (Pillow's Willow VR Studios, 2019), a multiplayer VR experience with several devices playing audio, on which I worked last year as an intern.

Another way to implement adaptive music is to use audio middleware. FMOD and WWISE are two industry-standard examples of middleware. I personally only have experience with the former, which I will thus use to describe the process. By using FMOD, the implementation process is effectively cut in two. First, the audio designer implements the adaptive music system in FMOD Studio, which can be done visually like in a digital audio workstation. Next, the audio events created in FMOD Studio are exported to be used in conjunction with an application programming interface (API) in the actual game code. Through this separation, audio designers in general don't have to worry about the

integration of the system in the code, which can be—out of my own experience—a quite technical job. That is, unless you're the audio programmer as well.

Although I am a more textual and code-oriented audio designer myself, I will gladly use FMOD in game projects, if that means I don't have to reinvent the wheel by writing my own code. The middleware has great tools available to use all of Michael Sweet's transition techniques (Bonkerk, 2019), and more. In my experience, the only drawback of FMOD in particular is the quite cumbersome usage of its API in the Unity engine, which has to do with the differences in programming languages that Unity and the API use (C# and C respectively).

As I said, my personal choice of implementation method is mostly based on the requirements of the project. Simple music systems for example do not require the overhead of using audio middleware, and using an available sound engine in the game engine can suffice in that case. Sometimes it isn't possible to use middleware because of licensing issues as well.

### 3.2. PIEN EN DE VISSENSTEEN

*Pien en de Vissensteen* is a student game project created by Daan Gijzen and Inge van Riezen. Their original concept was to create a small 3D fantasy adventure game with focus on the narrative, reminiscent of games in *The Legend of Zelda* series. The game takes place in a little fishing village where Vera, the antagonist, plans to steal an important artefact. The protagonist Pien must collect information about this plan to prevent Vera from succeeding in her plans. Eventually the two engage in a small boss battle at the village square.



As a member of Ocarime I worked on the design and implementation of the music system for this project. For the music design I collaborated with Amber Veerman and Stijn de Koning, who composed the music for the game as well.



### 3.2.1. MUSIC DESIGN

The game contains one level where the entire narrative takes place. We based the music system on the five locations of interest in the level (Gijzen, 2019) and the major events in the main quest. The overall feel of the soundtrack starts relaxed, but as the player gains more information about the plan of the antagonist, more tension is added in the music. Eventually this tension accumulates in the boss fight.

Every location has a different musical cue, which is composed of two vertical layers: a base layer containing percussion, and two or more melodic layers. These melodic layers are played back based on the progress in the quest, i.e. the state variable. All layers are looped when the player visits a location until they leave. Figure 2 depicts how the cues are ordered according to the location and state.

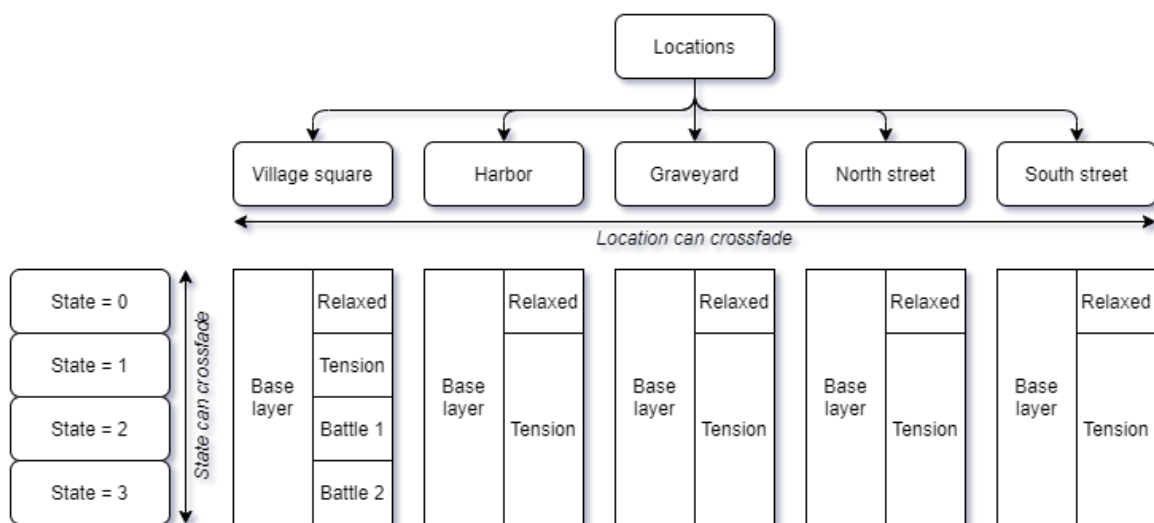


Figure 2: The musical cues used in *Pien en de Vissensteen*.

To make the transition between the cues as smooth as possible, the composers and I agreed on a few fixed parameters for the soundtrack such as general feel, tempo, time signature and length of the loop. In that way I could implement smooth transitions using horizontal resequencing with crossfades by synchronizing the beats of the several cues.

### 3.2.2. IMPLEMENTATION

Because the game is developed in Unity and the music system in itself is simple to implement, I directly coded the system in the game engine. Unfortunately the game hasn't been finished into a playable demo, so I implemented the audio system in a separate scene based on the concept art of the level map.

The map is divided in five areas which represent the five main locations. When the player enters such an area, the system triggers the accompanying cue, taking the current state of the game into account. The audio layers themselves are defined as data objects and are grouped into a set that links a game state to one or more audio layers. An area then holds a reference to this layer set, so it can play back the correct audio files.

The synchronization of musical cues is accomplished by starting all audio sources for all layers at the same time. Horizontal transitions are implemented by modifying the volume of those audio sources, fading in the new audio source and fading out the previous one upon switching to another area.

### 3.2.3. DISCUSSION

The music system used in *Pien en de Vissensteen* is a system that uses music to set the mood for the story, but adapts the used musical cues based on the actions of the player. Even though the system itself is simple in design, it can be used to play a variety of music that reflects the current narrative events without getting repetitive.

In retrospect I could have used audio middleware such as FMOD as well for this particular implementation. I however chose to implement the system directly in Unity, because of the simple design of the system which could easily be achieved with the available components. Moreover, while using FMOD could make the playback of the actual audio simpler, especially in terms of performance, the logic of the system still had to be coded in Unity, This would in turn include the overhead of a middleware API.

## 3.3. INVISIBLE WINGS

*Invisible Wings*<sup>1</sup> is a demo for an interactive visual novel with point-and-click elements. The visual novel is set in a modern fantasy world where humans, faeries, and their hybrid children were supposed to live together. Faeries are believed to be dangerous by humans, because of their touch magic, and so they are prejudiced about hybrids as well.



The story revolves around Dawn, a young hybrid with a great passion for figure skating, as she searches for answers to her own identity, and tries to blend in society as a human by tucking away her tail. She gets help from her best friend Aranea, a hybrid who fights for equal rights for hybrids in politics, but in turn has an aversion to faeries as well. During a skating competition Dawn gets distracted by Lilith, a faerie who hides her wings and tail to

---

<sup>1</sup> More information is available at: <https://invisiblewingsgame.com/>

join the contest as well, which leads to her tail coming loose, for which she is disqualified. Dawn initially despises Lilith and blames her for her own disqualification, but she apologizes to her a day later, against Aranea's urgent advice to avoid the faerie. The two eventually become more than just friends.

*Invisible Wings* is a game project I initiated myself, as opposed to working in a team with an often already established concept. This way I could shape the project in a way that closely resembles my artistic vision for narrative games and music, and pour in my passion for worldbuilding and storytelling as well.

Apart from my work as audio designer on the game, which included creating several sketches for compositions and designing and implementing the audio system using Unity and FMOD, I also worked on the development of the game itself, which gave me the opportunity to tightly integrate the audio system in the game.

### 3.3.1. MUSIC DESIGN

For the music system for *Invisible Wings* I took a lot of inspiration from *Life Is Strange*, because both games depend heavily on dialogue to convey the story. The function of the music is in the first place to convey the mood and setting of the story in the game, but does appear diegetically in the story on a few occasions.

I envision that most dialogue heavy scenes feature either no music or ambient music, which does not include a lot of rhythmic or melodic material, in order to give room to the dialogue. The music takes on a more prominent role in the different scenes that feature point and click interaction, but is still inferior to the dialogue triggered by interactions with items. In turn these interactions can trigger a new arrangement of the musical cue, depending on the unfolding of new information to the player, but this has to be decided on per scene.

There are two types of occasions where music is used diegetically. On the one hand there are a few objects in the point and click scenes that play music upon interaction, such as a radio in Dawn's bedroom or a jukebox in a restaurant. This music could also function as background music for dialogue if the interactions of the player invoke that. In that case I can decide to decrease the volume of the cue, or transition to a more ambient arrangement as described above, but this—as well—has to be decided on a per-case basis.

Another type of diegetic music is the use of background tracks for the skating performances of Lilith and Dawn, which reflects its real life counterpart in figure skating. It is noteworthy that I use some motifs from famous Tchaikovsky pieces in these compositions to further emphasize and allude to the clichés of this practice.

Because both of these skating scenes feature dialogue from Dawn, which require interaction from the player, I devised these musical cues to be strongly interleaved with the narrative events in these scenes. I will implement this by separating the linear parts of the musical cues with bridge transitions. If the playback of the cue reaches a bridge, which is then looped, the corresponding interactive dialogue is shown in the game. When the player in turn finishes the interaction, the cue continues playback by first finishing the bridge and

then starting the next linear part using phrase branching. Figure 3 depicts this behavior between the music system and game in a visual way. These two scenes are the only heavily scripted musical cues in the system so far.

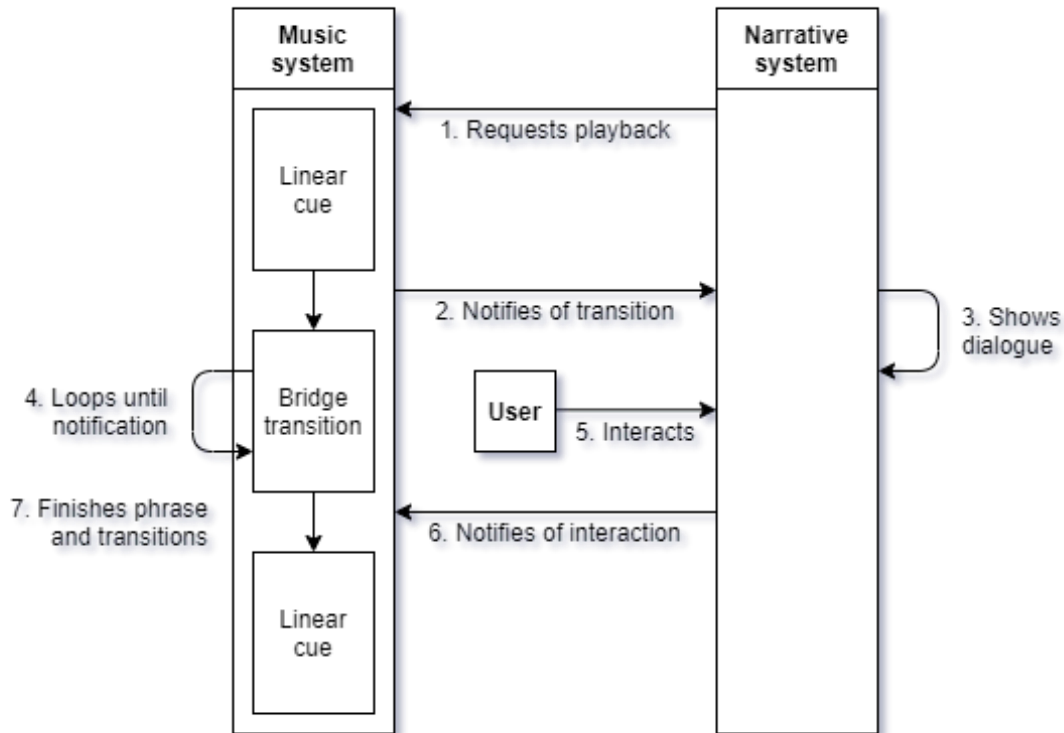


Figure 3: Flow diagram for the music system in the skating scenes.

### 3.3.2. IMPLEMENTATION

*Invisible Wings* is developed in Unity. Because the overall audio system for this projects not only includes the aforementioned music system, but also sound effects and dialogue audio, I decided to use FMOD as middleware to implement the audio, to not have to start from zero for every aspect of the audio system. However, I experienced that programming using the FMOD API in Unity has its own quirks, which are mostly due to the aforementioned difference in programming languages. To close this gap, I wrote several extensions that wrap the API code into higher level functions that adhere to the C# standard.

For the music in the skating scenes I will use the timeline marker callback feature<sup>2</sup> of the FMOD API. When such a callback is set in the code in the game engine, specific code can be executed every time when the playback of an FMOD event reaches a marker. Using this callback I can "listen" in my code when a bridge transition is reached by placing a marker at that point (figure 3, step 2), and in reaction let the narrative system show the next dialogue text (step 3). To notify FMOD in return when the dialogue interaction is finished (step 6), I use an event parameter which lets the playback leave the bridge transition loop.

<sup>2</sup> <https://fmod.com/resources/documentation-api?version=2.0&page=studio-guide.html#event-callbacks>

I also use the audio table feature<sup>3</sup> of FMOD Studio for playing back the dialogue audio clips. Using this feature, I can assign labels to all audio clips, which can be referenced from the code in Unity in order to play them back in an FMOD event instance. Because these audio tables are localized, i.e. per audio language exists a separate table, it is easy to add dialogue audio for other languages in the future.

### 3.3.3. DISCUSSION

As mentioned, the music system for *Invisible Wings* is inspired on *Life Is Strange*, where the music leaves a lot of room to convey the story through dialogue. Since I'm still working on this project at the time of writing and the music system isn't implemented yet, I can't clearly reflect on the effect that the music system will have on the narrative design. I do however think that using the same techniques as in *Life Is Strange* is the best approach for now to use in a dialogue heavy narrative game.

## 3.4. OCARIME: THE GAME

*Ocarime: The Game*<sup>4</sup> is an interactive portfolio website for the members of Ocarime. The game is modeled after a top down role playing game, such as the older *Final Fantasy* series. In the game the player meets the team members of Ocarime as non-playable characters. The player can listen to their music and view their portfolio by interacting with the environment and objects. The game itself acts as a portfolio item for the team as well, and shows what its members are capable of creating.



Although *Ocarime: The Game* is not a narrative game in the sense of this supportive narrative, I nevertheless want to discuss this project. Similar to the discussed narrative games and my own projects so far, it features music design in which the implementation is dependent of the function of the music in the game.

This project was initiated by the Ocarime team itself and is the result of an assignment where we had to define our team. While all team members took a part in the design of the game and creating content, I took a leading role in the development of the game engine

---

<sup>3</sup> <https://fmod.com/resources/documentation-studio?version=2.0&page=dialogue-and-localization.html>

<sup>4</sup> A playable demo is available at: <https://beta.ocarime.com/>

and audio system. Because I developed the game engine along with the audio system, I again had the opportunity to tightly integrate the two.

#### 3.4.1. MUSIC SYSTEM

For the design of *Ocarime: The Game* our idea was to give each team member their own room, where their portfolio is showcased in the form of paintings and a composition or soundscape, that is played diegetically through a speaker. These six compositions together form a nonlinear composition which is based on the position of the player. To create this "umbrella" composition we decided on some requirements for the musical cues that each of us would compose, such as tempo, length, and key signature. In this way I could implement smooth transitions using horizontal resequencing using crossfades by playing back all cues at the same time, leading to synchronization of the beats. This design is similar to the music system in *Pien en de Vissensteen*.

The actual music system is implemented using of audio sources and listeners that are placed in the game world. This way cues are heard less loud and muted as the player moves away from the source, or if the path between source and player is obstructed by objects or walls, leading to a diegetic display of the music.

#### 3.4.2. IMPLEMENTATION

Both the game engine and the audio system are implemented in HTML5 and JavaScript, so that the game could easily be run in a web browser and can be integrated with the website of *Ocarime*. For the playback of audio I created a system with audio sources and listeners, similar to how other game engines such as Unity have implemented their audio engine. The actual playback of audio is accomplished with the Web Audio API, a standard for controlling audio on the web.

#### 3.4.3. DISCUSSION

The music design in *Ocarime: The Game* is quite similar to that of *Pien en de Vissensteen*. Both make use of musical cues that are synchronized while playing back and smooth transitions based on horizontal resequencing using crossfades. However, the music system in this game can be considered a next iteration, because it makes use of the position of the audio sources that play back the cues. This way the music is made more diegetic, as opposed to the non-diegetic background music in *Pien en de Vissensteen*.

An improvement of the music system could be the implementation of loop regions, so that cues with reverb tails can be used. In the current implementation all cues are pre-rendered with reverb tails, so in this specific implementation it wasn't necessary to implement this functionality.

## 4. CONCLUSION AND REFLECTION

Music design and its implementation differs among narrative games, which is in the first place dependent of the function that the music in those games takes on. As seen in the critical review and discussion of my own projects, narrative games feature different functions for the music. First, music can be tightly coupled to the gameplay and narrative, such as in *Celeste*. Secondly, there is music that defines the setting and mood of the narrative, like in *The Elder Scrolls V: Skyrim* and *Pien en de Vissensteen*. And last, music can have a function that is not tightly coupled to the narrative, but emphasizes on elements from that narrative, like in *Life Is Strange* and *Invisible Wings*.

This musical role is in turn dependent of the intended gameplay and how the narrative is conveyed in the game. *Life Is Strange* and *Invisible Wings* for example heavily rely on dialogue between characters to convey the story, hence the role of the music is designed not to get in the way of that. Another example is *Skyrim*, in which the gameplay mainly focuses on exploring, and so the music can take on a more prominent role in the gameplay.

As an audio designer I applied the design strategy based on the reviews in my own projects. Exploring the different types of music design and the techniques used to implement those systems gave me a lot of insight into common practices in the field of narrative games. Since I'm currently still working on *Invisible Wings*, I'm now able to fully adapt this design method to create a fitting music design for the intended gameplay, as well as future projects on which I will embark.

## REFERENCES

### Articles

Bonkerk, N., 2019. *Experiences and Experiments in Music for Games*. Supportive Narrative, Music & Technology, Hogeschool voor de Kunsten Utrecht.

### Online articles

Couture, J., 2018. *Road to the IGF: Matt Makes Games' Celeste*. Gamasutra. Available at: [https://www.gamasutra.com/view/news/315116/Road\\_to\\_the\\_IGF\\_Matt\\_Makes\\_Games\\_Celeste.php](https://www.gamasutra.com/view/news/315116/Road_to_the_IGF_Matt_Makes_Games_Celeste.php) (Accessed: 5 March 2020)

Gijzen, D., 2019. *Week 5 & 6 - Project Vrij III*. gewoonDaan. Available at: <https://daangijzen.com/home/week-5-amp-6-project-vrij-iii-dutch> (Accessed: 23 May 2020)

Sweet, M., 2016. *Top 6 Adaptive Music Techniques in Games – Pros and Cons*. Design Music Now. Available at: <https://www.designingmusicnow.com/2016/06/13/advantages-disadvantages-common-interactive-music-techniques-used-video-games/> (Accessed: 9 May 2020)

### Online videos

Caddicarus, 2018. *Rayman 2: The Great Escape*. [Video] Available at: <https://youtu.be/-xNG3cdSlrU> (Accessed: 9 May 2020)

Calcom, 2018. *Among the GREATEST Games Ever Made? | Rayman 2: Revolution*. [Video] Available at: <https://youtu.be/OXDT5TV5b78> (Accessed: 9 May 2020)

Dontnod Entertainment, 2015a. *Life Is Strange Developer Diary - Creating Arcadia Bay*. [Video] Available at: <https://youtu.be/OVkJdxAPPJ0> (Accessed: 17 May 2020)

Game Score Fanfare, 2018. *The Anxiety of Celeste and its Music*. [Video] Available at: <https://youtu.be/jvETIcMXaEo> (Accessed: 12 March 2020)

Noclip, 2018. *The Music & Sound of Bethesda RPGs (Skyrim, Oblivion, Fallout)*. [Video] Available at: <https://youtu.be/3193ZsxChSc> (Accessed: 10 May 2020)

Taveras, J., 2014. *Life Is Strange Creative Director Jean-Maxime Moris Talks Atmosphere, Choices and Female Leads*. DualShockers. [Video] Available at: <https://www.dualshockers.com/life-is-strange-creative-director-jean-maxime-moris-talks-atmosphere-choices-and-female-leads/> (Accessed: 17 May 2020)

### Video games

Bethesda Game Studios, 2011. *The Elder Scrolls V: Skyrim*. [PC game] Rockville, MD: Bethesda Softworks.

Dontnod Entertainment, 2015b. *Life Is Strange*. [PC game] Shinjuku: Square Enix.

Matt Makes Games, 2018. *Celeste*. [PC game] Vancouver: Matt Makes Games.

Pillow's Willow VR Studios, 2019. *Exodus Burned*. [Multiplayer VR game] Eindhoven: Pillow's Willow.

Ubi Pictures, 1999. *Rayman 2: The Great Escape*. [PC game] Montreuil: Ubi Soft.