MySoundtrack: a tool for personalized and adaptive music listening while playing games

Daniel Filgueira Bezerra
Centro de Informática
Universidade Federal de Pernambuco
Recife, Brazil
dfb2@cin.ufpe.br

Filipe Calegario
Centro de Informática
Universidade Federal de Pernambuco
Recife, Brazil
fcac@cin.ufpe.br

Giordano Cabral
Centro de Informática
Universidade Federal de Pernambuco
Recife, Brazil
grec@cin.ufpe.br

Geber Ramalho
Centro de Informática
Universidade Federal de Pernambuco
Recife, Brazil
glr@cin.ufpe.br

***Abstract*—Adaptive soundtrack design for video games is an ever-evolving field of study: games that modify its music depending on what is happening around and with the player. Meanwhile, independent game developers struggle with composition of soundtracks, and adaptive soundtrack tools are not accessible to them, mostly because of their complexity. This forces most of them to collage soundtracks with permissive licenses they find online. Besides, there is a growing habit of players to mute the game's original soundtrack to listen to their own songs. This paper introduces MySoundtrack, an asset for Unity that allows the player to keep playing while listening to Spotify songs, chosen according both to the player musical preferences and to the intended emotions on each moment in the game. We review existing approaches on adaptive soundtracks, explain how MySoundtrack's prototype works and its design choices, and discuss future plans for the tool. Validation so far indicates interest and curiosity by game developers and players, indicating the relevance of the proposed plug-in.**

Keywords—adaptive music, video game music, mood playlists, soundtrack, personalized music

I. Introduction

In recent years, independent game development (hereby referenced as indie), grew and received a lot of attention. However, it is still a slow and costly activity, partially because of the diversity of specialties that an indie developer must learn so he may create a video game alone, or in a small group of people, and one of the rarest specialties is sound design. Often, there is no person in the group with the ability to compose and design soundtracks, and what happens is that developers search for free and open soundtracks to glue together.

On top of this, video game music may be non-linear, reacting to any arbitrary set of actions the player may do throughout the gameplay, which is very important to reinforce the emotion of each moment and increase immersion. This strategy, often referred to as adaptive music or interactive music, is common practice between major game companies. However, it is another level of complexity in game development that separates small teams from high budget projects even more, because of its complexity.

Meanwhile, there is a growing habit of players to mute original game's soundtrack to listen to their own songs, as demonstrated in [1] and witnessed on this work's Proof of Concept. This may hinder the soundtrack from having the impact that it was designed to have, because what the player is listening to may not have any emotional relation to the moment he is experiencing in the game.

This paper provides an alternative to adaptive music implementation in video games, that does not involve soundtrack composing or any technicalities involved in it. The game design associates the "music vibes" to geographic areas or moments in the game. When the player is in that area or moment, MySoundtrack triggers a search for songs in Spotify for those that match both the designed vibe and the musical preferences of the player.

II. Problem Characterization

This paper and its proposal focus on attacking two problems: the difficulty of soundtrack design when working on teams without a sound specialty (majority of indie teams), and the habit of players muting the game's music.

A. Complexity of soundtrack design on indie teams

When working on a game alone, or with a small group of people, sound is one of the rarest specialties present on the team. This often leads to use of free soundtracks online, which may result in a game without audio identity and a great struggle to find good soundtracks with permissive licenses and fit well with one another.

According to [2], [3] and [4], sound is the least present specialty on *Game Jams*. In these surveys, the frequency of participants is analysed based on specialties, such as programming, art, sound and design. In the Global Game Jam 2016, only 12% of the participants were skilled in audio, and in the *TasJam: Voices* of 2015, only 10%. This shortage of sound specialists often drives the event organizers to ask them to participate in different teams simultaneously. Game Jam scenarios are a good representation of the independent game industry, since most of their participants are novice game developers, and several of them begin the game development journey at these events. Therefore, the necessity from the indie community of tools that facilitate the audio composing process, or that offer an alternative for it, is latent.

Another alternative for composing sound pieces would be to use tools for design of adaptive soundtracks, which are often used by great game companies. However, they are most of the time not available for everyone, and when they are, for example [5], they are designed for music composers or someone with likewise speciality, which usually does not exist on indie teams.

B. Player habit of muting game's music

Currently, there is the growing habit of players listening to their own music while playing, and usually, that involves the muting of video games soundtrack. Consoles, such as *Playstation 4*, even have integration with Spotify and their own music streaming service, *PlaystationMusic*, to enable easier music listening while playing.

A survey was conducted in [1], with 156 participants, about the habit of silencing game's music while playing. Only 25% of respondents declared to never do it, while 11 % answered that they did it regularly and 64% declared to sometimes do it. The frequency of this practice is higher on younger respondents and also on people that play more often. The survey also shows a high contrast between the occurrence of silencing the game music for each game genre: casual/mobile and racing games were the genres in which participants declared to do it more often.

This may cause the great effort on game's soundtrack design to be counterproductive, specially on such small teams, with limited resources. Additionally, such a habit may harm the game experience, because of possible dissonance between the emotion of the song the player is listening to and the desired emotion of that game moment.

This culture goes beyond only playing, but also on content creation: gameplay *streamers* (people that record themselves while playing), usually enhance viewer's interaction by allowing them to choose songs to play in the background, strengthening and normalizing even further the habit of music listening while gaming.

III. Current Solutions (State of the Art)

The last years witness the growth of the research on adaptive soundtracks aiming to enhance and refine player experience. The goal is, most of the time, to fit the soundtrack's mood to each game moment, this is what makes it adaptive. There are two categories of methods for dealing with adaptive soundtracks: the playback-based methods and the procedural soundtrack generation.

The playback-based methods modify pre-existing sound pieces. They use an arsenal of already composed music pieces, which are played in an arbitrary sequence, or layered on top of each other. The Sequence Music Engine [6] is a good example of such a method. Implemented to develop the soundtrack of Kingdom Come: Deliverance (KC:D), the system can be bound to CryEngine and Unity. Sequence Music Engine is expected to be manipulated by the authors of the music soundtracks, and they should use the system to set up different tracks, which play on top of each other, but with variations, depending on in-game variables such as time of the day, weather or location. Kingdom Come uses orchestrated music, including gregorian chant, and Sequence Music Engine also handles smooth transition between the tracks. The goal of this feature is to build a seamless experience for the player, so when the game situation changes the soundtrack playback follows, making it look like the music was composed that way. Ultimately, the tool manages to merge and interpolate pre-existing complex tracks seamlessly, achieving outstanding results on KC:D, but with the presence of highly skilled music professionals to operate the tool.

Procedurally generating a soundtrack in real time is another option to achieve adaptive music in video games. The main approach of procedural, or algorithmic soundtracks, is to modify in runtime the parameters of the soundtrack generation algorithm according to what is happening in the game, leading to adaptation of the music to the game's circumstances. One of the first major games to utilize such a technique was Spore [7], a game where you play as an organism that is in constant evolution, you grow, feed, and evolve. Spore's musical tracks follow the player's play-style and the various situations of the ever-growing world around him. Another example is the game Doom [8] (2016), a first person shooter about slaying hordes of demons. The game composer Mick Gordon utilized various audio channels, each with a sequence of filters, and depending on the game's moment, those filters' parameters would be modified at runtime, modifying that audio channel and changing the overall soundtrack to a new mood.

It is important to ascertain that these approaches rely on the presence of specialized music professionals, or implementation of complex systems, often not available publicly, that use music theory to compose soundtracks procedurally. Hence, moving away from independent developers' capacity, who have limited financial and human resources.

IV. Our Solution

The main goal of MySoundtrack is to provide to game developers, especially to those without extensive musical knowledge or someone who can sound design on the project, an alternative to the process of composing or assembling soundtracks.

MySoundtrack can be easily used to select and play Spotify songs, based on the user's Spotify account, and on parameters defined by the game creator, so that the already existing habit of listening to music while playing does not ignore the game's emotional flow.

V. Method and Implementation

A. *Classifying tracks' overall emotion.*

The first step was to figure out how Spotify could identify the moods of a specific song. It is possible to retrieve information that can be useful in this sense using tracks' audio features, which are parameters that say a lot about the music tone, feeling and general aesthetic. These parameters are:

- Danceability: describes how suitable the song is for dancing. It takes into account the overall rhythm and beat of the song.
- Energy: represents the intensity of the song. Less energetic tracks feel slow and quiet, while higher ones feel fast and noisy.
- Loudness: overall volume in decibels.
- Speechness: the amount of presence of spoken text in the song. Poetries, for example, near maximum.
- Acousticness: detects if the track is acoustic.
- Instrumentalness: higher instrumentalness tracks consist of tracks with a lower presence of vocality.
- Liveness: higher liveness tracks are tracks in which are detected more audience presence.
- Valence: describes the track's positiveness. Higher valence tracks tend to be more happy or euphoric, while lower ones, more sad or angry.
- Tempo: consists basically of the song's pace. Consider directly the average beat duration.

The chosen parameters to implement the ranking algorithm was Energy and Valence. This choice was made based on [9], where it is demonstrated the circumplex

model of affect, a two-dimensional correlation between arousal and pleasure, as shown in "Fig. 1". This is translated from the context of psychology into audio features by facing arousal as energy, and pleasure as valence, as demonstrated in [10] by the examples of famous songs placed in this Arousal-Pleasure chart.

With this simple relationship established, four predefined vibes were chosen, one in each quarter of the chart, trying to cover major videogames feelings, in each chart extreme. Ranking results were surprisingly accurate, and the other audio features may be used in the future to refine the ranking based on player preferences, such as instrumentalness and loudness.

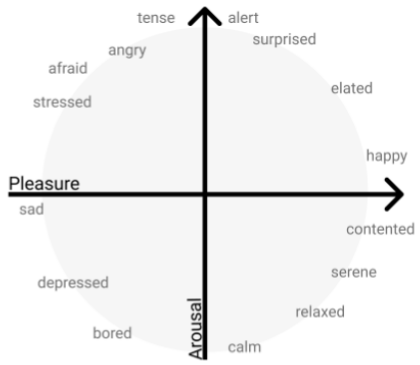

Fig. 1: Circumplex Model of Affect, by (Russel 1980)

B. MySoundtrack's design and basic logic

The first prototype was implemented using four predefined emotions, or so called vibes. Table I shows the chosen vibe names and energy-valence values.

TABLE I. PRE-DEFINED VIBES

Vibe Name	Audio Feature	
	Energy	*Valence*
Combat	1	0.4
Adventure	0.5	0.3
Sad	0	0
Romance	0.8	0.8

Pre-defined vibes with each chosen audio features values.

Thus, a Unity's custom editor script was implemented that enables game developers to choose which vibe corresponds to that area. This is shown in "Fig. 2".

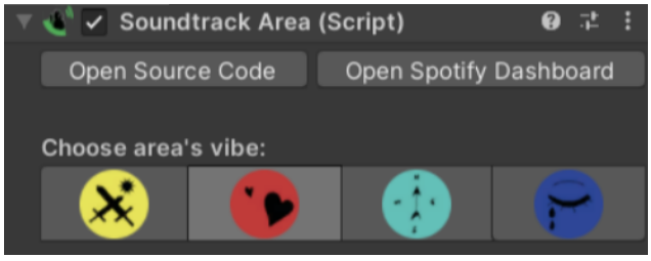

Fig. 2: Custom inspector, displaying the toggle control through which developer chooses the vibe of that area, and helper Hyperlinks. Icons, from left to right, correspond to: Combat, Romance, Adventure and Sad.

The result of level design using soundtrack areas, as shown in "Fig. 3", displays the trigger of that area, which will activate when the player enters it, and custom icon and colors to better indicate which vibe is selected.

C. Ranking Algorithm

MySoundtrack has two main classes, as shown in "Fig. 4", which are Unity's monobehaviours, C# code which runs attached to a game object inside Unity's scene.

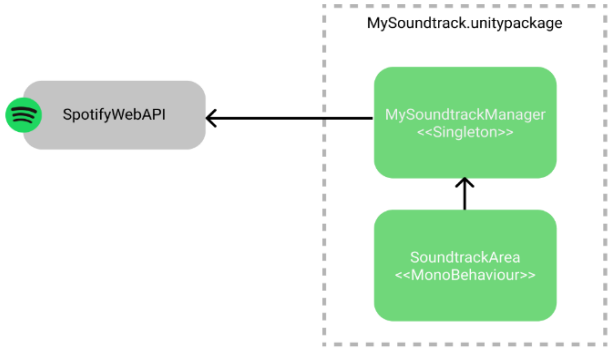

Fig. 4: Basic MySoundtrack's architecture.

When the player enters a soundtrack area, it requests MySoundtrackManager to rank the player's Spotify's songs based on the energy and valence of that area, as shown in *Algorithm 1*.

Algorithm 1 Getting best tracks for audio features

Input: E, the float value of the desired audio feature energy
V, the float value of the desired audio feature valence
T, the list Tracks to rank
S, the size of the returned ranked list
Output: R, the list of ranked songs, with the size of S

$\tau \leftarrow$ new empty list of tuples (track, grade)
for $t \in T$ **do**
  $a \leftarrow$ GetAudioFeatures(t)
  $grade \leftarrow |GetEnergy(a) - E| + |GetValence(a) - V|$
  $Insert(\tau, t, grade)$
end for
SortByGrade(τ)
$R \leftarrow$ Sublist(τ, 0, S)
return SelectFirstItemFromTuple(R)

The ranking calculation is quite simple: the lower the variable *grade* is, the closer the track is from the desired energy-valence combination. Since both energy and valence audio features range from 0 to 1, no weights were necessary in the calculation. They will be when implementation takes into account other variables, especially those that do not range from 0 to 1.

D. Validation and Proof of Concept

Systematic interviews were conducted with game development students and professionals, in order to validate the concept and gather ideas and feedback. The interview was composed of four parts: technical profile; personal and team experience in soundtrack implementation on game projects; musical habits, including while playing; and MySoundtrack presentation.

Currently, 23 interviews were held, with participants' ages ranging from 19 to 52. Among them, 3 identify themselves as women and 20 as men. Participants were professionally distributed as: 12 programmers (52%), 4 game designers (17%), 2 producers (8%), 1 sound designer (4%), 2 professors (8%) and 2 independent game company presidents (8%). 19 of the participants (82%) are or already

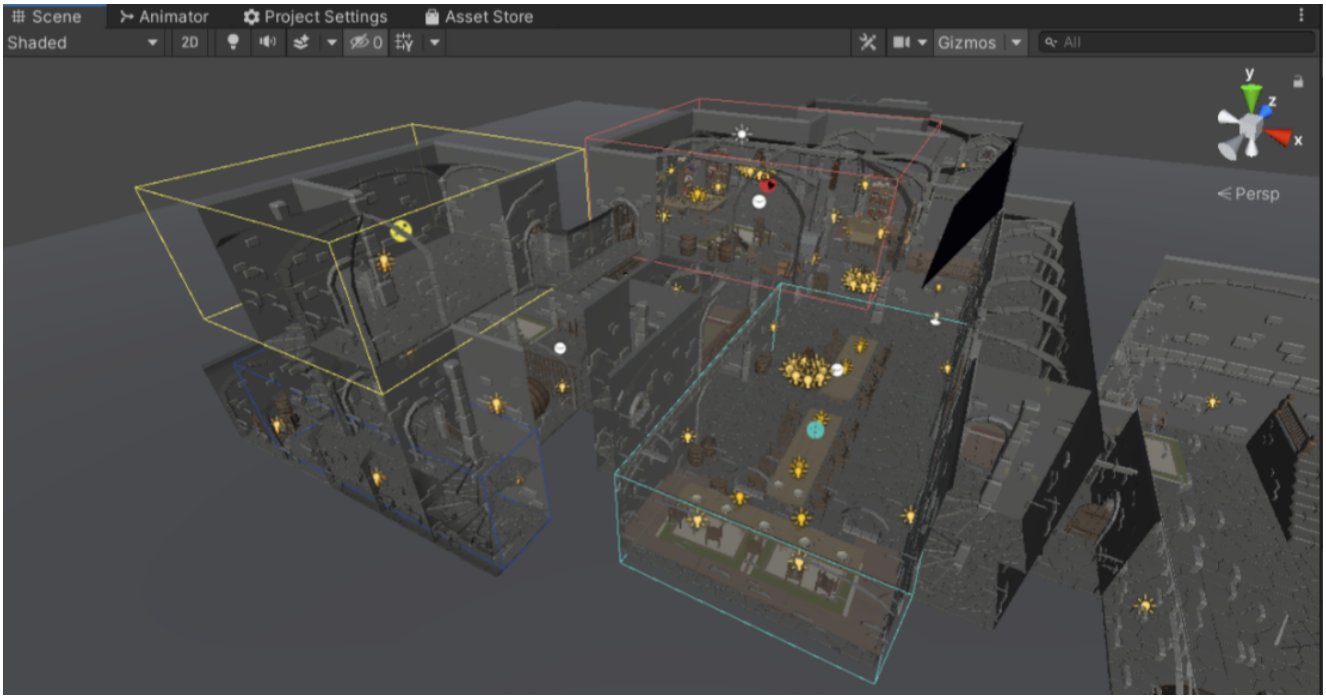

Fig. 3: Level Design view, displaying 4 inserted soundtrack areas, one of each available vibe. The map example resembles a dungeon, in which some rooms should have a specific vibe.

have been involved in indie game development, and 11 (47%) are *premium* Spotify users.

Participants were selected through chain recommendation: at the end of an interview, it was requested if the interviewee knew people who would gladly participate. This method gathered participants spread around 4 Brazilian estates, but there is still bias risk, regarding the small number of participants.

16 (69%) of the participants mentioned they themselves often listen to their own songs while playing, and would love to tether this habit to the game's narrative. 5 of them even stated that they used to turn off the game's soundtrack completely, while listening to their own songs or podcasts. According to them, the frequency of this habit increases on more casual games, specially mobile ones, and on games with repetitive game loop, frequently experienced in MMORPGs, for example. Another relevant result is that 10 (43%) of the participants declared frequent use of musical *bots*, on apps such as *Discord*, to play songs while players gather and play together, associating this habit to a social aspect as well. One participant in particular (P15) declared: "We always play with a bot playing songs in the background, but we have to select one by one. It would be great to use this tool [MySoundtrack] to make the selection process automatic and fit the game."

Among the 19 participants who have experience in indie game development, 14 (73%) stated that they hardly ever or never had professional support to compose soundtracks. According to them, the common approach is to look for music that has a permissive license, low or no cost, and that fits well to the game and to each other. The main complaints about this method is the resulting lack of the game's musical identity and high curating effort for regular musical results. This frustration can be exemplified by the affirmation made by participant (P04): "I've played games with music I've already used in personal projects. It feels bad", and by participant (P11): "to search for music online only works well when you have the money to spend on unique songs. Other than that, the results are pretty mediocre."

Among the least enthusiastic responses, there were arguments about the importance of the composed soundtrack to the game's narrative and experience. 7 (30%) people said that using Spotify songs could get in the way of story-driven games, especially when there is a lot of non-cutscene dialogue. Nevertheless, only 2 (8%) declared having no interest in MySoundtrack's features.

Results of the interviews suggest that the premises on which MySoundtrack bases itself are indeed relevant, and generated a powerful reflection on the social aspect of music listening while playing, which can be explored in the future. With this experimentation, it was also obtained a new priority on future works, and which feature holds more value to the end user. Two of future works were conceived during these interviews: filter copyrighted music, in order to enable content creators to play and advertise games that use MySoundtrack; and to support other music streaming platforms, aiming to expand the target audience beyond Spotify users.

VI. Future Works

Enhancements in UX and performance are already in progress, but also, throughout research, some ideas were validated and listed as future works. These features enhance and extend MySountrack by taking advantage of Spotify`s huge community, and their commercial and social presence.

A. Royalty-free songs for content creators

Mentioned in many interviews, a royalty-free option to MySoundtrack was said to be extremely relevant, sometimes even desired by those being interviewed. Such a feature consists of replacing normal player's songs ranking with songs that may be reproduced online without the risk of copyright claims. This is a recurrent discussion on *content creators* and most of all, streamers that worry about it and frequently search for permissive licensed Spotify playlist to use on the streams. The importance increases when it is stated that a relevant part of game's marketing, most of all indies, rely on sponsoring digital influencers to play the game and leave a positive review to the audience.

Of course this feature adds complexity to the ranking system, because most likely the player will not have enough permissive songs in its account, and it will be necessary to infer which DMCA free music both fit the player's taste and the game's situation.

B. Audio modification through audio filter

Another frequently used feature on video game sound design is to apply sound filters, in order to convey a higher immersion sensation. This happens, for example, in some games where the character enters underwater, or suffers a big impact, such as a nearby grenade, and the whole audio becomes muffled.

Since it is not a current option to make the songs play in-game, but keep it playing outside, on the player's Spotify device, it was briefly explored the possibility to have the game analyze which song is being played, and play an audio that would make that song feel like it was filtered like the example above. Such a feature seemed too complex to receive higher attention at this time, and was left as future investigation.

C. Create game playlist after game is played

Some game developers, while being interviewed, gave the much appreciated suggestion of when the game is over, MySoundtrack should generate a Spotify playlist that summarizes the game experience, much like an original soundtrack album.

This is a really interesting *nice-to-have* feature, and is validated by Spotify's efforts to personalize and reward user's experience on the app: Spotify Wrapped stands as the major evidence of this effort and its success, where the user receives an overall year review, with interesting statistics and even a custom playlist that represent that year.

VII. Conclusion

In this work, it was introduced MySoundtrack, a Unity's plug-in that enables adaptive and personalized music listening while playing, using Spotify. Usage and goals of adaptive soundtracks in game design were detailed, pondering about the difficulty of high-end soundtrack implementation, specially on indie scenarios. Additionally, it was also discussed the player's habit to often listen to music while playing.

The prototype logic was explained, passing through the theoretical base, design choices, implementation details and validation. And the MySoundtrack prototype was made available for anyone who wants to test it at [11]. Interviews so far validate the relevance of the tool and the premises on which it bases itself, while official usability testing is still to be done.

Finally, current working tasks and future works were listed and explained, discussing each one's relevance and feasibility.

Acknowledgment

The authors thank all research volunteers that provided insights, feedback and considerations. The authors also thank students and professors from the class *Criatividade Computacional*, which enabled the birth of MySoundtrack, in a welcoming and friendly community.

References

[1] J. McGowan. "Harmonious: An Emotion-Matching System for Intelligent use of player's own music libraries with game soundtracks," MSc project, Leeds Metropolitan University, 2011.

[2] K. De Salas, I. Lewis, and I. Bindoff. "Game jams as an opportunity for industry development," 1st International Joint Conference of DiGRA and FDG, 2011.

[3] T. Steinke, M. Linsenbard, E. Fiske, and F. Khosmood, "Understanding a Community: Observations from the Global Game Jam Survey Data," Proceedings of the International Conference on Game Jams, Hackathons, and Game Creation Events, 2016.

[4] M. Borg, V. Garousi, A. Mahmoud, T. Olsson, and O. Stalberg, "Video game development in a rush: A survey of the global game jam participants," IEEE Transactions on Games, vol. 12, no. 3, pp. 246–259, 2020.

[5] A. Gungormusler, N. Paterson-Paulberg, M. Haahr. "barelymusician: An adaptive music engine for video games,". InAudio engineering society conference: 56th international conference: audio for games, 2015.

[6] A. J. Sporka and J. Valta, "Design and implementation of a non-linear symphonic soundtrack of a video game," New Review of Hypermedia and Multimedia, vol. 23, no. 4, pp. 229–246, 2017.

[7] D. F. Kosak, "The beat goes on: Dynamic music in Spore," GameSpy, 20-Feb-2008. [Online]. Available: http://uk.pc.gamespy.com/pc/spore/853810p1.html. [Accessed: 04-Oct-2021].

[8] B. Smith, "Rip and Tear: Deconstructing the Technological and Musical Composition of Mick Gordon's Score for DOOM," The University of Adelaide, 2016.

[9] J. Posner, J Russel, and B. Peterson, "The Circumplex model of affect: An integrative approach to Affective Neuroscience, Cognitive Development, and psychopathology," Development and Psychopathology, vol. 17, no. 03, 2005.

[10] P. Helmholz, D. Siemon, S. Robra-Bissantz. "Summer hot, Winter not!–Seasonal influences on context-based music recommendations," ACIS 2017 Proceedings. 6, 2017.

[11] D. Bezerra, "Danielfib/MySoundtrack," GitHub, 29-Mar-2021. [Online]. Available: https://github.com/Danielfib/MySoundtrack. [Accessed: 04-Oct-2021].