Development of an immersive sound tool for RPG sessions using Pure Data and MobMuPlat.

Gabriel L. Rocha [1,3], **João P. M. Oliveira**[2,3], **Flávio L. Schiavoni**[2,3]

[1]Graduação em Ciência da Computação

[2]Programa de Pós-Graduação em Ciência da Computação (UFSJ)

[3]Arts Lab in Interfaces, Computers, and Everything Else (ALICE)
Universidade Federal de São João del-Rei (UFSJ)

gbr.cdc@gmail.com, joaopedromoliveira1996@gmail.com, fls@ufsj.edu.br

Abstract. *Role-playing games (RPGs) are an important niche within the gaming market. They aim to entertain players by immersing them in a fictional universe. To achieve this, narrative resources are used, and in some cases, different supporting tools are employed. Among these tools, the use of audio, such as soundscapes and soundtracks, can be considered. This article presents the development of software that allows an RPG game master to control various sounds to create environments for their players. This tool is developed using the Pure Data programming language and the MobMuPlat platform.*

1. Introduction

Role-playing games (RPG) are part of our society's culture as a playful entertainment experience present in the daily lives of many people. This game genre is a collective entertainment based on narrative experiences, aiming to introduce a group of players into an adventure in a fictional environment. In addition to entertaining, RPG can also be used in society for other purposes, such as a pedagogical aid tool [Pessôa 2020] or as a method for treating disorders [Vieira 2020].

The players' relationship with the fictional environment of the game's narrative proposes an immersive game environment. Immersion in an RPG game can include several different stimuli such as scenery, costumes, lighting and sound. In this context, elements such as soundscapes and soundtracks contribute to a better immersive experience for players in order to allow greater interaction between players and the narrative.

In this paper we present a proposal for tabletop RPG sound tool that works with different layers of sounds. These layers include a first set of sounds, which are related to the game's scenario, a second layer, related to non-player characters (NPC), and a third layer related to the actions of NPCs and players. More about this will be presented in Section 3.

Among the tools available to implement our sound system, we decided to use Pure Data and MobMuPlat, which will be better presented in Section 4.1 and 4.2 below. We also present in this section the prototypes developed in the context of this work. These prototypes were used in an experimentation context where it can be experienced in an RPG game. The details of this experimentation are presented in Section 5.

Certainly, the first prototype brought several questions, suggestions for improvements and difficulties encountered. Section 6 brings these discussions while Section 7 brings the considerations of this work so far.

2. Related Works

Ribas and Teixeira[de Oliveira Ribas and Teixeira 2016] also discuss sound design and immersiveness within the context of RPGs. The authors conducted a practical experiment to analyze the effectiveness of using sound resources during game sessions and obtained positive results. An interesting comment present in the paper is how the utilization of these resources can disrupt the flow of the game and that perhaps a separate person would be needed just to control the sound. This is a problem that we set out to solve through our design choices.

There are already some tools on the market that perform a function similar to our proposal, such as Tabletop Audio [1], Syrinscape [2] and Tabletopy[3]. Tabletop Audio gives users access to a set of high-quality ready-made scenes to use in their adventures. It also has a tab called Soundpad that allows you to create your own environments. Syrinscape is a very complete tool that also provides ready-made scenes, with the possibility of changing specific elements of them, such as the volume of each of the sounds that composes it. Scenes in Syriscape have predetermined states that change the parameters of their elements automatically, which makes them easier to use. This tool also has a global sound effects tab with various noises for the user to use whenever they want. With all these features, Syriscape comes very close to the proposal of the tool we set out to develop. Tabletopy provides on its page different sounds and sound effects that the user can use to set up their scene and represent events on the narrative.

Our tool, however, also has its differences. First, the tools mentioned above, despite providing some free resources, are paid. We wish to create a free and open source environment. Furthermore, our aim is to create a simple and versatile tool that best adapts to the user's needs. To do this, we define methods for organizing and transitioning between scenes in an automated way. We also intend to work on organizing the interface in a minimalist and highly customizable way.

3. Proposal

To start planning, we first got together with the authors and members of our research lab in a brainstorm session to decide what the requirements for this tool would be. These ideas were further discussed with other members of the computing course with experience playing role-playing games to form the basis of our project. First, we concluded that it was important for the tool to work on cell phones, as it is a technology that is currently easy to access and transport. Furthermore, a situation in which all game participants have cell phones is not unlikely. In this case, we can think of implementing controls over the network so that all players can influence the game sound. If everyone is using headphones, it becomes possible to work on sound spatialization using binaural audio[Roginska 2017]. A problem with spatialization is that it needs to be relative to the position of each character

[1]Available at: `https://tabletopaudio.com/`. Accessed 6/7/2023.
[2]Available at: `https://syrinscape.com/`. Accessed on 6/7/2023.
[3]Available at: `https://tabletopy.com/`. Accessed on 06/20/2023.

in the scenario. Because of this, the organization of sounds in 3D space needs to be unique for each player.

As for the tool's architecture, it is important for it not only to allow the use of audio resources dynamically, but also the organization of such resources in a way that makes sense within the context of creating adventures and RPG scenarios. Our goal is to build the soundscape of the different fantastic scenarios where RPG stories take place. Such scenarios have several elements that compose them, such as the people, streets and buildings of a city or the trees, plants and animals of a forest. Each of these elements may or may not have sound properties that will compose the landscape.

It is interesting to think that this tool has two distinct moments of use. First, we have the composition phase where the game master (GM), the one who narrates the history, will define which sounds he will use at each moment, as well as the characteristics (execution parameters) of these sounds. For this process, the tool should allow a high level of customization and versatility, so that the user can assemble his sound composition the way he wants. Here, it may also be interesting that the tool presents some ready-made resources that facilitate the creation. We may make some generic scenario templates available for users to modify. This also makes room for discussing collaborative creation processes. For this, it is necessary to create means for users to make the scenarios they created available to the community. This even brings the possibility of monetizing the tool by assigning prices to these ready-made scenarios.

The second moment of use is during the game session. In it, the GM will play all the sounds and effects he prepared during the creation of the scenarios for his adventure. In this context, the ease of use of the interface is the most important element. The user should be able to make complex modifications to the sound layer in a simple way. Sound control needs to be done in such a way that it doesn't disturb the flow of the game. The GM shouldn't have to search for sounds or change parameters manually to be able to make the changes they want.

To simplify the design of these spaces, we adopted a model with different sound layers, each layer associated with specific elements of the scenario. Such a structure allows sounds to be organized according to their functions, characteristics and importance within the scene. The scene, in turn, is configured as a set of sound resources that characterize an environment, for example: sounds that make up the scenery of a city, a forest, a cave, etc. Three layers were idealized in this work, namely:

- **Base event layer**: This is the basic layer of a scene. Its sound reproduction is constant and helps in the construction of the chosen environment. Taking a scene composed of a forest as an example, the base layer would reproduce sounds of wind passing through treetops. For this, the sound repetition resource (*loop*) is used with the intention that an audio is reproduced cyclically for an unlimited time. Another interesting resource that can be implemented in this layer are musical tracks, songs that can be played and that can contribute to the setting and immersion of the scene.
- **Sporadic events layer**: this is the layer that will receive sounds that are common to the context of the scene, but not constant or cyclic. In the previous forest example, the sporadic events layer could contain sounds from your fauna. Let us assume that the forest in question is inhabited by animals such as owls or crickets.

The constant reproduction of the hooting of these owls can give players the impression that these animals are chasing their characters. However, if these sounds are reproduced at longer intervals of time, this effect dissipates and helps to characterize the environment where the scene is located. In this case, features like timers are implemented in this layer.

- **Trigger events layer**: this layer is reserved for the reproduction of sounds that are a consequence of an event or a character's action. In the case of the forest example, the encounter with a creature not common to that environment can be characterized within this layer. If the characters encounter a dragon, for example, a roaring sound might be played at that specific moment. For this, a sound trigger mechanic through buttons is implemented in this layer.

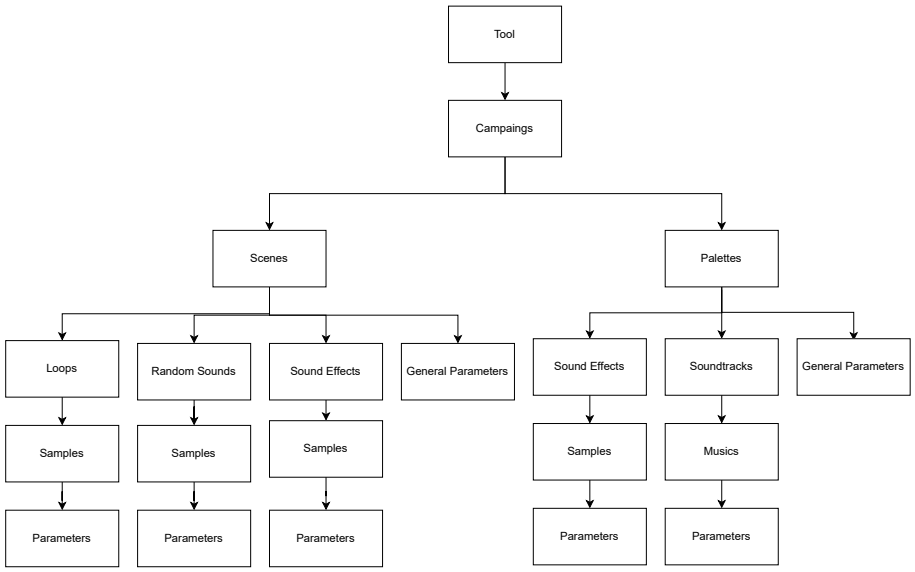

Figure 1. Tool metamodel

Figure 1 presents the meta model of the tool with all of its elements organized hierarchically.

Two main elements stand out in the tool's composition: scenes and palettes. A scene represents a certain moment in the RPG plot, whether it is related to a location or event. The GM will create or edit these scenes in the moment of composition and define the sound resources related to it, here called samples. These samples have three distinct classifications according to how they relate to the three layers presented previously. Samples from the first layer are Loops, played constantly. Those in the second layer are Random Sounds, played according to a frequency. Those in the third layer are the Sound Effects. They represent specific events, so they only play when triggered. The scene also has parameters such as general volume, to facilitate control by the GM. Each of the samples also has its specific parameters.

Furthermore, scenes can have different states. We define a scene state as the set of parameters used to play the sound resources of a scene at a given moment. These states can be defined by the user at the time of composition. With defined states, we can program automated transitions. This allows the GM to change current scene conditions by adding or removing elements, varying the volume or varying the playing frequency,

and switching between different scenes quickly without interrupting the game to change these parameters manually. If the light rain that was falling on the plain starts to turn into a thunderstorm, simply trigger a transition to increase the volume of the rain sound and the frequency of the thunder samples.

There are also battle sounds or spells, for example, that may be necessary regardless of the current scenario. Such sounds are part of the layer we call trigger events. Since these can be independent of scenes, we propose organizing them in a separate section called palette. A palette is a set of sound effects laid out in a simple way so that they can be played at the touch of a button. In these palettes, we also include the addition of soundtracks, which are part of the base events layer. It is quite common for GMs to use music during the game. At different times, whether when players are preparing to start the game or in simpler and more relaxed scenes of the narrative, a soundtrack is enough to create the necessary atmosphere. The possibility of separating these tracks in a palette gives the master more versatility in his composition. He can either assign tracks to the construction of the scene or leave them separated to play whenever he deems necessary.

4. Implementation and design decisions

There are currently a range of resources and architectures that can be used for the development of a sound tool such as audio editors, programming APIs [Schiavoni et al. 2012] or musical programming languages [Araújo et al. 2018]. Considering the creation of sound content itself, there are two possible paths to be followed: sound synthesis or the use of samples from sound recordings. In the case of synthesis, sounds are created from algorithms such as the generation of simple periodic waves such as sine waves, square waves, sawtooth waves and various types of noise that are modified by adding dynamic envelopes and controlling other parameters of these algorithms. Such a technique is quite efficient but has as it's limitations. The production of material by this means depends on the knowledge of techniques and synthesis algorithms such as frequency modulation (FM), amplitude modulation (AM) or others.

The use of samples is simpler since the production of new signals depends only on the recording of a certain sound. For this reason, the method chosen for this work was sound sampling. In addition, the use of new sounds is also possible with the use of files already recorded and available in free sound banks such as Freesound[4].

As we mentioned earlier, for the sake of practicality, we found it interesting that the tool can be run from cell phones. For this, a versatile possibility that would work for computers and cell phones is to use a web interface. This can be done through a Javascript library called WebAudio that allows manipulating sound resources through the browser. Another possibility we found is to work with the PureData(Pd) environment. Pd is already well known and used by artists and researchers to create multimedia applications. MobMuPlat provides a simple way to create interfaces to run Pd programs on Android and iOS phones. The following subsections describe these tools.

4.1. Pure Data

Pure Data(Pd)[Puckette et al. 1996] is an open source visual programming environment aimed at using multimedia resources. It allows the processing of sound, video, 2D and

[4]Available at `https://freesound.org/`. Accessed on 06/15/2023.

3D graphics, as well as input interfaces and sensors. As a visual programming environment, it facilitates use by artists and performers who do not have programming skills and facilitates prototyping.

4.2. Mobile Music Platform

Mobile Music Platform(MobMuPlat)[Iglesia 2016] is a mobile application that allows the execution and communication of a Pd program, called a patch, with an interface created from a multiplatform editor. In this way, it is possible to develop and test your program on the computer and transfer it to the cell phone to be used from the interface.

4.3. First prototype

We combined these two tools in the making of our first prototype. To begin with, it was necessary to create the patches used for the opening, execution and control of the samples that make up the scenes and palettes. Afterwards, we used the MobMuplat editor to create the screens. Finally, we establish the communication standards between the application patch and the interface. This is done by sending and receiving messages. Each interactive element available in the editor (buttons, knobs, sliders...) triggers a message that follows the pattern: /[name] [list of parameters]. These messages are handled by the patch and forwarded to the correct control parameters.

The figure above shows how one of the prototype screens was organized. Each screen has controls related to one of the scenes. When accessing the scene screen, just move the knob referring to the audio volume control to play it. The second layer audios also have a slider for their playing frequency. This frequency concerns the chance of that audio being played every second. At the bottom of the screen we have the set of sound effects to be used in the session. All share the same volume control and each effect is triggered from a button. The transition between scenes is done in a simple way. When accessing a scene page, the patch performs a fade out in all the others and a fade in in the current one.

As a first prototype, not all features presented in our proposal were implemented. The part of sound effects and tracks separated into palettes still needs to be built. We also didn't get to set up the state transition scheme within a scene and the controls need to be changed manually. This prototype was used to prepare some simple scenes to be tested on a real RPG table. Discussions regarding this experiment are presented in the section 5.

5. Experimentation

We held an RPG session with the participation of one of the authors and some volunteers. All participants had a good previous experience with the game. We asked the participant who was willing to play the role of a GM to create a simple adventure and suggest the composition of some scenes to be used.

Before the session took place, we talked with the GM to understand how he usually works with game sound. Normally, he uses several different applications such as Spotity, Youtube and the Syrinscape tool. When questioning the master about why he used different applications and not just Syrinscape, he said he had difficulties adding audio to the available ready-made scenes and working with them in a versatile way. He commented

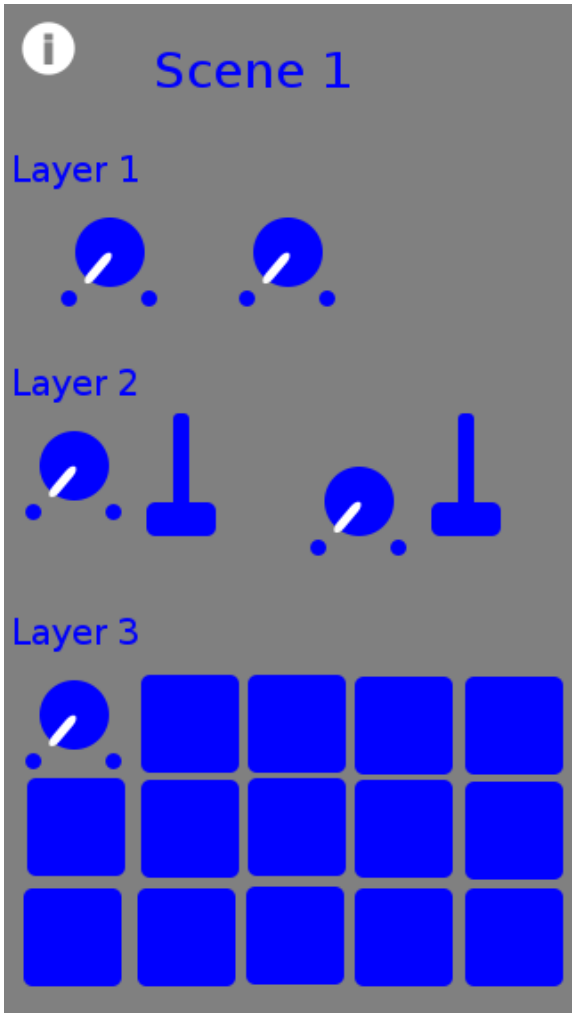

Figure 2. Prototype screen example

that a very important element for the game's sound design is the use of soundtracks. And this is a task that he manages to carry out in a simpler way in environments like Youtube.

The test game lasted 3 hours. During the session, we used the scenes and effects that we had prepared. The part of the soundtrack, however, was performed by the master from the Youtube platform, since our prototype did not include music for the scenes. At the end, we held a conversation to find out what was the opinion of the participants regarding the experience. All players had a very positive response. They stated that both ambient sounds and the use of sound effects at specific moments were very effective in enhancing the story experience. Here is a list of lessons learned in this test session:

- It's important to have easy a way to play soundtracks, as those are often the main resource used for the RPG sound.
- It's not necessary to have a sound for every action. Our GM doesn't like to use sound for combat, for example.
- Every GM have his own style, so customization is key to be able to serve different users.
- Because of the nature of RPGs, unexpected events can occur in the game. Having generic scenes and sound effects can be useful in this situation.

Figure 3. Picture of the RPG session

6. Discussion

RPG is an imaginative game. It aims to allow participants to get away from reality for a moment and live as a character within a fantastic universe. For this to happen, it is necessary to reach a high level of immersion. We can understand immersion as "the feeling of being surrounded by a completely different reality"[Murray 2017].

Ermi and Mäyrä [Mäyrä and Ermi 2011] discuss the gameplay experience in the context of electronic games from the element of immersion. They propose a model that brings three dimensions of immersiveness: sensory, imaginative and challenge-based. We can use this model to analyze immersiveness in RPG. The imaginative layer relates to fantasy, story, world and characters. It's the GM's job to keep the players engaged in their narrative, while the players must be willing to jump in and let the fantasy take over. The challenge-based dimension happens through the puzzles, traps and enemies presented by the master.

The dimension of sensorial immersiveness is the point where RPGs differ from electronic games. In video games, this dimension is given mainly through the quality and style of the graphics and the sound layer. These elements are not naturally present in RPGs. The only representations of the game world needed to play are the character sheets and rulebook, everything takes place within the imagination of the participants. However, some resources can be used to improve immersion such as drawings, maps and miniatures. If they are willing, players can make more effort in this regard by preparing an environment according to the theme, using costumes and objects to represent game items. The sound of the RPG enters into this context.

Therefore, sound is neither the only nor the main element capable of increasing the immersiveness of the participants. But, it is an important part of the sensory dimension

and has significant impact. This conclusion was reinforced by the positive response we got from all our test session participants. In addition, in the experience of the authors and volunteers who participated in the game, it is common for RPG masters to use some sound resource in their adventures. These features are often music to set the mood of the scene.

Although it worked for an initial test, our prototype still has some issues to be resolved. The prototype still cannot work the transition between states of a scene, which would have facilitated the master's control. In addition, we only focus on the application of ambient sounds and do not prepare ourselves to work on soundtracks. During the RPG section we had, it became clear that using music is a more frequently used resource and has more prominence than ambient sounds. However, everyone agreed that ambient sounds had a positive impact on the gaming experience. Finally, it only brings a few fixed scenes and we still haven't worked out a way to allow changing the resources of a scene and creating new ones. From the experience that the test participant who played the role of GM presented us with in using similar tools, the ease of modification and organization of resources must be an important objective of ours.

7. Conclusion

We present the proposal of a tool to help the sound in tabletop RPGs. We discussed the architecture and organization of the tool, technologies used, prototyping and the results of a test we were able to do with the help of some volunteers. This project is part of a course conclusion work that is still in its initial state. We have not yet managed to reach full implementation of all the ideas presented. However, experience with the prototyping process and using the prototype on a real table was enough to generate some interesting discussions.

We conclude that sound is an effective method to increase immersiveness on an RPG and that the development of our tool is moving in the right direction. Much of the sound layer is usually done through soundtracks and this is an issue that we had not paid attention to at the beginning. In addition, we reinforce our notion that versatility and ease of use are very important requirements in this context, as this was the main complaint of one of the test participants regarding his experience with similar tools. We will continue with the prototyping and testing process to implement the features that are still missing, such as the screen for creating scenes and defining transitions. Afterwards, we will use the result to proceed to the final implementation of the tool.

7.1. Acknowledgments

Agradecemos ao Professor Flávio Schiavoni pela paciência e dedicação em nossa orientação e a todos os integrantes do laboratório de pesquisa ALICE que nos acompanham nessa jornada da vida acadêmica. Agradecemos também à PROAE por dar o suporte necessário para continuar estudando nesta universidade e também a PROPE / UFSJ pelo apoio financeiro a este projeto. Agradecemos ainda o apoio da FAPEMIG e do CNPq que mantém o fomento da pesquisa deste projeto.

References

Araújo, R. R. d., Sandy, J. M. d. S., Cirilo, E. J. R., and Schiavoni, F. L. (2018). Análise e classificação de linguagens de programação musical. *Revista Vórtex*, 6:1–24.

de Oliveira Ribas, N. and Teixeira, N. S. (2016). Design sonoro no rpg de mesa: Uma estratégia para imersão. *Blucher Design Proceedings*, 2(9):3499–3509.

Iglesia, D. (2016). The mobility is the message: The development and uses of mobmuplat. In *Pure Data Conference (PdCon16). New York*.

Mäyrä, F. and Ermi, L. (2011). Fundamental components of the gameplay experience. *Digarec Series*, (6):88–115.

Murray, J. H. (2017). *Hamlet on the Holodeck, updated edition: The Future of Narrative in Cyberspace*. MIT press.

Pessôa, B. M. A. (2020). Entre professores e mestres: O rpg e as experiências docentes no ensino de história. *História & Ensino*, 26(2):337–356.

Puckette, M. et al. (1996). Pure data: another integrated computer music environment. *Proceedings of the second intercollege computer music concerts*, pages 37–41.

Roginska, A. (2017). Binaural audio through headphones. In *Immersive Sound*, pages 88–123. Routledge.

Schiavoni, F. L., Goulart, A. J. H., and Queiroz, M. (2012). Apis para o desenvolvimento de aplicações de áudio. In *Anais do IV Seminário Música Ciência Tecnologia*, São Paulo, Brasil.

Vieira, F. L. (2020). Role-playing game (rpg) como ferramenta terapêutica para intervenções de pessoas com transtorno do espetro autista. *Fórum Regional de Pesquisa e Intervenção (FOR-PEI)*, (2):38.