Procedural Music in Games

José Eduardo de Almeida Ayres[1]*, Pedro Arthur dos Santos Souza[1]†, Vitor Guerra Rolla[1]‡, Luiz Velho[1]

[1]Visgraf – Instituto de Matemática Pura e Aplicada
Estrada Dona Castorina, 110 – 22460-320 Rio de Janeiro, RJ

jeaayres@impa.br, pasouza@impa.br, vitorgr@impa.br, lvelho@impa.br

Abstract. *This workshop will bring to the audience an introduction to the Chuck audio programming language, to the Unity game engine within a hands-on experience how one can use such technologies to achieve a new level of immersion through procedural generated sounds responding to game events and visual effects. The workshop is intended to a broad audience ranging from programmers to ones with little to no knowledge in the field.*

1 Background and motivation

With the advent of video games in the eighties, sounds (and music) were a crucial piece of immersion giving depth to the game-play. Even if nowadays graphics are on the spotlight, sounds are still a significant area in development. New technologies, for instance, Chunity [1], allows developers to provide fantastic sound experiences through game-play.

Audio can deeply connect to our emotions contributing to the immersion of the player in the game. Video games became a broad entertainment platform that gave birth to new technologies which facilitate the developer to implement his/her ideas. One of these technologies is the Chuck programming language which offers total control over sound synthesis and playback to the developer.

The Chuck programming language provides a wide range of tools for generating sounds algorithmically and in conjunction with the Unity game engine [2], which provides a simple environment for game development. One can achieve great results without requiring much knowledge of such technology. Therefore these are perfect for a hands-on workshop where anyone interested in the subject can learn and produce good results.

Our research group is continuously experimenting and developing new ideas through these technologies (Chuck, Unity), having already presented various workshops on the subject and developed a few apps. As said by [3], even if you use commercial software to create sounds, ChucK has new ways to control and interact with them. It's like being able to put a new engine in your car.

2 Program

The workshop program will be essentially practical. We start by showing you how to use the basic features of Unity: installation, interface, game objects, assets, and scripts in C# (1 hour). Then we will show the basic features of ChucK programming: the standart library, the math library, oscillators, sampling, and envelopes (1 hour). Chuck is not difficult because it looks a lot like C. In the next 2 hours, we will develop a game project and create the background music and sound effects with Chuck and Unity.

3 Infrastructure and target audience

The infrastructure required for the workshop is, in a way, very simple. The required physical space can be a classroom or an auditorium with a projector and speakers, we're going to use our own computers to show examples. We will need internet connection, the software configuration will be part of the workshop program. If the participant wants, he can follow the examples in his computer, but we will do everything on-the-fly, and everyone can follow without a computer.

The workshop was developed to be simple and to reach a good range of people. To take more advantage of the workshop, it is good that the participant has some previous knowledge in programming, but it is not mandatory. The target audience of the workshop is people of all ages who have an interest in the development of computer music and its applications, i.e. computational sound synthesis, creation of synthetic musical instruments, on-the-fly audio programming, and production of sound and music in games. In this vast audience are contained many professionals, like musicians, programmers, designers, and mathematicians. Anyone who is interested in using the tools described as means of producing audiovisual experiences. Substantially, everyone at SBCM is welcome in our workshop.

This workshop will contribute to SBCM by presenting, promoting, and training its participants in the discipline of audio and video game programming. Concerning technology, this event will add to the dissemination of Chuck and Unity programming interfaces.

References

[1] Jack Atherton and Ge Wang. Chunity: Integrated audiovisual programming in unity. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 102–107, 2018.

[2] Daniel V. de Macedo and Maria Andréia Formico Rodrigues. Experiences with rapid mobile game development using unity engine. *Comput. Entertain.*, 9(3):14:1–14:12, November 2011.

[3] Ajay Kapur, Perry R Cook, Spencer Salazar, and Ge Wang. *Programming for musicians and digital artists: Creating music with ChucK*. Manning Publications Co., 2015.

*Supported by FAPERJ.
†Supported by CNPq.
‡Supported by FAPERJ.