
81.

Towards Improved Literacy in Computer Programming Among Artists

Elliot Gertner (Ventura College) & Ventura Patrician Waterman (Santa Ana College)

Abstract

Artists are creative people who excel in creating very complex 3D images, animation and sound presentations but find it difficult and cumbersome to deal with low-level programming details. Visual programming techniques (also known as visual studios) are visual only to the coders with computer science backgrounds; to the artists, visual studios still look like a medical prescription and taste like medicine. Recent releases in visual game programming environments and game engines such as Unity, Unreal, and GameSalad appear to be promising to advance the art of programming to non-programmers. In this paper we share our plans and preliminary experience in those game engines to build a community college level course in programming for artists and other non-CS majors.

Introduction

Games/Interactive Media production requires both artists and computer programmers; however finding programmers to team with has become an increasingly difficult task. According to the Bureau of Labor Statistics, the demand for programmers is twice as large as the demand for multimedia artists. This trend continues today as there is a severe shortage of programmers. As a result, artists are finding themselves having to take on some of the responsibilities that were once handled solely by a programmer. Job postings for digital artists are now requesting applicants to possess a “working knowledge of computer programming”.

In pursuit of a game design course for artists and other non-programmers, we have reviewed existing literature on game development. The benefits of learning by using digital game design are described in *Digital Games* (Clark 2015). Preliminary experience in teaching high-school teachers to teach game design are described in *The Iterative Design of Eight Week Course* (Aubrecht, 2015). An interdisciplinary video game design program has been developed in the University of Texas as a collaborative program between the Computer Science and Arts Department (UT-RT, 2013-2016). An attempt to improve communication between Computer Science and Arts students has been developed in California State University Long Beach as an interdisciplinary course in Software Engineer (Gertner 2016). In addition to the research experiences described above we have also reviewed commercial instruction book of the type, *Sams Teach yourself game development in 24 hours* (Unity, Unreal, GameSalad). We contrast these high-level game engines with very low-level programming library such

as the Simple DirectMedia Library (SDL 2013) which is suitable to use with general programming languages. Using these experiences referenced above we plan to develop a community level course for artists, to teach them programming skills.

In the remaining portion of this paper we propose an outline of the course and invite the conference proceedings readers and conference audience to share comments with us. In particular, we focus on the logic control tools in game design. We found these tools in game engines are very high level but rigid in contrast to the programming language tools such as Simple DirectMedia Layer (SDL) that are low-level but very general – one can program anything (SDL 2013). The challenge is to find a happy medium, a general programming interface that is understood and liked by both the artists and computer programmers.

Course Outline

This section reviews two example of course outlines. Existing courses in video game programming are very similar to the courses in film production. Although the course below starts with an introduction to computer programming and ends with a game demo, the bulk of the material is about tools for digital media; see an example below.

Game engine-based course outline:

- *Introduction to Computer Games*
- *Animation Techniques*
- *Simulating Physics*
- *Graphical Sprites, Game Worlds, Game Intelligence*
- *Analysis, presentations and demos of Sample Games and Student Projects*

That the above looks like a movie production course is not a coincidence because the faculty in Arts have a lot of experience in movie production. This experience naturally leads to a course that looks like a video game production. Although such courses do teach some skills for jobs in video game studios, they do not teach programming. And the gap remains between the artists and computer programmers.

We have examined tutorials for game engines such as GameSalad, and we found that the programming tools are often hidden in advanced sections that are frequently overlooked by both the faculty and students. Furthermore, programming tools are very rigid and do not lead themselves to general programming. For example, GameSalad limits the behaviors to the following three types: rules, persistent behaviors and action behaviors. These behaviors appear to be very limited to a computer programmer. This is in contrast to programming with SDL, which is oriented towards the programmers.

SDL Book Outline:

- *Introduction to SDL*

- *Drawing*
- *Exploring movement*
- *Handling game states*
- *Data-driven design*
- *Game examples: creating Alien attack, creating Conan the caveman, etc...*

SDL programming and game engine programming are two different worlds that are vastly apart, in spite of the fact that both intend to help an artist or a programmer to implement a video game.

Discussion

In this presentation we focus on the programming logic control tools in game engines such as Gamesalad, Unreal and Unity and compare it against generic programming tools and libraries such as the Simple DirectMedia Layer (SDL). We find that game engines are very high-level and easy to program but are not flexible, while SDL interfaces are very low-level but are difficult to program and are typically taught in the computer science departments. In this presentation, we are in search of a happy medium that is acceptable to both artists and computer programmers.

References

Aubrecht, M., Schutter,B., & Butler,D. (2015) The Iterative Design of Eight-Week Course Aimed at Developing a Community of Gameful High School Teachers. Gaming GLS11 Conference Proceedings, Madison, Wisconsin, July, 2015.

Clark, D., Tanner-Smith E., & Killingworth (2015). Digital Games, Design, and Learning: A Systematic Review and Meta-Analysis. Gaming GLS11 Conference Proceedings, Madison, Wisconsin, July, 2015.

Dickson, P. (2015). Using Unity to Teach Game Development: When you've Never Written a Game. Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education. Pages: 75-80.

Gertner, E., & Moore, A. (2016). "Let's Make a Game!" Bridging The Computer Science/Fine Arts Divide Using High Impact Practices. 19th Annual CSU Teaching and Learning Symposium. San Jose, California, October, 2016.

UT-RTF (2013-2016) Game and Mobile Media Applications Program (GAMMA). University of Texas Interdisciplinary undergraduate certificate program in video game design.

SDL (2013) SDL Game Development, book by Shaun Mitchell, Packt Publishing, 2013.

Unity, (2013). Geig, M., "Unity Game Development in 24 Hours. Sams Teach Yourself Series.

Unreal, (2015). Cookson, A., DowlingSoka R., Crmpler, C., Johnson, and T. Unreal Engine 4 Game Development in 24 Hours.

Gamesalad, (2015). Cross, J. Learn Mobile Game Development in One Day Using Gamesalad: Create games for iOS, Android and Windows Phone.