Combining games with theatre to create an interdisciplinary learning experience for Computer Science students

Joe Geigel  
Department of Computer Science  
Golisano College of Computing and Information Sciences  
Rochester Institute of Technology  
jmg@cs.rit.edu

Marla Schweppe  
School of Design  
College of Imaging Arts and Science  
Rochester Institute of Technology  
mkspbh@rit.edu

ABSTRACT

In this paper we describe our experiences in combining theatre with games programming as a means of providing a collaborative experience for Computer Science students. With the goal of creating a theatrical production in a virtual space, we’ve designed a set of companion courses, geared toward those looking to work in the electronic entertainment industries, which emphasizes teamwork, cooperation, and successful collaboration between artists and technologists. This paper focuses on the Computer Science course, where students work together towards adapting a gaming engine for theatrical purposes.

Categories and Subject Descriptors

K.3.2 [Computing Milieux]: Computers and Education-Computer and Information Science Education [Computer Science Education]

General Terms

Design, Experimentation

Keywords

Interdisciplinary coursework, theatre, games education.

1. INTRODUCTION

Imagine a computer science course where students develop a theatrical performance for a live audience. Like most theatre, excitement is in the air. The ‘crew’ of programmers is anxious and excited, wondering how the audience will respond to their performance. The difference is that in this performance, the action takes place in a virtual gaming world; actors are all avatars carefully created and designed by artists; the performance is realized using motion control devices, data gloves and a virtual stage manager. The audience sits at computers, rather than in seats at physical theatre, where they watch and respond to the performance.

Not the usual programming classroom, this classroom is alive. Students are seeking solutions to problems, asking for components to be completed so they can be tested or integrated into the whole. Everyone plays a key role. Everyone has to come through to make the event a success; these students are fully engaged, depending on each other, and excited about the end result. This is what Virtual Theatre is all about. A unique educational experience for Computer Science students that combines theatre with games programming and the goal of teaching skills beyond the traditional CS course.

For the Computer Science student looking to enter the electronic entertainment industry, the skills required for success expands well beyond programming and algorithms. Although technical expertise is important, the interdisciplinary nature of these industries additionally demands practice in a number of non-technical skills. Qualities such as teamwork, collaboration, independent thinking, experience with a large project, flexibility, and problem solving have been reported as being critical in the success of those looking to work in technical positions in both the computer animation [1] and the game development[2,7] fields. In addition, successful projects in these fields rely on effective communication and uncompromised cooperation between artists and technologists.

Courses that foster such collaboration and address these non-technical issues have been reported in Animation [4], Image Synthesis [5], Virtual Reality [6] and game programming [2,7]. Common themes shared by these courses include: a capstone experience; collaboration of artists and programmers working on a single project; and a team based approach. In our work, we embrace theses themes and, adding a theatrical component, we create a learning experience with the goal of further sharpening students’ collaborative and teamwork skills.

The courses, which are an integral component of the larger project (Virtual Theatre), create an educational experience that provides an opportunity for students interested in working in the electronic entertainment industries to experience, first hand, the type of collaboration between artists and technologists that is crucial to success in that industry.

The education component of the project is implemented as a set of companion courses, some offered in the School of Design and others offered by the Department of Computer Science. The focus of this paper is the courses taught in Computer Science.

The remainder of the paper is structured as follows: In section 2, we discuss our motivations for using theatre as an application domain for the courses. This is followed by a brief overview of the Virtual Theatre project and the details of the Virtual Theatre CS course, in Sections 3 and 4 respectively. Results and observations from the course offerings are discussed in Section 5 and we finish the paper with Conclusions and Future work in Section 6.
2. THEATRE AS AN EDUCATIONAL MOTIVATOR

Theatre, by its very nature, is a collaborative art. Theatre can be defined as a place of action; field of operation [8]. The telling of a theatrical story takes place in a common field of operation and involves a number of actors, directors, designers, and technicians, who create the storytelling space and the action that is viewed within it. This collaborative environment provides a natural foundation for a learning experience that emphasizes teamwork and cooperation between artists and technologists.

In addition to the collaborative aspects of theatre, the discipline of theatre teaches a number of other highly valuable skills that can contribute to student success, regardless of their area of study. [9] We list some of these skills in Table 1 below. Not surprisingly, this list mirrors the skills previously mentioned as critical to those looking to working in the electronic entertainment industries.

<table>
<thead>
<tr>
<th>Table 1 - Skills learned from working in the theatre</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Oral Communication Skills</td>
</tr>
<tr>
<td>• Creative Problem Solving Abilities</td>
</tr>
<tr>
<td>• Motivation and Commitment</td>
</tr>
<tr>
<td>• The Ability to Work Independently</td>
</tr>
<tr>
<td>• Time-budgeting Skills</td>
</tr>
<tr>
<td>• Initiative</td>
</tr>
<tr>
<td>• Promptness and Respect for Deadlines</td>
</tr>
<tr>
<td>• Respect for Colleagues</td>
</tr>
<tr>
<td>• Adaptability and Flexibility</td>
</tr>
<tr>
<td>• The Ability to Work Under Pressure</td>
</tr>
</tbody>
</table>

Others in the field of Computer Science have certainly recognized the value of theatre as an educational stimulus. Brenda Laurel uses theatre as a metaphor for Human Computer Interaction [10]. Others have used a more direct approach, combining computer graphics education with traditional theatre [11,12]) which serve as a motivation for our approach.

3. THE VIRTUAL THEATRE PROJECT

The Virtual Theatre Project is an interdisciplinary project between the department of Computer Science and the School of Design at the Rochester Institute of Technology. The goal of the project is to explore distributed theatre and enable the presentation of a theatrical performance in virtual space, where actors, crew, and audience can share and participate, possibly from different physical locations, in a single theatrical performance over a distributed network [13].

The Virtual Theatre project serves as both a research project and educational experiment. There are three major facets to project:

1. Artistic – Exploration of means to create engaging distributed performance and to create the aesthetic elements of such a production.
2. Technical – Defining, specifying, implementing, and documenting, the technical components to enable such a performance
3. Educational – Allow for first hand student involvement the research aspects of the project, both technical and artistic and to create a collaborative experience for students in both camps.

Computer Science students are responsible for the technical aspects of the project. The goal of the technical team is to create a distributed VR framework to support this theatrical interaction. Rather than starting from scratch, the framework is built on top of a gaming engine which provides layered access to low level capabilities (graphics, sound, networking) required by such a system. Similar to the spirit of Machinima [14], (filmmaking using gaming technology), our framework adapts the functionality provided by a gaming engine and wraps it in the language and processes of the theatre, as illustrated in Figure 1. Control of, and interaction with, the system during a production is done at this more intuitive layer.

The framework assumes a shared object oriented database model where items in the virtual space are represented by software objects, which are distributed amongst all those accessing the shared space. 3D visual representations of these objects are created off-line by the design team using a modeling package and imported into the system prior to the production. The design team is also responsible for creating stage elements, defining lighting, and generating scripted animations for the avatars on the virtual stage.

During an actual performance, individual workstations, each resident with his or her own instance of the framework and assets, are networked together and configured to share a single virtual space as shown in Figure 2.
4. VIRTUAL THEATRE COURSES

Logistically, the ultimate outcome of the courses is to create the technology and assets to realize a particular theatrical performance in a virtual space. Art students are tasked to design and create the aesthetic elements of the performance (models, textures, animation sequences) whereas the CS students are responsible for the design and implementation of the technical infrastructure to allow real time interaction and manipulation of these models. Unlike the courses presented in [4,5,6], we did not strive for a common classroom experience for both artists and technologists. Instead, students were instructed in their own discipline; however, their work was guided and shaped by constraints and requirements defined by the other half. As such, the project was implemented via a set of companion courses, some specifically for design students, and others for CS students, rather than a single course consisting of both sets of students. Efforts were made to have the companion courses offered at the same time, when possible.

The production to be realized in a given year’s offering of Virtual Theatre is decided upon prior to the start of the school year. This production is chosen based upon technology and devices available. All course activities included the definition of the elements that student would be creating is driven by production needs both on the art and the tech side.

The remainder of this section will provide details of the course taken by Computer Science students.

4.1 Pre-requisites

The Virtual Theatre course is listed as an advanced course in the area of Computer Graphics. Within this area, the CS department also offers Graphics I, which focuses on fundamentals of graphics; Graphics II, which focuses on rendering and global illumination; Computer Animation, which takes a low level look at the algorithms and techniques of animation; and Procedural Shading. All graphics courses are co-listed for both Graduate and Undergraduate students. Students studying graphics are also encouraged to take AI for Interactive Environments, which focuses on the use of artificial intelligence methods in 3D environments, as well as game programming courses offered by our Information Technology department.

Graphics I is the only listed prerequisite for the course. Although it is helpful for students to have taken some of the more advanced graphics courses, this is not required as there is much flexibility in the work that any individual student will be tasked to do. In fact, since there are several required tasks involving areas that are non-graphics related (e.g. networking, audio), we do allow students with no graphics experience to enroll in the course with permission of the instructor. Such students should show an expertise in one of these non-graphics areas and an interest in graphics, games, or 3D environments.

4.2 Learning Outcomes

In designing the course for Computer Science students, we had the following goals in mind:

- Provide students with a rich and relevant technical experience.
- Give students a taste of a real-world working environment. Thus, a major emphasis was placed on having students collectively work on a large-scale project from which any individual student will have the responsibility for a smaller piece.

- Define the project such that collaboration with artists is a requirement for success of the project as a whole.

Given these goals, the following student learning outcomes were defined for the course:

- Students will be able to describe the components of a distributed virtual reality system and the latest advances in designing such components.
- Students will be able to apply existing graphics/VR toolkits, APIs, and software packages in the construction of a large scale project.
- Students will be able to specify, design, implement, and document an integral component of a larger software project related to computer graphics and VR.
- Students will be able to successfully participate in a interdisciplinary team based project with responsibilities assigned within and between individual teams.

Unlike other courses that involve capstone projects in gaming (e.g. [3][7]), an outcome of this course involves the application of a gaming engine as opposed to the creation of one. This is due, in part, to the short time span of the course (10 weeks) and the fact that the skills that we hope to develop in the students are addressed by collaborative nature of the course. Since the focus of the work for a particular individual is quite specific, the learning curve of working with the chosen gaming engine to address each task is manageable given the timeframe of the course.

4.3 Team Assignments

The primary goal of the Computer Science course was to build the Virtual Theatre layer as mentioned in Figure 1. Towards this end, students are divided into a number of teams; each team responsible for a particular technical component of this layer. Table 2 lists some of the teams and their responsibilities.

![Table 2 - Responsibilities for student teams](image)

<table>
<thead>
<tr>
<th>Team</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset Import</td>
<td>Creation of tools and classes for importing models from design team.</td>
</tr>
<tr>
<td>Audience Participation</td>
<td>Creation of software objects representing audience members and providing capabilities for audience participation.</td>
</tr>
<tr>
<td>Audio / Music</td>
<td>Creation of a subsystem for distributed audio and music.</td>
</tr>
<tr>
<td>Character Motion</td>
<td>Creation of classes to enable playback of predefined animation sequences created by design team.</td>
</tr>
<tr>
<td>Motion Capture</td>
<td>Creation of classes for interface with motion capture devices.</td>
</tr>
<tr>
<td>Networking</td>
<td>Creation and management of networking infrastructure.</td>
</tr>
<tr>
<td>Staging / Lighting</td>
<td>Creation of software classes representing staging and lighting commands and cues.</td>
</tr>
</tbody>
</table>
The instructor plays the role of technical director, keeping teams on track with regard to serving the ultimate needs of the entire production and the production schedule. Task requirements for work to be performed by each of the teams are predefined based upon the needs of the production. Given these requirements, each team must design a solution for the task at hand, develop an implementation plan and document the final deliverable. Students are pointed to relevant papers and articles from the VR, graphics, and animation literature to assist them in their design decisions.

Teams are hand assembled by the instructor, based upon the interests and the expertise of the students taking the course in a given quarter. Graduate students are expected to take leadership roles in the teams. The goal is to have as many teams as grad students. With each course offering, enrollment has easily supported natural team assignments. On the occasion when grad leaders could not be assigned, the instructor stepped in as the director for the team.

4.4 Course Delivery

The course is given in a computer laboratory consisting of 20 Windows based PCs equipped with advanced graphics hardware. Software to be used by the class is preloaded on these machines. The software is also available for students to install on their own personal laptops, though the use of laptops is not a requirement for the course. In addition, students have access to our Virtual Reality laboratory, which houses a number of VR devices along with specialized, dedicated hardware that is used to interface with these devices. Currently, our list of VR devices include a single node Flock of Birds Motion Capture device, several datagloves, a head mounted display, and a full body motion capture system.

Classes are scheduled 4 hours weekly, divided into 2 two-hour periods. Lectures are given only in first 2-3 weeks of course, with the goal of familiarizing students with the processes of theatre, providing details on the given production being targeted, and instructing on the use the gaming engine and toolkit being utilized. The remainder of class periods is used as working sessions. Although it is expected that a good deal of the work will be done outside of class time, the working sessions assure guaranteed time when teams will have time together to meet.

It is important that each team makes steady progress as the quarter advances. With the goal of keeping the teams on track, each team is required to give short demos periodically during the quarter (“checkpoints”) illustrating the progress made. Each team leader, in conjunction with the instructor, will determine the expectations for the checkpoint demos.

The course culminates with a performance demonstration that takes place during finals week. In preparation for this demonstration, team leaders integrate the deliverables from all of the teams into a single unified framework. Assets created by the design team are imported and a demo performance is presented.

4.5 Student Assessment

Student assessment for the course is performed on two levels. The first involves direct instructor evaluation of student work. This is performed on a per team basis where each team is evaluated based on the design, implementation, and documentation of their assigned component given the requirements. Teams are also graded on the incremental demos shown during the checkpoint sessions. Secondly, students provide peer evaluations as input to the instructor. In these peer evaluations, students are asked to judge each team, as a unit, and each member within their respective team. In addition, grad students are evaluated on their leadership qualities in directing the work of the team.

5. RESULTS

We have been offering the Virtual Theatre course annually, in the spring since 2004. For each course offering, a one-act improvisational theatrical piece was created as a target performance. A list is given in Table 3. The story of and interaction required by each piece was influence by the set of VR devices at our disposal at the time of the course.

Table 3 - Virtual Theatre Productions

<table>
<thead>
<tr>
<th>When Offered</th>
<th>Production</th>
<th>Interface Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spring 2004</td>
<td>What’s the Buzz?</td>
<td>Single node FOB motion capture device, SDT dataglove keyboard / mouse</td>
</tr>
<tr>
<td>Spring 2005</td>
<td>Getting By</td>
<td>Full body motion capture keyboard / mouse</td>
</tr>
<tr>
<td>Spring 2006</td>
<td>Critics</td>
<td>Full body motion capture SDD Dataglove P5 Dataglove keyboard / mouse</td>
</tr>
</tbody>
</table>

As a direct outcome of the course over the years, we have built two versions of the Virtual Theatre System (Table 4), each using a different gaming engine as a foundation. In the spring of 2004, we built upon MUPPETS, a distributed, collaborative virtual environment originally designed for enhancing student education in the areas of programming and problem solving [15]. We continued with MUPPETS in the Spring of 2005, building upon the work done in 2004. In an attempt to give students experience in working with a commercial engine, we switched to RenderWare Graphics[16] in 2005, using the RakNet library[17] for networking.

Table 4 - Virtual Theatre Systems

<table>
<thead>
<tr>
<th>Engine</th>
<th>Development Language</th>
<th>When used</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.0</td>
<td>MUPPETS</td>
<td>Java</td>
</tr>
<tr>
<td>V2.0</td>
<td>RenderWare (graphics) RakNet (networking)</td>
<td>C++</td>
</tr>
</tbody>
</table>

Enthusiasm among the students was overwhelming, greatly exceeding our expectations. In a sense, and quite unexpectedly, the theatrical metaphor extended to the overall feeling of the project. Consistently we’ve observed that, by the end of the quarter, technical team boundaries spontaneously evaporated with all students working together equally towards the goal of completing the system. The project became more than a course, with the drive to finish mimicking that of stage hands on opening night.

1 http://muppets.rit.edu
Most of the collaboration between artists and the students in the class focused on importing and adapting the assets created by the design team for use in the gaming engine. It was of utmost importance that the artistic aesthetics of the final models be preserved given the constraints of the gaming system. Students had to work together to create innovative solutions to achieve this, some which involved modification of the models, whereas others involving the creation of software and tools for automating the conversion.

Interactive sessions, during which these problems were addressed, offered the richest collaborative experience. During these sessions, technologists learned the thinking process of the artists and learned how to specify and implement tools to meet the needs of these users. The artists, on the other hand, learned to adapt their work to the needs of a real-time, interactive VR environment. In addition to peer evaluations, students were also asked to evaluate the course. Table 5 lists some of the more popular responses, in order of frequency of replies, when students were asked “What have you learned in Virtual Theatre?” Note that although technical content is high on the list, many of the other goals of the course are well represented.

Table 5 -- Student response on what they learned

- Technical Content
- Teamwork
- Integration on a Large Scale project
- The importance of deadlines
- How to work with artists
- How to work independently
- Problem solving
- Importance of communication
- Importance of backups
- That class can be fun

Finally, in several cases, the work started in the Virtual Theatre class has been used as the basis for a Graduate project in Computer Science. Aspects of the system have been extended as graduate work involving both motion capture and behavioral models.

6. CONCLUSIONS

We feel that collaborative nature of the course, in both the interaction between technical teams and between technologists and artists, resulted in an enhanced learning experience for all involved. Student work, excitement, and especially interactivity, consistently exceeded expectations. Defining student and team tasks as a portion of a larger project resulted in a communal feeling towards the project as a whole with all contributing and taking responsibility for the final product. Finally, placing the project in a theatrical paradigm, a natural structure for collaboration, further enhanced the interactive experience for both students and instructors.

We plan on offering the course again in the Spring of 2007. In addition, we are considering applying the same model and course structure to domains other than theatre, where interaction in a virtual 3D space would be appropriate.

7. ACKNOWLEDGMENTS

Acks will go here.

REFERENCES

Columns on Last Page Should Be Made As Close As Possible to Equal Length