GAME 1023
Bob Pendleton

Game Programming & Graphics

Course Description

This course is intended for programmers who want to enter the field of game design and development, but who do not have training in game programming techniques including 2D and 3D graphics, sound, and animation needed to complete a game programming project.

Topics covered in the course include: Object oriented design of games, discrete event simulation and state machines, managing game time, game GUI implementation, and introduction to 2D and 3D graphics, sound programming, and game design.

Course Length: 60 hours

Course Rationale

Video games are a rapidly growing high technology industry. The demands that video games place on computing hardware have driven the development of computer graphics and sound hardware for several years. Unlike more traditional IT software development the largest markets for video games are not on PCs and do not run on common PC operating systems. For that reason this class will emphasize portable game development skills and cross platform tools that are applicable to many game platforms and not just the PC.

Course Objective

Teach basic concepts of game development including:

1. Understand 2D and 3D graphics concepts.
2. Develop input management.
3. Write event based programming.
4. Write time based animation.
5. Perform basic computer sound concepts.
6. Design object based organization of game programs.
7. Understand game design.

Instructional Methodology

This is a hands on course. Students will be required to develop a simple game program that demonstrates the use of animation and sound. Students will also be expected to participate in a group design project. Grades will be determined based on the number of objectives met by the student projects.

Students who chose to do so will have the opportunity to demo their projects in class. Public demos are completely voluntary. Private demos must be scheduled with the instructor.

Registration Prerequisite:

C/C++ programming experience.

Textbooks and Materials

Required Textbook:

Required: Game Programming In C++: Start To Finish by Erik Yuzwa, ISBN: 1584504323; Recommended: Computer Graphics, Principles and Practice by Foley, van Dam, et al. ISBN: 0-20-184840-6

Students are required to have a PC or Mac with a C/C++ development environment installed.

An Internet connection and the ability to download and install development tools and libraries is also required. Many of the resources used in this class are only available on the Internet.

Communication

Instructor	Bob Pendleton
Office Hours	TBD
Phone	512-218-8279
Email	Bob@Pendleton.com
Web Page	www.gameprogrammer.com

Students are strongly urged join the instructors mailing list at http://www.gameprogrammer.com/mailinglist.html.

Assessment and Grading

Grading Policies

College work must exhibit higher order thinking skills including analysis, synthesis, and evaluation. Mere knowledge about a situation or demonstration of comprehension of the material is not sufficient to prepare you for employment consideration. As a Video Game Development student, you must consistently apply higher order thinking in order demonstrate mastery of the material covered in this course. Grades are given for results not for effort. Read the definitions for each grade noted below, as this is really how grades are determined.

Grading is based on an absolute scale - you are not competing with anyone else, but you will be challenging yourself. There are no distributions of grades; hence, all of you can earn an A in this course. Note: Students earns grades, faculty members do not give them. Your final grade will be based on both individual and learning team performance.
Your final grade will be based on the points that you earn during the course. You may receive “fractions” of points on some assignments. When calculating your final grade, I will use the standard rounding convention – meaning that scores with a fraction of ½ or greater will be rounded up, those with a score of less than ½ will be rounded down. I will use the following grading scale to calculate your letter grade.

The grade will be based on developing a game that meets the following requirements:

- Interactive
- Loads image files
- Loads sound files
- Uses animation
- Plays sounds in response to events.
- Splash Screen
- Game Play Screen
- Game Over Screen

Extra Credit:

- Plays background music
- Multiple levels
- Attract Mode

Grade will be computed as:

final score = ((required features + extra credit) / 8) * 100

Score	**Grade**
90 and above	**A**
80-89	**B**
70-79	**C**
60-69	**D**
59 and below	**F**

Minor adjustment to final grade may be made based on class participation or exceptional qualities of the students project.

Course Policies

Attendance

Students are expected to attend lectures and participate in discussions. Failure to do so is a waste of your money.

Withdrawals

You may withdraw from the class up to the last day allowed by ACC policy.

Incomplete

A grade of incomplete ("I") will be given if the student can show that extenuating circumstances prevented them from attending class or completing the project on time. The student may take as long to make up the incomplete as is allowed by ACC policy. The instructor is willing to meet students at a mutually convenient location to view your game demo.

Studying

This is a hands-on programming class. Students are expected to develop their projects over the entire length of the class. My experience is that students fail this class due to:

1. Procrastination. Putting off the start of development and not leaving enough time to finish the project.
2. Biting off more than they can chew. The project can be made as simple or complex as you chose. If you make it overly complex it will take to long to complete.
3. Underestimating the time needed to create or obtain the graphic and sound resources need by the game. This part of the task takes a deceptively long time.

Expect to have to experiment with different approaches to solving problems and to have to research problems on the Internet.

Classroom Etiquette

The instructor reserves the right to expel rude, abusive, or disruptive students from the class.

Course Schedule

The course schedule is approximate and will vary depending on the needs of the students. At the start of each session there will be time for open discussion and topics questions about any previous topic.

1) Class organization. Tools, toolkits, anatomy of a game program.
2) Discussion of student game concepts.
3) Graphic hardware, video modes and pixels, using what you have, not what you want. Working with image files.
4) Basics of interactive animation. Time, speed, and contact.
5) Working with fonts. More animation, double buffering and dirty rectangles.
6) Programs Objects. Splash Screen, play screens, end game screens.
7) 2D & 3D graphics.
8) More 3D graphics.
9) 3D Math for games.
10) Programs Objects. Splash Screen, play screens, end game screens.
11) Event driven programming == Discrete Event Simulation. Everything is an event. Basics of game GUI implementation.
12) Sound. What it is and how a computer represents it.
13) Working with sound. Loading files. playing sounds, mixing, and latency.
14) Putting it all together examining a complete game.
15) Intro to game AI. Pointers to further reading.
16) Game genre's. Brain storming a game concept to be designed in class.
17) Game design: Part 1
18) Game design: part 2
19) Game design: Part 3
20) Open discussion on student projects. Public demos of student projects for students who chose to participate.

Video Game Development Program Philosophy

The Video Game Development Program has been designed, developed and implemented in partnership with leading video games studio managers and directors in Austin. The video games industry has undergone significant changes in how games are developed. They are rarely developed by few persons working in isolation. Today’s games are often developed by teams of 50 to 200 on schedules from 2 to 3 years with budgets of $10M to $20M. The large publishers drive the game development funding and schedules. Consequently, it is critical that personnel in the industry communicate and collaborate effectively.

This drove the certificate requirements definition. Students are required to successfully complete courses in four categories:

1. The base industry courses: Video Games Industry, Business of Video Games and Video Games Development.
    a. Students will understand what drives the industry, why games are developed, what is needed for success and how to get from idea to delivery.
2. The course specialization courses: Video Game Programming, Video Game Art, Video Game Design and Video Game Production.
    a. Students will understand the requirements, objectives, limitations and goals of the different disciplines in a studio. This is essential for communication and collaboration.
    b. Students in these core courses will be cross-discipline in order to build an understanding and appreciation of how different discipline teams collaborate and contribute to the final product.
3. The five specialization electives.
    a. Students will develop skills in the discipline in which the student will seek employment.
4. Non-specialization electives
    a. These are optional courses that will give you a deeper understanding of what other disciplines do and how they function. They will help you understand how to work with others on the team and to get the 'big picture.' These courses do not count towards the Video Game development certificate.
5. Capstone Project
    a. This multi-person team project will simulate the real video game development environment. Students will develop a concept, turn it into a design, implement the programming and art required and produce it on the committed schedule. Go/no go milestones and final "publisher" acceptance reviews will mimic the industry. The students will have a deliverable for their portfolio that can be used for employment purposes.

Throughout the program each course will focus on knowledge transfer, skill building and teamwork. There will be a heavy emphasis on projects that will broaden and deepen each student's portfolio development. Portfolios are critical to demonstrating an individual's capabilities. Some projects will individual, many will be team based. How much a student gets out of each course will largely be determined by how much the students puts into the course. Video game development is highly complex, difficult work. The courses are designed to prepare students for that environment. So, come expecting to work hard.

The program is designed to reinforce key concepts such as teamwork, collaboration, and cooperation across all disciplines in the games development and management process. Many concepts are repeated throughout the program because they are extremely important to successful game development.