

# GAME 2014

## Multiplayer Game Programming I

### *Course Description*

This course is intended for those who have a strong desire to be a programmer in the games industry. With the multiplayer aspect of games growing increasingly important in the delivery of top titles, every game programmer should understand the basic concepts taught in this course.

### *Course Schedule/Location*

May 24 – July 12      MW6:00-9:00 p.m.      42 Hours      HBC 301.10

### *Course Objective*

Teach basic concepts of multiplayer game development including:

- Understanding the differences between peer-to-peer and client-server architectures.
- Identifying the what and why of DirectPlay vs. sockets
- Creating, transmitting and processing network messages
- Usage of reliable vs. unreliable message delivery
- Handling latency and packet loss
- Representing and maintaining consistent game world state
- Cheat detection and prevention

### *Instructional Methodology*

This is a very hands-on course. Students will be required to add multiplayer capability to an existing game of their own, or to a simple game that is provided. Lecture time will be kept to a minimum to focus the majority of class time on implementing the concepts that are taught. Students will learn primarily by doing, and via interactive discussion of problems and solutions.

### *Registration Prerequisites:*

C/C++ programming experience.

GAME 1010 Video Game Programming or equivalent background highly desirable.

## ***Textbooks/Materials***

There is no required text book. Copies of lecture notes/slides will be provided. Selected articles may also be provided.

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

The Direct X SDK will need to be installed (a CD will be provided).

An Internet connection is highly recommended, as email is the preferred form of out of class communication. See below for contact info.

## ***Communication***

**Instructor**                      Jay Lee  
**Phone**                              512-278-1336  
**Email**                                [gunnerjay@earthlink.net](mailto:gunnerjay@earthlink.net);

## ***Evaluation***

The course will be an intensive hands-on experience. Students will gain most by attending all the meetings if humanly possible, actively participating in the interactive discussions, and of course completing the project. The course is taught in a lab so that students work on their project and interact with the instructor during class to get maximum benefit.

### *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  $\frac{1}{2}$  or greater will be rounded up, those with a score of less than  $\frac{1}{2}$  will be rounded down. I will use the following grading scale to calculate your letter grade.

The grading scale is based on a 100-point (or percentage) scale:

*How points and percentages equate to grades*

|          |              |  |
|----------|--------------|--|
| <b>A</b> | 90 and above | <b>A</b> = Excellent performance. Work is exemplary and worthy of emulation by others. Student is in full attendance and constructively contributes to the learning environment. |
| <b>B</b> | 80-89        | <b>B</b> = Above average performance. All assignments are complete and exhibit a complete understanding and an ability to apply concepts.  |
| <b>C</b> | 70-79        | <b>C</b> = Average performance. Accomplishes only the minimum requirements. Oral and written communication is at an acceptable level for a college student.                      |
| <b>D</b> | 60-69        | <b>D</b> = Demonstrates understanding at the most rudimentary level. Work is minimally passing.  |
| <b>F</b> | < 59         | <b>F</b> = Work is not passing, characterized by incompleteness, lateness, unsatisfactory demonstration of understanding and application.  |

The breakdown for the four areas of the student's final grade is as follows:

|                              |       |
|------------------------------|-------|
| Attendance and Participation | nn%   |
| Assignments                  | nn%   |
| Quizzes/Tests                | nn%   |
| Project Deliverables         | nn%   |
|                              | ----- |
| Total:                       | 100%  |

***Classroom Etiquette***

The instructor reserves the right to expel rude, abusive, or disruptive students from the class. Please provide notice as early as possible should attendance to a meeting not be possible.

***Course Schedule***

The course schedule is approximate and will vary depending on the needs of the students and the progress being made on student projects

| <b>Module</b> | <b>Topic</b>  |
|---------------|---|
| 1             | Intro and Overview. Class Introductions. Overview of multiplayer programming. Discussion and setup for class project. |
| 2             | Crossing the Great Divide. Single player vs. Multiplayer programming. Different architectural approaches.             |

| <b>Module</b> | <b>Topic</b>   |
|---------------|--|
| 3             | Making the connection. Hosting a game. Joining a game. When does the game start? |
| 4             | Packets. Sending, receiving. Processing. Flow Control.                           |
| 5             | Latency. Measuring performance, bandwidth utilization, resends                   |
| 6             | Keeping game world state synchronized  |
| 7             | Reliable vs. unreliable message delivery   |
| 9             | Resource usage. Measuring bandwidth and performance                              |
| 9             | Handling unexpected situations – huge latency, huge packet loss                  |
| 10            | Hacking and cheating. Why should we care? Detection and prevention.              |

### **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.