

Name of Course: GAME 1010 Video Game Programming

Number of Hours: 42

Course Description:

This course serves as an introduction for any student in the Video Games Program to the role of the programmer in the development of a game, what games look like from a programmer's viewpoint, and how programmers translate game designs, artistic specifications and production requirements into code. It will include hands-on programming using C# on the Microsoft XNA Platform (for Windows and Xbox 360) to drive the concepts home. Cross-discipline collaboration, communication and compromise are key concepts in game development.

Course Rationale:

This course is designed to introduce the student to the role of the programmer in the game development team, and to give them an understanding of how engineers interact with game designers, artists, and producers. The class is geared towards non-programmers, and will balance high-level production, architecture, and engineering concepts with hands-on programming and development tasks.

Course Learning Objectives:

Students will write a simple 2D C# XNA game. They will spend the first 2/3 of the course developing the basic game which is identical for all students. The last 1/3 of the course they will customize this game according to a design spec provided either by themselves. This design document is a feature commitment that the Final Project must fulfill to receive an A. Students aren't required to become fully conversant in C#, but they must be comfortable enough with their game engine to customize at least the art and game "rules". By working to a design spec, students will learn how engineers influence design and interact with designers. By changing game art, students will learn how to manage a simple art production pipeline and work within set restrictions.

Prerequisites:

None. However, it is preferred that you have a fair competency with computers, as well as working knowledge in high school level Math and English. An appreciation for video games also helps.

Required Materials:

None. Email account and access to a computer at home is recommended.

Summary:

This course introduces students to the role of programming within the game development industry.

Primary topics include:

- What do games look like from the programmer's perspective?
- What tools do programmer's use?
- What is "code" and how does one write it?
- How does a programmer translate a design into game code?
- How does a programmer import art resources (graphics and sound) into a game?

Secondary topics include:

- What kinds of tasks to programmers undertake in comparison to artists and designers?
- What are different roles a programmer can perform within the field of game development?
- What training does one need to become a game programmer?
- Anything else students request.

Structure:

The class takes place over 14 weeks(11 for Summer semester). 1 week we will dedicate to introducing the course and installing tools. We will have guest speakers for 2 other weeks. We will dedicate the last 3 weeks to work on individual "Final Projects". In the remaining 8 weeks, we will cover the following topics:

- Fundamentals – what is a program; how to we display information; how do we perform repeated tasks (loops); how do we make decisions (flow-of-control).
- Classes – what is a class; what do classes look like in C#; how do we break a game design down into classes?
- Game States – what is a game state; how to do we handle state changes.
- Graphics – basic UI principles and basic graphics theory (what is a pixel); how do we draw graphics in C# XNA.
- Resource management – what are some basic rules of resource management; how do we load resources in C#.
- Game Programming: laying out the map.
- Game Programming: player movement and actions.
- Game Programming: AI Entities.

Philosophy:

The best way to learn is by doing. In order to understand the role of the engineer in game development, we're going to write a simple game from start to finish. We will use C# for XNA. XNA programs can run on Windows platforms and the Xbox 360. We're using C# because it's much like C++ (most common game development language) but easier to learn.

Evaluation:

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 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:

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

There will be no traditional exams in this course. Your grade will be based on the quality of your projects and documents (e.g. in-class programming tasks and project documentation), class participation, and read assignment questions/discuss.

Students receive grades based on performance in 3 areas:

Attendance/Participation and Reading Assignments (33%): This includes arriving on time, providing explanations for missed classes, and staying on-task in labs (i.e., no web browsing, etc). Leaving class early, without instructor consent, will result in an unexcused absence.

Lab scores: Students will complete 8 labs during the quarter. The instructor will grade these based on the following factors (most important to least important): absence of bugs, completion of all features, and code cleanliness.

Final projects: Each student must take the game built in the 8 labs and customize it in some way. Non-trivial code modification and asset customization are the expected minimum.

Attendance and Participation	34%
Lab scores	33%
Final project	33%
Total	100%

Grade Policy and Scale:

Your final grade for the class will be calculated by averaging the points received for each of the three areas and then weighting that area average according to the table above. Once this weighting and averaging has occurred, the following table will be used to determine your final grade for the class. No curve will be applied.

3.5 - 4.0	A
2.5 - 3.4	B
1.5 - 2.4	C
0.5 - 1.4	D
0.0 - 0.4	F

Late Assignment Policy:

For each day that an assignment is late, 10% will be deducted from the grade for that assignment. This represents the loss of a full letter grade per day that the assignment is late. The maximum number of points deducted in this way will be 30%. If your absence is excused, the lab can be graded with no loss of points in the follow class session.

Incomplete:

A student may receive a temporary grade of "I" (Incomplete) at the end of the semester only if **all** the following conditions are satisfied:

1. The student is unable to complete the course during the semester due to circumstances beyond their control.
2. The student must have earned at least half of the grade points needed for a "C" by the end of the semester.
3. The request for the grade must be made in person at the instructor's office and necessary documents completed.
4. To remove an "I", the student must complete the course by two weeks before the end of the following semester. Failure to do so will result in the grade automatically reverting to an "F".

Freedom of Expression Policy:

It is expected that faculty and students will respect the views of others when expressed in classroom discussions. As a course with a high level of discussion, it is imperative that everyone in the class feel comfortable expressing their views.

Academic Integrity:

A student is expected to complete his or her own projects and tests. Students are responsible for observing the policy on academic integrity described in the Current ACC Student Handbook.

"Acts prohibited by the college for which discipline may be administered include scholastic dishonesty, including but not limited to cheating on an exam or quiz, plagiarizing, and unauthorized collaboration with another in preparing outside work. Academic work submitted by students shall be the result of their own thought, research or self-expression. Academic work is defined as, but not limited to tests, quizzes, whether taken electronically or on paper; projects, either individual or group; classroom presentations, and homework."

The penalty assessed for violations will be in accordance with the current ACC Student Handbook policy. See <http://www.austincc.edu/handbook/policies4.htm> for more information.

Attendance and Participation Policy:

The official college policy states that students are expected to attend classes and will be held responsible for all material covered in class. Regular attendance helps ensure satisfactory progress towards completion of the course. Participation in this case means actively participating in the class assignments and discussions. As you can see above, Attendance and Participation will account for 34% of your final grade in the class. A significant portion of class time is devoted to labs. Completing labs outside of class will be difficult, further effecting your grade.

Withdrawal Policy:

It is the student's responsibility to complete a Withdrawal Form in the Admissions Office if they wish to withdraw from this class. The instructor may withdraw students from this class if their absences exceed 10% of the total number of class meetings. The last date to withdraw for this semester is provided in the ACC Academic calendar for the semester in which the student is enrolled. It is not the responsibility of the instructor to withdraw the students from their class even though the instructor has the prerogative to do so under the above listed circumstances.

A grade of "W" will be automatically assigned if the student initiates a withdrawal through the Admissions and Records office, in accordance with the requirements of that office. If the student fails to complete the work and also fails to properly withdraw, a grade of A, B, C, D, or F will be assigned in accordance with the work that was completed.

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.
 - 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.
 - Students will understand the requirements, objectives, limitations and goals of the different disciplines in a studio. This is essential for communication and collaboration.
 - 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.
 - Students will develop skills in the discipline in which the student will seek employment.
4. Non-specialization electives.
 - 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.

- 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.