

| | | | | | | | | | | | |
|--|---|----------------------------|-----|---------------|-----|--------------|-----|--|------------|-----------------------|-------------|
| Objective | This course provides students with an in-depth introduction to technologies and techniques used in the game industry today. At semester's end, students will have: <ol style="list-style-type: none"> 1. Implemented a variety of core game systems (incl. rendering, input, sound, and collision/physics) 2. Developed a strong understanding of essential mathematics for games 3. Written multiple-fully functional games with Unreal 4 and C++, both individually and as a team. 4. Learned the critical thinking required to continue further study in the field | | | | | | | | | | |
| Concepts | 3D mathematics for games. Working with Unreal Engine 4. Intermediate C++. 3D graphics. Collision detection. Events and scripting. Introduction to animation and A.I. Implementing game play. Getting a job in the game industry. | | | | | | | | | | |
| Prerequisites | CSCI 104 or ITP 365x | | | | | | | | | | |
| Instructor | Sanjay Madhav | | | | | | | | | | |
| Contact | Any questions related to the course and material should be posted on Piazza. <i>Email: madhav@usc.edu (Only for non-course questions or prospective students).</i> | | | | | | | | | | |
| Office Hours | Monday – Thursday 2:30 – 4:30PM in OHE 530H | | | | | | | | | | |
| Lab Assistant | Zachary Metcalf (contact via Piazza) | | | | | | | | | | |
| Lecture | Monday 5 – 6:50PM in KAP 267 | | | | | | | | | | |
| Lab | Wednesday 5 – 6:50PM in KAP 267 | | | | | | | | | | |
| Course Structure | <p>Every week, we will cover a specific game programming concept in lecture. For the first few weeks of the semester, students will work on an individual game assignment in Unreal 4.</p> <p>After this individual game assignment is complete, students will break into groups of three and work on a larger game project. Further details regarding this project are provided on the next page. The individual assignments must be completely <i>individually</i>. And of course, the group project will be completed with a group.</p> <p>There is a midterm exam, but no final exam. There will also periodically be “pop” quizzes. These quizzes will generally be 15-20 minutes in length, and can occur during any class session. There are a total of six quizzes, with the lowest quiz grade being dropped.</p> | | | | | | | | | | |
| Textbooks | Required: <i>Game Programming Algorithms and Techniques</i> . Sanjay Madhav. ISBN-10: 0321940156. | | | | | | | | | | |
| Grading | <p>The course is graded with the following weights:</p> <table> <tr> <td>Individual Game Assignment</td> <td>20%</td> </tr> <tr> <td>Group Project</td> <td>45%</td> </tr> <tr> <td>Midterm Exam</td> <td>25%</td> </tr> <tr> <td><u>Quizzes (6 total, lowest grade dropped)</u></td> <td><u>10%</u></td> </tr> <tr> <td>TOTAL POSSIBLE</td> <td>100%</td> </tr> </table> | Individual Game Assignment | 20% | Group Project | 45% | Midterm Exam | 25% | <u>Quizzes (6 total, lowest grade dropped)</u> | <u>10%</u> | TOTAL POSSIBLE | 100% |
| Individual Game Assignment | 20% | | | | | | | | | | |
| Group Project | 45% | | | | | | | | | | |
| Midterm Exam | 25% | | | | | | | | | | |
| <u>Quizzes (6 total, lowest grade dropped)</u> | <u>10%</u> | | | | | | | | | | |
| TOTAL POSSIBLE | 100% | | | | | | | | | | |

Grading Scale Letter grades will be assigned according to the following scale:

| | |
|--------------|----|
| 93%+ | A |
| 90-92% | A- |
| 87-89% | B+ |
| 83-86% | B |
| 80-82% | B- |
| 77-79% | C+ |
| 73-76% | C |
| 70-72% | C- |
| 69 | D+ |
| 67-68 | D |
| 66 | D- |
| 65 and below | F |

Half percentage points will be rounded up to the next whole percentage. So for instance, 89.5% is an A-, but 89.4% is a B+.

There is no curving. Students will receive the grade they earn. Extra credit is generally not offered.

Group Project Starting in Week 6, students will break into groups of 3-4 to work on a larger 3D game of their own design. Before commencing work on their project, students must have their proposed game approved by the instructor. Working on a project which is not approved is not allowed.

The primary learning objectives of this project are two-fold: first, we expect you to apply and expand your knowledge of gameplay programming in Unreal 4. By the end of this project, every team member should be confident and knowledgeable regarding a wide range of aspects of how gameplay code works in Unreal 4.

Second, as this is a fairly lengthy project (~11 weeks in length), the other primary learning objective is to learn the process of how to manage group projects of longer duration. Specifically, we will utilize Agile/Scrum and the online task board Trello. This will give students “real world” game development experience.

Must greater detail on how the project is graded, as well as the weekly schedule is provided in the lab guide available on Blackboard.

Policies *Make-up policy for exams:* To make up for a missed exam, the student must provide a satisfactory reason (as determined by the instructor) along with proper documentation. Make-ups are only allowed under extraordinary circumstances.

Late Assignments: Late individual games will be accepted with a 15% penalty per day late, up to a maximum of two days late (for -30%). An individual game more than two days late will be given a 0.

The final project will *not* be accepted late. It will be graded based on the submission at the required time of presentation.

| | |
|-----------------------------------|--|
| Software | <p>Students will be able to setup their own PC and/or Mac computers to use Unreal 4. This will require purchasing an Unreal 4 license at least for a single month (it is \$20/month, but you are allowed to purchase one month and then cancel once you have all of the source code).</p> <p>ITP also offers Open Lab use for all students enrolled in ITP classes. These open labs are held beginning the second week of classes through the last week of classes. Hours are listed at: http://itp.usc.edu/labs/.</p> |
| Academic Integrity | <p>USC seeks to maintain an optimal learning environment. General principles of academic honesty include the concept of respect for the intellectual property of others, the expectation that individual work will be submitted unless otherwise allowed by an instructor, and the obligations both to protect one's own academic work from misuse by others as well as to avoid using another's work as one's own. All students are expected to understand and abide by these principles. SCampus, the Student Guidebook, (www.usc.edu/scampus or http://scampus.usc.edu) contains the University Student Conduct Code (see University Governance, Section 11.00), while the recommended sanctions are located in Appendix A.</p> |
| Plagiarism | <p>In this class, all homework submissions will be compared with current, previous, and future students' submissions using MOSS, which is a code plagiarism identification program. If your code significantly matches another student's submission, you will be reported to SJACS with the recommended penalty of an F in the course.</p> <p>It is okay to discuss solutions to specific problems with other students, but it is not okay to look through another student's code. It does not matter if this code is online or from a student you know, it is cheating. Do not share your code with anyone else in this or a future section of the course, as allowing someone else to copy your code carries the same penalty as you copying the code yourself.</p> |
| Students with Disabilities | <p>Any student requesting academic accommodations based on a disability is required to register with Disability Services and Programs (DSP) each semester. A letter of verification for approved accommodations can be obtained from DSP. Please be sure the letter is delivered to me (or to TA) as early in the semester as possible. DSP is located in STU 301 and is open 8:30 a.m.–5:00 p.m., Monday through Friday. Website and contact information for DSP: http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html, (213) 740-0776 (Phone), (213) 740-6948 (TDD only), (213) 740-8216 (FAX) ability@usc.edu.</p> |
| Emergency Preparedness | <p>In case of a declared emergency if travel to campus is not feasible, USC executive leadership will announce an electronic way for instructors to teach students in their residence halls or homes using a combination of Blackboard, teleconferencing, and other technologies.</p> <p>Please activate your course in Blackboard with access to the course syllabus. Whether or not you use Blackboard regularly, these preparations will be crucial in an emergency. USC's Blackboard learning management system and support information is available at blackboard.usc.edu.</p> |

Course Outline

Week 1 – Course Intro, Game Programming Basics, C++ Crash Course (8/25 and 8/27)

- Course Introduction
- The Game Loop
- Game Objects: What they are and different representations
- Intermediate C++ Concepts

Reading: *Madhav*: Ch. 1

Homework: Setup Unreal 4; Follow Tutorial Series (not graded, but essential)

Week 2 – Vector Math and Games (9/3)

- Vectors
- Applications
- Why vector math is valuable

Reading: *Madhav*: Ch. 3 (pp. 41-58)

Lab: Begin work on UnrealShmup (Individual)

Week 3 – Rendering, Part 1 (9/8 and 9/10)

- Color systems/formats
- Rendering Pipeline
- Matrices, Coordinate Spaces and Projections
- Painter's Algorithm and Z-Buffer

Reading: *Madhav*: Ch. 3 (pp. 58-end), Ch. 4 (pp. 65-75)

Lab: Continue work on UnrealShmup (Individual)

Week 4 – Quaternions and Input/Sound (9/15 and 9/17)

- Euler Rotations and Quaternions
- Human Input Devices and Event-Based Input Systems
- Sound basics

Reading: *Madhav*: Ch. 4 (pp. 88-91), Ch. 5, Ch. 6

Lab: Continue work on UnrealShmup (Individual)

Week 5 – Artificial Intelligence (9/22 and 9/24)

- AI state machines
- A* and basic path finding
- Navigation Meshes

Reading: *Madhav*: Ch. 9

Lab: Finish work on UnrealShmup (Individual)

UnrealShmup DUE Sunday, 9/28 @ 11:59PM

Week 6 – Collision Detection and Game Physics, Part 1 (9/29 and 10/1)

- Rays and Planes
- Raycasting
- Types of Collision Geometry
- Numeric Integration

Reading: *Madhav*: Ch. 7 (pp. 127-134 and pp. 148-153)

Lab: Group Project: Form groups; Create product backlog; Begin Sprint 1

Week 7 – Midterm Exam on Monday, 10/6

Lab (10/8): Group Project: Continue Sprint 1

Week 8 – Camera Systems (10/13 and 10/15)

- Field of view
- Basic camera styles
- Follow cameras

Reading: *Madhav*: Ch. 8

Lab: Group Project: Continue Sprint 1

Week 9 – Rendering, Part 2 (10/20 and 10/22)

- Lighting, Shading, and Texture Mapping
- Advanced Mapping, Shadows, Particle Effects
- Vertex and Pixel Shaders overview

Reading: *Madhav*: Ch. 4 (pp. 76-88)

Lab: Group Project: Continue Sprint 1

Week 10 – Collision Detection and Game Physics, Part 2 (10/27 and 10/29)

- Sphere/Sphere (swept) and Sphere/Plane Collision
- AABB Collision

Reading: *Madhav*: Ch. 7 (pp. 134-148)

Lab: Group Project: **Sprint 1 Review**; Begin Sprint 2

Week 11 – User Interface Systems (11/3 and 11/5)

- Event-based UI systems
- Implementing a HUD

Reading: *Madhav*: Ch. 10

Lab: Group Project: Continue Sprint 2

Week 12 – Event-Based Systems and Scripting Languages (11/10 and 11/12)

- Closer look at event-based systems
- Scripting Languages
- Case Study: UI Mods in *World of Warcraft*

Reading: *Madhav*: Ch. 11 (skip “Implementing a Scripting Language”)

Lab: Group Project: Continue Sprint 2

Week 13 – Animation and Assorted Gameplay (11/17 and 11/19)

- Evolution of Animation systems (2D to key framed to skinning)
- Skeletons and Poses
- Gameplay Systems for different genres

Reading: None

Lab: Group Project: **“E3” (Sprint 2 Review)**; Begin Sprint 3

Week 14 – Basic Networking (11/24)

- Separating players
- Packet writer and Game services

Reading: *Madhav*: Ch. 12

Lab: No lab due to Thanksgiving

Week 15 – Conclusion (12/1 and 12/3)

- Where to go from here?
- Engines, mobile game, multiplayer games, etc.
- How to get into the game industry

Lab: Group Project: Continue Sprint 3

Finals Week – **Final Group Project Presentations on Wednesday, 12/10 @ 4:30PM**
