CS 381: Game Engine Architecture

Sushil J. Louis
http://www.cse.unr.edu/~sushil/class/381
classroom: MS321
Lab: SEM 321 (Linux)

Catalog Description

CS 381, Game Engine Architecture
Credit(s): 3

Introduction to the technical elements of modern videogame and the pipeline for assembling them, plus issues of interface design, quality assurance, and business practice.

Prereq(s): CS202 or consent of instructor

Course Objectives

This course introduces students to game engine architecture and the game development pipeline by teaching students how to design and implement a 3D game engine based (this semester) on the open source Ogre graphics engine. The course will cover the following topics.

- The Ogre graphics engine
- Game engine architecture
- Adding 3D models to a Scene
- Keyboard interaction
- Game entity management
- Simple game physics
- Simple AI
- Interaction design
- If time permits
  - Sound
  - Multiplayer networking

Office hours

- MTW from 11:00 until noon, and by appointment (sushil@cse.unr.edu)
Course Arrangement

The course has two components: Lecture and Laboratory. Both lecture and lab components are mandatory. Note specifically that the laboratory component is mandatory. All assignments, homework, quizzes, and exams are required.

Textbook

- **Required:** Ogre Documentation and Manual, [http://www.ogre3d.org/documentation](http://www.ogre3d.org/documentation)
- **Required:** Blender Wiki/Manual, [https://www.blender.org/support/](https://www.blender.org/support/)

You will have to use the information above, class notes posted on the class web page, and your own notes to study for tests and to do the assignments.

Required Materials

- ECC computer accounts

Unique Class Procedures/Structures

- One lecture (in LME316) and one lab (in SEM 321) per week
- Lab portion requires coding in lab
- Required in class demonstrations of student assignments
- Peer review of final projects
- No textbook
- Expected independent learning from online tutorials

Important Dates

- **Exam 1:** Tentative: Feb 22, 2017
- **Exam 2:** Tentative: April 10, 2017
- **Final Exam and Project Presentations:** Monday May 15, 2017, 10:15 a.m. - 12:15 p.m. in SEM321. If you cannot make this time, please drop this course.

Assignments

Assignments require writing programs to integrate and “glue” existing and newly created components into a game engine. You will be using the linux boxes in SEM 321. Each programming assignment will be demonstrated in class/lab and graded on the demonstration.

The set of assignments will culminate in a 3D game engine that you will use to build an exemplar 3D computer game.
Course Rules

- Late programming assignments or exercises will not be accepted.
- Exams/Tests/Quizzes (if any) are individual efforts. A severe penalty will be given for collusion or other form of academic dishonesty. The usual penalty for academic dishonesty on assignments or an exam is failure in the course.
- Each assignment will specify whether the assignment is an individual effort or a group assignment. Maximum group size is two (2).
- For individual assignments, **do not show, exchange, or copy code**. Using another person’s code or having another person ”ghost write” a lab will be considered academic dishonesty. For group assignments, if you are in a group **do not show, exchange, or copy code outside your group**.
- The final project is a group project.

Preliminary Information on Grading and Assessment

- Students will be assigned letter grades. Your grade will be one of A, B, C, D, or F. We will use the +/- grading system.
- There will be a number of programming assignments. No late assignments will be accepted.
- There will be one or two midterm exams. Midterms take place in your regular classroom.
- The **Final Exam: is on Monday May 15, 2017 from 10:15 - 12:15 in SEM231B**. If you cannot make this date, please drop the course.
- There will be announced and **unannounced** quizzes.

Tentatively, your final grade will be based on:

<table>
<thead>
<tr>
<th>Item</th>
<th>Final Grade Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm exams (10% each)</td>
<td>20%</td>
</tr>
<tr>
<td>Assignments</td>
<td>50%</td>
</tr>
<tr>
<td>Final Project &amp; Exam</td>
<td>25%</td>
</tr>
<tr>
<td>Quizzes and Exercises</td>
<td>5%</td>
</tr>
</tbody>
</table>

Preliminary Course Outline

This outline is approximately in order.
<table>
<thead>
<tr>
<th>Topic</th>
<th>Textbooks</th>
<th>Lectures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>Syllabus</td>
<td>1</td>
</tr>
<tr>
<td>GFX: Setting up Ogre</td>
<td>Tutorial 1</td>
<td>1</td>
</tr>
<tr>
<td>GFX: Coordinate Systems, Scenes, Entities, lights</td>
<td>Tutorial 1, Tutorial 2, Tutorial 3</td>
<td>4</td>
</tr>
<tr>
<td>GFX/UI: Input/Output Systems</td>
<td>Tutorial 4, Tutorial 5,</td>
<td>1</td>
</tr>
<tr>
<td>GFX/UI: Cameras, mice, keyboards</td>
<td>Tutorial 4, Tutorial 5,</td>
<td>1</td>
</tr>
<tr>
<td>GFX: Ogre architecture</td>
<td>Tutorial 6</td>
<td>1</td>
</tr>
<tr>
<td>Game Engine Architecture</td>
<td>Notes</td>
<td>1</td>
</tr>
<tr>
<td>Entity Management: Creation</td>
<td>Notes</td>
<td>1</td>
</tr>
<tr>
<td>PHX: Entity movement: Physics</td>
<td>Notes</td>
<td>2</td>
</tr>
<tr>
<td>AI: Entity smarts: Unit AI</td>
<td>Notes</td>
<td>1</td>
</tr>
<tr>
<td>AI: Entity Group smarts</td>
<td>Notes</td>
<td>1</td>
</tr>
<tr>
<td>UI: Game Interaction (UI)</td>
<td>Notes</td>
<td>2</td>
</tr>
<tr>
<td>Net: Game Networking</td>
<td>Notes</td>
<td>2</td>
</tr>
<tr>
<td>Brainstorming a game</td>
<td>Notes</td>
<td>1</td>
</tr>
<tr>
<td>Designing and finishing a complete game</td>
<td>None</td>
<td>8+</td>
</tr>
</tbody>
</table>

**Course Learning Outcomes**

- Students demonstrate they can integrate existing components to build a 3D game engine
  - Strategies and Actions:
    * Lectures, assignments, and projects covering the game engine architecture and existing game engine components in particular python ogre.
  - ABET Criteria covered:
    * (1) An ability to apply knowledge of computing, mathematics, science, and engineering.
    * (5) An ability to analyze a problem, and identify, formulate and use the appropriate computing and engineering requirements for obtaining its solution.
    * (11) An ability to use current techniques, skills, and tools necessary for computing and engineering practice
    * (13) An ability to apply design and development principles in the construction of software systems or computer systems of varying complexity
  - Program Objectives covered:
    * (2) Have peer-recognized expertise together with the ability to articulate that expertise as computer science professionals
    * (3) apply good analytic, design, and implementation skills required to formulate and solve computer science and engineering problems
  - Assessment:
    * Programming assignments and projects
• Test/Exams

- Students demonstrate they can learn and use computing tools in the pursuit of a broader project goal.
  
  - Strategies and Actions:
    - Point students to online tutorials for Blender - a graphical modeling tool - and ask them to use the tutorials to modify, texture, export, and incorporate 3D models into their developing game engine.
  
  - ABET Criteria covered:
    - (1) An ability to apply knowledge of computing, mathematics, science, and engineering.
    - (11) An ability to use current techniques, skills, and tools necessary for computing and engineering practice
  
  - Program Objectives covered:
    - (4) Students demonstrate that they can function, communicate, collaborate and continue to learn effectively as ethically and socially responsible computer science professionals
  
  - Assessment:
    - Assignments that deal with texturing, modifying, and importing models into python ogre.

Statement on Academic Dishonesty

Cheating, plagiarism or otherwise obtaining grades under false pretenses constitute academic dishonesty according to the code of this university. Academic dishonesty will not be tolerated and penalties can include filing a final grade of "F"; reducing the student’s final course grade one or two full grade points; awarding a failing mark on the coursework in question; or requiring the student to retake or resubmit the coursework. For more details, see the University of Nevada, Reno General Catalog.”

Statement of Disability Services

Any student with a disability needing academic adjustments or accommodations is requested to speak with the Disability Resource Center as soon as possible to arrange for appropriate accommodations.”

Statement for Academic Success Services

Your student fees cover usage of the Math Center (784-443 or http://www.unr.edu/mathcenter/), Tutoring Center (784-6801 or www.unr.edu/tutoring-center), and University Writing Center (784-6030 or http://www.unr.edu/writing-center). These centers support your classroom learning; it is your responsibility to take advantage of their services. Keep in mind that seeking help outside of class is the sign of a responsible and successful student.
Statement on Audio and Video Recording

Surreptitious or covert video-taping of class or unauthorized audio recording of class is prohibited by law and by Board of Regents policy. This class may be videotaped or audio recorded only with the written permission of the instructor. In order to accommodate students with disabilities, some students may be given permission to record class lectures and discussions. Therefore, students should understand that their comments during class may be recorded.

Communications

If I need to communicate with the class as group I will post a message on our class web page. You are required to check the class web page and your email every day.