CSCI 397 – Spring 2009
Game Programming

Ben Coleman
coleman@cs.moravian.edu
214 PPHAC

Office Hours: Mon, Tues 2:00-4:00
or by appointment
Office Phone: 610-625-7781

Course Description

This course focuses on the mathematics and algorithms necessary to create various types of computer games. Topics include advance programming in Java, the mathematics of game programming, artificial intelligence, event-loop programming, and 2D graphics.

Course Goals

- Implement large programs using advanced Java features.

- Apply patterns of design and testing to improve program development.

- Utilize trigonometry and vector mathematics to solve game-related problems.

- Apply artificial intelligence techniques to create autonomous agents.

- Use a graphics library to render appropriate visual representations of two-dimensional scenes.

Course Texts

The required physical textbook for this course is *Programming Game AI by Example* by Mat Buckland. In addition, we will utilize the advanced Java tutorials at `java.sun.com` and the online version of *Killer Game Programming in Java* by Andrew Davison. Finally, I will distribute a variety of articles in class taken from relevant sources.

You should expect to spend at least an hour before each class session working through the readings. This means reading the text for detail, not just skimming through the material before class.

Responsibilities

Your timely attendance is expected at each class meeting. You are also responsible for the contents of reading assignments, handouts, class activities, and class email.

If you have a disability that may affect your participation in this course, please contact me immediately to discuss academic accommodations.

Graded Materials

- **Homework** – Primarily during the fist half of the semester, homework will be assigned nearly every class session. Some problems will be traditional pencil and paper work and others will be small programming assignments. These assignments will be due the next class session.

- **Programs** – You will be assigned five programming tasks that require you to implement small, game-like applications, and you will be given approximately two weeks to finish each one. You should not plan to complete these programs in a single sitting, but rather complete them in a number of shorter sessions.

- **Project** – Approximately half way through the semester, I will provide formal specifications for the course project that will incorporate the ideas from the homework and programming assignments. This project will require a significant amount of time, and there will be a number of intermediate deadlines during this project.

- **Tests** – Two tests will be given during the semester on Wednesday, March 11 and Friday, April 17.

- **Final** – The final will be cumulative and will be held on Tuesday, May 5 beginning at 8:30 a.m.

Academic Honesty

Except on tests, you are encouraged to discuss the material and work with other students in the course. Specifically, on homework, programs, and the project, you may discuss any portion of the assignment with your fellow students. However, you must write your own code and produce your own write-up of the material. Ultimately you are responsible for the material on the tests.

Grade Determination

- Homework – 25%

- Programs – 20%

- Project – 20%

- Tests – 20%

- Final – 15%

CSCI 397: Game Programming

Spring 2009 Schedule

Date	Reading(s)	Topic(s)
M Jan 19		• What does it take to write a game?
W Jan 21	• Davison Ch1: 1-10, Ch3: 15-31	• Basic Animation
F Jan 23	• Buckland Appendix B	• Java and UML Review
M Jan 26	• Buckland Ch1: 1-17	• Points and Trigonometry
W Jan 28	• Buckland Ch1: 17-28	• Vectors
F Jan 30	• Davison Ch1: 10-17	• Timers
M Feb 2	• Davison Ch1-3	• Mouse and Keyboard Input
W Feb 4		• World and Screen Coordinates
F Feb 6	• Davison Ch4: 1-19	• Sprite Graphics
M Feb 9	• Buckland Ch2: 43-69	• State Machines
W Feb 11		• State and Singelton Patterns
F Feb 13	• Buckland Ch2: 69-83	• Messaging
M Feb 16	• Buckland Ch3: 85-91	• Agent Behavior Model
W Feb 18	• Buckland Ch3: 91-99	• Simple Behaviors
F Feb 20	• Buckland Ch3: 99-112	• Advanced Behaviors
M Feb 23		• More Advance Behaviors
W Feb 25	• Buckland Ch3: 113-124	• Flocking
F Feb 27	• Buckland Ch3: 124-132	• Efficiency Issues
M Mar 1 – F Mar 6		• Spring Break
M Mar 9		• Behavior Wrap-Up
W Mar 11		• Test #1
F Mar 13		• Java Collections and Iterators
M Mar 16	• "Crashing Into the New Year"	• Collision Detection
W Mar 18	• "Pool Hall Lesson"	• Collision Detection Part 2
F Mar 20	• Book Excerpt	• Collision Response
M Mar 23		• Collision Wrap-Up
W Mar 25	• "The Science of Debugging Games"	• Debugging

Date	Reading(s)	Topic(s)
F Mar 27	• "The Magic of Data-Driven Design"	• Data-Driven Design
M Mar 30		• Project Overview
W Apr 1	• Buckland Ch5: 193-209	• Graph Implementation
F Apr 3	• Buckland Ch5: 209-231	• DFS and BFS
M Apr 6	• Buckland Ch5: 231-241	• Dijkstra's Algorithm
W Apr 8	• Buckland Ch5: 241-248	• A* Algorithm
F Apr 10 – M Apr 13		• Easter Break
W Apr 14		• Graph Wrap-Up
F Apr 17		• Test #2
M Apr 20	• Buckland Ch8: 333-342	• Navigation-Graph Generation
W Apr 22	• Buckland Ch8: 342-377	• Path Planning
F Apr 24		• Path Smoothing
M Apr 27	• Davison Ch5: 1-21	• Sound API
W Apr 29	• Davison Ch5: 22-74	• Sound Design
F May 1		• Review