CSCI 330 – Spring 2011
Game Programming

Ben Coleman
coleman@cs.moravian.edu
214 PPHAC

Office Hours: W 2:30-4:00; R 1:30-3:30
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 for games, event-loop programming, and 2D graphics.

Course Goals

Upon completion of this course, a successful student will be able to:

- 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 readings for this course are from two textbooks and a collection of articles:

- *Programming Game AI by Example* by Mat Buckland. This book is available through the bookstore.
- *Killer Game Programming in Java* by Andrew Davison. This book is available online, and relevant chapters will be distributed in class.
- In class I will distribute a variety of other articles.

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.

Graded Material

- **Homework** - The goal of homework problems is for you to practice using the current course content and to explore the topics in more detail. Problems will be assigned nearly every class session and generally will be due the next class.

  Homework problems will be graded on a scale between zero and three:

  - **3**: You completed the problem perfectly or nearly perfectly.
  - **2**: Your solution had non-trivial problems.
  - **1**: You tried the problem, but either didn't get very far or made serious mistakes.
  - **0**: You failed to turn anything in for the problem.

  At the end of the semester, your average homework problem score will translate into an actual letter grade as follows:

≥ 2.5	A
≥ 2	B
≥ 1.5	C
≥ 1	D
< 1	F

  Essentially, this scale means that you must earn threes on at least half of the problems to be in the A range (with the remainder of your scores being twos). Plusses and minuses will be used within each range.

- **Programs** – You will be assigned a number of 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.

  For each of these assignments you will meet with me to demonstrate the functionality of your program and to discuss the quality of your design, code, and testing. I will assign a grade at the end of this meeting.

- **Project** – In late March, 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.

  You will demonstrate your game in class on Wednesday, April 27 or Friday, April 29, and your grade will be determined by the number of working features.

- **Tests** – Two tests will be given during the semester on Friday, March 4 and Monday, April 11. You may only re-schedule a test for college-approved absences or documented illness. In either case, you must contact me *before* the beginning of the test.

- **Final** – The final will be cumulative and will be held on Wednesday, May 4 beginning at 8:30 A.M. Any change to the final exam schedule must be approved by both me and the dean of students.

Grade Determination

- Homework – 25%
- Programs – 20%
- Project – 20%
- Tests – 20%
- Final – 15%

Course Policies

- **Late Policy** – I understand that life sometimes gets in the way of getting work done. Consequently, late assignments will be accepted without penalty in the class after the assignment was due. However, this policy should not be used as a crutch, and if you frequently use it I will deduct from your grade. After the next class session, late work will not be accepted unless there are exceptional circumstances.
- **Extensions** – In a similar vein, I am generous with extensions on work if you approach me *before* the day the assignment is due.
- **Absences** – Your attendance is expected at each class meeting, but I understand that students occasionally get sick, have obligations outside Moravian, and even over sleep. If you do miss class, please send me an email explaining your absence – preferably before the class session. Regardless of your reason for missing class, you are responsible for the contents of reading assignments, handouts, class activities, and class email.
- **Academic Honesty** – Except on tests, you are *encouraged* to discuss the material and work with other students in the course. Specifically, on homework and programming assignments you may discuss any portion of the assignment with your fellow students. This policy does not allow you to copy another student's work verbatim – you must produce your own code or write-up of the material. Work together to learn the concepts, but keep in mind that you are ultimately responsible for the material on the tests.
- **Disabilities** – If you have a disability that may affect your performance in this course, please contact me immediately to discuss academic accommodations.

Schedule

Date	Reading(s)	Topic(s)
M Jan 17		• What does it take to write a game?
W Jan 19	• Davison Ch1: 1-10, Ch3: 15-31	• Basic Animation
F Jan 21	• Buckland Appendix B	• Java and UML Review
M Jan 24	• Buckland Ch1: 1-17	• Points and Trigonometry
W Jan 26	• Buckland Ch1: 17-28	• Vectors
F Jan 28	• Davison Ch1: 10-17	• Timers
M Jan 31	• Davison Ch1-3	• Mouse and Keyboard Input
W Feb 2		• World and Screen Coordinates
F Feb 4	• Davison Ch4: 1-19	• Sprite Graphics
M Feb 7	• Buckland Ch2: 43-69	• State Machines
W Feb 9		• State and Singelton Patterns
F Feb 11	• Buckland Ch2: 69-83	• Messaging
M Feb 14	• Buckland Ch3: 85-91	• Agent Behavior Model
W Feb 16	• Buckland Ch3: 91-99	• Simple Behaviors
F Feb 18	• Buckland Ch3: 99-112	• Advanced Behaviors
M Feb 21		• More Advance Behaviors
W Feb 23	• Buckland Ch3: 113-124	• Flocking
F Feb 25	• Buckland Ch3: 124-132	• Efficiency Issues
M Feb 28		• Behavior Wrap-Up
W Mar 2		• Java Collections and Iterators
F Mar 4		• Test #1
M Mar 7 – F Mar 11		• Spring Break
M Mar 14	• ”Crashing Into the New Year”	• Collision Detection
W Mar 16	• ”Pool Hall Lesson”	• Collision Detection Part 2
F Mar 18	• Book Excerpt	• Collision Response
M Mar 21		• Collision Wrap-Up
W Mar 23	• ”The Science of Debugging Games”	• Debugging
F Mar 25	• ”The Magic of Data-Driven Design”	• Data-Driven Design
M Mar 28		• Project Overview

Date	Reading(s)	Topic(s)
W Mar 30	• Buckland Ch5: 193-209	• Graph Implementation
F Apr 1	• Buckland Ch5: 209-231	• DFS and BFS
M Apr 4	• Buckland Ch5: 231-241	• Dijkstra's Algorithm
W Apr 6	• Buckland Ch5: 241-248	• A* Algorithm
F Apr 8		• Graph Wrap-Up
M Apr 11		• Test #2
W Apr 13	• Buckland Ch8: 333-342	• Navigation-Graph Generation
F Apr 15	• Buckland Ch8: 342-377	• Path Planning
M Apr 18		• Path Smoothing
W Apr 20	• Davison Ch5: 1-74	• Sound
F Apr 22 – M Apr 25		• Easter Break
W Apr 27		• Game Demonstrations
F Apr 29		• Game Demonstrations • Review

The details of this syllabus and schedule are subject to change based on our progress through the material.