CSC481: An Introduction to Artificial Intelligence with AI Game Development

Syllabus – Spring 2010

Time: Section 1 T Th 12:30-1:45, Location: Tyler Hall Rm 106
Webpage: http://homepage.cs.uri.edu/faculty/hamel/courses/2010/spring2010/csc481
Prerequisites: CSC301

Instructor:
Prof. Lutz Hamel
email: hamel@cs.uri.edu
office: Tyler Hall, Rm 251

Course Description

Advanced graphics hardware and novel rendering algorithms have revolutionized the look and feel of interactive computer games. With the ever-increasing computational power of desk- and laptop computers, not to mention game consoles, we are about to witness another revolution in games: the very smart NPC (non-player character). The increase in computing power of today's machines allows game designers to implement sophisticated AI techniques for their synthetic creatures that are beginning to blur the difference in the experience between a multi-player game and a single player game populated with smart synthetic creatures.

In this course we will examine AI techniques which are beginning to be routinely deployed in games including the A* path finding algorithm, rule based reasoning, neural networks, and genetic algorithms. We will also be concerned with knowledge representation and problem formalization.

To make the theory accessible (and fun!) we will be using a mod of the Quake II game engine that allows us to replace the brains of the Q2 monsters with AI routines of our own choosing and design. We will design synthetic creatures that will have special roaming abilities, that will be able learn to pick up loot, and make tactical decisions based on learned behavior.

The goal is that you take two things away from this course:
1. A solid understanding of popular AI techniques that can help solve problems in artificial (and real) worlds.
2. A better understanding of how some of the complex behavior in today's interactive games is generated and managed.

Texts

AI Game Development, by Alex J. Champandard, New Riders Games, 2003.
AI Game Programming Wisdom 4, by Steve Rabin, Charles River Media, 2008.

Software

We will be using public domain software in this course, the software will be provided through the course website. A Windows-XP based computer is required to perform the homework and programming assignments.

Grading

Homework, Quizzes, and Programming Assignments   50%
Presentation(s)   20%
Final Project   30%

Policies

- Check the website (often)! I will try to keep the website as up-to-date as possible.
- Class **attendance, promptness, participation,** and **adequate preparation** for each class are expected. If you are absent, it is your responsibility to find out what you missed (e.g. handouts, announcements, assignments, new material, etc.)
- **Late assignments** will be accepted with a penalty of 5% per day late. No assignments will be accepted if they are more than 7 days late.
- **Make-up quizzes** and **exams** will **not** be given without a valid excuse, such as illness. If you are unable to attend a scheduled examination due to valid reasons, please inform myself, or the department office in Tyler Hall, prior to the exam time. Under such circumstances, you are not to discuss the exam with any other class member until after a make-up exam has been completed.
- All work is to be the result of your own individual efforts unless explicitly stated otherwise. **Plagiarism, unauthorized cooperation or any form of cheating** will be brought to the attention of the Dean for disciplinary action. See the appropriate sections (8.27) of the University Manual.
- **Software piracy** will be dealt with exactly like stealing of university or departmental property. Any abuse of computer or software equipment will subject to disciplinary action.

Tentative Schedule

1. AI, Agents, Embodiment
   - The role of AI in games
2. Quagents
   - The Quake II mod
3. Planning vs. Reacting
   - Searching: the A* algorithm
   - Rule Based Systems
   - Knowledge Representation
   - Quagent map navigation
4. Quake II maps and level editing
5. Logic and Formal Reasoning
   - Axioms and inference
   - Map navigation revisited
6. Learning
   - Sub-symbolic learning
   - Symbolic learning
7. Evolutionary Computation
   - Learning classifier systems
   - Evolutionary defense strategies
8. If time permits -- finite state automata & emergent behavior
9. Student reading presentations from "AI Game Programming Wisdom 4"
10. Final student game project presentations