# Game AI

## Overview

The purpose of this course is for undergraduates in Computer Science and Computational Media to gain a breadth of understanding in the toolbox of AI approaches employed in digital games. This involves learning some basic topics covered in other AI courses, but with a focus on applied knowledge within the context of digital games.

Game AI is distinct from "academic AI" in that the end behavior is the target. Game AI programmers are less concerned with the underlying algorithms and more so with the end result. For example, if having an AI ‘cheat’ provides a more entertaining experience, than cheating will likely be a main component of the design. There are also characteristics of many games that focus Game AI on specific problems, like navigation through a virtual world, tactics, and believable behavior. Academic AI researchers are more concerned with rational behavior, knowledge representations, robust multi-agent communication, etc. However, there are overlaps between the two domains, where the desired behavior requires less cheating and more realistic decision-making. This course will survey topics related to this overlap, with a focus on applying what we review in depth through implementations in digital games

This course also observes the difference between AI as a technical challenge for opposing forces AI in games and the integration of AI as a key aesthetic component of the gaming experience. Lectures and projects will explores both of these views of Game AI.

This syllabus should be considered a living document subject to change throughout the course of the semester. There are multiple places in the class schedule to accommodate student interests in particular subjects.

## Programming Requirement

Students are required to have solid programming skills. Experience with Java or the ability to pick it up as part of the course is required. Development using SVN or other version control methods is highly encouraged, but not required or supported by the class. Students are expected to pick up pre-existing code bases and develop their AI code within that code base as part of the class.

## Required Books

Millington’s *Artificial Intelligence for Games* and Buckland’s *Programming Game AI by Example*.

## Grading Policy

The assignments will be weighted as follows: Project 1 (10%), Project 2 (20%), Project 3 (25%), Project 4 (35%). Projects 1-3 will require designing and implementing AI algorithms in existing game engines. Project 4 will require students to design and build a new game, using AI algorithms as a central aspect of the game. Class participation and good teamwork on team projects will account for 10% of your grade. Late work will not be accepted under any circumstances.

At various points throughout the semester, competitions may be performed pitting assignment solutions against each other. Winning a competition will receive an extra 2 points to the final grade. Second place will receive an extra 1 point to the final grade. Third place will receive an extra 0.5 points to the final grade.

## Regrade Request Policy

If you feel like the grade for a given project was not fair, please submit within 1 week via email of receipt of your grade a word document or PDF containing the following: (a) no more than 1 page of information, (b) a comparison between what you submitted and the grading criteria given for the assignment and the feedback given for your assignment, and (c) what you feel your grade SHOULD be given this comparison. Please keep in mind that a requested regrade will prompt me to revisit your project in much greater detail. Your grade may change for the better or worse depending on what I see, but will be responsive to any reasonable and well-founded requests. Requests submitted a week after your grades have been returned to you will not be accepted.

## Honor Code

Students are expected to follow the GT honor code as described here. Some points to keep in mind: Plagiarizing is defined by Webster’s as to steal and pass off (the ideas or words of another) as one's own: use (another's production) without crediting the source. If caught plagiarizing, you will be dealt with according to the GT Academic Honor Code (<http://www.honor.gatech.edu/plugins/content/index.php?id=9>). Submitting any unattributed work other than your own is a violation of the Academic Honor Code.

I encourage you to discuss the assignments, ask questions about how to program, etc. with the instructor, TA, and other students. But the code you submit must be your own. Unauthorized copying of anyone else's code is a violation of the Academic Honor Code. Unauthorized reuse of code from online is a violation of the Academic Honor Code. You’re taking this class to learn how to think about and create Game AI code on your own.

Learning about algorithms from text or online sources is permissible. Copying code verbatim from online or another student is not permissible.

I heavily encourage students to use assets that encourage fair use (e.g. Creative Commons licensed audio, textures, images, etc.) or to produce their own. Unattributed use of other instantial assets, such as graphics, text, or audio, or use of such assets without reappropriating them in a meaningful way to make a clear unique contribution on the student's part is a violation of the Honor Code. Reuse of outside instantial assets is permissible, in other words, but must be done in a manner that makes it clear that you have made a major and significant contribution to the project in question. Failure to do so will result in a failing grade.

Failure to cite your sources is an Honor Code violation. Unauthorized use of any previous semester course materials, such as tests, quizzes, homework, projects, and any other coursework, is prohibited in this course. Using these materials will be considered a direct violation of academic policy and will be dealt with according to the GT Academic Honor Code. For any questions involving these or any other Academic Honor Code issues, please consult me or visit [http://www.honor.gatech.edu](http://www.honor.gatech.edu/).

## Schedule

Week 1:

* Introduction to AI and Game AI
* Character movement

Week 2:

* Holiday
* A\* search

Week 3:

* A\* search continued
* Flocking
* Character formations

Week 4:

* Scripting
* Finite state machines
* Decision trees and behavior trees
* Rule/production systems

Week 5:

* Decision making with planning
* Hierarchical task networks
* Markov Decision Processes

Week 6:

* Procedural content generation: introduction
* Optimization techniques

Week 7:

* Optimization techniques, continued.
* Weapon and entity generation

Week 8:

* Level generation
* Terrain and cityscape generation

Week 9:

* Mission/quest generation

Week 10:

* Machine learning in games: decision tree learning
* Machine learning in games: neural networks

Week 11:

* Spring break

Week 12:

* Machine learning in games: reinforcement learning

Week 13:

* Believable character agents

Week 14:

* Interactive narrative

Week 15:

* Camera control

Week 16:

* Final project presentations

## Projects

**Project 1:**

Knowing the best place for an agent to move at any given time is a key element to almost any game AI. First person shooters, sports games, real-time strategy games, turn-based strategy games, etc. all share this common problem of needing to evaluate the current situation and decide where in space to move next and what action to take. This individual assignment focuses on movement in a semi-continuous 2D environment: Ms. Pac-Man.

This project will be using the IEEE Ms. Pac-Man vs. Ghost Team infrastructure for implementation. The description and distribution of the software can be found here: http://www.pacman-vs-ghosts.net/

This is an individual student project. You are to implement the 4 ghosts in a way that replicates the behavior of the original game. See http://gameinternals.com/post/2072558330/understanding-pac-man-ghost-behavior for a description of the original ghost behavior AI. (Note that when you read about the ghosts you will see that there was an error in the code resulting in ghosts that occasionally path planned to the wrong location. You may choose to replicate the bug as a feature, or correct the bug in your own implementation. In your write up, please indicate whether you are implementing the bug/feature or not.)

You will be given 2 weeks for this assignment with weekly-suggested milestones:

W1: Download and run game. Experiment with different example ghosts and understand how they work. Implement at least the Red ghost.

W2: Implement the remaining 3 ghosts.

**Project 2:**

Game AI can involve making real-time decisions about where is safe to move, how combinations of actions can lead to superior results, and how the behaviors of other agents may affect the environment – all within a dynamic and continuous environment. This individual assignment focuses on decision-making and movement within a continuous 2D environment: Super Mario Bros.

This project will be using the IEEE Super Mario Bros. Competition infrastructure for the Gameplay Track. The description and distribution of the software can be found here: http://www.marioai.org/

For this project, you will modify an existing A\* agent (http://www.doc.ic.ac.uk/~rb1006/projects:marioai by Robin Baumgarten, winner of the 2009 competition). The existing agent has a goal to reach the end of the level. For your project, modify the source code so that the agent collects the most coins. You may find the following interview transcript interesting: http://aigamedev.com/open/interviews/mario-ai/.

This is an individual student project. You are responsible for obtaining the code and using the resources provided with the competition. Note that there is an active development community here that is a valuable resource: http://groups.google.com/group/mariocompetition/topics.

You will be given 4 weeks for this assignment with weekly-suggested milestones:

W1: Download kit and run the A\* agent. Figure out how the agent works.

W2: Design and implement your changes to the A\* agent for coin collecting.

W3: Implement your design for personality.

W4: Tweak your agent to be competitive and work on your writeup.

**Project 3**

Procedural Content Generation is the use of algorithm (procedures) to create novel, and sometimes customized, game content from scratch. Examples of PCG include generation of levels, maps, tree, cityscapes, weapons, monsters, and quests. PCG is often used as a design-time tool to roughly sketch out level content to be refined by human designers. PCG can also be done at run-time to incorporate individual player differences such as skills or preferences. In this project, we look at run-time PCG to create Mario Bros. game levels customized to individual players’ play styles. This includes (a) learning a model of the player’s play style, and (b) using the model to create a custom level. Fortunately, the first part is already done for you. You must focus on designing and implementing algorithms that use the player information to create something that will evaluate well.

This project will be using the IEEE Super Mario Bros. competition infrastructure for the Level Generation Track. The description and distribution of the software can be found here: http://www.marioai.org/.

The rules for level-creating algorithms will follow the rules of the competition.

This is a team project. You are responsible for obtaining the code and using the resources provided with the competition. Note that there is an active development community here that is a valuable resource: http://groups.google.com/group/mariocompetition/topics

You will be given 4 weeks for this assignment with weekly-suggested milestones:

W1: Get the game running, test the RandomLevel, CustomizedLevel, and MyLevel. Make some simple changes to MyLevel.

W2: Design your mapping from player model (read in from "player.txt" to a GamePlay object) and start implementing your design.

W3-4: Implement, and tweak.

**Project 4**

This project is design to allow students to delve deeper into Game AI development by designing a game that has AI as a core aesthetic component (e.g. Nero, Spy Party, Black and White, etc.). Being creative about how to use AI in games so that it is not simply used to provide intelligent opponents but becomes an intrinsic part of the game mechanics is a difficult task. For example, designers like Ken Stanley and Peter Molyneux have been leaders in the field of taking risks and incorporating real time machine learning into gameplay. Even more importantly, the ML techniques used are core to the experience their games provide. Your task is to design a game that incorporates AI as a key element in the gaming experience that your game provides. This is an opportunity to take risks, to try new things, and approach game design in a fundamentally different way. This will be a joint effort executed by you and the teammates assigned to you in class. Implementation can be done in any platform in any language as long as a) the source code can be turned in to us and b) we can access the hardware needed to run it. I encourage all groups to shoot for projects that focus on interesting user interactions and experiences that are created by the AI and game interactions rather than on flashy, 3D visuals, 5.1 surround sound, mega-tastic networked FPS, etc. etc. games. Scoping your work is heavily important here.

Flash and Processing have been common environments in the past. I will also make available an XNA game engine that is particuarly tuned to Role Playing Games (e.g., Zelda, Final Fantasy).

This is a team project. Final projects will be graded along the final criteria:

* Creativity: How interesting, innovative and, most importantly, well thought out is your design? Does it make sense and is it consistent?
* Incorporation: What is the relationship between AI and the user experience? Is there a deep connection?
* Appropriateness: Does the choice of algorithm match your design?
* Technical execution: How well does the final game work? The AI? Are there any major bugs?
* Documentation: Thorough documentation of both the evolution of your design as well as technical decisions.

You will be given 5 weeks for this assignment.