PACMAN VS. GHOSTS

ASSIGNMENT 1 ~ CS 4150/5150 ~ FALL 2013

This is an individual assignment. You may use late days on it if you wish.

Deadline: **October 1, 11:59pm**

DESCRIPTION

For this assignment, you will be creating AI to control the ghosts in Pacman. There are two main phases to this assignment:

1. Creating an AI controller for the ghosts, and
2. Playtesting the AI controller with your friends.

Note that there are additional requirements for graduate students than for undergraduate students. Undergraduate students may attempt to complete the graduate student requirements as well for extra credit. No more than 20% extra credit will be given.

You will be turning in:

1. Source code for your assignment.
2. A short writeup that has a paragraph or two about how your controller works and a playtesting report. Please turn this in as a PDF.
3. A readme text file containing: your name, the names of any people who helped you with the assignment, references to any external sources you used, a list of all the files you created for this assignment and their location, any special instructions necessary for getting your code to work, and how many late days (if any) you wish to apply to the assignment.

PHASE 1: CREATING GHOST AI

We will be using the Pacman vs. Ghosts framework for this assignment, which has historically been used in game AI research competitions:

http://www.pacman-vs-ghosts.net/software

This framework provides a simple interface for you to write a Java agent to control the ghosts, and also comes with several example agents for both Pacman and the ghosts (including a pass-through controller that permits a human to control Pacman). It also provides some important pathfinding functions that you are permitted to use in the assignment. Note that an important part of this class is becoming comfortable with using other engines and frameworks; it is your responsibility to read the documentation and example code and learn how

this framework works. Feel free to ask for help with this on Piazza. **Note: You are not required to register with the competition website for this assignment. As far as I know, the competition is no longer active.**

It is not necessary to alter the framework in any way, other than changing a line in Executor.java that determines which controllers will be used in the game. Please place all of your code for this assignment in the *pacman.entries.ghosts* package.

MIMICKING TRADITIONAL GHOST AI

Everyone is required to implement a controller that mimics the original ghost behavior for Pacman. You can find information about how the original "AI" worked in several places online; one of these sources is here:

http://gameinternals.com/post/2072558330/understanding-pac-man-ghost-behavior

In your assignment writeup, please credit any sources you use for researching how the original pacman AI worked. Note that some sources use different names for the ghosts than the framework uses.

Because of the way the framework is set up, it may not be possible to perfectly replicate the original AI design. Try to come as close to the original as possible.

It is recommended that you begin by implementing a simple AI that has the ghosts move entirely randomly, and then build up your controller from there. Partial credit can only be given if your code is readable, compiles, and executes, so make sure that you always have something that works that you can turn in!

GRADUATE STUDENTS: DESIGNING YOUR OWN AI

In addition to re-implementing the original Pacman ghost AI, graduate students are also required to design and implement their own AI for the ghosts using one of the techniques we have discussed in class (e.g. decision trees, finite state machines, rule-based systems, behavior trees). Your goal in creating this AI should be making it fun to play against, not impossible to beat. Be creative with your approach! You may wish to investigate options such as giving the ghosts more of a personality, having them patrol certain territories, or having them collaborate with each other.

Undergraduate students may choose to also design their own AI for the ghosts; extra credit will be awarded to undergrads who take this option (see the Evaluation section later in this assignment for more details).

PHASE 2: PLAYTESTING

Both undergraduate and graduate students are required to playtest the game with the controllers. This will be a comparative playtest. Undergraduate students should compare their controller to the built-in random ghost AI controller. Graduate students should compare their custom controller with their controller that mimics traditional ghost AI. This will be a small playtest, to give you a sense of how they are run in the industry. Note that the results you get from a very small population playing your game are not enough to come to a scientifically valid conclusion, but this usually isn't the point of the test—you want to get a sense for what people like and don't like about your game, not find a statistically significant result.

Find five people who are willing to play your game. It is preferable for them to not be in this class (and thus not know anything about what you are testing); however, if you can't find people outside class to play, asking your classmates is fine. Ask each of your participants to play against both versions of your AI. **Do not tell**

them which AI is which before they play, and do not give them any information about how the AIs work. Have a notebook with you while you watch them play.

For each play session, take notes while you observe your participants playing the game. Do they seem to be struggling? Do they seem frustrated? Is there behavior you see in the ghosts that seems unusual, broken, too easy, or too difficult? Make sure you take note of which AI the player is playing against when you see these behaviors. After the player has played against both AIs, ask them what they think about the two AIs. Did one seem easier to play against? Was one more fun to play against? Were there any moments during their play that were especially memorable? Write down what you ask them and how they answer.

Graduate students: you may wish to run some of your playtests while you are developing your custom controller, to help inform your design decisions.

When you have finished your playtest, write up your notes into a short report. Make sure to mention not only what your participants reported, but also *why* you think they responded to each AI the way they did. If you altered your controller as a result of the playtest, report what changes you made, and which participant played which version.

ASSIGNMENT WRITEUP

As noted above, you must create a short writeup to accompany your code. This writeup **must** include:

- A paragraph for each of your controllers that explains how it works; in the case of the controller that mimics the original AI, note any places that you deviated from the original specification and state why you needed to do so.
- Your playtesting report.

SUGGESTED TIMELINE

You have three weeks to complete this assignment. Please start working on this early! If you wait until the last minute to start making your controller, you will not have enough time to complete the playtesting portion of the assignment. The following is a *suggested* timeline for you to follow:

Week 1: Familiarize yourself with the Pacman vs. Ghosts framework, examine and run several of the built-in controllers, start working on your own controller. Begin with the simplest behavior and work up from there. Graduate students should aim to have their first controller mostly finished by the end of week 1.

Week 2: Continue work on your controller. Graduate students should start planning and implementing their custom controller in this week.

Week 3: Finish your controller(s). Run your playtests and write your playtesting report.

EVALUATION

This assignment will be graded according to the following rubric. For undergraduate students, the mimic controller is worth 60% of the grade, the playtest report is worth 30%, and code style is worth 10%. For graduate students, the mimic controller is worth 30%, the new controller is worth 30%, the playtest report is worth 30%, and code style is worth 10%. **In order to earn partial credit, your code must compile. Code that does not compile will earn a failing grade on this assignment.**

Undergraduates who implement a new controller will receive extra credit, up to a maximum score of 120%.

Any evidence of copying or cheating on this assignment will result in a grade of zero and a report being filed with OSCCR.

	Excellent (9-10)	Good (6-8)	Not Good (3-5)	Poor (0-2)
Mimic Controller	All ghosts perform as expected (within the limitations of the framework)	Most ghosts perform as expected, but there are some bugs	Ghost AI is not implemented to target pacman; ghosts do not work as expected but there is evidence of an attempt	Little to no evidence of effort to make the ghosts work
New Controller (grad only)	Well-crafted AI that produces an engaging experience OR highly ambitious and functioning AI that does not produce an engaging experience but has an accompanying explanation in the playtest report on why it failed	Produces a reasonable play experience; AI is well-developed but might have some bugs in implementation	Many problems with AI implementation, though it's clear an attempt was made to do something interesting beyond the original AI	Little to no evidence of effort in designing a new AI system
Playtest Report	Well-written report that provides both the data from the playtests and analysis/opinion about why the design of the system led to results	Well-written report that recounts data from the playtests, but does not provide adequate reflection; may have some grammatical errors	Playtest was not performed as required (e.g. too few participants, or does not compare AI systems); does not report on what was required in the playtest sessions	Poorly written report (e.g. significant grammatical and spelling problems to the point that it is hard to read); evidence that playtesting did not occur or was carried out entirely incorrectly
Code Style	Code is well-commented; code is formatted clearly and legible (e.g. appropriate variable names); good code re-use (if appropriate); controller classes and any supporting code are placed in the required package	Some deficiencies in style, but overall code is still legible (i.e. does not mean all of the requirements for "excellent" but does meet many of them)	Very few comments; poorly chosen variable names, lack of code re-use	Completely illegible code; lack of comments; very poor coding style

ADDITIONAL RESOURCES

1. The Pacman-vs-Ghosts framework comes with tutorials, a javadoc for many of the important features of the framework, and reference agents; study these carefully.
2. There are several papers associated with the competition held at the Computational Intelligence and Games conference in 2012 (CIG 2012). You are welcome to read these papers to gain inspiration for your agent; however, be aware that some of these controllers are highly advanced and re-implementing them would be quite ambitious. Papers can be found here: http://geneura.ugr.es/cig2012/proceedings.html

SUBMISSION INSTRUCTIONS

Turn in a .zip file on Blackboard containing:

- All source code, including the original framework code. Students who are creating two controllers should have two separate classes, which should be named to make it obvious which is the mimicking controller and which is the one of your own design. If you create any supporting classes, make sure they are in the same package as your controller.
- Your readme file.
- Your project writeup, including the playtest report.

Assignments must be turned in via Blackboard. **Emailed assignments will not be accepted.**