PROCEDURAL LEVEL GENERATION

ASSIGNMENT 3 ~ CS 4150/5150 ~ FALL 2013

Undergraduate students may work in pairs for this assignment.

Deadline: **November 5, 11:59pm**

DESCRIPTION

For this assignment, you will be creating a level generator for a Mario-like game. This is another branch of The Mario AI Competition, which has been held at the IEEE Computational Intelligence in Games conference each year for the last four years. You will be using the level generation framework from the 2010 version of this competition.

You will be turning in:

1. Full source code.
2. An assignment writeup, including playtesting report, in PDF format.
3. Several screenshots characterizing the levels your generator can create.
4. A readme text file containing: your name(s), 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, complete instructions necessary for getting your code to work, and how many late days (if any) you wish to apply to the assignment.

For undergraduates: only one person is required to submit the assignment.

GENERATING YOUR LEVEL

Your goal with this assignment is to create a level generator for Mario-style levels. Aim to create engaging and interesting levels; they do **not** necessarily need to look like levels that would be made by a human.

You may use any of the approaches we have discussed in class for content generation (e.g. grammars, evolution, constraint satisfaction), or you may choose your own. You must justify your choice in your writeup (it is acceptable for your choice to be based on the one you're most interested in learning more about, just make it clear why you're making the decisions you're making).

Optional: the framework creates a file containing data from a player's previous play-through. You may choose to use this data to guide how your generator works, in an effort to create a level that is personalized for a particular player.

There is no single correct answer to this assignment. Be creative, but be careful to take small steps towards your goal. If you are too ambitious at first you can easily get lost. **You are encouraged to share your high level strategies with the rest of the class, but do not share code.**

If possible, please avoid modifying the framework itself. Your generator code can live entirely in its own class, in dk/itu/mario/level/generator. You should only need to modify CustomizedLevelGenerator.java, as well as optionally creating new classes as helpers.

ASSIGNMENT WRITEUP & PLAYTESTING REPORT

You should write up a brief description (a couple of pages is sufficient) of how your generator works, why you chose the technique you did, and what you learned from it. You should also conduct playtests with **five** individuals throughout your development process. **At least two playtests must occur before your level generator is complete.**

In your playtesting report, describe what configuration of the generator you gave the players, their reactions, and how their reactions influenced your future decisions in refining the generator. Report directly on the feedback they give you, but also reflect upon it by integrating it into your overall writeup.

SUGGESTED TIMELINE

You have approximately two weeks to complete this assignment. Make sure you familiarize yourself with the framework early. As with assignment two, cooperate with your classmates and feel free to share your discoveries on how the framework works with the class via Piazza.

Week 1: Familiarize yourself with the framework. Begin writing a simple level generator.

Week 2: Run initial playtests. Refine the generator, and write up your report.

EVALUATION

This assignment will be graded according to the following rubric. **In order to earn partial credit, your code must compile. Code that does not compile will earn a failing grade on this assignment.**

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)
Level Generator (60%)	A well-crafted level generator that creates levels that are guaranteed to be playable.	A level generator that still creates playable levels, but might not be quite so sophisticated or bug-free as an "excellent" generator.	Generator is trying to do something interesting in creating levels, but has significant problems (e.g. unplayable levels).	Little to no evidence of effort going into creating the generator (e.g. completely unplayable levels, or just a flat line with no level features).
Code Style (10%)	Code is well-commented; code is formatted clearly and is legible (e.g. appropriate variable names); good code re-use (if appropriate).	Some deficiencies in style, but overall code is still legible (i.e. does not meet 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.
Writeup (30%)	Well-written, justifies the techniques used and the goals. Shows strong evidence of reflection upon the AI implementation. Integrates results from the playtesting sessions into the report.	Minor flaws in writing (e.g. spelling and grammatical errors), but still effectively argues for and justifies the techniques used. Shows evidence of reflection upon the AI implementation. An incomplete playtesting report, but playtesting was still performed.	Flawed argument that does not clearly or correctly state why a generation technique was chosen, but does still meet the requirements of the writeup (i.e. describing the technique, describing strengths and weaknesses to approach). Incomplete playtesting report.	No writeup, or so poorly written as to be unintelligible. No playtesting conducted.

RESOURCES

1. The generator framework is available here: http://www.ccs.neu.edu/course/cs5150f13/MarioLevelComp.zip
2. This paper describes the entrants to the 2010 competition: http://sokath.com/main/files/shaker-mariocompetition10-tciaig.pdf
3. The Mario AI competition has resulted in several academic papers. Here is a link that points to several of them; however, there are more out there if you are interested in finding them. Most have been published at IEEE CIG in the last three years: http://www.marioai.org/RelatedPapers
4. There is a Google Group for the Mario AI competition here: https://groups.google.com/forum/#!forum/mariocompetition. However, bear in mind that you are using an older version of the framework, so some of the details in this group may no longer be valid.

SUBMISSION INSTRUCTIONS

Turn in a .zip file on Blackboard containing:

- All of your source code, including the original framework code.
- Your readme file.
- Your project writeup and playtesting report.

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