Scripting in Super Mario

The goal of Project 3 is to learn how to use Behavior Trees, which are very commonly used in computer games for defining the behavior of characters and groups of characters.

In this project you will work with the a Java version of Super Mario that you can download from here.

This version is played on randomly generated maps, so you have a practically infinite number of different levels to play or test your AI. Follow the instructions from the link above to configure the code in your favorite Java IDE. If you run the "ch.idsia.scenarios.Main" class directly, you can play the game by hand by controlling Mario using the keyboard (arrow keys + 'a' + 's'), if the screen is too small, press 'z' to zoom in.

For the purposes of this class, we are not interested in finding the optimal behavior, but with authoring an acceptable behavior for Mario. As you will see, the game comes with several controllers already defined (like the "ForwardJumpingAgent.java" one that simply moves to the right and jumps constantly (with which you can surprisingly complete some easy levels!). You can use them as an example of how to interface your controller with the game.

Try to create, using Behavior Trees, a controller for Mario that tries to collect as many coins as possible, and kill as many enemies as possible. Do not code a fast forward controller that simply advances and jumps. In case enemies are too hard to kill (e.g. flying enemies, or enemies that are in a hard to reach place), it is ok if you controller skips them (same for coins).

As you will discover, Behavior Trees are not the best idea for this kind of game. But precisely for that reason I've chosen it. The goal of this project is for you to explore the possibilities of BTs and their limits. Discover what is easy and waht is hard to do with BTs.

You are free to use any Behavior Tree library you can find online. But I strongly recommend implementing everything from scratch, since that will be much faster and easier for you. BTs are VERY easy to implement, and in the time you take to locate and learn a library, you can already have your own version implemented

Specific Project Tasks:

- Download the Super Mario source code from here
- Set the Super Mario in your favourite Java IDE (Netbeans, Eclipse, etc.)
- To test that you have it properly setup, you can try to run the file ch.idsia.scenarios.Main. If you see Super Mario launching and you can control it with the arrow keys, you are good.
- Read the documentation at the bottom of this page (https://code.google.com/archive/p/marioai/wikis/GettingStarted.wiki)to familiarize yourself with the code. But maybe the easiest way is just to look at example controllers (that can be found in the package "controllers").

In this project you will just have to create a new class that extends the "BasicMarioAIAgent" class.

- Implement a controller that uses Behavior Trees (as explained in class) for controlling Mario to collect as many coins, and to kill as many enemies as possible.
- Create a short 3 minute video demoing your project, and send it to the instructor (do not send the video file! please just send a link (e.g., DropBox link, Youtube link, etc.). You will have to present this video in class.
- Turn in the source-code of your project, plus a 1-2 page description of what you did before midnight the due date. Submissions will be done via Blackboard

Notes:

- The goal of the project is to learn how to use Behavior Trees, not to create the best Mario AI. If you create a very good Mario AI that doesn't use Behavior Trees, I will be very impressed but it will NOT count towards your grade.
- In the description of your project, please tell me where in the project I can find the code you added.