Inteligência Artificial para Jogos

2016-2017

Laboratory Guide

Lab 7 – MCTS Optimizations

In this Laboratory, you will continue working on the scenario from the previous lab, performing optimizations in the MCTS algorithm. However you will also have to deal with other characters taking decisions in the game.

In the new version of the game, enemies will attack the main character when he approaches them. Enemies are also stronger. The new dragon (a fire dragon) is immune to fireballs, skeletons deal 5 damage, orcs deal 10 damage, and the dragon will deal 20 damage. On the positive side, the character has now 200 seconds to achieve its objective of getting all coins without dying. There is also a new LevelUp action that increases the level of the character when enough XP has been reached. Leveling up will increase the maximum HP by 10, and will fully heal the character. Level 2 requires 10 XP, while reaching level 3 requires a total value of 30 XP. Skeletons give 5 XP each, Orcs 10 XP and the dragon gives 20 XP.

To deal with the enemies' initiative to attack in MCTS simulations, when simulating future states of the world, a check will be made to detect if the main character is too close to an enemy. When this happens, the next player to play in that state will be player 1 and the only action considered in that state is forcing the main character to perform a sword attack. If this does not occur, then the next player in that state will be player 0 (the main character) and the normal actions will be considered. This mechanism is already implemented, but you must be sure to always call the CalculateNextPlayer method after a new state is created and action effects are applied. See the example below.

```
WorldModel newState = parent.State.GenerateChildWorldModel();
action.ApplyActionEffects(newState);
newState.CalculateNextPlayer();
```

1) Explore the the source code

a) Integrate the source code of the previous labs in the new Unity project..

b) Start by analyzing the new classes and changes to older classes. A new LevelUp action was added to the DecisionMakingActions. There were changes in the Current/FutureStateWorldModel to deal with actions of other players (as described above). There were also changes to the GameManager and to the actions to deal with the new rules of the game. Additionally, the autonomous character only reinitializes the Decision Making processes after 10 seconds (you can adjust this value) or if a new action is executed. Finally, there is a new method in the Reward class that will help you calculate the reward for a particular node.

2) Change implementation of Fireball

a) Change the implementation of the Fireball action (inside DecisionMakingActions folder) so that the fireball does not work for the dragon. You will spend the mana, but the dragon will not die.

3) Implement MCTS Biased Playout

a) Implement an optimization to the playout process that consists in biasing the random selection of actions. There are several options here. You can use heuristic information about the actions themselves or extract heuristic information from features of the child states to determine the best actions. Decide what approach and what heuristics will be used. Implement this in the MCTSBiasedPlayout class.

b) Try out the resulting behavior. You will quickly realize that you need to adjust some parameters. What are those?

4) Implement MCTS-RAVE

a) Implement an optimization to the selection process that consists in combining AMAF/RAVE information with traditional MCTS information to select the best child. Implement this in the MCTSRAVE class.

b) Try out the resulting behavior and try to detect any differences regarding the quality of the behavior. Experiment different values for the parameters that influence the calculation of beta.

5) Combine MCTS-RAVE with Biased Playout

a) Implement a new version of the algorithm that combines the two techniques described above.

b) Try out the resulting behavior. You should also experiment again new values for the parameters that influence calculation of beta (used by MCTS-RAVE).