TÉCNICO LISBOA

Inteligência Artificial para Jogos

2016-2017

Laboratory Guide

Lab 6 – Vanilla Monte Carlo Tree Search

In this Laboratory, we will continue working on the scenario from the previous lab, but this time we will explore the use of vanilla Monte Carlo Tree Search in order to consider multiple sequences of actions in order to select the best action to perform.

The main goal of the scenario is now to get the coins of all chests spread around the map. The character has a maximum time of 100 seconds to get them all. If at any time the character's hp falls at 0 or bellow, or if the time is exceeded the character will lose.

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 classes inside the DecisionMaking/MCTS Folder. A couple of changes were also made in the CurrentStateWorldModel and a FutureStateWorldModel was created. This new class specifies the termination conditions of a future game state.

2) Implement MCTS

a) Implement the missing methods in the MCTS class. For now do not worry about different players. We will assume that it is always the character turn to play. So ignore the player properties inside the node and reward.

b) In adition to the standard algorithm, you will need to implement a mechanism that limits the number of iterations performed per frame. Moreover, you will need to implement additional information that can be used for debugging, such as the TotalProcessingTime, the best action sequence and other properties.

3) Examine resulting behavior and experiment different parametrizations

a) Try out the resulting behavior and examine if the resulting behavior is according to expected.

b) Experiment different values for the total number of iterations, and see what are the effects. What happens when the value is lower? Can you explain it?

c) Try out different values for the time termination condition. For instance, try out a value of 200 seconds, and a value of 30 seconds. What is the effect?

4) Use GatewayHeuristic to estimate action duration

a) One of the problems with the algorithm is that it is using the Euclidean distance to determine the duration of movement actions, and thus is not properly evaluating some actions. To mitigate this problem, reimplement this estimation using the Gateway Heuristic. This is performed in the WalkToTargetAndExecuteAction. To estimate duration divide the estimated distance by the character's maximum velocity (instead of using a fixed constant value). Examine the impact of this change in the character's behavior.