Enhancements for Monte Carlo Tree Search in The Mario AI Framework

Emil Juul Jacobsen
ejuu@itu.dk

Rasmus Greve
ragr@itu.dk

Supervisor:
Julian Togelius
juto@itu.dk

May 22, 2013

IT University of Copenhagen

Abstract

(Bare copy/paste-ish fra projektbasen, skal skrives om!) In this experiment we explore different implementations and enhancements of the Monte Carlo Tree Search algorithm for an AI, in order to evaluate their performance and results in the Super Mario AI Benchmark tool. We have implemented the basic MCTS algorithm in the Mario AI Framework and characterised the performance and identification of the strengths and weaknesses of the algorithm relative to the framework. We have identified a set of refinements and alterations of the algorithm and through implementation and evaluation of these individually we came up with compositions that greatly increase the performance of the AI.

1 Introduction

In this experiment we explore different implementations and enhancements of the Monte Carlo Tree Search algorithm for an AI, in order to evaluate their performance and results in the Super Mario AI Benchmark tool.

2 Background

- Om MCTS [1]
- Om UCB og UCT [1] måske også [4]
- (kort!) Om The Mario AI Framework [5]

3 Approach and Improvements

3.1 Monte Carlo Tree Search with UCT

[1]

3.2 Softmax Backup

3.3 High domain knowledge

3.3.1 Limited actions

3.3.2 Hole detection

3.4 Macro actions

[3]

3.5 Heuristic Partial Tree Expansion Policy

3.6 Checkpoints

3.7 (Combination)

4 Results

Her er noget tekst

Her er noget mere tekst, som det ses på billede 1

5 Conclusion

6 Some Sums DEMO

Here are a few sums[1] I know.

[1] Additions of sets of numbers

Figure 1: Mario being followed

$$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2} \tag{1}$$

$$1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6} \tag{2}$$

$$1^3 + 2^3 + 3^3 + \cdots + n^3 = \frac{n^2(n+1)^2}{4} \tag{3}$$

I can find the sum of the first 10 squares easily with formula (2) above.

7 A Cool Relationship DEMO

Take a look at formulas (1) and (3) on page 3 of section 6. Notice that the right side of (3) is the square of the right side of (1).

References

[1] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," Computational Intelligence and AI in Games, IEEE Transactions on, vol. 4, no. 1, pp. 1–43, 2012.

[2] S. R. K. Branavan, D. Silver, and R. Barzilay, "Non-linear monte-carlo search in civilization ii," in Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three, 2011, pp. 2404–2410.

[3] E. J. Powley, D. Whitehouse, and P. I. Cowling, "Monte Carlo Tree Search with macro-actions and heuristic route planning for the Physical Travelling Salesman Problem," in Computational Intelligence and Games (CIG), 2012 IEEE Conference on, 2012, pp. 234–241.

[4] T. Pepels and M. H. Winands, "Enhancements for Monte-Carlo Tree Search in Ms Pac-Man," in Computational Intelligence and Games (CIG), 2012 IEEE Conference on, 2012, pp. 265–272.

[5] S. Karakovskiy and J. Togelius, "The mario ai benchmark and competitions," Computational Intelligence and AI in Games, IEEE Transactions on, vol. 4, no. 1, pp. 55–67, 2012.