Comparing game tree search techniques for general videogame AI (GVGAI)

Alastair Rayner - 1507516

***Abstract*—The abstract goes here.**

I. Introduction

THIS Literature review will cover what questions I will be asking for my dissertation topic as well all the literature I have found that is related to my research questions.

II. Research Questions

TODO: Simplify research questions

- How does game tree search techniques compare for GVGAI?
- Where does GVGAI succeed best in set games?
- Where does each tree search technique do well in each game?
- What are the most challenging areas for GVGAI in the GVGAI competition?
- What are the strengths and weaknesses of different search techniques and how can they be improved?

III. Hypothesis

//TODO: Hypothesis Does visualising the search space in GVGAI help identify the challenges.

IV. Literature Review

A. The General Video Game AI Competition

In most modern video games the AI is tailored specifically for that game and can't easily be modified for use in a different game type. However this is what GVGAI aims to create an AI that can play any game.

There have been quite a few AI competitions before in video games, such as Unreal Tournament [1], Super Mario Bros [**?**], Starcraft [2] [3]. However most of the winning AI strategies used in those games are very domain specific and it is often more about knowing the game than developing good general AI [3].

Another competition that was similar to GVG-AI was the General Game Playing (GGP) competition [4]. However almost all of the games in the GGP are board games, and the Game Description Language (GDL) used is not designed for video games.

The GVG-AI Competition is a competition framework that proposes the challenge of creating controllers for general video game playing. The controllers must be able to play a wide variety of video games, many of them will be completetly unknown to the controller. This means the controller must have some general AI to discover the mechanics and goal of the game, so it can increase it's score and win the game. [5], [3]

The framework contains a library of 2D Java based video games some of which are based of classic arcade games, there are currently as of writing this, 62 games that AI controllers can be tested on.

B. Challenges and Goals of GVGAI

The goal of GVGAI is to create a generally intelligent agent that is able to win any game it is placed in, when it doesn't know the game. During the tournament a completely new set of games are used, to avoid the agents becoming too deomain specific. Another challenge is the time limit that an agent can choose an action, this avoids the agent spending too long deciding a task and not making an action. [6]

C. Competition & Rules

The winning conditions are decided by three factors:

- Number of games finished with a victory
- Total sum of points
- Total time spent

The fist objective to be considered is the number of victories, however in case of a tie, the next objective is the number of points. Then if those two are a tie then the final decider is the total time spent before the win [3].

In the competition the agent will play 10 unknown games and 5 levels per game. Furthermore each level is played ten times, so each agent plays roughly 500 total games in the tournament [6].

D. The GVGAI Framework

The Framework is developed in the Java Environment The controllers are allowed upto 40ms to compute the agents action(s) [7], [5].

The GVGAI framework comes with quite a few sample agents;

- doNothing
- Heuristics
- Human
- olets
- repeatOlets
- and more.

These sample agents provide useful insights into how new agents can be created for the competition by applying common AI techniques. The *HUMAN* player and the *REPLAYER* can

be used for debugging the game and to help the programmer get a better understanding of how the game can be played.

The framework uses a Video Game Description Language (VGDL) to describe a wide variety of video games. The VGDL is based on a python version developed by Schaul (2014) called PyVGDL [6]. Furthermore in the GVG-AI Competition the AI agent does not have access to the whole games description, where as in GGP the agent was able to see the whole game description. This means that the agent has to analyze and simulate the game in order to figure out the rules and goal of the game.

To create an agent in

The framework has an StateObservation object that has an interaction set that consists of *up, down, left, right* and *use*.

E. Game Search Techniques

Alpha beta pruning
minimax
Breath First Search
Depth First Search
MCTS

Because of vanilla MCTS's success in the GVGAI competition there have been quite a few papers that propose modifications to the vanilla MCTS to try and improve the successfulness of the algorithm.

OLETS
Evolutionary Algorithms (RHEA)

F. Visualizing search trees in GVGAI

G. Goal Orientation

This paper [8]

H. Analyzing the Robustness of General Video Game Playing Agents

This paper [9]

I. Efficient Implementation of Breadth First Search for General Video Game Playing

This paper proposes an efficient implementation of Breath First search, however it only works well for deterministic game sets. The paper proposes a method of BFS where a node that has already been visited in other nodes will not be expanded, this is stored in a hash function. The algorithm uses hash codes to improve the efficiency and performance.

[10]

J. HyperHeuristic

Hyper Heuristic methods are [11]

V. Conclusion

The conclusion goes here.

References

[1] P. Hingston, "A new design for a turing test for bots," in *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*. IEEE, 2010, pp. 345–350.

[2] S. Ontanón, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss, "A survey of real-time strategy game ai research and competition in starcraft," *IEEE Transactions on Computational Intelligence and AI in games*, vol. 5, no. 4, pp. 293–311, 2013.

[3] D. Perez-Liebana, S. Samothrakis, J. Togelius, T. Schaul, S. M. Lucas, A. Couëtoux, J. Lee, C.-U. Lim, and T. Thompson, "The 2014 general video game playing competition," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 8, no. 3, pp. 229–243, 2016.

[4] M. Genesereth, N. Love, and B. Pell, "General game playing: Overview of the aaai competition," *AI magazine*, vol. 26, no. 2, p. 62, 2005.

[5] D. Perez, "The general video game ai competition," http://http://www.gvgai.net/, 2017.

[6] T. Schuster, "Mcts based agent for general video games," Ph.D. dissertation, Masters thesis, Department of Knowledge Engineering, Maastricht University, Maastricht, the Netherlands, 2015.

[7] D. Perez-Liebana, S. Samothrakis, J. Togelius, S. M. Lucas, and T. Schaul, "General video game ai: Competition, challenges and opportunities," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[8] B. Ross, "General video game playing with goal orientation," 2014.

[9] D. Pérez-Liébana, S. Samothrakis, J. Togelius, T. Schaul, and S. M. Lucas, "Analyzing the robustness of general video game playing agents," in *Computational Intelligence and Games (CIG), 2016 IEEE Conference on*. IEEE, 2016, pp. 1–8.

[10] S. Ito, Z. Guo, C. Y. Chu, T. Harada, and R. Thawonmas, "Efficient implementation of breadth first search for general video game playing," in *Consumer Electronics, 2016 IEEE 5th Global Conference on*. IEEE, 2016, pp. 1–2.

[11] A. Mendes, J. Togelius, and A. Nealen, "Hyper-heuristic general video game playing," in *Computational Intelligence and Games (CIG), 2016 IEEE Conference on*. IEEE, 2016, pp. 1–8.

Appendix A
First appendix

Appendices are optional. Delete or comment out this part if you do not need them.