

Project Proposal

G54IRP/COMP4027

Project Title: AI for General Video Game Playing
4262648 Benjamin Charlton (psybc3)

22nd October 2018

1 Background and Motivation

Game playing has often been used in Artificial Intelligence (AI) research to help push the field forward, developing new techniques to tackle harder and harder problems. Many of these AI techniques developed have since been applied to further applications proving that this field of study is not just producing novel solutions.

Game Playing is a good testbed to test AI techniques as games have many features that are helpful to testing AIs. Some of the following reasons show why games make good AI problems:

- Clear win states leading to easy objective evaluation
- Limited actions creating fewer output options
- Well defined documentation allowing for precise and simple simulations
- Testing against different game types as defined in Game theory, such as imperfect information, stochastic outcomes, asymmetric gameplay etc.
- Ability to test and rank AIs against human players

The history of AI game playing begins near the start of artificial intelligence as a field, in the 1950s. Strachey created a draughts player for one of the first general computing machines (Manchester Ferranti Mark I) which by 1952 could “play a complete game of draughts at a reasonable speed”[2]. Prinz wrote a simplified chess player for the Manchester Machine as well which could solve the mate-in-two, which if there was a checkmate solution in 2 turns it could find it[2].

Currently most simple board games have been solved with AI techniques. In 1997 IBM managed to beat the reigning world champion, Garry Kasparov, using Deep Blue. Deep Blue used a combination of alpha-beta search trees with specialised hardware to generate the possible moves[1]. DeepMind recently took a long standing goal of AI game playing by beating a Go champion with AlphaGo[7]. This was done with a combination of Monte Carlo Tree Search and 2 Deep Neural Networks trained with reinforcement learning, one was a value network to assess the board predicting the winner and the other a policy network to predict which move would be played next by an expert. These networks were combined in the Monte Carlo Tree Search, the policy network would predict the next moves to play and the value network would evaluate each leaf in the tree. AlphaGo was then extended by DeepMind to create AlphaGoZero which didn’t train with human data only self play[8].

Artificial Intelligence has been a key component for video games as designers have tried to create compelling single player experiences. Often AI in video games aren’t highly complex often being simple agents optimised for making the player experience fun rather than optimising to win. For this reason in game AI is often not comparable to academic AI research.

In recent years research into video game playing AIs have begun, due to many simpler games being played to a high level via AI techniques and the field moving towards tackling more complex systems. One of the most notable examples is OpenAI who have created a team of 5 bots that can play a limited version of DotA 2[5]. This was done using an LSTM neural network and has managed to beat several amateur and semi-pro teams. While it cannot currently beat a professional team it does hold up against them quite well.

DeepMind has also been working on video game playing making a model that could play a sample of Atari 2600 games[3]. While introducing a new form of reinforcement learning called Q-Learning they have shown that it can successfully learn to play video games. Although not a general AI they did show that they could use the same model and hyperparameters to train for each game, showing that game playing AI aren’t always specialised in design. General AI, the ability to create an AI that is deemed generally intelligent that it can complete a wide variety of tasks in many fields, is the “holy grail” of the AI field. Often this level of intelligence is compared to humans, or other animals, as they are good at adapting their knowledge to unseen problems. Many tests have been thought up to test how to compare the general intelligence of AIs such as the Turing test, Coffee Test or Robot College Student test[4].

- General AIs

1.1 Motivation

- Motivation

2 Aims and Objectives

This research project sets out to create and implement a General AI to play video games. To test and evaluate this goal the solution developed will be tested against the GVGAI framework using the competition rules.[6] The project will produce an AI to play in the GVGAI framework as well as a research paper describing the performance of the AI created. The objectives of this project are as follows:

Background Research Perform research into the current state of AI game playing, general AIs, and current well performing solutions to the GVGAI competition. The result of this background study will be used to direct the rest of the project and determine a suitable method to tackle the problem with.

Design a General AI Design an AI suitable to playing general video games as presented in the GVGAI framework. The design and approach will be heavily influenced by the research performed earlier in the project, taking account of other previous methods both in and out of the general game playing field as well as how the problem is presented in the framework.

Develop a General AI Implement and develop the design created to tackle the problem. This will be the primary output of the project. Due to the heavy reliance on the background research it isn't clear what this stage will involve.

Evaluation and Analysis of Solution Compare and analyse the performance of the solution against the other entries to the GVGAI competition and potentially against human players. If the solution performs well there may be the ability to test them in unseen environments, such as Video Games that aren't shown in the training sets.

3 Work Plan

To help outline the project scope and schedule, a gantt chart (found in the appendix) has been created. High level descriptions of all of the subtasks have been written out as follows. Some parts of the work plan (specifically development) are currently very vague as the initial research needs to happen to acquire the knowledge and insight of how to proceed.

Updates to the work plan will happen to break down these tasks in to more concrete steps with accurate timeframes once the direction of the project is better known. Additional updates will also come to better reflect how long certain parts of the project will take if adjustments need to be made due to events such as increase of scope or additional commitments.

Documentation

D1–Project Proposal Draft Write the project proposal draft for approval and feedback from supervisor, due 12th October.

D2–Project Proposal Finalise Any revisions to the project proposal that need to be made after supervisor feedback, due 22rd October.

D3–Preliminary Ethics Checklist Fill out and submit the preliminary ethics checklist, no further ethical checks should be required so no further time is scheduled.

D4–Interim Report Outline Sections Create a basic outline of how the interim report will look with respect to the sections included and their content

D5–Interim Report Related Work Write the related work section of the interim report, discussing previous research in General AI and game playing. This will happen alongside the research steps that correspond to this task.

D6–Interim Report Draft Create a draft of the interim report with all sections roughly finished.

D7–Interim Report Finalise Add the finishing touches to the interim report summarising the work so far, due 7th December.

D8–Research Paper Structure Sections Outline the sections of the research paper and what content will be in each section.

D9–Research Paper Abstract Write the abstract for the research paper.

D10–Research Paper Draft Create a draft of the entire research paper with all sections roughly finished.

D11–Research Paper Video Demo Create the demo video that needs to be submitted alongside the research paper.

D12–Research Paper Finalise Final touches to the research paper, due 12th April.

Research

R1–Background Research Background research into the problem and history of the field for the Background section of the project proposal, as well as research into the module structure to help plan out the project.

R2–GVGAI Basic Research Basic research into the structure and aims of the GVGAI competition.

R3–GVGAI Framework Review Look into the structure and features of the GVGAI framework, happening alongside implementing a random agent. This will help with development of this agent and inform the complexity of the problem for future planning.

- R4–GVGAI Papers for Related Works** Background research into previous attempts at the GVGAI competition, looking into successful methods and analysis of the problem to better formulate a new approach to the problem.
- R5–Other Related Works** Background research into further related works to the problem, looking into AI game playing and general AI problem solving.
- R6–Research for Solution** Rough allocation of time to allow further research into a chosen method. This will be broken into subtasks with better timeframes depending on the chosen method to tackle the problem.

Development

- S1–Install GVGAI Framework** Install the components of the GVGAI framework.
- S2–Implement Random Agent** Implement an agent that performs random actions, this won't score well but will show that the framework is working and will help with familiarity of the framework.
- S3–Run available previous agents** If available, find and run some previous solutions to the GVGAI problem.
- S4–Develop Solution** Rough allocation of time to allow development of the chosen method. This will be broken into subtasks with better timeframes depending on the chosen method to tackle the problem. Currently vague as the sub tasks involved in this step will depend on the method chosen to tackle the problem.

Presentation

- P1–Presentation General Outline** Begin work on the presentation outlining the general content and flow.
- P3–Create Demo for Presentation** Create a demo of the project to show how it performs.
- P3–Finalise Presentation** Make final adjustments to the presentation and practice delivering it in its final form.
- P4–Present Presentation** Present the presentation and demo during the demo day.

Miscellaneous

- M1–Create Git repository** Create a Git repository on the School of Computer Science Git server to allow for management of the code and documentation.
- M2–Install LaTeX** Install LaTeX and any other dependencies to allow production of documentation through out the project.
- M3–Create Meeting Minute Document** Create a document to keep track of meetings with project supervisor. This will be added to after every meeting but initial setup happens during this time.
- M4–Plan 2nd Stage of the Project** Time set aside to plan out the 2nd stage of the project involving a the research and development of the solution. This is mostly breaking down tasks R6 and S4 into precise subtasks.
- M5–Print and Collate Final Hand In** Time to print and collate the final documentation required for the main hand. This will involve properly packaging both digital and physical files.
- M6–SET/SEM Questionaries** Module feedback to be completed. Student Evaluation of Teaching and Student Evaluation of Modules to comment upon supervisor teaching and the module as a whole respectively.

Other Commitments

- C1–Welcome Weeks** First weeks of the academic year, time set aside to allow for settling in as well as running various welcome events.
- C2–Christmas Holiday** Time off after Autumn term, Partially set aside to all for a break and time to relax.
- C3–Autumn Exams** Potentially could have exams during the entire 2 weeks so is set aside to allow for revision and the exams themselves. This can be updated to give a better reflection of when exams will be at a later date.
- C4–Easter Holiday** Time off after Spring term, As no teaching is happening will give more time to concentrate on coursework and the presentation.

4 Appendix

References

- [1] Murray Campbell, A Joseph Hoane, and Feng-hsiung Hsu. Deep blue. *Artificial intelligence*, 134(1-2):57–83, 2002.
- [2] Jake Copeland. A brief history of computing. http://www.alanturing.net/turing_archive/pages/Reference%20Articles/BriefHistofComp.html#MUC, June 2000.
- [3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- [4] Luke Muehlhauser. What is agi? <https://intelligence.org/2013/08/11/what-is-agi/>, August 2013.
- [5] OpenAI. Openai five. <https://blog.openai.com/openai-five/>, 2018.

- [6] Diego Perez-Liebana, Spyridon Samothrakis, Julian Togelius, Tom Schaul, Simon M Lucas, Adrien Couëtoux, Jerry Lee, Chong-U Lim, and Tommy Thompson. The 2014 general video game playing competition. *IEEE Transactions on Computational Intelligence and AI in Games*, 8(3):229–243, 2016.
- [7] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.
- [8] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

Work Plan

