IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Issue 4, April 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

9940 572 462 | 6381 907 438 | ijircce@gmail.com | www.ijircce.com

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 4, April 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1104102 |

Building A Gaming Agent

Shilpali Bansu , Govindnarayan Singh, Akshata Madiwal, Yash Jaiswal, Akanksha Jadhav

S.R.Chunamari

Department of Computer Engineering, A C Patil College of Engineering, Kharghar, Navi Mumbai, India

ABSTRACT - The aim of General Game Playing (GGP) is to create intelligent agents that can automatically learn how to play many different games at an expert level without any human intervention. The traditional design model for GGP agents has been to use a minimax-based game-tree search augmented with an automatically learned heuristic evaluation function. The first successful GGP agents all followed that approach. In here we describe, a GGP agent employing a radically different approach: instead of a traditional game-tree search it uses Monte-Carlo simulations for its move decisions. Furthermore, we empirically evaluate different simulationbased approaches on a wide variety of games; introduce a domain-independent enhancement for automatically learning search-control knowledge to guide the simulation playouts; and show how to adapt the simulation searches to be more effective in single- agent games.

KEYWORDS: AI Agent, Monte Carlo Tree Search, Algorithms, Evaluation Function

I. INTRODUCTION

Gaming agents are basically artificial intelligence programs that learn and play game efficiently without human intervention. For many games like chess, Othello computer programs are designed such that they play the game with best move but they cannot learn other games based on learning one game. Gaming agents knows as GGP (General Game Playing) are capable of playing previously unknown games of a wide variety by being told nothing but the rules of the game which are very different from programs which are designed to play specifically a game General Game Playing is concerned with the development of systems that can play well an arbitrary game solely.

Systems able to play arbitrary, unknown games can't be given game-specific knowledge. They rather need to be endowed with high-level cognitive abilities such as general strategic thinking and abstract reasoning. This makes General Game Playing a good example of a challenge problem which encompasses a variety of AI research areas including knowledge representation and reasoning, heuristic search, planning, and learning. This is very interesting field in AI of building game playing agents capable of matching wits with the strongest humans in the world. In-fact, it is a very interesting and research challenge to
build intelligent software agents that can learn a strategy for game playing simulating human behavior.

Moreover as per Dresden group of GGP , "A General Game Playing system, if well designed, would be able to help in other areas, such as in providing intelligence for search and rescue missions". Since game playing is to provide fun to humans nothing much can be done in learning the game but it can be used in real world problems. Any useful task can be cast as a digital game where task (may be driving, business related- paying etc) which provide a well contained environment to operate in.

II. LITERATURE SURVEY

1. Survey of Existing System

The simplest way to gain insight into areas of success and opportunities for improvement is through feedback collected from survey participants. The primary objective of this survey is to conduct a comparative analysis of sharing applications that are available across all platforms. This analysis will include descriptions of the applications, as well as an evaluation of their respective advantages and disadvantages.

a. Cadia-Player

CADIA-Player: A General Game Playing Agent (PUBLCATION AUTOR AND DATE:- Hilmar Finnsson Master of Science December 2007 WINNER OF GGP COMPETITION DATED 2007) CADIA

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 4, April 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1104102 |

PLAYER, a GGP agent. It does not require any a prior domain knowledge nor does it use heuristic evaluation of game positions. Instead, it relies exclusively on MonteCarlo based simulation search for reasoning about its actions, but guided by an effective search- control learning mechanism.

b. Ary

Ary, a general game playing program (PUBLICATION DATE:-JANUARY 2010 AUTHORS:- TRISTANCAZENAVE(PARISDAUPHINE UNIVERSITY) JEAN MEHAT(UNIVERSITY OF

VICENNES) WINNER OF GGP COMPETITION
HELD ON 2004) Ary uses a simple Monte-Carlo method to explore the game tree. The program executes random play outs by playing random legal moves until the end of the game. At the end of the thinking time, the move with the best mean reward is played.

c. **Developing Game Playing Agents** Developing Game Playing Agents That Adapt to User Strategies (PUBLISHED DATE :- DEC 2014
Conference: IEEE Symposium on Intelligent Agents) This describes the development of a novel webdelivered computer game, Boundary, where human players vie against each other or computer agents that use adaptive learning to modify playing strategies.

d. Flux Player

Flux player: A Successful General Game Player (AUTHORS:-STEPHAN SCHIFFEL AND MICHAEL THIELSCHER CONFERENCE PAPER · JANUARY 2007 DEPART2 MENT OF COMPUTER SCIENCE DRESDEN UNIVERSITY
OF TECHNOLOGY STEPHAN.SCHIFFEL,MIT@INF.TUDRESDEN.DE WINNER OF GGP COMPETITION HELD ON 2007)
An approach to General Game Playing which combines reasoning about actions with heuristic search. IT Uses techniques for constructing search heuristics by the automated analysis of game specifications.

2.3 This article discusses the importance of gaming agents in GGP Competition.

General Game Playing is a project of the Stanford Logic Group of Stanford University, California, which aims to create a platform for general game playing. It is the most well-known effort at standardizing GGP AI, and generally seen as the standard for GGP systems. The games are defined by sets of rules represented in the Game Description Language. In order to play the games, players interact with a game hosting server that monitors moves for legality and keeps players informed of state changes.
Since 2005, there have been annual General Game Playing competitions at the AAAI Conference. The competition judges competitor AI's abilities to play a variety of different games, by recording their performance on each individual game. In the first stage of the competition, entrants are judged on their ability to perform legal moves, gain the upper hand, and complete games faster. In the following runoff round, the AIs face off against each other in increasingly complex games. The AI that
wins the most games at this stage wins the competition, and until 2013 its creator used to win a $10,000 prize. So far, the following programs were victorious:

III. PROPOSED SYSTEM

1. Introduction

Our Approach is to design a gaming agent will be capable of playing games that can be single player game like puzzle or multi player games like carrom, chess without providing any knowledge to the system. In simple words, the agent will have zero knowledge about any game, rules of the game will also be unknown until the game starts.The agent will be provided the environment to play and rules during the game. Initially, the agent will make mistakes and it will learn by its own mistake leading to provide the optimal solution and becoming the champion in that game. Unlike other agents, database will not be provided to our system initially. We Will use the traditional Algorithms and will even design our new algorithm.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 4, April 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1104102 |

2. Architecture

An Architecture of the game playing agent is shown below. At the center of the ecosystem is the game manager.

- Game manager maintains a database of game descriptions and match records, and it maintains some temporary state for matches while they are running.
- Game manager communicates with game players. It also provides a user interface for users who want to schedule matches, and it provides graphics for spectators watching matches in progress. Communication between the Game Manager and game players takes place through HTTP connections. communication model assumes that each player is running on an Internet connected host listening on a particular port.
- An info message is used to confirm that a player is up and running.
- A start message is used to initialize an instance of a game. • A play message is used to request a move from a player.

The process of running a match goes as follows.
Upon receiving a request to run a match, the Game Manager first sends a start message to each player to initiate the match. Once game play begins, the manager sends play messages to each player to get their plays; and it then simulates the results. This part of the process repeats until the game is over.

The Manager then sends a stop message to each player The Game Manager checks that these actions are legal, simulates their effects, updates the state of the game, and then sends play messages to the players to solicit their next actions. The second argument in the play message this time is a list of the actions received in response to the preceding play message.

1. Algorithm

GDL for game description:-
GDL is a logic programming language. It is similar to other logic programming languages, such as Prolog; but there are some important differences. • The semantics of GDL is purely declarative (there are no procedural constructs like assert, retract, and cut).

- GDL has restrictions that assure that all questions of logical entailment are decidable

1. Alpha–beta pruning Algorithm:-

Alpha–beta pruning is a search algorithm that seeks to decrease the number of nodes that are evaluated by the minimax algorithm in its search tree. It is an adversarial search algorithm used commonly for machine playing of two-player games (Tic-tactoe, Chess, Connect 4, etc.). It stops evaluating a move when at least one possibility has been found that proves the move to be worse than a previously examined move. Such moves need not be evaluated further. When applied to a standard minimax tree, it returns the same move as minimax would, but prunes away branches that cannot possibly influence the final decision.

2. Minimax Algorithm:-

Minimax is a kind of backtracking algorithm that is used in decision making and game theory to find the optimal move for a player, assuming that your opponent also plays optimally. It is widely used in two player turn-based games such as Tic-Tac-Toe, Backgammon, Mancala, Chess, etc.
In Minimax the two players are called maximizer and minimizer. The maximizer tries to get the highest score possible while the minimizer tries to do the opposite and get the lowest score possible.
Every board state has a value associated with it. In a given state if the maximizer has upper hand then, the score of the board will tend to be some positive value. If the minimizer has the upper hand in that board state then it will tend to be some negative value. The values of the board are calculated by some heuristics which are unique for every type of game.

3. **MONTE CARLO / Stochastic Search:-** The basic idea of Monte Carlo Search (MCTS) is simple. In order to estimate the value of a nonterminal state, we make some probes from that state to the end of the game by selecting random moves for the players. We sum up the total reward for all such probes and divide by the number of probes to

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 4, April 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1104102 |

obtain an estimated utility for that state. We can then use these expected utilities in comparing states and selecting actions. Monte Carlo can be used in compulsive deliberation fashion to evaluate the immediate successors of the current state. However, it can also be used as an evaluation function for heuristic search.

4. **Algorithm for selecting the moves (UCT) :-** The Upper Confidence Bounds applied to Trees (UCT) algorithm is a generalization of the UCB algorithm that can be applied to game trees. The algorithm uses simulations to gradually build a game tree in memory where it keeps track of the average return of each state- action pair played, Q(s, a). It offers an effective and sound way to balance the exploration versus exploration trade-offs.

3. Requirement Analysis

1. Hardware Requirements

- RAM: 8 GB
- CPU: Intel i5/AMD RYZEN 5 and Above

2. Software Requirements

- OS: Windows 7+ 64 bit.
- Programming Language: Java, HTML
- Software Tools: Eclipse IDE 2022-12
- Backend: csv, json
- Jdk version :17 and above

IV. CONCLUSIONS

This research paper focuses on the development of an AI based Gaming Agent in which various algorithms are used to develop AI. The games present are also simple and played world wide. We have developed this project using java and the database are stored in json format for easy use. Games developed can be played by users in any circumstances without any error. Whole project is based on developing the user friendly interface to play games by various AI agents as well as humans

REFERENCES

1. Koller, D., Pfeffer, A.: Representations and solutions for game-theoretic problems. Artificial Intelligence 94(1), 167–215 (1997)
2. M. R. Genesereth, N. Love, and B. Pell,"General Game Playing: Overview of the AAAI competition." AI Magazine, vol. 26, no. 2, pp. 62–72, 2005.
3. R. Coulom, "Efficient selectivity and backup operators in Monte-Carlo tree search," in The 5th International Conference on Computers and Games (CG2006), 2006, pp. 72–83. [Online]. Available: http: //remi.coulom.free.fr/CG2006
4. George F.luger. Artificial Intelligence Structures and Strategies for Complex Problem Solving Fifth edition [M]. BeiJing: China Machine Press,2006
5. S. Schiffel and M. Thielscher, "Automatic construction of a heuristic search function for General Game Playing," in Seventh IJCAI International Workshop on Nonmontonic Reasoning, Action and Change (NRAC07), 2007
6. Clune, J.: Heuristic evaluation functions for general game playing. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp.1134–1139 (2007)

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 4, April 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1104102 |

7. Nathaniel Love, Timothy Hinrichs, David Haley, Eric Schkufza and Michael Genesereth, "General Game Playing: Game Description Language Specification" in Technical report, Stanford Logic Group Computer Science Department Stanford University, 2008.[1]
8. Yngvi Bj¨ornsson and Hilmar Finnsson. Cadiaplayer: A simulation-based general game player. IEEE Transactions on Computational Intelligence and AI in Games, 1(1):4–15, 2009.
9. Kaiser, Ł., Stafiniak, Ł.: First-order logic with counting for general game playing. In: Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco (2011)
10. Yuxia Sun, Ziyang Zhang and Xiaoyan Wang, "The Research of UCT and Rapid Action Value Estimation in NoGo Game", 2016 28th Chinese Control and Decision Conference(CCDC 2016), pp. 5192-5196, 2016.
11. C. B. Browne, "A class grammar for general games", Advances in Computer Games, pp. 167182, 2016.
12. C. Browne, "Modern techniques for ancient games", 2018 IEEE Conf. Comput. Intell. Games, pp. 490-497, 2018.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.379

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details