International Journal of Scientific Research in Computer Science and Engineering
Technical Paper
Vol.6, Issue.6, pp.31-34, December (2018)
E-ISSN: 2320-7639

Extending Specialized Systems to a Generic Approach of Game Playing

Vaibhav Kataria

Dept .of CSE, Maharaja Surajmal Institute of Technology, GGS Indraprastha University, New Delhi, India

**Corresponding Author: vaibhavk4999@gmail.com, Mob.: +91-9582-316-713*

Available online at: www.isroset.org

Received: 07/Dec/2018, Accepted: 22/Dec/2018, Online: 31/Dec/2018

Abstract— Better decision making requires better analysis of the situation at hand. A program which can analyze the current situation better, can take better decisions, thus, increasing the likelihood of winning more games. Specialized game players are very narrow. They might outperform humans in a particular game and know nothing about any other game. They do only part of the work. Most of the definition, analysis and design is done in backend, within the systems by their programmers.

Generic Game Bot are systems which are able to accept descriptions of any game at runtime and can use such definitions to play that game effectively, without any human intervention or supervision, i.e., they do not know the rules of the games beforehand. Generic Bot as such should be capable of playing simple games (like Snake) and complex games (like Chess), games in static or dynamic environments, games with complete and partial information, games with different numbers of players, with simultaneous or alternating turn of play, with or without communication among the players, and so forth.

Keywords-General Game Playing; Specialized Systems; Game Bot; Reinforcement Learning; Classification of Games; OpenAI; Generic game Player; Game Description Language.

I. INTRODUCTION

General game playing (GGP) is the design of intelligent programs that are able to play more than one game successfully. For many games like chess, checkers, etc., computers are programmed to play these games using a specially designed algorithm, which cannot be transferred to another game environment [8]. For eg., a chess-playing computer program cannot play any other game (say, tic-tac-toe). A generic game player, if well designed, would be able to help in other areas, such as in providing intelligence for search and rescue missions, reducing the effort of programming a smart system and providing a base for further research [1][4][8].

A similar project on General Game Playing was taken by Stanford Logic Group of Stanford University, California, in 2005, to establish a platform for general game playing. It is the most widely known effort at standardizing GGP AI, and generally taken as the standard for GGP systems. The games are defined by sets of rules represented in the Game Description Language.

The Game Description Language is a high-level, rule-based linguistics for communicating the rules of arbitrary games to general game-playing systems, whose challenging task is to learn to play previously unknown games without human intervention. Originally designed for deterministic games with complete information about the states in the game, the language was recently extended to include randomness and imperfect information. [2][3][8]

However, defining the extent to which this enhancement allows to describe truly arbitrary games was left as an open problem. And this is the major drawback of GDL which gives us a good field for research.

There hasn’t been any major landmark in the field of General Game Playing due to the vast variation and differences in the games including Rules, Moves, State changes, Goal State variation. [2][3]

If we achieve the accuracy and efficiency that the researchers have been fantasizing about, major implementations can be found in daily life as well. The Power to think beyond the already coded-in knowledge, and dug deep in to the artificial intelligence will help us in a lot of domain ranging from everyday chores to rescue missions.

For the sake of this paper, we classify the games in two categories, One, in which the future states are random or doesn’t depend upon the player’s move, in other words, the states that appear before the user are random and cannot be foreseen, no matter what the user chooses the future state will remain unchanged. Generally, Simple games fall into this category, which have basic rules and moves within a specific environment. Other, the games in which the player’s move matters and forces change in the future states and to succeed in such games, user needs to carry each step with minimum loss to maximize the score and reach the goal state. These games involve dynamic changes, and for each decision the future states have to be considered, while in the previous category the future states have no relevance for current position and choice. More Complex games fall under this category

In this paper, we will analyze the previously proposed algorithms and the effect of modifications on the performance of the bot. Also, the development of specialized game player to attain maximum efficiency for comparison with a more intelligent algorithm involving implementation of reinforcement learning, q- learning and genetic algorithm. Section I contains the introduction of need, scope and development of Generic Game Player, Section II contains work, starting with the development of game-specific bot, taking the game of T-rex (Chrome- Offline Page Game) followed by the development of a generic approach to the game, Section III involves the methodology used in the development of the Generic Bot using the specialized systems to attain maximum accuracy and introduce us to the libraries used in the projects as well as the algorithmic approach used in the development of the generic bot. Deducing results after the modifications and training the generic player over data generated and studying the effects of various algorithms.

II. Related Work

For the development of the generic game player, we need something to compare it to, so as to know the efficiency of the generic player, for this, we implement a specialized system to play T-rex (Category One) using brute force method and ask the game actor to jump over the obstacles whenever the pixel ahead of the actor turn black. In simple terms, we have divided the t-rex's path in three parts, and the obstacle to be identified and avoided if it is nearer than 20 pixels.

Actual game actor is as shown in the fig.1,

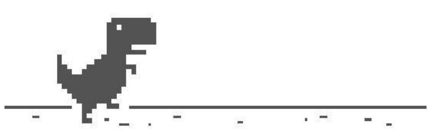


Figure 1

Now there are four categories that decide which move has to be chosen, these categories are 0-point, 1-point, 2-point & 3-point as in figure 2.

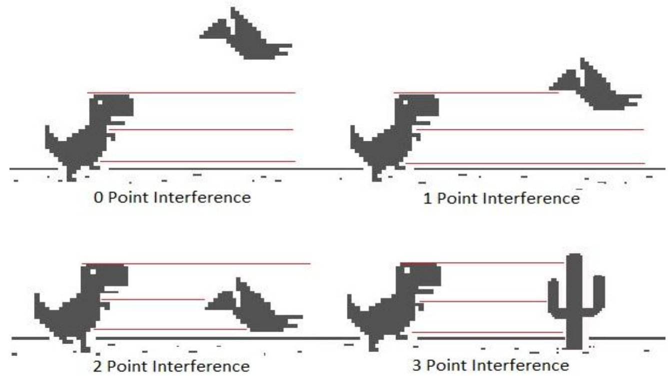


Figure 2

For every 0.2 seconds or less for accuracy of a decision, a snapshot of the game is analyzed on the basis of the point interference, refer table 1

Interference	Action
0-Point	No Action
1-Point	Duck
2-Point	Jump
3-Point	Jump

Table 1

Now for the Bot to function, we need screenshots after every few milliseconds, and analyze the level of interference of the actor with the obstacle.

The pixels are counted and on the actor's level, if the color of the pixel distant from the object at about 15-20 pixel is indistinguishable with that of the actor, then a case of obstacle arises.

The specialized system created now is used to train the generic player, not how to choose the action but to identify the actor and use the specialized system to identify the moves that are taken and other keys that actually have an impact on the game.

III. Methodology

The specialized system performs the decisions on the case-based method, as when a situation of collision arises the specialized systems avoids it.

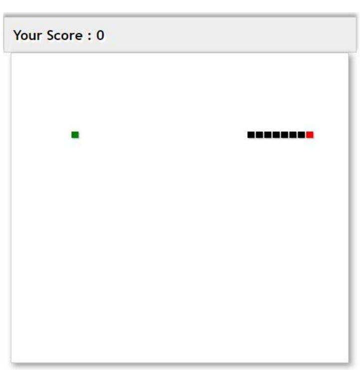
During the progress of this project, two approaches that held much potential and field to explore were:
Using the concept of GDL as the base of the Game-Player for decision making: Game Description Language introduces a lot of complexity with increase in the number of games and the number of dimensions' (comparing the game

of t-rex and Mini-militia), and the common moves of these games are given higher priority [2][3]. Based on OpenAI, libraries such as Gym and Universe allow us to play multiple games without explicitly being programmed, but the performance of these libraries is really low, Implementation of reinforcement learning, neural networks, genetic algorithm can improve the performance of the game-player [1][5].

Using game-specific Bot to show the generic game player to learn to identify actor and moves at runtime

This paper is more oriented towards this approach. In this, we are dependent on the learning capabilities of the bot. For this, a dedicated game-specific bot is to be constructed and that is used to teach the game player to identify the actor in different game (for eg., the common portion of screenshots of the game while the game-specific bot is used, and the game player can see the common part that doesn't change often, for instance, t-rex game that we see on chrome page, at different intervals, it is observed, that the t-rex remains the same with minor difference between its legs), and the game play in this approach is used to teach the generic-game bot, this we may call knowledge base for the game player. The advantage of using such technique will improve the time complexity, and will teach the generic-game bot only while the accuracy is low. The idea of making game specific bot will be highly general and unique, for each obstacle that the actor encounters it will try to avoid it and by doing this it would be able to explore the possible moves in the game (for eg, T-rex game requires jump and duck, while in the game of flappy bird only a single tap is required to adjust the height of the bird to avoid obstacles), the extent of this paper for now is limited to simpler games. The Specific Bot trains the generic bot and force it to work on a strategy, until its accuracy reaches minimum of 0.75/1.0. When the generic bot achieves the minimum accuracy level, game-specific bot is replaced by generic one to check the performance of the training received.

To take this system to a more generic approach, we used the specialized system to identify the actor in the game, like in the game of T-rex and Snake, as in figure 1 and figure 4, respectively



Snake Game, Figure 4

The generic player uses the successive snapshots to figure out the common pixels and their behavior as the moves are made and this sub-part in the images is generally the actor in such games (Category One).

The Generic Bot reads the keyboard and Mouse events and the changes in the game state, to determine what move is actually beneficial and which is not. Also, the data generated by specialized game is used for the generic bot's advantage by using that data to test the performance ratio of the generic bot.

The Generic Bot approaches the game with a dictionary similar to the concept of GDL, with the moves identified by a key element, which is the game state, to look for the move stored in the dictionary. As the Bot fails to perform, it explores more possibilities of the game by using other keys which might provide with another functionality.

Now, the generic bot has started to explore the moves and the states, when it reaches an accuracy ratio >=0.75 (when the generic bot can predict the next move of the game and recognize the game state with at least 75% accuracy), the generic bot kicks in and the specialized system is put to rest. And then the generic bot is left to work on itself, using learning algorithms (reinforcement learning / q-learning).

The same generic bot when subjected to game of Snake, it failed to identify the actor at first.

After subsequent modifications to look at the screen capture, to look for a particular pattern, the bot came to recognize the Snake in the game, and its natural movement and when it touches the food-pixel, the pattern of the snake modifies. The Bot learns to avoid itself and the wall and create an exception for the last row and column to find a way back to the beginning and look for the highlighted-pixel again.

The generic bot used a similar technique by which the Snake Game-specific bot was programmed, but could identify the other parameters of the game, and more importantly the plausible moves and the final state.

1: create database with the category of obstacle and move chosen using Specialized Bot.

2: Using the game state changes and the screen capture, generic bot analyses the data received for pattern matching to identify the actor and the moves being taken.

3: using the data generated in Step 1 to test the accuracy of the bot, and allow the generic bot to step in once it reaches the accuracy level of 75%.

4. Once the generic bot is deployed, it then learns by itself to explore the game beyond the specialized system.

IV. RESULTS AND DISCUSSION

Generic Bot subjected to specialized player and reinforcement learning proves to be a great decision. And Figure 5 shows the score achieved within the allowed time frame.

Results for T-Rex, Figure 5

The algorithm for the player works outstandingly well, and it is observed, as the generic bot is subjected to more games, the time requirement increases substantially.

When the same algorithm was applied to the game of Snake, the algorithm was able to win the game, but the time complexity increased to

$$T(n) = O(n^2)$$

Which resulted in slower performance of the bot, and ultimately falling out of time frame.

The idea to use the specialized bot proved to be a boon and time requirement to create a Deep-Learning Bot is reduced to a great extent.

V. CONCLUSION AND FUTURE SCOPE

The work done so far has allowed to make faster algorithms for a game player, which can provide quick results. However, the goal of creating a generic game player is still a mile away and a lot of optimisations to be taken care of. The generic game player, so far, is intelligent enough to identify the game environment, game actor, moves, and the goal state.

Though it still is a wide area of research and experimentation, there hasn't been any major landmark in the field of General Game Playing due to the vast variation and differences in the games including Rules, Moves, State changes, Goal State variation.

If we achieve the accuracy and efficiency that the researchers have been fantasizing about, the Power to explore and learn, to think beyond the already coded-in knowledge, and dig deep in to the artificial intelligence will help us in a lot of domain ranging from everyday chores to rescue missions.

REFERENCES

[1] M. Swiechowski & J. Mandziuk, "*Specialized vs. Multi-game Approaches to AI in Games*", Proceedings of the 7th IEEE International Conference on Intelligent Systems, pp. 243-254,2014

[2] S. Schiffel & M. Thielscher "*Representing and Reasoning About the Rules of General Games With Imperfect Information*", Journal of Artificial Intelligence Research Vol. 49, pp. 171-206 , 2014

[3] J. Romero, A. Saffidine & M. Thielscher, "*Solving the Inferential Frame Problem in the General Game Description Language*", Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence,2014.

[4] M. Genesereth & M. Thielscher, "*General Game Playing*", 2014

[5] M. Swiechowski & J. Mandziuk, "*Self-Adaptation of Playing Strategies in General Game Playing*", Computational Intelligence and AI in Games, IEEE Transactions on Vol. PP(99), 2013

[6] A. Saffidine, H. Finnsson & M. Buro, "*Alpha-Beta Pruning for Games with Simultaneous Moves*", The Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012

[7] S. Haufe, S. Schiffel & M. Thielscher, "*Automated Verification of State Sequence Invariants in General Game Playing*", Artificial Intelligence Vol. 187-188, pp. 1-30 , 2012

[8] H. Finnsson, "*Simulation-Based General Game Playing*", PhD Thesis Reykjavík University, 2012

[9] X. Sheng & D. Thuente, "*Using Decision Trees for State Evaluation in General Game Playing*", KI Vol. 25(1), pp. 53-56, 2011

Authors Profile

Mr. Vaibhav Kataria pursuing Bachelor in Computer Science and Engineering from Guru Gobind Singh Indraprastha University, New Delhi, India. A Technology enthusiast whose interests include Big Data Analytics, Data Mining, IoT and Computational Intelligence based education, with prime focus on Artifical Intelligence and Internet of Things.

