2019 IEEE Pune Section International Conference (PuneCon)
MIT World Peace University, Pune, India. Dec 18-20, 2019

Solving the Classic Snake Game Using AI

Shubham Sharma
Computer Engineering)
RMDSSOE
Pune, India
Shubhamsharma91097@gmail.com

Saurabh Mishra
Computer Engineering
RMDSSOE
Pune, India
kabirmish123@gmail.com

Nachiket Deodhar
Computer Engineering
RMDSSOE
Pune, India
deodhar.nachiket@gmail.com

Akshay Katageri
Computer Engineering
RMDSSOE
Pune, India
katageriakshay@gmail.com

Parth Sagar
Computer Engineering
RMDSSOE
Pune, India
parthsagar_rmdssoe@sinhgad.edu

***Abstract*— In this paper we did survey of various papers based on the classic snake game and compared their various traits and features.**

In this paper we introduce an AI bot to enhance the skills of the player and the AI bot uses the algorithms further discussed in this paper. Player can follow the simultaneously running AI bot to play the game effectively.

In this we use the classic snake game, for that we present different algorithms or methods for AI bot. It includes three searching algorithms related to artificial intelligence, Best First Search, A* Search and improved A* Search with forward checking, and two baseline methods Random Move and Almighty Move [1].

***Keywords— Best First Search, Breadth First Search, Greedy Search, A* Search, A* with Forward Checking, Snake Game, Artificial Intelligence, Random move, Almighty move, Automated Bot*.**

I. INTRODUCTION

A. Information

The most beloved and common game 'The Snake' invented in 1970's was the most famous game at that time and is still played by many people. This game was first introduced in Nokia phones. This game gave the pioneer base for other mobile phone games.

We will be showing different algorithms and methods for the game play and distinguishing between them with the help of graph.

The snake begins with length 1. In each iteration snake leads a stage, furthermore, it can go in typical four directions. There are few sorts of sustenance in the game. Eating the fruit which increases the length of snake by 1. There is constantly one fruit in the game play .As soon as the snake eats the respective fruit , a new fruit gets generated at any randomized position on the game board. The goal of our play is to augment the score. The snake dies in two conditions :

1. Snake hits its head on the wall.

2. Snake hits its own body.

The previously mentioned straightforward methodology will insure the snake is alive, yet without advancing toward the fruit proficiently it can't get a high score. In this manner, we have to structure a progressively astute controller, which is theme of the paper.

The snake is typically inbound in the board which is of 2-dimensions. The snake approaches the destined path with its head in front followed by its body. The play finishes whenever the snake slams into the walls of the playground or the snake body itself.

This game can be classified in two ways:

1. Some plays just have a solitary user, and the target is to pick up as much as conceivable avoiding an impact. Since the zone is of limited parameter and the snake doesn't diminish and the play is ensured to finish in the end. The player plays on its own intelligence to augment the score. This user advances the game endeavouring its own gaming logic.

2. The game play we are proposing consists of an AI bot. This AI bot will function as a shadow in the game. This AI bot will advance into the game using the intellect gained by the proposed algorithms and by training in the training environment [2].

B. Similar Work

Gaming giants have demonstrated numerous effective applications to structure AI or on the other hand produce substance for PC games. A valuable worldview is to utilize EAs to enhance the experience of the play. Gallager and Ryan proposed a standard based bot for playing the Pacman game. The game describes using the entity (Ms. Pacman) to gather supplements and not to get caught from phantoms in labyrinth. The bot initially decides its state, investigate or return, in light of the separation among Pacman and the phantom. At that point, it decides the new course of development in light of the sort of its area (for example passageway or crossing point) probabilistically. The separation limit and the certainties structure a lot of experiences of the controller. An advancement procedure (ES) was connected to advance estimations of these parameters. Notwithstanding structuring AI controllers for game play, EA can be used to produce game substance. Various users could have various inclinations when playing this game. Some players like to play the game slowly while some rush to complete a stage quickly. The generator generates a level by mixing up some of the zones. The proportions of zones were improved by intelligent

transformative streamlining, where clients assume the job of assessment [7].

II. Related Work

A developmental methodology has likewise been proposed for preparing specialists to play an adjusted rendition of the Snake game where few sustenance things diminish the score and snakes length. A weighted blend of rating capacities scores every conceivable iterative step, and the most astounding move is taken. The methodology utilizes four rating capacities.

1. The smoothness capacity finds, for every cell, the most modest number of moves expected to arrive at the cell, and takes on the biggest of these qualities. This evaluations how far the snake can move while constraining heading changes.

2. The space capacity assesses to the quantity of playing zone. A bigger worth shows a place that gives a more noteworthy degree for protective development.

3. Two capacities gauge the potential outcomes of arriving at each fruit type securely. An inaccessible fruit provides worth of null. An accessible sustenance type creates an estimation of the space capacity separated due to separation to the fruit [2].

III. ALGORITHM

Here we show case the algorithms that will be used in training the AI bot to take appropriate and best decision during the game.

A. Best First Search –

It is the combination of BFS (Best First Search) and DFS(Depth First Search).It takes into account the closest distance between the snake and the fruit. Closest distance is measured using Manhattan distance.

Manhattan Distance=|p1-p2|+|q1-q2|,

where (p1,q1) and (p2,q2) are the co-ordinates of snake head and fruit.

The limitation is that, it guarantees the snake till length 4 , because for length greater than 4 sneak can bite itself.

B. A Search-*

A* search algorithm is dependent on the cost of the path to reach the current fruit from the starting, and the heuristic distance from the head of the snake to the next fruit.

Cost means the number of pixels the snake has traversed till now, and Heuristic distance which is dependent on heuristic function which finds the optimal minimum distance between the head and the fruit.

f(n)=g(n)+h(n),

Where g(n) is cost and h(n) is heuristic distance.

This helps algorithm to find the minimum time consuming path, the highest distance is limited. The heuristic path gives different possibilities. If no path is less that the min path then the algorithm gets stuck [1].

C. A Searching with forward checking-*

A* algorithm has some limitations. It only checks the path till the fruit is reached, with the knowledge of the previous path cost. It has nothing to do with the consequences after the fruit is reached. If the head sticks at the dead end or to the snake body with no path to the destination fruit the Algorithm stops to work.

Therefore A* with forward checking is the solution , it checks for the path to reach any required position after the goal is reached as a precondition after selecting the path.

D. Random Move-

It is the first Comparison method introduced. It considers only the movement of the head of the snake , the body is not considered. It selects the next possible move randomly by just considering that the game should not terminate.

Although this is not the best method , but it is considered as the good comparison method.

E. Almighty Move-

1. Almighty Move for **N*N map** , Where **N is Even**. It guarantees that the snake will surely eat **(N^2-2) Fruits.**

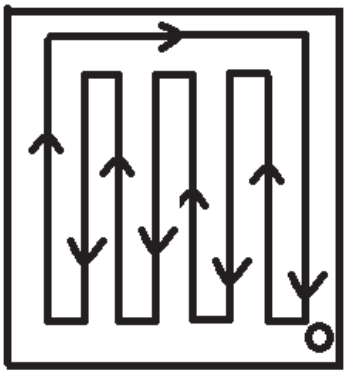

Fig. 1. Almighty Move for **N*N map** where **N is Even**

2. Almighty Move for **N*N map** , Where **N is Odd**.

It guarantees that the snake will surely eat Number of fruits greater than **(N^2- 4N+4),** but less than **(N^2-2).**

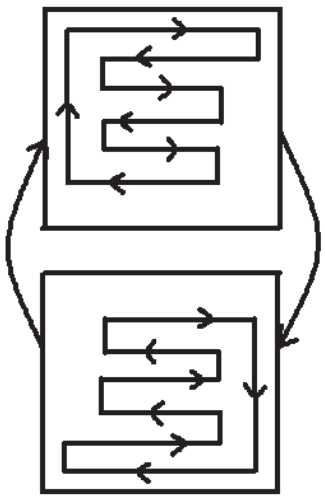

Fig. 2. Almighty Move for **N*N map** , Where **N is Odd**

IV. COMPARISON

This figure shows the output for different algorithms, when run on the board of size 10*10.

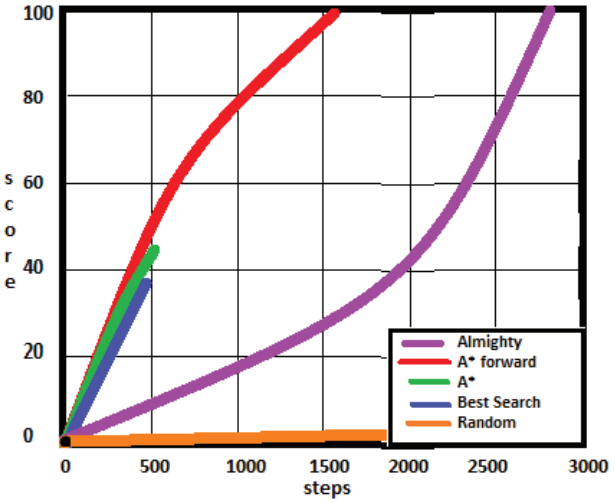

Fig. 3. Output for different algorithms, when run on the board of size 10*10

Different colours represent different algorithms. It also represents the steps required or taken by the respective algorithm to reach their maximum capable score.

This table represents the range of score and the range of steps required to achieve the score mentioned below. This data is collected by performing 100 iterations of the game for each of the respective algorithms [1].

TABLE I. RANGE OF SCORE AND THE RANGE OF STEPS REQUIRED TO ACHIEVE THE SCORE MENTIONED

ALGORITHMS	SCORES	STEPS REQUIRED
ALMIGHTY MOVE	100	2300-2700
A* WITH FORWARD SEARCH	40-80	250-1250
A*	24-40	200-300
BEST FIRST SEARCH	14-26	100-200
RANDOM MOVE	3-7	200-800

TABLE II. COMPARISON BETWEEN THE PAPERS W.R.T. TECHNOLOGY AND THE BEST CASES APPLIED

NAME OF PAPER	TECHNOLOGY	BEST CASE
AUTOMATED SNAKE GAME SOLVERS VIA AI SEARCH ALGORITHMS	A*, A* WITH FORWARD REFERENCE, ALMIGHTY MOVE, BEST FIRST SEARCH	ALMIGHTY MOVE
SNAKE GAME AI: MOVEMENT RATING FUNCTIONS AND EVOLUTIONARY ALGORITHM-BASED OPTIMIZATION	SNAKE EVOLUTIONARY ALGORITHM	EVOLVED AND HEURISTIC CONTROLLER
PLAYING THE GAME OF SNAKE WITH LIMITED KNOWLEDGE: UNSUPERVISED NEURO-CONTROLLERS TRAINED USING PSO	NEURO CONTROLLERS, PARTICLE SWARN OPTIMIZATION	THE NEUROCONTROLLER METHOD CLEARLY OUTPERFORMEDTHE HAND OPTIMIZED AGENTS, DEMONSTRATING THE APPROACH'S FEASIBILITY

The table above shows the ccomparison between the papers w.r.t. technology and the best cases applied

V. METHOD

Working of AI Bot using AI algorithms is decided by analysing the algorithms and comparing them on the basis of their performance. The number of iterations for each algorithm is decided by running the AI Bot for 100 times on the board of size 10*10.

SEQUENCE	ALGORITHM / MOVE	NUMBER OF ITERATIONS
1	BEST FIRST SEARCH	4 [0-4]
2	A* WITH FORWARD CHECKING	34 [4-38]
3	ALMIGHY MOVE	62 [38-100]

Best First Search is used for 4 iterations, i.e. to eat the first 4 fruits. As it finds the shortest distance between the snake head and the fruit .For snake of size 4 , it is not possible to that its head touches its body , therefore there is no probability that the snake dies.

A* with forward checking is used for the next 34 iterations, i.e. to eat the next 34 fruits. It increases its length from 4 to 38. It is decided by performing the iterations for 100 times. A* with forward checking checks for the conditions after the fruit is consumed for sustenance.

Almighty move is used for the next 62 iterations, i.e. to eat all the remaining 62 fruits. It increases its length from 38-100.Here the maximum score is reached. There is no possibility of the failure of Almighty move, but the reason why it is not used from the first iteration itself is that the number of steps required increases to a large extent, thus increasing the time complexity[1].

VI. DISCUSSION

A. Basic Concept

There are two snakes in the game play :

1. Manual snake which is operated by the player

2. AI Bot which is operated by the AI algorithms.

To achieve the maximum score in the optimal time and space complexity the manual player has to copy the moves of

the AI Bot. This can be used as a training phenomenon for the visual computer game players. AI Bot is trained to achieve the maximum score possible in the minimum number of steps.

This can also be used in other games of bigger size, which are the part of "Electronic Sport" to train the players.

B. Concept Expanded

(1) On the white colour 10*10 board , there are two different colours of snakes:

- Manual player snake is black in colour.
- AI Bot snake is gray in colour.

The AI Bot runs as a shadow in the background of the play simultaneously for the player to follow it. The two snakes can overlap each other and have no impact on each other.

(2) When the game starts, the AI Bot is initiated to play at the same instance. The manual player starts the play after the time interval of approximately 1second. The manual player can follow the AI Bot precisely when it is one step ahead.

(3) Initially the fruit is black in colour when it is generated. As soon as any of the two snakes eats the fruit for the first time, its colour changes to red. Then the snake which is left to eat the fruit eats the fruit and the fruit disappears.

When the snake eats the black fruit and the fruit turns red, at the same instance another black fruit is generated. This is the medium of sustenance for the snake which is ahead in the game [2].

VII. CONCLUSION

Henceforth after studying and analyzing the concept, we conclude that for AI Bot to achieve the maximum score in the minimum number of steps use BFS for first 4 moves, use A* with forward search for next 34 moves, and use Almighty move for the last 62 moves. The other conclusion made is we can use the AI Bot to train the players for "Electronic Sports". Performing this training Bot in the snake game can lead to a future of training Bots in bigger size games. This can be the future for the efficient gaming and training environment.

REFERENCES

[1] Automated Snake Game Solvers via AI Search Algorithms Shu Kong, 80888472, skong2@uci.edu Joan Aguilar Mayans, 87286425, joana1@uci.edu.

[2] Snake Game AI: Movement Rating Functions and Evolutionary Algorithm-based Optimization Jia-Fong Yeh, Pei-Hsiu Su, Shi-Heng Huang, and Tsung-Che Chiang Department of Computer Science and Information Engineering, National Taiwan Normal University, Taiwan {40247006S, 40247031S, 40247032S}@ntnu.edu.tw, tcchiang@ieee.org.

[3] M. Gallager and A. Ryan, "Learning to play Pac-Man: an evolutionary, rule-based approach," Proceedings of IEEE Congress on Evolutionary Computation, pp. 2462□2469, 2003.

[4] N. Cole, S. J. Louis, and C. Miles, "Using a genetic algorithm to tune first-person shooter bots," Proceedings of IEEE Congress on Evolutionary Computation, pp. 139□145, 2004.

[5] N. Böhm, G. Kókai, and S. Mandl, "An evolution approach to Tetris," Proceedings of The Sixth Metaheuristics International Conference, 2005.

[6] C.-Y. Cheng, Y.-H. Chen, and T.-C. Chiang, "Intelligent level generation for Super Mario using interactive evolutionnary computation," The 27th Annual Conference of the Japanese Society for Artificial Intelligence, 2013.

[7] T. Ehlis, "Application of genetic programming to the snake game," Gamedev.Net, 2000.

[8] A heuristic snake AI controller (written in Chinese), http://blog.csdn.net/fox64194167/article/details/19965069.

[9] LearnSnake: Teaching an AI to play Snake using Reinforcement Learning (Q-Learning), https://italolelis.com/snake.