International Journal of Artificial Intelligence & Robotics (IJAIR) E-ISSN : 2686-6269
Vol.3, No.2, 2021, pp.58-66

Optimization of Breadth-First Search Algorithm for Path Solutions in Mazyin Games

Bonifacius Vicky Indriyono [1]*, Widyatmoko[2]
[1,2]Dian Nuswantoro University, Semarang, Indonesia
[1]bonifaciusvicky@gmail.com*; [2]admoko.widy06@gmail.com
*Corresponding author

ABSTRACT

A game containing elements of artificial intelligence, of course, requires an algorithm in its application. One example of a game that includes elements of artificial intelligence is the Labyrinth game. Maze is a simple educational game. This game is known as finding a way out of the maze to arrive at a predetermined goal. The labyrinth encounters numerous obstacles along the way, such as dead ends and parapets, to reach the target location. In this game, players are required to think logically about how to find the right maze path. The obstacle faced in this game is that sometimes players have difficulty finding a way out, especially if the game level has reached a high level in the process of finding a way out. To solve this problem, a graph tracing technique is needed. The Breadth-First Search (BFS) strategy can be used in conjunction with various graph search algorithms. An example of a broad search method is the Breadth-First Search Algorithm, which works by visiting nodes at level n first before moving on to nodes at level n+1. The advantage of the Breadth-First Search algorithm is that it can find a solution as the shortest path and find the minimum solution if there is more than one solution. This study will discuss how to find a path for the Labyrinth using the BFS algorithm. The result of applying this BFS algorithm is the shortest route solution raised so that the Labyrinth can arrive at the destination point through the route provided.

Keywords : Artificial Intelligence, Graph Search, Maze Game, Algorithm, Breadth-First Search

This is an open-access article under the CC–BY-SA license.

Article History

Received : Nov, 3rd 2021	Revised : Nov, 27th 2021	Accepted : Nov, 28th 2021

I. INTRODUCTION

The rapid growth of technology has a favorable impact on the increase in the number of technological applications in a variety of domains of need, which is a good thing. The increasing use of these technologies has led to new innovative products, both hardware, and software, which are developed based on human needs for a new system. The advent of game application products, particularly game apps that integrate aspects of artificial intelligence, is a basic example of innovation resulting from the application of this technological breakthrough. Games are a software product that is quite popular with many people, ranging from children to adults, due to the variety of ways to play, genres, and visual appearances.

A maze game is an example of a game application that incorporates elements of artificial intelligence due to advancements in information technology. In its most basic definition, a maze is a network of complicated and winding routes. As a sort of simple game, the maze game (also known as a maze) needs players to navigate their way through a maze of obstacles, including dead ends and barrier walls, to reach their destination. This path is made up of skipped boxes for each row or column in the data table. However, the player may encounter difficulties searching for the path because of the large number of roads obstructed by obstructions, resulting in the player arriving at the goal for an extended period.

The problems in the labyrinth game described can be overcome by providing the right path solution and free from obstacles so that it does not take a long time to arrive at the destination. The use of a search algorithm is required in order to develop a path solution that can be employed in a labyrinth game. To determine the quickest path solution for the maze game, the widened search method Breadth First Search (BFS) was used in this study. The result was then tested. In general, the BFS algorithm is used to execute a broad search by visiting the nodes in the network in the order in which they were discovered. It is necessary to use a queue in the BFS algorithm, and this queue serves as a storage area for the nodes that have already been visited. [1].

The BFS algorithm has been widely used in research to provide alternative solutions for the shortest path solution. The BSF algorithm is used for intelligent route planning in the field of industrial robotics; as demonstrated in research [2], the results of this study provide the ideal level of robot paths that may be enhanced efficiently. To discover the ideal path for uncrewed aerial vehicles in another study, researchers evaluated various path solution search techniques, including the BFS algorithm [3]. The BFS algorithm is also used for routing in optical access networks [4]. Based on the literature review results, there is no research that uses the BSF algorithm to create the shortest path solution for the maze game.

The purpose of the discussion in this paper is to understand the rules and processes in the BFS algorithm used in the Labyrinth game program when searching for paths. The findings of this study are expected to bring new insight and understanding into how the rules in the BFS algorithm for pathfinding work and aid in the development of path solutions that may be used in maze gaming software.

II. METHOD

Generally speaking, a research method is defined as a scientific process of obtaining data that has specified aims and applications [5]. The methodologies that were employed in this study will be discussed in detail in this part.

A. *Research Method*

The method used to support the implementation of this research is a case study. The case study method is one type of research that can answer some of the main ideas about a particular event. The case study method allows the investigation of a specific event, situation, or social condition [6].

Fig. 1. Action Research Steps

The general steps in the case study method are shown in Fig. 1 is defining and determining research questions: The first step is to assess research questions. In this initial step, the researcher will determine what goals to be achieved in the research. Researchers will perform data processing using the algorithm used so that research objectives can be achieved. Data collection: At this stage, secondary data collection is carried out in the form of literature studies from journals, proceedings, and books related to the research topic being carried out. Following the determination of data analysis techniques, the data gathered during the data collection process will be analyzed using the techniques determined, with the outcomes of the analysis being consistent with the research topic.

The data collection technique is a technique used to collect data. This technique refers to how its use can be demonstrated through questionnaires, interviews, observations, tests, documentation, etc. The data collection method is a procedure carried out by researchers in collecting research data. Data collection methods are the main thing in research activities because research aims to obtain data [7]. The data collection approach used in this study was secondary, and it consisted of a review of the literature. Literature study takes the form of examining past researchers' scientific work, which may be found in both journals and proceedings where the issues explored are relevant to the current research project. Additionally, to the scientific work of other researchers, a review of books that examined search algorithms, Breadth-First Search algorithms, and Labyrinth games was undertaken as part of this literature study. It is believed that this learning will increase participants' understanding of the topics presented. The data has been gathered from journals and proceedings that discuss the BFS algorithm method, the creation of the Labyrinth board model, and the starting and finishing nodes for the Labyrinth.

This data analysis method responds to the problem that has been identified or defined. The answer is achieved by processing the data that has been collected. It was decided to use qualitative analysis approaches to assess the data that was gathered for this investigation. Qualitative analysis is a process designed to comprehend better the symptoms or difficulties experienced by research subjects, including their behavior, activities, and perceptions of such symptoms or issues. When employing this strategy, the researcher can construct numerous challenges, such as how the Breadth-First Search algorithm can be used to find the graph or the shortest path between the beginning point and the destination point in a maze game.

B. *Research Flow*

The entire series of processes carried out in this research, including all stages of the application design, follow the rules of the waterfall software development method. Fig. 2 shows the stages of the entire series of research processes carried out.

C. *Path Search Flowchart Design*

Path search is carried out with the use of the BFS algorithm as a solution. Numerous steps must be completed to find this path. Fig. 3 depicts a flowchart illustrating all of the phases involved in the pathfinding process. It will be easier for researchers to incorporate this diagram into their applications if they start with this one.

 E-ISSN : 2686-6269

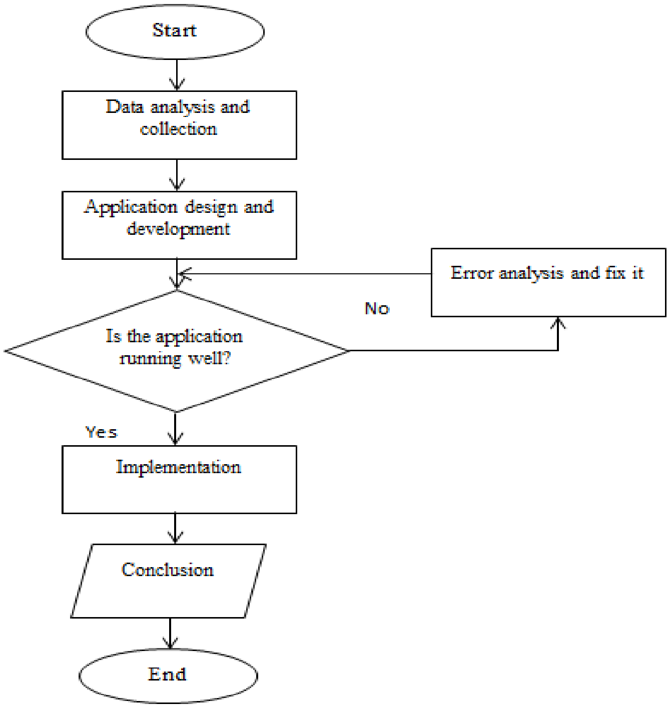

Fig. 2. Research Stages

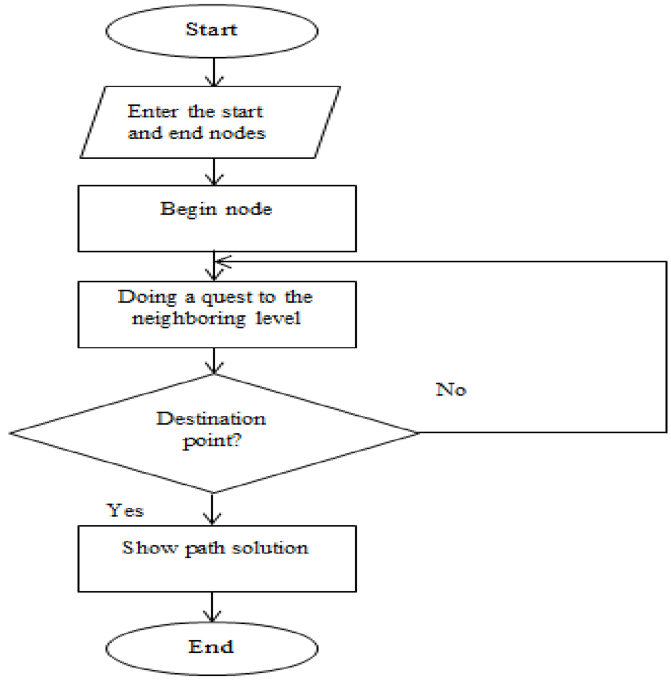

Fig. 3. Pathfinder Process Flowchart

III. RESULT AND DISCUSSION

After the data is collected and the system has been designed, the next stage is to implement software.

A. Maze Games

One form of the instructional game that is easy to understand both in terms of appearance and how to play is the maze game. Although it appears to be simple, this game requires a great deal of thinking to complete. To avoid reaching a dead end, the players must think logically to find the correct way to follow. The interface of this maze game takes the shape of a grid of boxes (columns and rows), the size of which can be customized to suit the player's preferences. A series of branching and winding paths are present in the interface, some of which are inaccessible to players to reach the end goal due to several walls acting as barriers to their progress [8]. The level or level of the game lies in tracing the path that the player must traverse to find the right path to arrive at the predetermined goal [9].

On the surface, this maze game appears to be quite easy, requiring players to just move objects through convoluted paths by using the arrow buttons on their keyboard or using the mouse. Players must be able to direct the object to arrive at the destination. Fig. 4 shows the appearance of the shape of the maze path.

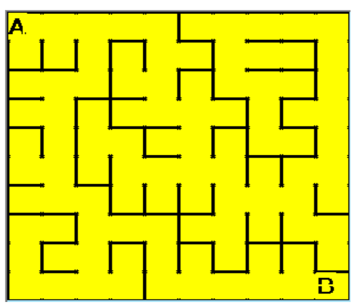

Fig. 4. Labyrinth Path View

B. Artificial Intelligence

Artificial Intelligence (AI) is the intelligence possessed by machines that can act like humans [10]. Artificial intelligence can also be interpreted as an essential part of computer science. The working principle of this AI concept is to make computers able to carry out work like humans themselves [11].

 E-ISSN : 2686-6269

Artificial intelligence is viewed from various perspectives [10], and Fig. 5 depicts the concept of putting artificial intelligence into action. According to an Intelligence point of view, it is envisaged that artificial intelligence would enable computers to perform tasks that would otherwise be performed by humans [10]. When looking at artificial intelligence from a research perspective, it is essential to remember that artificial intelligence is a science and learning about how to make computers perform and ordinary humans [10]. From the standpoint of business, artificial intelligence (AI) Those who hold this point of view believe that artificial intelligence is a powerful tool that can be applied to solve human business issues [10]. Let's look at artificial intelligence from the perspective of programming. We can say that it encompasses the learning of symbolic programming, applying logic in problem-solving, and developing search solution algorithms [10].

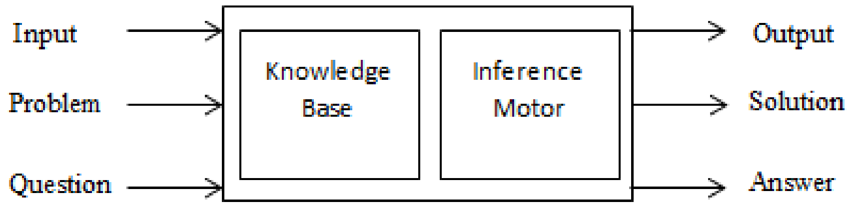

Fig. 5. Artificail Intelligence Concept

C. Breadth-First Search

Breadth-First Search (BFS) is one of the search algorithms that are currently in use. The BFS algorithm operates in such a way that it searches for nodes in a large preorder. This algorithm's search strategy is implemented in practice by traveling from one node to another and finally to all of the nodes that are neighbors to the node that is being searched for. BFS is a graph-searching approach that is used in a wide variety of applications nowadays. The search technique begins with the root node and then proceeds by tracing all of the nodes in the surrounding area [12]-[14]. It is searched for surrounding nodes that each nearest node in the process has not yet checked. This step is repeated indefinitely until the ultimate destination node has been identified and reached. The following are the steps involved in the BFS algorithm in general [15]:

- Adds the root node to a queue of other nodes.
- Take the first node in the queue and determine whether or not it is a solution by checking the node's properties.
- If a node is found to be a solution, the search process is terminated, and the results are displayed.
- If the node is not a solution, the procedure of putting all child nodes into a queue is carried out as a result of the failure.
- Once the queue is empty and all node checks have been completed, the search process is considered to be complete.
- If the queue is not empty, the search procedure is resumed from the beginning, starting at step 2.

Basically, the BFS algorithm is a strategy used in the search process in a graph. There are two main processes, namely visiting a vertex in the graph and checking the vertex. Then see all the nodes that are neighbors to the current node and then check again. Fig. 6 shows an illustration of the search process in the BFS algorithm. There is a graph (G) that has nine vertices (Node A - vertex I).

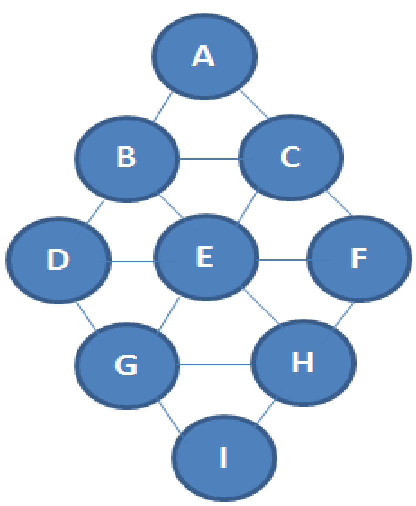

Fig. 6. Illustration of Graph Tracing with BFS

Based on Fig. 6, it can be explained that the tracing process starts from the initial node (A), so that from the process of tracing all nodes, the results can be obtained as shown in Table I.

DOI: 10.25139/ijair.v3i2.4256

E-ISSN : 2686-6269

TABLE I
TRACING PROCESS USING BFS ALGORITHM

Loop to	Node (V=Vertext)	(Q=Queue)	Visited								
			A	B	C	D	E	F	G	H	I
0	A	{A}	T	F	F	F	F	F	F	F	F
1	A	{B,C}	T	T	T	F	F	F	F	F	F
2	B	{C,D,E}	T	T	T	T	T	F	F	F	F
3	C	{D,E,F}	T	T	T	T	T	T	F	F	F
4	D	{E,F,G}	T	T	T	T	T	T	T	F	F
5	E	{F,G,H}	T	T	T	T	T	T	T	T	F
6	F	{G,H}	T	T	T	T	T	T	T	T	F
7	G	{H,I}	T	T	T	T	T	T	T	T	T
8	H	{I}	T	T	T	T	T	T	T	T	T
9	I	{}	T	T	T	T	T	T	T	T	T

Table 1 illustrates how the tracing process uses the BFS algorithm by describing the number of iterations/iterations that are performed. One of the columns in the table has the words "visited," which describes which nodes have been traced. The information is of the boolean type, with T denoting true and F denoting false. It is possible to explain this by noting that the search begins with node A, the 0th iteration. The search process is continued from node A, which subsequently discovered nodes B and C, resulting in the inclusion of the two nodes (B and C) in the queue because they are regarded to have been visited by the search process. This method is repeated indefinitely until all of the nodes have been explored. The flow chart of the graph tracing process with the Breadth-First Search algorithm is shown in Fig. 7.

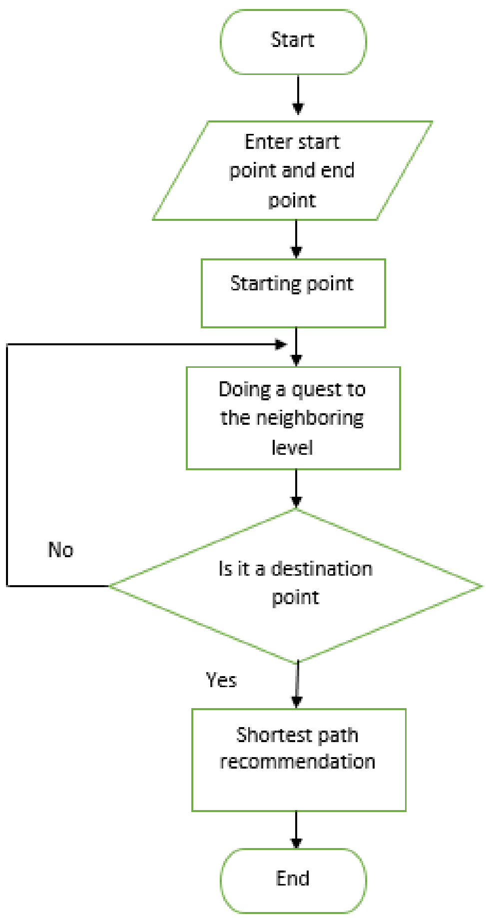

Fig. 7. BFS Algorithm Flowchart

DOI: 10.25139/ijair.v3i2.4256

E-ISSN : 2686-6269

D. Implementation

When all of the information has been gathered, and a system has been built, the next step is to put that system into operation through computer software. With the development of this program, it is possible to ensure that the BFS algorithm can be optimized as a path solution in maze games. The first time you view this game, it will appear to be a representation of Fig. 8.

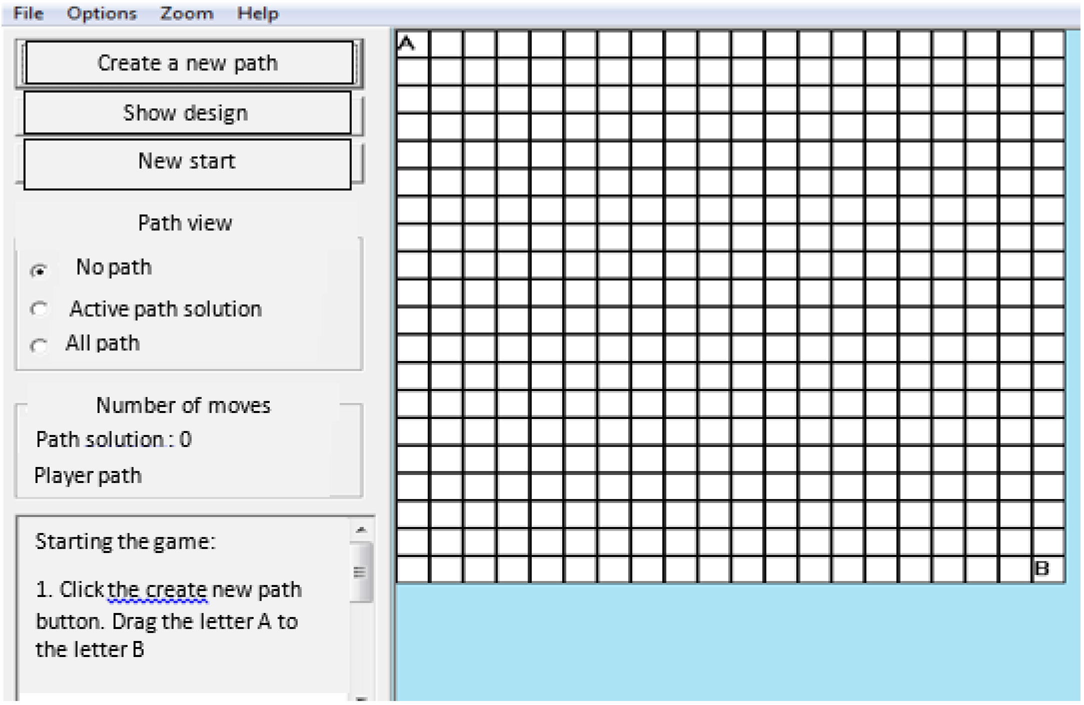

Fig. 8. Initial View of the Application

Furthermore, to set the size of the path consisting of columns and rows, players can make settings through the Options menu. The display of the settings dialog box is shown in Fig. 9.

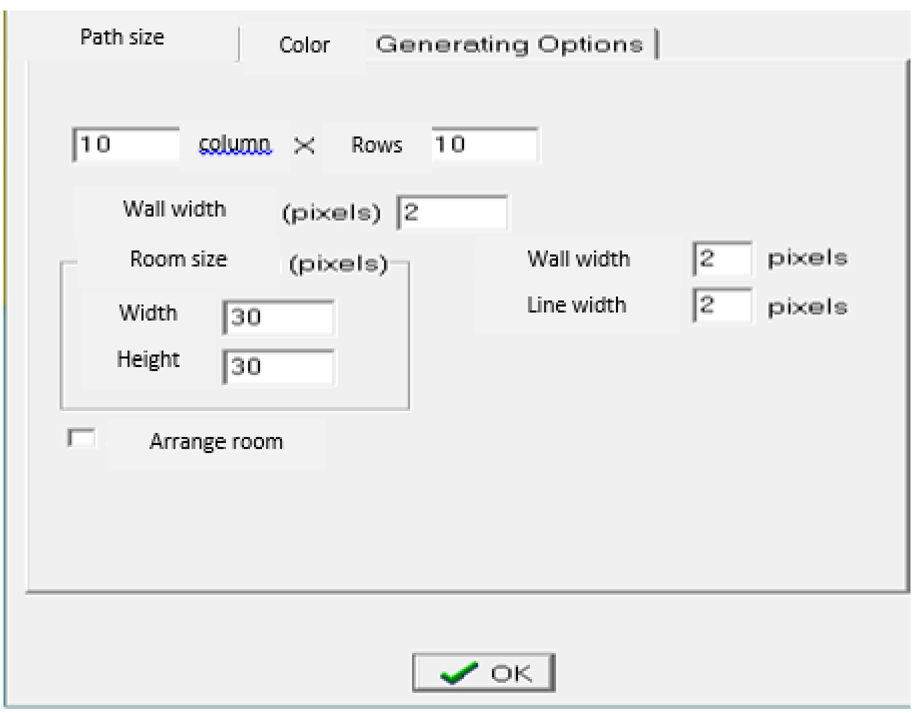

Fig. 9. Settings Dialog Box Display

DOI: 10.25139/ijair.v3i2.4256

 E-ISSN : 2686-6269

The path design consisting of columns and rows is obtained from these settings, as shown in Fig. 10.

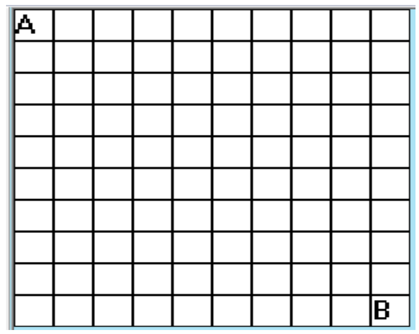

Fig. 10. 10 x 10 Strip Design Display

By pressing the Create New Path button on the path that has already been defined, the design of obstacle paths can be formed on the path that has already been defined. Fig. 11 depicts the design of the obstacle course that players of a predetermined size must navigate through.

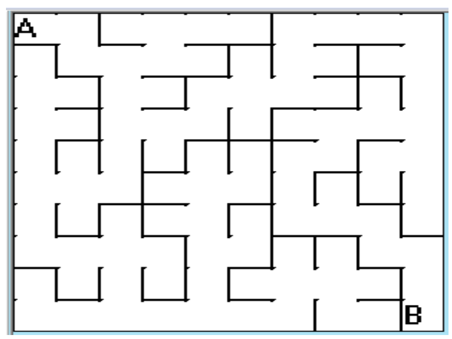

Fig. 11. Obstacle Path View

E. Breadth-First Search Algorithm Analysis

From a path that contains many obstacles, as shown in Fig. 10, players are required to reach the final destination from the starting point (A) to the endpoint (B). To be able to pass through many obstacles, the right path solution is needed. From Fig. 10, to arrive at the endpoint (B) it takes 30 steps of the tracking movement. Fig. 12 shows the resolution of the solution with the BFS algorithm.

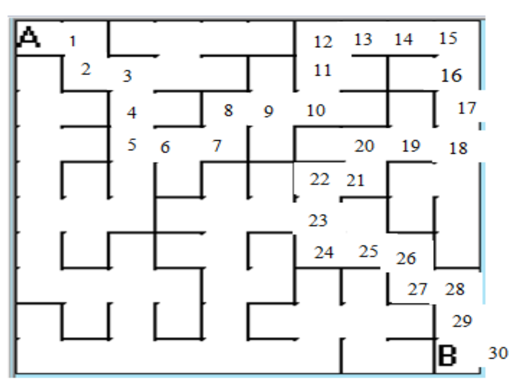

Fig. 12. Path Search Completion Analysis

Based on Fig. 12, it can be explained briefly how tracing the exit path is carried out. Search on all nodes at level N and visit first before visiting the nodes at level N+1. The process starts from the initial node (A) to the end node (B) and then moves to the next level, as well as from right to left until it reaches the destination. Fig. 13 shows the implementation of the results of the analysis of the search process in the maze game application. The path solution can be displayed by pressing the Active Path Solution option from the start menu window.

DOI: 10.25139/ijair.v3i2.4256

 E-ISSN : 2686-6269

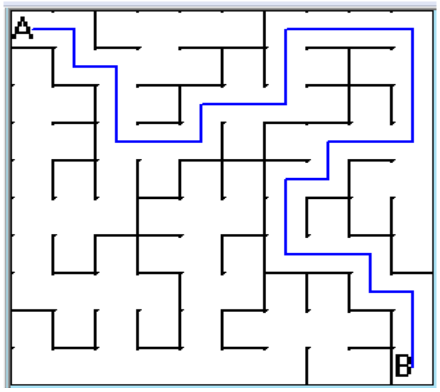

Fig 13. Path Search Completion Results

F. Finite State Machine

The criteria in the Labyrinth Third-Person Shooter Game with the Breadth-First Search Alghoritma can be explained on the Finite State Machine as follow Fig.14

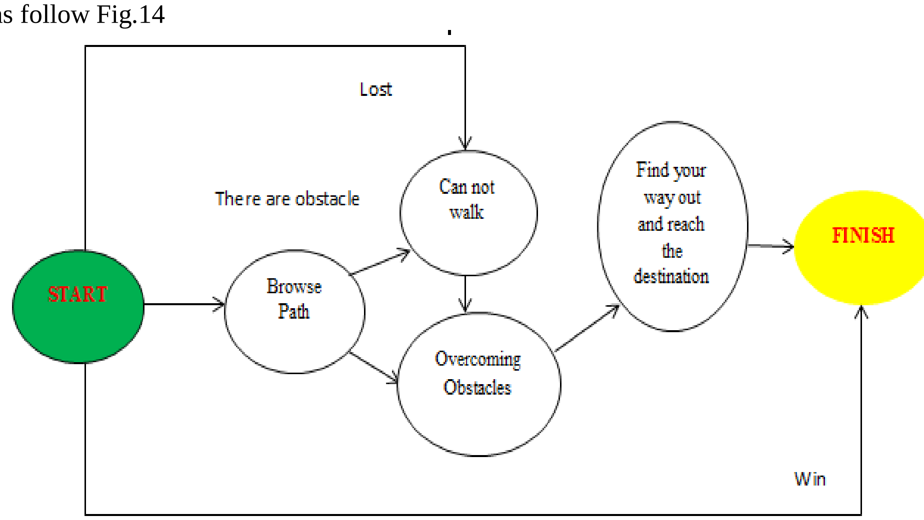

Fig. 14. Finish State

G. Application of the Breadth-First Search algorithm

The Breadth-First Search (BFS) algorithm steps are presented in code 1 to display the BFS algorithm phases of the path solution.

```
Procedure TMaze.DrawSolutionPath;
  var
    i:integer;
    p:Tpoint;
    flag:boolean;
    begin
      with canvas,Pen do
      begin
        if showSolutionpath then flag:=noerase {color:=Solvedpathcolor}
        else flag:=erase{color:=bkcolor};
        style:=solvedpathstyle;
        pen.width:=pathwidth;
      end;
      p:=startpoint;
      for i:=0 to buildmoves.count-1
      do drawonepath(p,getdirfromchar(buildmoves[i][1]),flag);
      invalidate;
    end;
```

IV. CONCLUSION

The Breadth-First Search algorithm can optimally assist in finding the fastest route/path for the Labyrinth to arrive at the endpoint (destination) with various Labyrinth board design models. The A* algorithm uses a graph and applies heuristic functions to expand its state space in solving a problem. The heuristic function is a function that assigns a value to each node that directs A* to get the desired solution. This heuristic function will make the A* algorithm get the solution if there is a solution. In other words, the heuristic is an optimization function that makes the A* algorithm is better than different algorithms. The BFS algorithm has stages of determining all paths to find a solution to another possible node step by step. The BFS algorithm begins its search from the side where the data is stored. The search for a solution will continue when one route has been completed until a solution is discovered. Next, the path produces a solution that can be compared to how fast and effective the path is for the exit for the maze. This algorithm will search all routes in the maze until it reaches the point where solutions are in various ways. The result can be determined the best route to reach the destination.

REFERENCES

[1] D. Zai, H. Budiati and S.B. Berutu, "Simulation of the Shortest Route to Tourism Locations in Nias Using the Breadth-First Search and Tabu Search Methods", InFact Journal ,Vol. 1 No. 2, Nov. 2016.
[2] B. Fu et al., "An improved A* algorithm for the industrial robot path planning with high success rate and short length," Rob. Auton. Syst., vol. 106, pp. 26–37, 2018.
[3] S. K. Debnath et al., "A review on graph search algorithms for optimal energy efficient path planning for an unmanned air vehicle," Indones. J. Electr. Eng. Comput. Sci., vol. 15, no. 2, pp. 743–750, 2019.
[4] G. L. Andrade and D. H. Thomas, "An Optimized Breadth-First Search Algorithm for Routing in Optical Access Networks," IEEE Lat. Am. Trans., vol. 17, no. 7, pp. 1088–1095, 2019.
[5] Sugiyono, Educational Research Methods with Quantitative Approaches, Qualitative, and R & D. Bandung: Alfabeta. 2013.
[6] D.J. Hodgetts and E.E. Stolte," Case-based Research in Community and Social Psychology: Introduction to the Special Issue", Journal of Community & Applied Social Psychology, Vol. 22 No. 5, p. 379-389, Aug. 2012.
[7] S. Arikunto, Research Procedure A practical approach. Jakarta: Rineka Cipta. 2010.
[8] I.,G.,S. Astawa, April, 2012. "The Use of Advanced Tracing Intelligence Methods in Solving Labyrinth Game Problems". Journal of Computer Science. Vol 5 No 1. ISSN Print 1979-5661. ISSN Online 2622-321X
[9]] R.B. Sirait, Desember, 2013. "Labyrinth Game Application Design Using Backtracking Algorithm". Pelita Informatics Budi Darma. Vol. V. No. 2. ISSN : 2301-9425.
[10] Billy; I. Kuswardayan ; W.N., Khotimah, "Implementation of Artificial Intelligence in Game Defender of Metal City by Using Finite State Machine". Journal Of Engineering Pomits. Vol. 6, No. 2. ISSN- Online 2337-3520. ISSN Print 2301-928X.
[11] E. Wijaya,,2013. "Analysis of the Use of the Breadth First Search Algorithm in the Concept of Artificial Intelligence". TIME Journal, Vol. II No 2. P 18-26, 2013. ISSN 2337 – 3601.
[12] R. Zhou and E. A. Hansen, "Breadth-first heuristic search," Artif. Intell., vol. 170, no. 4–5, pp. 385–408, 2006.
[13] D. C. Kozen, "Lecture 4. Depth-First and Breadth-First Search," in The Design and Analysis of Algorithms, Springer, New York, NY, 2011.
[14] F. Zhang et al., "An adaptive breadth-first search algorithm on integrated architectures," J. Supercomput., vol. 74, no. 11, pp. 6135–6155, 2018.
[15] A. N. Putri, "Search the blind breadth first search algorithm in 3d game engine maze third person shooter android based on intelligent agent". Transformatika Journal, Vol. 14, No. 1, Jul. 2016.