

Implementation of the Breadth First Search Algorithm in the Darkness Maze Game Based on Desktop Intelligent Agent

Ramadhani Suciansyah*
 STMIK Widya Cipta Dharma,
 Samarinda, 75123, Indonesia
 Vjdanny19@gmail.com
 *Corresponding author

Bartholomius Harpad 
 STMIK Widya Cipta Dharma,
 Samarinda, 75123, Indonesia
 bartolomius@wicida.ac.id

Hanifah Ekawati 
 STMIK Widya Cipta Dharma,
 Samarinda, 75123, Indonesia
 hanifah@wicida.ac.id

Received: 2021-03-01; Revised: 2021-05-06; Accepted: 2021-06-19; Published: 2021-09-01

Abstract- This research was developed with a multimedia development stage consisting of 6 stages, namely Concept, Design, Material Collecting, Assembly, Testing and Distribution. Analysis of system design using UML (Unified Modeling Language) which consists of Use Case Diagrams, Activity Diagrams and Sequence Diagrams. In the development of this 3D game using the Unity Engine as software designed for the creation and development of video games assisted by typing some coding programs. The test method uses Black-Box testing and Beta Testing. Testing and Distribution (Distribution). Analysis of system design using UML (Unified Modeling Language) which consists of Use Case Diagrams, Activity Diagrams and Sequence Diagrams. In developing this 3D game using the Unity Engine as software designed for the creation and development of video games assisted by typing some coding programs. The test method uses Black-Box testing and Beta Testing. Testing and Distribution . Analysis of system design using UML (Unified Modeling Language) which consists of Use Case Diagrams, Activity Diagrams and Sequence Diagrams. In developing this 3D game using the Unity Engine as software designed for the creation and development of video games assisted by typing some coding programs. The test method uses Black-Box testing and Beta Testing. The final results of this study can be applied with the help of typing coding so that the path points that have been created can provide information about adjacent paths so that they can find an exit path.

Keywords- Breadth First Search Algorithm, Intelligent Agent, Desktop Based.

I. INTRODUCTION

Games are very popular in society, from children to adults as a form of entertainment that supports social interaction (Anggara, 2010). With the development of information and communication technology as well as sophisticated computers and hardware, it has shown the potential to make games on the computer very good. One type of game that is very popular in all circles of society is the Third Person Shooter maze (abbreviated as TPS) in this Unity 3D Game which contains a game centered on

weapons and gun-fighting projects from a third-person perspective. In a game (Tarmiandi, 2016).

The maze game is a game to find a way out, from several paths in a game area. (Hong, 2010) The game template is a square or rectangle whose size can be adjusted according to the user's wishes. (Hendi. 2015) In the maze game This is a game that looks very simply but is able to solve a logic puzzle more easily which completes it and its appearance from a third person perspective or the third person shooter genre (Roedavan, 2014; Samuel, 2010). This game has 3 levels (Longleat Maze, Masano Labyrinth, Villa Pisani la Labyrinth) where the higher the level of this stage, the more difficult the stage level will be (Hakim, 2012)

Games in general use Artificial Intelligence or artificial intelligence in the system to make the game more interesting to play. Artificial Intelligence is applied to the game in the form of an enemy or opponent's non-player character (NPC) intelligent agent that must be faced in the game, one of the applications of artificial intelligence in games for smart decision making is the first wide search (Breadth First Search) which is the method This search begins by entering the initial node or root node (node 0) to the next node. This search is carried out to find a solution for all nodes that have the same level to determine the final result (Goal) at that level. If, node or node is not a solution path, the search will continue to repeat until it finds a solution path.

Based on this background, the theme of this research was taken with the title "Implementation of the Breadth First Search Algorithm in the game" The Darkness Maze "with the Genre of Third Person Shooter Using Unity 3D based on Intelligent Desktop Agents.

II. LITERATURE REVIEW

A. Game

According to Henry (2010), a game or game is something that can be played with certain rules so that some win and some lose, usually in the context of not being serious with the aim of refreshing. Playing games can already be said to be the lifestyle of today's society.

Starting from the age of children to adults also like video games. It's all because playing video games is fun.

B. Third Person Shooter

According to Henry (2010) Third Person Shooter is a game that is almost the same as First Person Shooter, only in a third person perspective where we can see the body of the character we are playing.

C. Game Engine Unity 3D

According to Pranata, et al, (2015), Unity 3D is a game engine developed by Unity Technologies. This software was first launched in 2005 and is one of the many game engines used by many professional game developers in the world (pressman, 2010). unity 3d is a game development tool with integrated rendering capabilities (shalahuddin, 2011; siswanto, 2010) by using its sophisticated features and high working speed. unity 3d can create an interactive program not only in 2 dimensions, but also in 3 dimensions. (Putri. 2016)

D. Blender 3D

According to Hendratman (2015), Blender is an OSS (Open Source Software) or other term software that can be used in various OS (Operating Systems). It used to be developed commercially and globally, but is now released under the GPL (GNU General Public License). The features included in this Blender software include Photorealistic Rendering, Modeling, Realistic Materials, Rigging, Animation, Sculpting, UV Unwrapping, UV Mapping, Compositor, Simulations, Game Creation, Camera and Object Tracking, Library of Extensions and Flexible Interface.

E. Artificial Intelligence

According to Russel and Norvig (2010), in the English meaning, artificial means artificial, while intelligence means an adjective that means intelligent. So, artificial intelligence is an intelligence that is created or an intelligent imitation. Intelligent intent here is the intelligence to think and act like humans.

F. Breadth First Algorithm

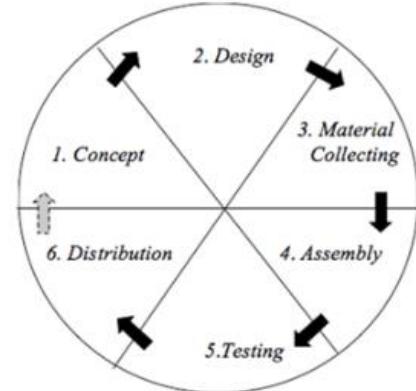
According to Suyanto (2014), Breadth First Search (BFS) is a search carried out on all nodes in each level sequentially from left to right. If no solution is found at one level, then the search is continued at the next level. And so on until a solution is found. With this strategy, it can be guaranteed that the solution found is the best (Optimal). But Breadth First Search (BFS) must store all nodes that have been generated. This should be done for tracing back when a solution has been found.

III. RESEARCH METHODS

According to Nugroho, (2011), multimedia is a combination of several elements (audio, animation, text, video images, graphics, sound) that can create a dynamic and interactive presentation and have tools for navigation

for users and can be online. (internet) and offline. Multimedia in this case can also provide more text but also bring text to life by including sounds, images, music, animation and video.

According to Binanto (2010), the multimedia development methodology consists of six stages, namely concept, design, material collecting, assembly, testing and distribution.



Picture 1. Multimedia Development Stage

A. Concept

The concept stage (filling) is the stage for determining the objectives and who is the user of the program (audience identification). The goals and end users of the program affect the multimedia nuance as a reflection of the identity of the organization that wants information to reach the end user. User characteristics including user capabilities also need to be considered because they can influence design creation. In addition, this stage will also determine the type of application (presentation, interactive, etc.) and application objectives (entertainment, training, learning and others). The basic rules for the design are also determined at this stage, for example the size of the application, targets and others. The output from this stage is usually a narrative document to reveal the project objectives to be achieved.

B. Design

Design (designing) is the stage of making specifications regarding the program architecture, style, appearance and material / material requirements for the program. Specifications are made as detailed as possible so that at the next stage, namely collecting and assembly materials, new decision making is no longer needed, it is enough to use the decisions that have been determined at this stage. However, in practice, project work at an early stage will often experience additional materials or reduction of application parts or other changes.

C. Material Collecting

Material collecting is the stage of collecting the components needed in the development of control structure learning applications on the concept of multimedia-based algorithms designed in the form of images, text, and sound according to the needs that are done. These materials include clip art images, photos, animation, video, audio and others which can be obtained

free of charge or by ordering to other parties according to the design. This stage can be done in parallel with the assembly stage. However, in some cases, the material collecting and assembly stages will be carried out linearly and not parallel.

D. Assembly

The assembly stage is the stage of making all multimedia objects or materials. Making applications at the design stage, such as storyboards, sections of the flow and / or navigation structures.

E. Testing

The testing phase is carried out after completing the assembly stage by running the application / program and seeing whether there is an error or not. In this study, testing using black-box and -beta-testing.

F. Distribution

The stage of multiplying and distributing the results to users. Multimedia needs to be packaged properly in accordance with its distribution media, whether through CD / DVD, download, flash or other media.

IV. RESULTS AND DISCUSSION

A. Implementation of the Breadth First Search Algorithm on the Point Line

At each point the point path which is implemented using the Breadth First Search Algorithm has its own point path (node) so that when the player's character stands right on a point path, the player gets information from that point by means of the system displaying an option menu (queue status)) the path closest to the player character's point path, so that the player can choose the path that he wants to pass. When the player character selects the closest path, information such as navigation will appear. navigation itself functions as directions so that the player's character does not find dead ends making it easier to find an exit (exit node).

To see clearly Picture 2. implementation of the breadth first search algorithm in the gameplay of this game used a map sketch taken from the longleaf maze map (Level1).



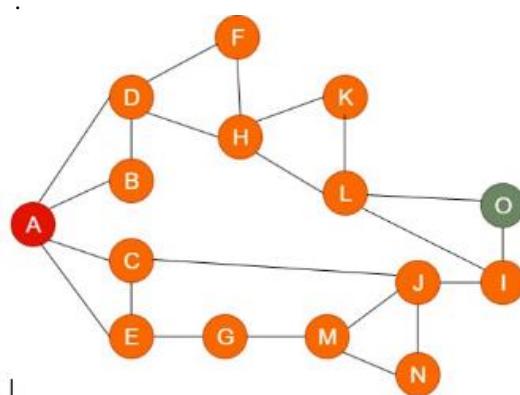
Picture 2. Gameplay stands on a point track

As for Picture 3. The Sketch Level Stage 1 is converted into a breadth first search algorithm tree to make it easier to find the output results so it is easier to find a solution path.



Picture 3. Sketch Stage 1 Level

To find the most optimal path or output from a BFS Algorithm Tree at Stage Level 1 using the breadth first search algorithm is to use the basic work steps of the Breadth First Search Algorithm and the queues it uses as shown in Picture 4.



Picture 4.BFS Algorithm Tree on Stage Level 1

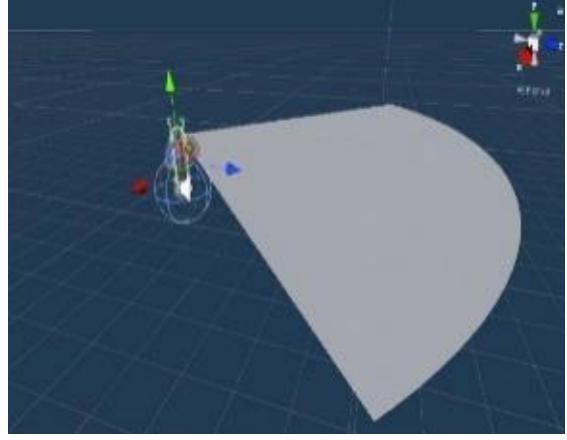
To find the most optimal path or the following steps for the Breadth First Search Algorithm:

1. Put the end (root) node into the queue.
2. Take a node from the start of the queue, then check if it is a solution.
3. If the node is a solution, the search is completed and results are returned.
4. If node is not a solution, enter all neighboring nodes (child nodes) into the queue
5. If the queue is empty and every node has been checked, the search is complete and returns no results.
6. Repeat the search from step two.

So, enter the initial root node into the queue, take node A and put it in the queue then check if node A is a solution node if node A is a solution then node search is complete and if node A is not a solution then enter it into the queue list (queue status) show node which are adjacent or neighboring vertices to node A, namely node D, node B, node C and node E. because, node A is not a solution, then Node A is Output Only and enter node B, node C, node D and node E in the queue list. Repeat the search until the node is a solution. So, we get some of the most optimal outputs or paths from sketch level 1, namely nodes:

1. ADFHKLO
2. ADFHLO
3. ADHKLO
4. ADHLO
5. ABDHKLO
6. ABDHLO
7. ACEJIO
8. ACEGMNJI
9. AEGMJIO
10. AEGMNJI

It can be concluded that the application of the Breadth First Search Algorithm at sketch level 1 gets the 10 most optimal lines. At this stage, it explains the writing of coding so that the algorithm used can be applied to the game that has been created as shown in Picture 5.



Picture 6. Enemy Radius

```
Navigation.cs  X
Assembly-CSharp  Navigation
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  [SerializeField]
5  {
6      public GameObject Informasi;
7      public GameObject DeletedNav;
8      public GameObject DeletedNavi;
9      public GameObject DeletedNav2;
10     public GameObject DeletedNav3;
11
12     void Start()
13     {
14         Cursor.visible = false;
15         Cursor.lockState = CursorLockMode.Locked;
16     }
17     void OnTriggerEnter(Collider other)
18     {
19         DeletedNav.SetActive(false);
20         DeletedNavi.SetActive(false);
21         DeletedNav2.SetActive(false);
22         DeletedNav3.SetActive(false);
23         Cursor.visible = true;
24         Cursor.lockState = CursorLockMode.None;
25     }
26     void OnTriggerStay(Collider other)
27     {
28         Informasi.SetActive(true);
29         Cursor.visible = true;
30         Cursor.lockState = CursorLockMode.None;
31     }
}
```

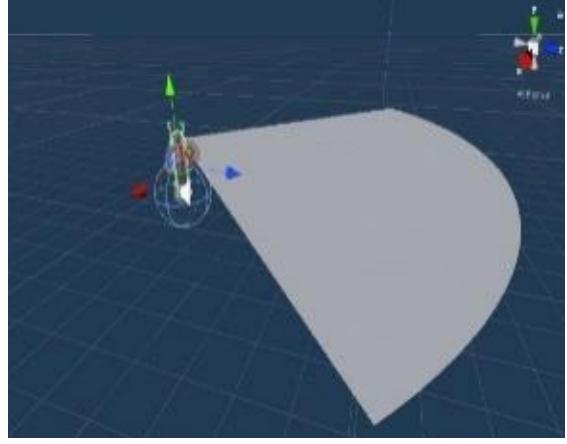
Picture 5. Coding Algorithm BFS

B. Implementation of Intelligent Agents on the Enemy

On an enemy that has been implemented with an intelligent agent that is anything that can sense its environment through sensory equipment, acts in accordance with its environment and by using its actuator. When the actuator or player enters the enemy's radius, the enemy will chase and attack the player.

In the enemy radius movement in Picture 6, when the player enters the enemy radius, the enemy character

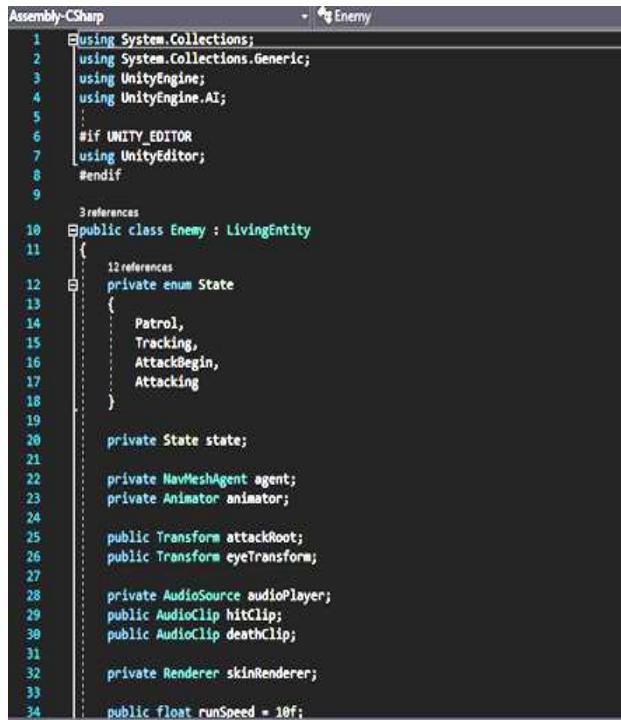
sensor will move and the enemy sensor character has hearing where when the player walks behind or in the enemy character's blind spot, the enemy character will immediately chase and try to attack the player.



Picture 6. Enemy Radius

The Enemy Radius object is created using a 3-dimensional plane object then transformed after that adjust the coordinates of the x axis, y axis and z axis so that it forms a 45-degree axis and adds typing coding so that the enemy character can move through sensors triggered by the surrounding conditions of the character enemy.

coding enemy in Picture 7, is coding to make the enemy do what it should be, such as idle movement (the enemy is stationary), run movement (the enemy is running), attack movement (the enemy attacks) and movement is edited (the enemy is dead) and coding to make the enemy have sensors, which act according to their environment and by means of actuators, propulsion devices or actuators themselves, which are the enemy's radius and hearing when hearing footsteps or hearing gunshots. When the sensors are active, the sensors will send a signal to the actuator or the player character enters the enemy's radius so that the enemy character will immediately chase and attack the player character. In this coding the enemy character also gives 40 hit points to the player character,



```

Assembly-CSharp
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.AI;
5
6  #if UNITY_EDITOR
7  using UnityEditor;
8  #endif
9
10 3 references
11  public class Enemy : LivingEntity
12  {
13      12 references
14      private enum State
15      {
16          Patrol,
17          Tracking,
18          AttackBegin,
19          Attacking
20      }
21
22      private State state;
23
24      private NavMeshAgent agent;
25      private Animator animator;
26
27      public Transform attackRoot;
28      public Transform eyeTransform;
29
30      private AudioSource audioPlayer;
31      public AudioClip hitClip;
32      public AudioClip deathClip;
33
34      private Renderer skinRenderer;
35
36      public float runSpeed = 10f;

```

Picture 7. Coding enemy

Russel.S. and Norvig. P. 2010. Artificial Intelligence: A Modern Approach. New Jersey: Alan Apt.

Pranata, et al. 2015. Easy to make games and their financial potential with Unity 3D.

Pressman, RS 2010. Software Engineering: a practitioner's approach. McGraw-Hill. New York.

Daughter, AN. 2016. Optimization of Breadth First Search Algorithm in a 3D Third Person Shooter Maze Game Engine Based on an Android Smart Agent.

Roedavan, R. 2014. Unity Game Engine Tutorial. Informatics Publishers. Bandung.

Samuel, H. 2010. Smart with Games. Jakarta: Gramedia.

Shalahuddin, M. 2011. Structured and Object Oriented Software Engineering. Bandung: Informatics.

Siswanto. 2010. Artificial Intelligence. Yogyakarta: Graha Science.

Suyanto. 2010. Artificial Intelligence Searching, Reasoning, Planning, and Learnig. Bandung: Informatics Publishers.

Tarmiandi ,. 2016. Implementation of the Breadth First Search (BFS) Algorithm in the Search for the Shortest Route of Boarding House in Central Semarang

Yudhanto, PA 2010. Designing Edu-Games Product Promotion through Events. Final report. Indonesian Computer University Bandung.

V. CONCLUSION

Can be used as a reference for making games using Unity 3D, especially for game developers of Indonesia's younger generation. The result is a game the darkness maze with the genre of third person shooter (TPS) using the breadth first search algorithm to find a solution or patch finder which functions as a direction and creates a non-player character that has a sensor and actuator (smart agent) that can feel where the player is located through the environment around him. Based on the results of beta testing, in general, the games made get a percentage value of 82.5%. Beta testing shows the level of satisfaction of friends who have been willing to become respondents to this game and the Black-Box test that is carried out shows that each button in the game is running well.

REFERENCES

Anggara, FD. 2010. Implementation of the Breadth First Search (BFS) Algorithm in word word games

Binanto, I. 2010. Digital Multimedia Theory and Its Development. Yogyakarta: Andi.

Judge, Z. 2012. "The History of Adobe Photoshop". Jakarta: PT. Gramedia General Library.

Hendi, H. 2015. The Magice Of Blender 3D Modeling. Bandung: Informatics.

Hong, B. 2010. Breadth First Search (Flowchart). Retrieved (26 September).

Nugroho, A. 2010. Object-Based Software Engineering using the USDP Method. Andi. Yogyakarta.