A* PATHFINDING APPLICATIONS IN TWO-DIMENSIONAL AI VIDEO GAMES

Tomi Tri Sujaka, Kurniadin Abd Latif, Sirojul Hadi, Hasbullah, Rifqy Hammad
Department of Computer Science, Faculty of Engineering, Bumigora University
tomi_tri@universitasbumigora.ac.id; kurniadin@universitasbumigora.ac.id
sirojulhadi@universitasbumigora.ac.id; hasbullah@universitasbumigora.ac.id
rifqi.hammad@universitasbumigora.ac.id

Abstract – A widely varieties of game genre already exist in the world, one of them is a platform game-type, this type of game in which players are required to jump between the Platform, through various obstacle, so that players can survive and continue the game to the next level. This type of games has a wide variety of game mechanics such as jumping higher with the help of an object such as trampoline, beat or slay the enemy AI (Artificial Intelligent enemy) using a weapon or Environment that exist in the game, The Development of the game with fast phase in-game movement speed hence the player reflect will be tested, this game is 2D based, which will be made using Unity and the assets that available at Unity Store and some are creator made.

Keywords: A*, pathfinding, game, unity, asset.

I. INTRODUCTION

Along with the growth and development of technology both in terms of the hardware and software, the development of games is also growing rapidly both in terms of entertainment and business, therefore today's games have extraordinary prospects, games with PC and Mobile platforms are widely played and loved by children. as well as adults, the genre of those game is widely diverse, starting from strategy to action/adventure genre games, and of course the way the game is played is different, one of which is single player offline or online multiplayer. The Six-Step Commutation method is a hall sensor reading method on a Brushless Direct Current motor. This method determines the timing of the correct and appropriate reading on the hall sensor. Hall sensor is a sensor used to detect magnetic fields. The working principle of the hall sensor on a Brushless Direct Current motor is that when the south or north magnetic pole approaches the hall sensor.

With well-developed in terms of video games and game engines, a lot of game engines may have appeared with different features such as Unreal Engine4(U4), Frostbite – Game Engine, Unity 3d, Source Engine, RPG Maker, and any other video games engines available in the market/web, but Unity emerged as one of the popular game engines when it comes to video game developers, both for professionals and indie developers, hence the author is interesting to make games using the software.

A* Pathfinding (A Star Pathfinding) is a computer algorithm which function can be utilized to find / reach the desired path, A* Pathfinding was also appearing as one of the assets in Unity, which will later be utilized.

II. METHOD

2.1 The Basic Pathfinding

When Building Pathfinding based game systems, the main concern of the video game developer is accuracy and efficiency, (but when it comes to both of this parameter it depends on the concept of the ai, such as if the developer want the enemy to be sluggish the code and parameter has to be tuned to less smart and underperform but if the developer want the ai to be précised and sharp then the parameter and code must be tuned to its peak and as precise as possible), this requires a lot of optimization at some point. Pathfinding is divided into 3 components:

1. Spatial Representation, such as a Graph, is a depiction of walkable zones (accessible zones) that are interconnected (for instance roads, floors, earth etc.)
2. Goal estimation, also known as heuristic, is the rendezvous point where the desired goal may be achieved.
3. The agent is the one who's responsible for finding the searched object based on the available heuristics.

2.2 The Grid

The basic method and easy to deploy and understand, as well as frequently utilized on how to explain the A* pathfinding algorithm works (this method was utilized in the making of this video game). A uniforms of grid graph is applied over the desired applicable area, creating net-like nodes (grey and white in the image below, the gray block is a barrier). Quite effective and seems simple, but also will use a moderate number of memories. which is a problem that will be experienced when searching between objects a to object, we must search through all nodes from mid to edge of the applicable grid (lots of additional searches), Here is an example Grid image of A*(A Star) Pathfinding as shown at Figure 1.

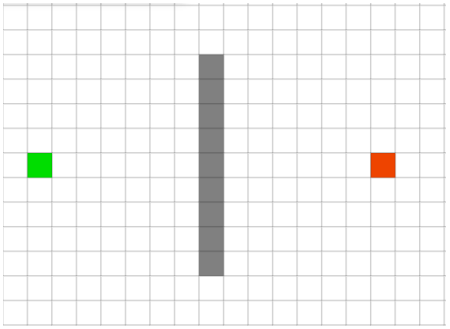

Figure 1. A*Pathfinding illustration

2.3 Game Design Document

Game Design Document is s process of scheming or designing video game both content and rules at the precreation stage. And designing the game layout, environment, storyline, and characters. During the creation stage. At the game development (creation stage), there is a document that contains the entire description of the game to be designed. A game design document or Game Design Document (GDD) is a description of how a video game in general will be created/finished. Every detail in the game must be mentioned in it. If it's not in this document, it shouldn't be in the game. The problem is, there is no general game design document (GDD) that fits all game genres, such as games like counter strike that has no storyline, just online multiplayer mode.

2.4 A* Pathfinding

A* pathfinding is a graph-type search of algorithm that were able to find a path with utilizing the least amount of memory from a starting point to the desired destination point (from 1 or many destinations). The A* algorithm is the result of the development of basic breadth first search method, this method evaluates each point by combining g(n), the number of paths taken from starting point to the desired destination, and h(n), is a heuristic function that considers the shortest path from start n to destination.

1. A is the node that is currently running in the search for the shortest search algorithm.
2. Nodes are small dots as a representation of the pathfinding area, the shape can be a triangle, a square or a circle.
3. Open list is a place to save node data that could be accessed through the starting point or node that is currently running.
4. Closed list is a place to store node data before A, which is also part of the shortest path that has been successfully fetched or applied.
5. Travel path (f) is the value gained from the sum of the g values, which is the sum of the values of each node in the shortest path from the starting point to A, and H is the number of approximate values from a node to the destination node, so that it can be formulated, as seen below:

   $$f(x) = g(x) + h(x) \quad (1)$$
6. The destination node is the final node, that has been set as the goal.
7. Obstacle is an attribute that act as a barrier therefore a node cannot be traversed by A.

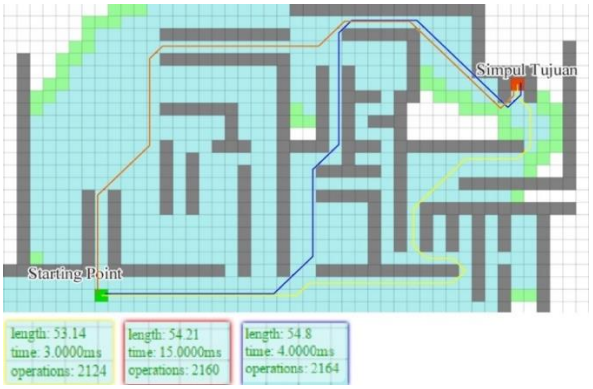

Figure 2. A* Pathfinding Enumeration Result

Figure 2 shows Starting point nodes marked with the green color main job is to look for the red Destination Node by avoiding existing obstacles by accessing the Open List, specifically passing through nodes that do not have obstacles and accessing the Closed List The shortest path from each node. The Pictures above indicate that, it can be concluded that the Yellow Line looks for the fastest path to the destination node by taking 3.0000ms per node and at the same time calculating 2124 computing operations on the map while looking for the destination node, blue come second, with 4.0000ms, and red took the longest with 15.0000ms.

2.5 High Concept Document

The development of this game is to aims as an entertainment in term of video games and the introduction of the A* Pathfinding algorithm which is an interesting algorithm implemented in a video game, the main reason was the features provided and the user interface is quite friendly for novice developers, A* Pathfinding appears as a Plug-in (3rd party software) for Unity.

2.6 *Game Treatment Document*

The design of the video game aims is also to provide process provisions for the software that will be made so that the making of the application does not deviate from the rules and analysis results that have been applied to the design of this game (game design documents).

2.7 *Game System*

The game system section discusses the preparations game rules in the game, the game rules apply to both enemies and players, in general the game rules of this game are as seen bellows:

a. This game is played as a single player.
b. The gameplay of this game is played wave by wave which means the game has different enemies by each wave and different stats.
c. Live Points is available as a counter measure of how many chances the player must play. If the Health Points (HP) character of the player touches the number / amount 0 on the last Life Points, which mean the player will be given the option to retry (repeat the game starting from wave 1) or game over (exit the game).

Table 1. SWOT

	Strength	*Weakness*
	based on Genre Video Game that are easy and also had relatively low required hardware specification	Is a single player video game, the *game* come in 2-dimensional state, and also this video *game* is an offline game
Opportunity The target of this video game platform is an Android-type device, which is the most widely used mobile device at this time.	S – O Strategy: The game has low hardware specifications, older mobile devices with adequate hardware specification could run this game flawlessly such as Samsung Galaxy IV mini	W – O Strategy: Mobile device that doesn't equip with discrete GPU could run this game with just integrated GPU that also come with CPU
Threat There are a lot of developers for mobile devices, which makes the competition for video game in general even tougher.	S – T Strategy: the game is quite lightweight if integrated with another software that could make this directly played from the cloud without taking longer to load, the game data/assets.	W – T Strategy: collaborated with another existed developer that could develop multiplayer mode and online features integrated with this game.

III. RESULT AND DISCUSSION

The launching stage (both beta) is the stage where the application or game has been used by the end user (beta). Before it can be used properly by the user (golden stage), the system must go through testing phase to ensure that there is no fatal error that arise when the user put to use the application or game.

3.1 Making Game Assets

The video Game assets is manually made for every single part (such as prefab, game object, etc.) to keep the originality intact in the video game, a couple of software were utilized to create these assets, the following of which can be found below:

a. Adobe Photoshop
b. Adobe Audition
c. Unity Game Engine

3.2 Making Main Character

Adobe Photoshop CS 5 is utilized to create and manage character as can be seen below:

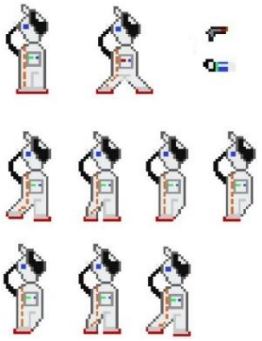

Figure 3. Creating Main Character

3.3 Making Background, Platform and Rigid Body Application

To create and processing game object such as background assets, main character, and platform, ai, etc. Adobe Photoshop CS5 were utilized, then the rigid body is applied later in unity as shown in Figure 4.

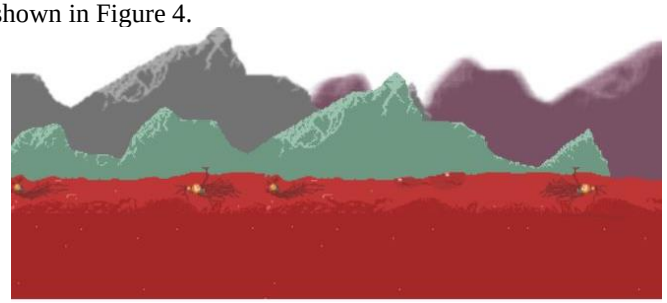

Figure 4. Making Video Game Background

3.4 Sound and Voice Recording

Assets for audio processing are made using adobe audition, which had previously been recorded using a mic, there are several audio files in it, The illustration can be show at Figure 5, including:

a. Shooter character.
b. Footstep sound of the main character
c. Respawn sound
d. Gun fire and attack from the enemy
e. Background music
f. The voice of the main character when defeated

To keep the originality intact the assets were made using our own voice, free audio from the internet and daily activity sound that previously has been recorded then.

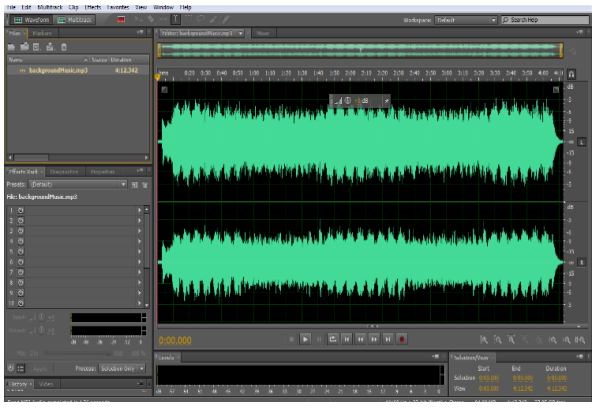

Figure 5. Audio Editing

3.4 Making Sprite

Sprite creation is a converting process of file type from original images file type (whether jpg or png) into sprites in Unity, all game sprites are done in the Unity Sprite Editor, sprites are utilized when the type of game is in 2D mode and has different sequences of images included such as characters in stationary sequence, running and jumping sequence therefore after every images are inputted in order, the images going to produce the craved animation.

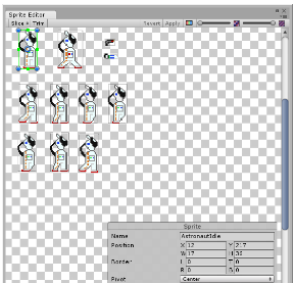

Figure 6. Image Conversion to Sprite File (Unity File)

3.5 Application of A* Pathfinding on AI

A* Pathfinding is a third-party asset that could work as plugin in unity which can be downloaded from http://arongrenberg.com/astar/download, the Inspector panel of A* Pathfinding in unity look like, the image below:

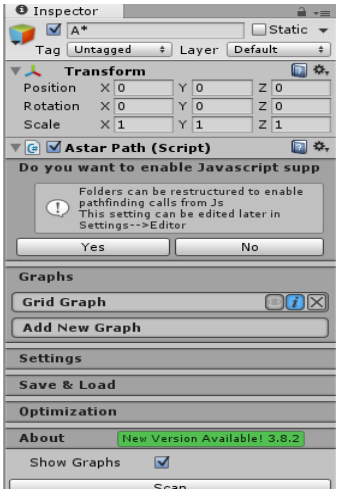

Figure 7. Application of A* Pathfinding

A* Pathfinding is applied to AI, that previously was made using Adobe Photoshop and converted into sprites and then given the required and desired parameters such as respawn position, weapon attributes, damage given to the main character, movement speed and health points for the AI, then there are features such whether the AI shoot or hit the main character as their main source of weapon.

After A* Pathfinding is applied, a selection of A* gizmos for Unity will be available, this will make it easier for the user to see the movement performed by the object that A* Pathfinding is applied to.

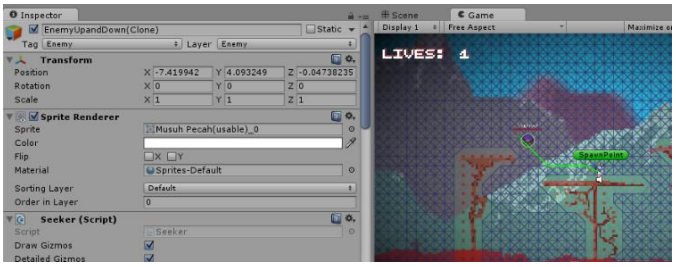

Figure 8. *Gizmo's* mode in *A* Pathfinding*

The blue dots are the points where the AI can move freely (except for platform as the obstacles, but some enemy could go through, and ignore rigid body) to find the main character and the green lines are the distance traveled.

3.6 *Game Script*

Game Script is a technical document that contains hundreds of code sheets that contain functions and game description in video games which of them are the visual description (the game visualization), the dialogues that appear in the video game, when the button is pressed so that the character could move and shoot, etc.

In this game there are several code scripts written using the C # programming language which are compiled using the Mono-Develop Text Editor, which is useful in moving graphic type of assets in game creation.

```
using UnityEngine;
using System.Collections;
using UnityStandardAssets.ImageEffects;

public class GameMaster : MonoBehaviour {

public static GameMaster gm;

    [SerializeField]
    private int maxLives = 3;
    private static int _remainingLives = 3;
    public static int ReaminingLives
    {
        get{return _remainingLives;}
    }
```

Figure 9. C# Code that Inputted in the game

3.7 *Publishing Game*

The Publishing or Exporting game process is the last stage in video games making, this is done to make the players to play and try this game, this game will go through the publishing process into a file type that can be read and used by windows-based computers or Android smartphones that available around the world. the associating file type is going to be .EXE and .APK, .EXE files which going to run on Windows based operating systems computer and. APKs for Android smartphones.

IV. CONCLUSION

The development of this game is aimed to introduce A* Pathfinding and how to apply it to AI in the form of Unity Assets. There are several elements that been utilized such as, Level Design, World Design, Character Design, Level Design and System Design. At process of making this video games, there are several elements that being amalgamated such as animation, images, and sound. The graphic assets were made Using Adobe Photoshop CS5 software (for Sprite assets), Adobe Audition (for Audio assets) and Unity 3D software is utilized to create and process these graphic assets into a game. Based on the results of the research and discussion as well as the conclusions that have been put forward, there are still shortcomings that can be further refined in subsequent research, therefore there are suggestions that can be given for future development: more interesting items need to be added as well as additional gameplay (such as jetpack, double, etc.) to add to make the game more fun to play. The game concept that has been designed, it is necessary to add more rules in the game so that the game is more challenging to play (such as rogue as mode, punishment for dead such as lost gold coin, etc.). This game also is much better state when the development for cross-platform is implemented and online game features such as PVP (Player Vs Player) and CO-OP (Cooperation) were developed and introduced.

REFERENCES

[1] Felner, Ariel, Roni Stern, Solomon Shimony, Eli Boyarski, Meir Goldenberg, Guni Sharon, Nathan Sturtevant, Glenn Wagner, and Pavel Surynek. "Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges." In International Symposium on Combinatorial Search, vol. 8, no. 1. 2017.

[2] Foead, Daniel, Alifio Ghifari, Marchel Budi Kusuma, Novita Hanafiah, and Eric Gunawan. "A systematic literature review of A* pathfinding." Procedia Computer Science 179 (2021): 507-514.

[3] Zikky, Moh. "Review of A*(A star) navigation mesh pathfinding as the alternative of artificial intelligent for ghost's agent on the Pacman game." EMITTER

International journal of engineering technology 4, no. 1 (2016): 141-149.

[4] Aydoğan, E. Bilgin, and Yılmaz Atay. "Unity Based A* Algorithm Used in Shortest Path Finding Problem for Helicopters." In 2021 International Conference on Control, Automation and Diagnosis (ICCAD), pp. 1-5. IEEE, 2021.

[5] Erhard-Olsson, Niklas. "Procedural Generation of 2D Levels for a Motion-based Platformer Game in Unity with Large Amount of Movement." (2018).

[6] Martins, Raianedos Santos, Filipe de Carvalho Pinto Raulino, Aquiles Medeiros Filgueira, and Akynara Aglaé Rodrigues Santos da Silva Burlamaqui. "SGDDEdu-Model of short game design document for digital educational games." (2019).