# Implementation of Behavior Tree in *Halo 2*

Fakhri Nail Wibowo 13519035[1]
*Program Studi Teknik Informatika*
*Sekolah Teknik Elektro dan Informatika*
*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*
*[1]13519035@std.stei.itb.ac.id*

***Abstract*—Games used to be a niche, where the only ones who play and create them are computer scientists intended for educational purposes. Now with the boom of technology, they started becoming more and more popular. Along with the improvement of technology, came the improvement of the mechanics, logic, and experience. With it also came to an improved NPC or AI to be able to experience a game in its fullest glory without having to have someone to play with. The AI can be implemented using a type of one of the topics in Discrete Math, that is Behavior Tree. In this paper, the author looks at how behavior tree is implemented in games, particularly *Halo 2*.**

***Keywords*—behavior tree, *Halo 2*, artificial intelligence.**

## I. Introduction

Video games are slowly becoming a more important part of society, whether it is for educational or recreational purposes. Even though it is very common now to own and play one it was not always the case, it was started by the scientists and in the research labs.

In the 1950s, computer scientists used video games as part of their research and for recreation. In 1952, British professor A.S. Douglas created a game called *OXO* which simulates a game of tic-tac-toe. Douglas created it as part of his doctoral dissertation at the University of Cambridge. In 1958, *Tennis for Two* was created by William Higinbotham.

As technology and computer become faster and smaller, video games began making their way to a mainstream audience. Home consoles started making their appearances in people's homes, arcade games were there for those who can not afford their own game. And then, with the power of personal computers, people have another way to enjoy video games. Until now, the video game industry continues to grow globally.

With the growth of the industry, came along the innovation of video games. When first created, it was limited in many aspects, starting from the graphics, accessibility, mechanics, and many more.

One of the most notable innovation is the artificial intelligence (AI) of video games, or usually called non-player characters (NPCs). This allowed players to test their skills against a computer, without having to have another player to play with. Although called AI, there are differences between academic AI and video games AI. There are many methods to create AI by using decision trees, finite state machines, and many more.

*Halo* is an American military science fiction media franchise. *Halo* focuses on a first-person shooter genre. Its first iteration came out in 2001 published by Microsoft Corporation and it sold more than five million copies and a lot of sequels. The sequel *Halo 2* became one of the most popular titles as well, especially in online multiplayer. *Halo* franchise is one of the pioneers in the industry in many aspects, one of them is in its AI. *Halo* uses a behavior tree to model its enemy behavior.

This paper will try to explain how AI or NPC in *Halo* 2 works using a behavior tree.

## II. Theoretical Basis

### A. Graph

A graph is a collection of nodes and lines. Dots represent discrete objects, while lines represent the connection between said objects.

In math, a graph can be defined as a set (V.E), where V is a set of vertices or nodes and E is a set of paired vertices, whose elements are called edges or lines.

There are two types of graphs, simple graph, and non-simple graphs. A graph without a loop and parallel edges is called a simple graph, while a graph with either one of loop or parallel edges is called a non-simple graph.

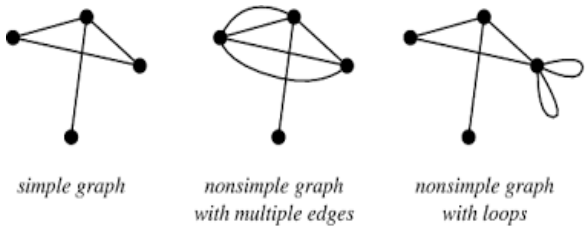



*Figure 1.Example of a simple and a non-simple graph (Source:* Simple Graph -- from Wolfram MathWorld mathworld.wolfram.com (google.com) *accessed on 10th December 2020)*

Based on its edges, a graph can be identified into two types, directed graph, and non-directed graph. A directed graph is a graph in which edges have orientations. An undirected graph is a graph in which edges do not have orientations.

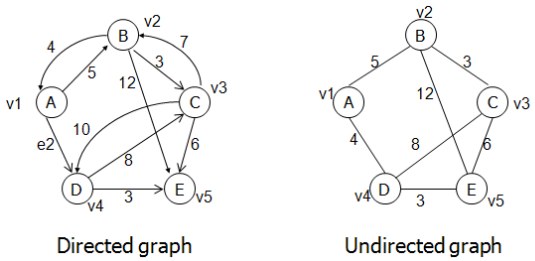



*Figure 2. Example of a directed and undirected graph (Source: GRAPH ( Graf ) | A. ERFAN PRATOMO global sites & blog! (wordpress.com) accessed om 10th December 2020)*

### *B. Tree*

A tree is an undirected graph containing no closed loops. If there is a graph G = (V, E), V cannot be an empty set, unlike E.

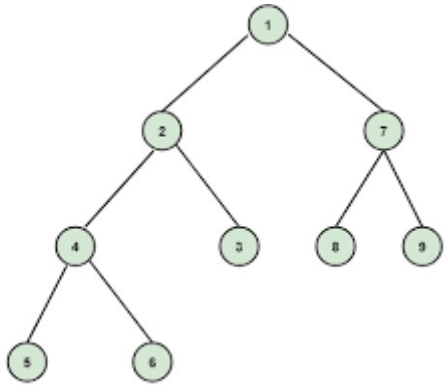


*Figure 3. Example of a tree (Source: Discrete Mathematics Binary Trees ... javatpoint.com (google.com) accessed on 10th December 2020)*

If G = (V, E) is an undirected, simple graph with n vertices, then it has these properties:

1. G is a tree.
2. Every paired vertex in G is connected with a single path.
3. G is connected and has n-1 edges.
4. G does not have a circuit.
5. The addition of an edge will make one and only one circuit in G.
6. All of the edges in G are bridges.

There are some important terminologies in a tree, such as:

1. Root

The first node from where the tree originates.

2. Edge

The line connecting paired nodes.

3. Parent

The node with an edge to the other node and which is the first node on the path from the other node to the root.

4. Child

The opposite of parent.

5. Siblings

The nodes which belong to the same parent are called siblings.

6. Degree

The total number of children of that node.

7. Internal Node

The node which has at least one child.

8. Leaf Node

A node that does not have any child.

9. Level

The number of edges between a node and the root added by one.

10. Path

A sequence of edges between nodes.

11. Depth

The length of a path from the root to a node.

12. Height

The largest depth of any node in the tree.

13. Directed Tree

A directed tree is a type of tree, in which the edges have an orientation.

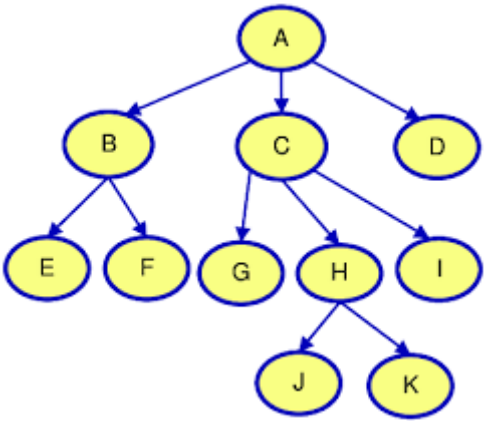



*Figure 4. Example of a directed tree (Source: Example of directed tree | Download ... researchgate.net (google.com) accessed on 11th December 2020)*

14. Ordered Tree

A rooted tree where the order of the nodes is important and it has criteria for each corresponding order.

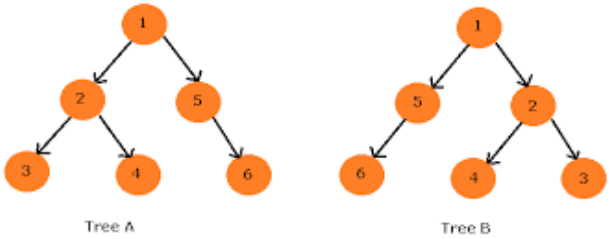



*Figure 5. Example of an ordered tree (Tree A) and an unordered tree (Tree B) (Source: Ordered Tree vs Unordered Tree ... notesformsc.org (google.com) accessed on 11th December 2020)*

### *C. Behavior Tree*

A behavior tree is used in video games to model the behavior of non-player characters (NPCs) or artificial intelligence (AI). It is used in many high-profile games and even mainstream and free game engines, such as Unity and Unreal Engine.

A behavior tree is usually represented using a directed tree. It has three different node types, root, control flow, and execution, with each node, returns either Success, Failure, or Running. It is

aimed to represent how something behaves.

The execution of the tree starts from the root which sends a signal that allows the execution of a child. Usually, it executes every tick of the game, running through from node to child every time.

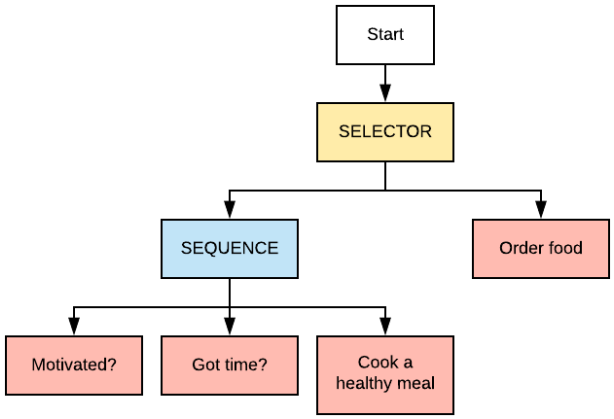



*Figure 6. Example of a behavior tree (Source: Behavior Trees - Introduction (zhaytam.com) accessed on 10th December 2020)*

There are three main types of behavior tree node, composite, decorator, and leaf nodes.

Composite nodes are control flow nodes. It is used to control the nodes of which it is composed, how they run, and when to stop. There are mainly three most used types of composite nodes.

The first is Sequence, Sequence contains one or more children. Upon execution, it will visit each child in order, and when that succeeds it will continue according to the order. If one of them fails, it will immediately return failure to the parent. In order for it to return success, it has to succeed at its every node. This node is analogous with an AND gate.

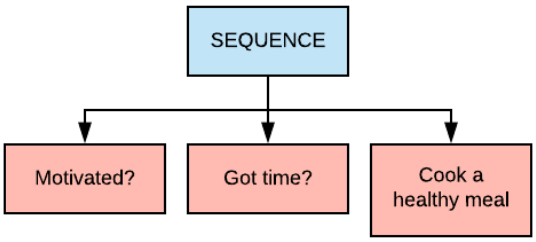



*Figure 7. Example of a sequence (Source: Behavior Trees - Introduction (zhaytam.com) accessed on 10th December 2020)*

The sequence above will go through the nodes from left to right, checking every condition whether it is true or not. If the player is motivated and has time, he will cook a healthy meal.

The second type is Selector, it works as an opposite to the Sequence node. Upon execution, it executes every child until one of them succeeds, otherwise, it fails.

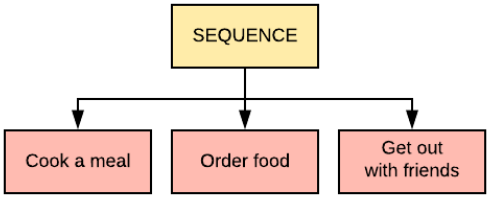



*Figure 8. Example of a selector node (Source: Behavior Trees - Introduction (zhaytam.com) accessed on 10th December 2020)*

In the example above, it first goes through the cook meal node and checks if it is satisfied or not, if not then it will continue until the last node. Meaning, it can only achieve one outcome out of three possible outcomes.

The third one is Random Selectors/Sequences. As can be seen from the name, it acts as a Selector or Sequence, except the order in which nodes are processed is random.

The second type of node is Decorator. Decorator nodes are also a control flow-node. It can only have a single child, though the child does not have to be a leaf node. It is used as utility nodes, an easy example would be a Repeater, which runs the child node a number of times.

Other than a Repeater, there are several other types, such as:

1. Inverter

This node is analogous to a NOT gate, it inverts the result of their child node.

2. Succeeder

A succeeder will always return success, it does not matter what the result of their nodes is.

3. Repeat Until Fail

Similar to a repeater, except this node will run the child node until its child return failure.

The third type of node is Leaf. There are mainly two types of leaf nodes, one to perform an action and the other to check a predetermined condition. They are usually user-defined.

Other than nodes, a behavior tree also needs a Blackboard. Blackboard is used to store data that is global and can be accessed by all nodes. With this, each node can talk to each other.

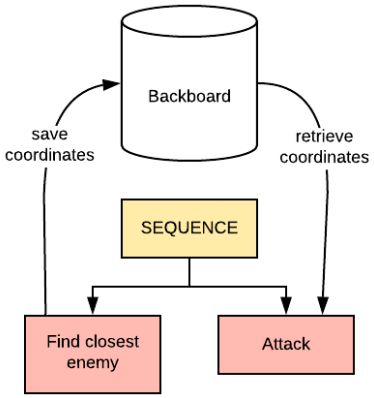



*Figure 9. Example of a blackboard and how it works (Source: Behavior Trees - Introduction (zhaytam.com)*

*accessed on 10th December 2020)*

## IV. IMPLEMENTATION OF BEHAVIOR TREES IN *HALO 2*

*Halo 2* is a game of the first-person shooter published by Microsoft. It is a part of the famous *Halo* franchise and the sequel to *Halo: Combat Evolved*. This franchise is considered a pioneer in many aspects, namely its combat behavior of the enemy using advanced artificial intelligence techniques.

In this game, the player plays as a character named Master Chief and an alien Covenant Elite called the Arbiter. The player will face various types of enemies, each with a different trait and the player has to go through them to complete the level.

There are seven types of enemies, each with its characteristics and personalized behavior.

The first one is Elites, they are designed to be the leader of the pack and be more intelligent than their Brute counterparts. This is shown with the fact that they are more likely to run away if their health is low.

*Figure 11. Elites in* Halo 2 (*Source:* Halo AI - Examining the Behavioral Dynamics of the Covenant Enemies (studyofai.com) accessed on 11th December 2020)

The second type is Brutes. Brutes are less intelligent but more aggressive than Elites. They have a specific mode which makes them more likely to attack and be more aggressive when low on health.

*Figure 12. Brutes in* Halo 2 (*Source:* Halo AI - Examining the Behavioral Dynamics of the Covenant Enemies (studyofai.com) accessed on 11th December 2020)

The third type is Grunts. Grunts are known to fleeing from the player if their Elite is killed. They also have a specific mode that will make them seek the player and explode on contact.

*Figure 13. Grunts in* Halo 2 (*Source:* Halo AI - Examining the Behavioral Dynamics of the Covenant Enemies (studyofai.com) accessed on 11th December 2020)

The fourth type is Jackals. Jackals have shields to protect their allies, but they also often carry long-range weaponry.

*Figure 14. Jackals in* Halo 2 (*Source:* Halo AI - Examining the Behavioral Dynamics of the Covenant Enemies (studyofai.com) accessed on 11th December 2020)

The fifth type is Hunters. Hunters are slow and massive, though they are powerful. They have an enrage mechanic that triggers when their partner is killed.

*Figure 15. Hunters in* Halo 2 (*Source:* Halo AI - Examining the Behavioral Dynamics of the Covenant Enemies (studyofai.com) accessed on 11th December 2020)

The sixth type is Flood Parasite. They have a specific mechanic that allows them to multiply when they get shot at.

*Figure 16. Flood Parasite in* Halo 2 (*Source:* Halo AI - Examining the Behavioral Dynamics of the Covenant Enemies (studyofai.com) accessed on 11th December 2020)

The last type is Drones. Drones have many ways to mobilize, they can walk, crawl, and fly. They tend to stick together.

*Figure 17. Drones in* Halo 2 (*Source:* Halo AI - Examining the Behavioral Dynamics of the Covenant Enemies (studyofai.com) accessed on 11th December 2020)

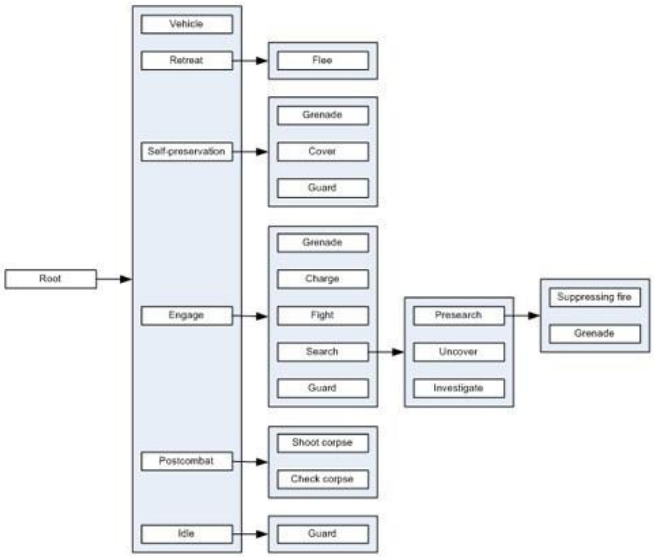



*Figure 18. Example of a behavior tree implemented in* Halo 2 *(Source: Isla, 2005)*

The above diagram is a behavior tree of enemies from *Halo 2*. It has several selector nodes, the root is the first one and its child is also a selector node. This will run every tick of the game and will go through the tree deciding which path is the best and most fitting to take. It will continue until a leaf or an action node is selected and an action is performed.

In a closer look, the tree consists of several levels. The first level selects the type of action the NPC is going to do. The second level consists of individual behaviors, some unique to a type of enemy.

To pick an action node, there are a requirement or a way to select depending on which action node. For example, in combat, there will be different situations, one of which is vehicular-based combat. For vehicular-based combat, there are three NPC roles, the first is infantry, the second is vehicle drivers, and the last is passengers. With each role come its special action nodes, the infantry will be restricted to a few selected nodes intended for other roles, and this applies to the drivers and passengers as well.

On top of that, *Halo 2* also has various types of enemies, each with its unique traits. For example, a particular enemy type named Brute is designed to act aggressively and does not shy away from combat. Another example is a Grunt who will abandon the battle and run away if their leader is killed first.

Though having such a big number of actions come with its challenges, and in this case, that is how to determine which one to choose. *Halo 2* uses a priority-based list to solve that, there are priorities to each action, which are calculated at runtime based on the current situation of the game. This means as battle evolves over time, enemies' actions will change accordingly and will try to re-evaluate their action's priority.

Even with the priority system in place, there will be a situation where the priority list does not fit the context. For example, if the enemy has the choice to enter an empty vehicle or to stand ground and shoot at the player it will choose to shoot the player, but the second the player itself decides to enter into a vehicle, the enemy will determine that entering a vehicle too is the best course of action. This is called an impulse trigger, which will redirect the current execution to a different area of the tree that

fits the context best, making it easy to switch between different behavior types while considering the context.

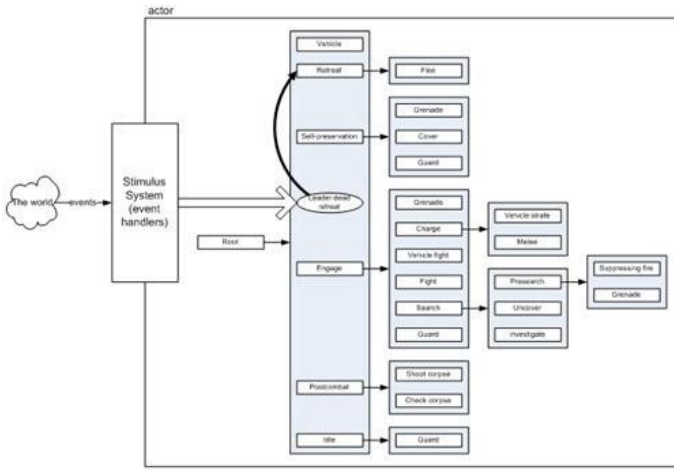



*Figure 19. Example of a stimulus behavior, a type of impulse trigger (Source: Isla, 2005)*

To interrupt and change the execution of the tree, there is an event-driven approach called stimulus behavior implemented. It is an impulse trigger and it allows them to override the enemy behavior.

There are also two more commands, designed to make the behavior more natural, believable, and unpredictable.

The first one is the style command. Style command limits the possibilities of action nodes from which the enemies can choose from. For example, Bungie team has aggressive and defensive styles, which will disallow a few action nodes that are not suitable for them.

The second is order commands, these will tell the AI where to go, and when combined with style commands, they create a team movement. Order commands tell the direction and style tell how to do them.

By implementing a behavior tree, it is also possible to create a personalized behavior for each type of enemy. The parent node can act as a generalized behavior and the children nodes can act as a specialized behavior for each type.

Also, using a similar mechanic to a blackboard mentioned above, the AI can identify, sense, and remember their surroundings, context, and behavior history. Particularly with behavior history, meaning there was no need to run through the entire tree at every tick of the game, which will slow down the game a lot.

## V. Conclusion

A behavior tree is one of many ways we could implement to create an AI or NPC in video games. It is easy to understand the logic of it because of its intuitive visual representation. It is also modular, by using parent nodes and child nodes which can be customized according to needs.

## VI. Acknowledgment

First of all, the author wants to express his gratitude to Allah SWT, for only with His will can the author work on and finish the paper. Next, the author wants to thank his family for always being there and always being supportive regardless of condition. The author also wants to thank his friends for supporting him throughout the course and the semester. Last but not least, the author wants to thank Fariska Zakhralativa Ruskanda, S.T., M.T., as the lecturer of IF2120 Discrete Mathematics in Bandung Institute of Technology and the entire lecturer team of course IF2120 for their guidance in this last 6 months, which without their help this paper would not be possible.

The author also would like to express his apologies for any mistakes, whether intentional or not. For a human is always susceptible to mistakes.

## References

[1] General Tree Definitions and Terminology (kent.edu) accessed on 11th December 2020

[2] Behaviour Trees: The Cornerstone of Modern Game AI | AI 101 - YouTube accessed on 11th December 2020

[3] asdas Behavior Trees - Introduction (zhaytam.com) accessed on 11th December 2020

[4] sadds Video Game History - Timeline & Facts - HISTORY accessed on 11th December 2020

[5] sada The Behaviour Tree AI of Halo 2 | AI and Games - YouTube accessed on 11th December 2020

[6] sadsa Outsmarting The Covenant: The AI of Halo 2 | by Tommy Thompson | Cube | Medium accessed on 11th December 2020

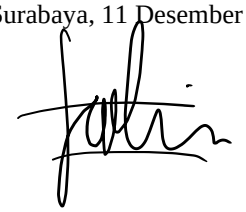
[7] sadvs Halo AI - Examining the Behavioral Dynamics of the Covenant Enemies (studyofai.com) accessed on 11th December 2020

[8] asda Gamasutra - GDC 2005 Proceeding: Handling Complexity in the Halo 2 AI accessed on 11th December 2020

## Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Surabaya, 11 Desember 2020



Fakhri Nail Wibowo 13519035