Decision Tree and Its Implementation in Different Types of Games

Ignatius David Partogi - 13518014[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13518014@std.stei.itb.ac.id

***Abstract*—Video games have been the life of many people now. People play games for many reasons. For example, people like playing games as a hobby, to relieve stress, or even as a way to make money (this is called e-Sports). While it is true that many people like playing games, not as many people have the curiosity about how the mechanics of the video game is possible to work in synergy. Decision Tree is a very important principle in making many games. Decision Tree can be used to explain about the gameplay mechanics and therefore, Decision Tree can describe the identity of the game.**

***Keywords*—Decision Tree, Gameplay Mechanics, Implementation, Video Game.**

I. Introduction

Video game is something many people like to play. It is fun, filled with good graphics and interesting gameplay mechanics so that people do not get bored easily. Video games have many genres since not every people likes to play the same type of video game. People like to play games as a hobby and to relax after a long and stressful life. With the massive development of technology, video games also has been significantly improved. Current video games have long stories, wide selection of gameplay mechanics, and realistic graphics, making them better choices for relieving stress. Furthermore, video games started to be used as a way to gain money. People like to hold tournaments or competitions for some games, particularly online ones. The tournament itself rewards thousands of dollars, making it a somewhat profitable for some gamers to dedicate on. Because of the increase of video game players, making video games becomes a very profitable job.

The concept of video game itself was founded by Ralph H. Baer in 1972. Ralph, who at that time was working as a contractor engineer at Sander Associates, tried to make a video game that could be played at televisions, because computers were rare and expensive back then [1]. The first video game that Ralph released was title Pong, Pong was very successful and became a legend among people at that time. At the same year, Ralph also released a video game console named Magnavox Odyssey, which then became the ancestor of video game consoles [2].

The succession of Ralph and his video games were heard throughout the world. Since then, many video game companies started to emerge, like Atari, Sega, and Nintendo. Video game industries became very competitive, making many video game companies which could not compete go bankrupt and be bought by other companies.

Now, the video game industry has improved significantly compared to the era of the first generation of video games. Video game companies are also divided into two classifications: Video game companies, companies who make video games, like Ubisoft, EA, and Capcom, and game console company, who make and provide platforms for playing games, like Sony and Microsoft. The existences of internet also boosts video game industries significantly, emerging a lot of famous online games like DoTA, Counter Strike: Global Offensive, Minecraft, Mobile Legends, and many more. With the advanced technology that we possess, games are able to be made with beautiful and realistic graphics, long and complex stories, and smooth gameplay, though this is done with the trade of more expensive cost for each game.

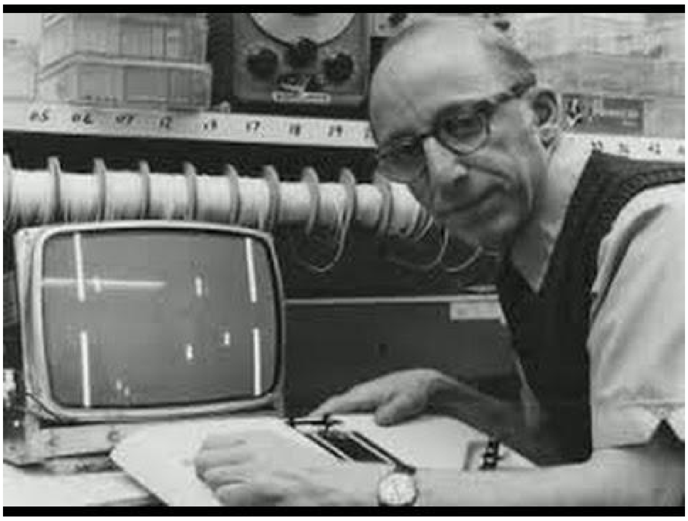

Image 1.1. Photo of Ralph H. Baer
Source: *solopos.com*

II. Theoretical Framework

A. Tree

The concept of tree was founded and has been used since 1957 by Arthur Cayley, a British mathematician to count the amount of chemical compounds [3]. This invention has helped many people solve many problems in different types of studies.

Tree is a connected, undirected graph that consists of no circuits [1]. A connected graph is a graph whose nodes are all connected and there is at least one way to get to every node using the edges, and an undirected graph is a graph whose edges do not have any direction [3]. This means that if an edge e connects two nodes x and y, x is connected to y and y is connected to x. A tree will always be a tree, no matter how each node and edge is arranged.

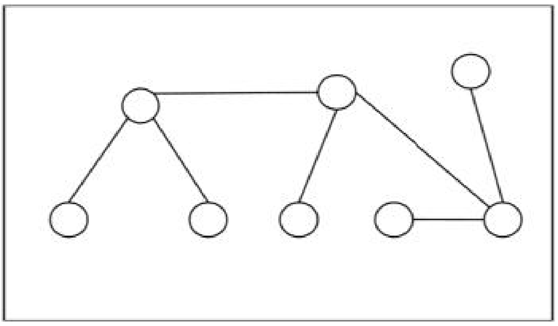

Image 2.1. An Example of a Tree
Source: *tutorialspoint.com*

B. Subtree

Subtree T' is a tree that has nodes and edges that can form a subset of a tree T [3]. This means that T' has to be a tree, and T' has to be one of the subsets of T.

C. Rooted Tree

Rooted Tree RT is a tree whose one node is considered to be a root, then every node that is connected to RT by edges is arranged so that it looks like a tree's root [3]. One tree can form many different rooted trees. Each of the rooted tree's edge is an edge with a direction which is away from the root of the tree.

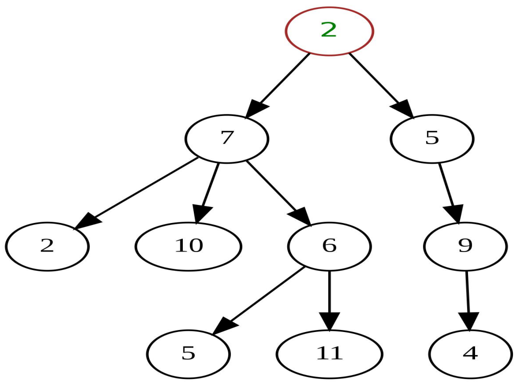

Image 2.2. An Example of a Rooted Tree
Source: *en.wikipedia.org*

Terms that are commonly known for using the concept of rooted tree:

1. Parent/Children Relationship Tree
The concept of this tree related to rooted tree is that the root of the tree is the parent, and the nodes exactly below the root are the parent's children. This goes on with the subtrees until the bottom of the tree.
For example, in Image 2.2, node 2 (root) is the parent, whilist node 7 and node 5 are the children.

2. Leaf
Leaf L is every subtree of a rooted tree RT which does not have any nodes connected below it [3].
For example, in Image 2.2, node 2 (leaf), node 10, node 5, node 11, and node 4 are all leaves of the tree.

3. Level
Level of a node of a tree can be defined by how 'far' it is from the root of the tree. The root of a tree has a level of 0, a node that is connected directly to it has a level of 1, then it goes on recursively until it reaches the desired node.
For example, in Image 2.2, the level of node 6 is 2 and the level of node 11 is 3.

4. Depth
Depth of a rooted tree is the highest level of a tree available. This can be obtained by counting each leaf of the tree and look at any which has the highest level.
For example, in Image 2.2, the depth of the tree is 3, Because the highest level of any node of the tree is 3.

D. Ordered Rooted Tree

Ordered Rooted Tree is a rooted tree whose value of all of the nodes is in order from the left side of the tree to the right side of the tree [3].

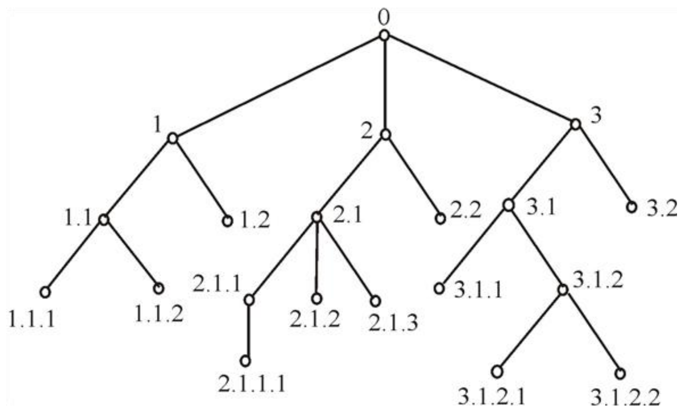

Image 2.3. An Example of an Ordered Rooted Tree
Source: *cheng.com*

E. n-ary Tree

n-ary Tree is a tree whose nodes have a maximum of n number of adjacent nodes each [2]. An n-ary tree is classified as a complete tree if all of its non-leaf nodes have n number of adjacent nodes each [2]. An n-ary tree with 2 as the value of n is called binary tree [2].

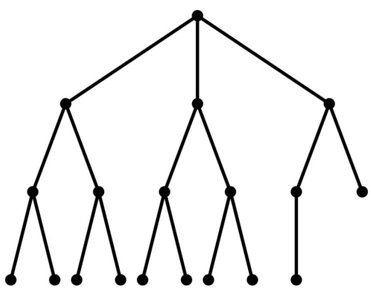

Image 2.4. An Example of an n-ary tree with n = 3
Source: *researchgate.net*

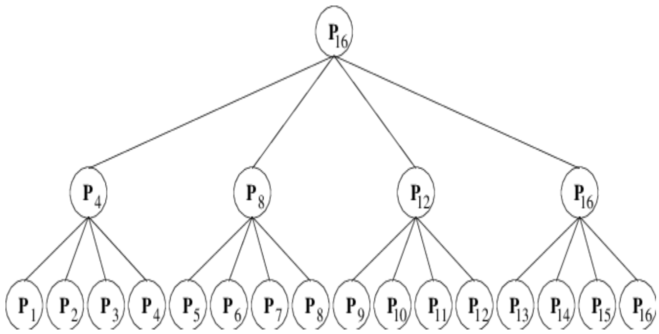

Image 2.5. An Example of a Complete 4-ary Tree
Source: *researchgate.net*

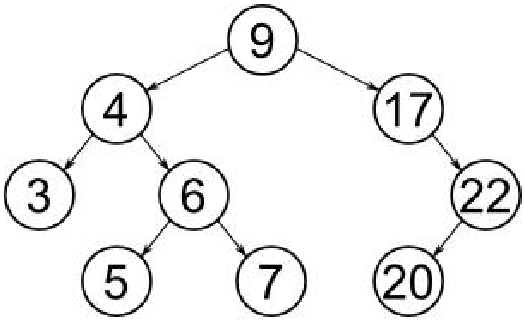

Image 2.6. Binary Search Tree, an Example of a Binary Tree
Source: *topjavatutorial.com*

III. Problem Limitations

Due to the many variations and complexities of video games available, the author limits the problem into games that have simple attributes and do not take a long time to play (1-10 minutes/game), but are replayable, because they do not focus on the story, but rather to the gameplay. The author decides to choose two types of games. The games are Rock-Paper-Scissors and a simple turn-based monster battle game. The author has made sure that these two types of games use the concept of decision tree.

IV. Decision Tree Implementation on Rock-Paper-Scissors Game

A. Video Game Description and History

Rock-Paper-Scissors is a game where each player has to choose one choice between rock, paper, and scissors. The game itself is instant and the result can be determined immediately after both players choose their choices. Rock-Paper-Scissors is a multiplayer game, meaning that the game requires more than one player in order for it to be played. However, despite being a multiplayer game, only two players can play one Rock-Paper-Scissors game at a time.

Rock-Paper-Scissors was derived from China, written in Han dynasty (sometime between 206 BC until 220 AD) by a Ming dinasty writer named Xie Zhaozhi, in a book titled *Wuzazu* [4]. The original name of Rock-Paper-Scissors was shoushiling [4].

Japanese history also mentioned this game frequently, in the name of *sansukumi-ken* [4]. The word ken in sansukumi-ken means “fist”, while the word *san* and *sukumi* means “three” and “deadlock”, respectively [4]. Etymologically, *sansukumi-ken* means “three-way deadlock fist game”, meaning that it is a game where A wins against B, B wins against C, and C wins against A [3].

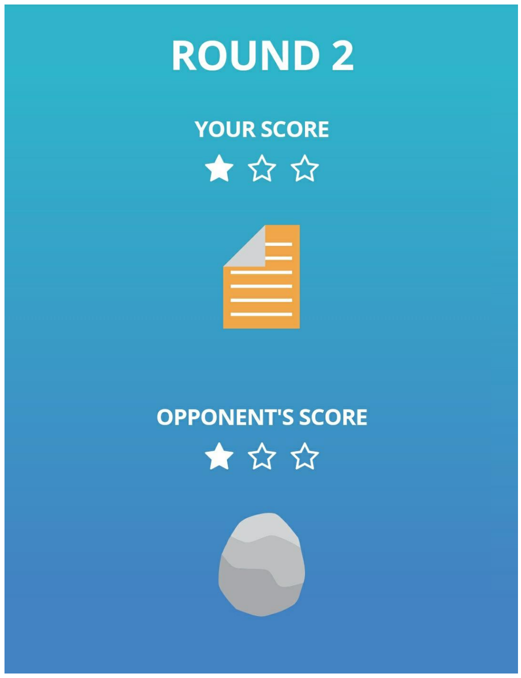

Image 4.1. An Example of Rock-Paper-Scissors Game
Source: *rpsgame.org*

B. Video Game Mechanics

In Rock-Paper-Scissors game, each player starts by choosing between rock, paper, and scissors. After each player has chosen, both show their choices to each other at the same time. The result is determined based on each player’s choice in comparison to the other player.

The choice matchups between rock, paper, and scissors is as follows:

1. Rock beats scissors;
2. Paper beats rock; and
3. Scissors beat paper

If a player chooses a choice that beats the other player’s choice, the player wins the game, the other loses the game, and vice versa. However, if both players choose the same choice, the game is considered a draw, meaning that both players neither win nor lose.

For example, if Player 1 chooses rock and Player 2 chooses scissors, then Player 1 wins. If Player 1 chooses scissors and Player 2 chooses paper, Player 2 wins the game. If both Player 1 and Player 2 choose paper, the game ends in a draw.

C. Decision Tree Implementation in The Game

Image 4.2 is the decision tree implementation of Rock-Paper-Scissors. In the image, the players choose between rock, paper, and scissors. The decision tree focuses on Player 1 choosing between rock, paper, and scissors first. After Player 1 chooses, the decision tree compares what Player 1 chooses to what Player 2 chooses. Finally, after comparing between two choices, the decision tree shows the result of the game that corresponds to what both players choose.

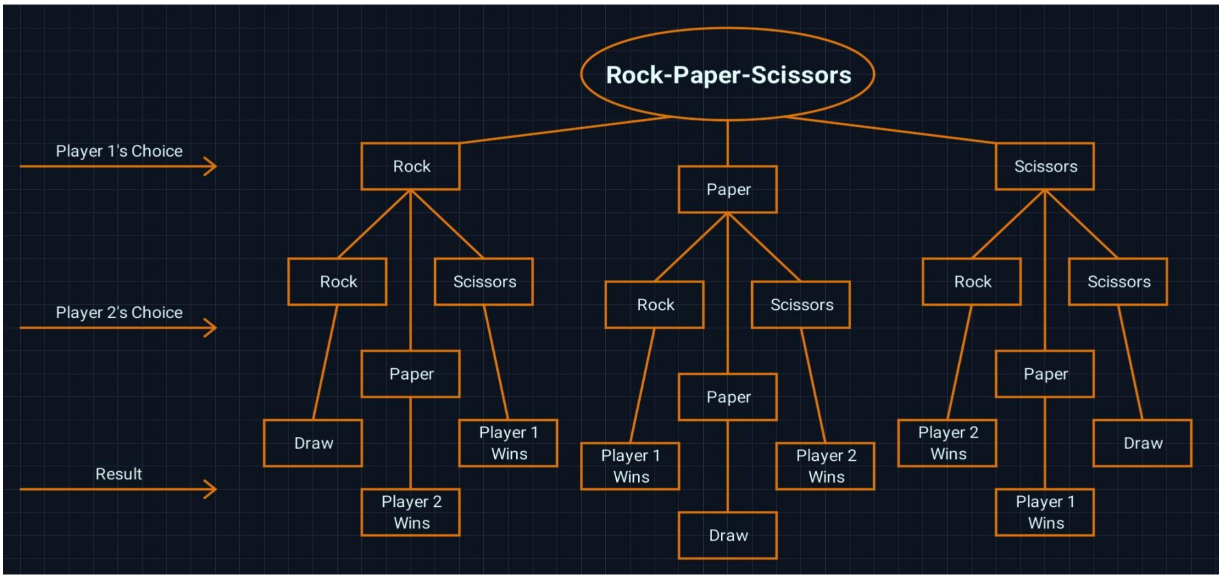

Image 4.2. Decision Tree Implementation of Rock-Paper-Scissors Game

The root of the decision tree is the title of game. While it explicitly states the title of the game, it also means the start of the game. The three nodes that are directly connected to the root of the decision tree by edges represent the possibility of choices that Player 1 can make. Each said node then connects itself using edges to three nodes that represent the possibility of choices that Player 2 can make. This also symbolizes the comparison between two choices. After both choices are compared, the decision tree then shows the result, represented by the one node that is connected directly to the node that both players go with.

V. Decision Tree Implementation on a Simple Turn-Based Monster Battle Game

A. Video Game Description

The game that the author means does not have any connection with any existing turn-based monster battle game in the world. The "simple turn-based monster battle game" here means a custom-made game with a simple mechanic of turn-based monster battle. The game is made for research purposes, to show where the decision tree plays a role in the game. While the game is simple, the game's decision tree representation is capable of representing most turn-based monster battle games, albeit to a lesser extent, with fewer attributes and commands. Due to the game being turn-based, the game is classified as a multiplayer game, meaning that the game requires at least two players to be able to be played properly. In this game, the battle can only be commenced between two players, so the game can only be played by two players at a time.

Image 5.1. Logo of Pokemon, The Most Successful Turn-Based Monster Battle Game in The World
Source: *id.wikipedia.org*

B. Video Game Mechanics

The game mechanics is straightforward and simple. The game starts with two players choosing 6 monsters each. After both players choose monsters, each player chooses one of the monsters to be sent out to the battlefield to battle. Each player then have a Rock-Paper-Scissors game to determine who gets the first turn. After the turn is decided, the player who gets the first turn moves first.

The player has three different types of commands: Attack, Skill, and Switch. Command "Attack" lets the player attack the other player's monster using his or her own monster on the field. After attacking, the system will determine whether the opponent's monster survives the attack or not. If the opponent's monster survives, the turn player's turn ends and the opposing player takes his or her turn, and vice versa. If the opponent's monster does not survive the attack, the system will analyze the opponent's monster inventory to see if there is still a monster that can go to battle. If at least one monster is available for battle in the opponent's monster inventory, then he or she chooses one monster for battle. After a monster is sent out, the turn player's turn ends and the opponent takes the turn.

Command "Skill" lets the player activate one of the four available skills that a monster learns. Each skill activates various different effects, but the variations of the skill are not relevant to the topic, so the detail of the skill is not specified. After command is chosen, the player chooses one of the four skills available to the monster to be activated. The skill chosen is then activated, granting the monster the effect of the skill. The turn ends after the skill is activated.

Command "Switch" lets the player switch the monster on the field to another one in the player's monster inventory. The purpose of this command is to avoid some dangerous skills that the player knows the opponent's monster can learn, or to avoid some matchup disadvantages. After the command is chosen, the player chooses one monster in the player's monster inventory to be switched by the one on the battlefield. The system then switches the two monsters. After the switch mechanic is executed, the turn ends. This means that by

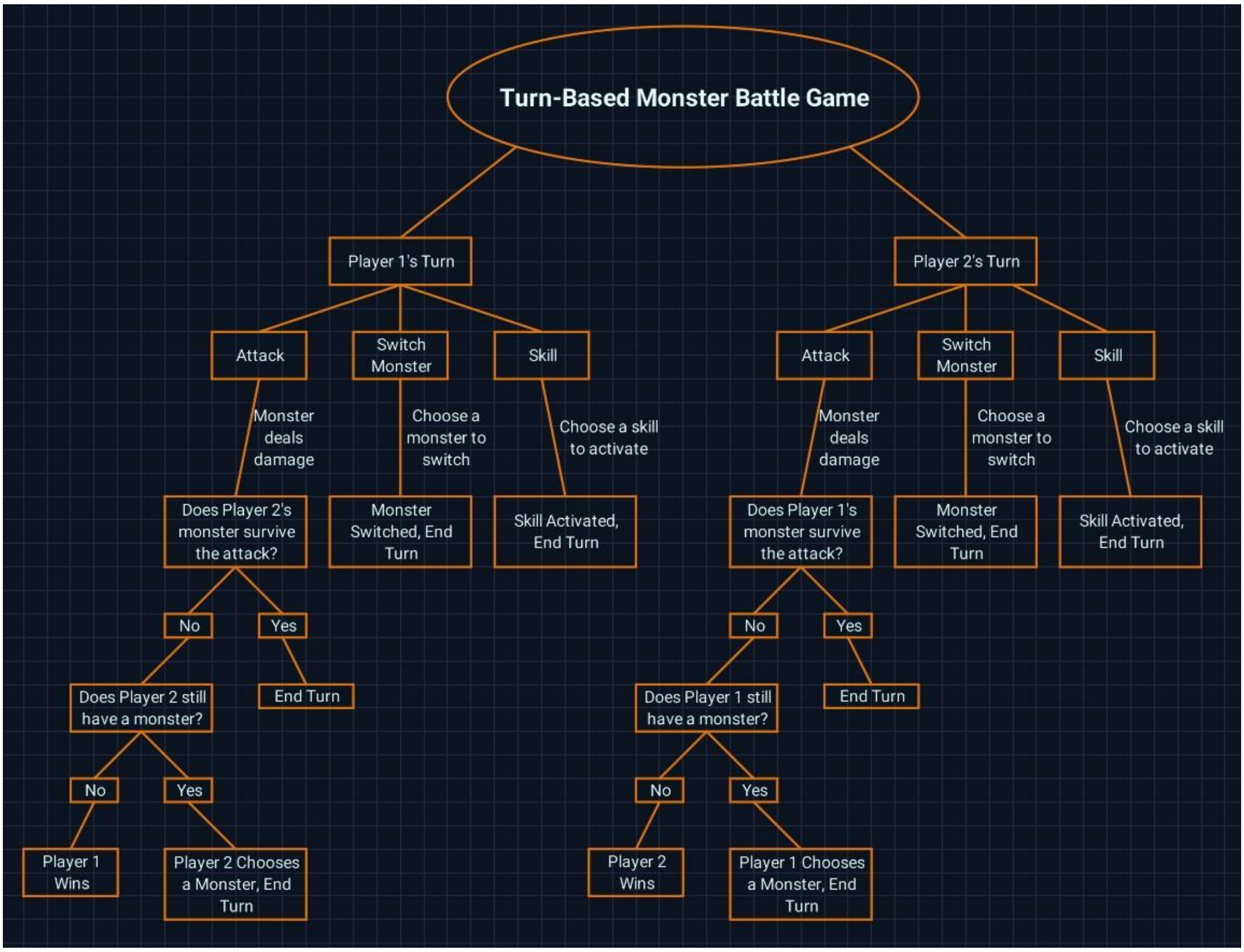

Image 5.2. Decision Tree Implementation of a Simple Turn-Based Monster Battle Game

switching monster, the player gives the opponent one free turn to attack or activate skill.

C. Decision Tree Implementation in The Game

Image 5.2 is the decision tree implementation of the battle section of the video game. The image shows that there are two separate, identical subtrees that represent each player's turn. The decision tree starts by showing three available commands, Attack, Skill, and Switch. Each of the chosen commands will execute the respective adjacent nodes.

The decision tree works as follows: If Player 1 moves first, Player 1 chooses one of the three available commands. After Player 1 chooses, the decision tree reads the choice and executes the adjacent node of the choice. The way the decision tree works is the same way the system of the game works as explained in Section B of this chapter. After Player 1 chooses, the choice is then processed into the corresponding adjacent node.

If Player 1 chooses "Attack", then Player 1's monster deals damage to Player 2's monster. After dealing damage, the decision tree checks battle condition to see whether Player 2's monster survives the attack or not. If it survives, decision tree ends Player 1's turn, whilist if it does not, the decision tree checks Player 2's monster inventory to see if there is any monster that are capable of battling. If there is at least one monster in Player 2's monster inventory, then Player 2 chooses one monster to be sent out to battle. If no monster is available in Player 2's monster inventory, Player 1 wins the game.

If Player 1 chooses "Skill", Player 1 chooses one between four skills that his or her monster has access. After choosing a skill, decision tree activates the skill and ends the turn.

If Player 1 chooses "Switch", Player 1 chooses one of his or her monster in monster inventory. After choosing a monster, the decision tree switches the monster chosen with the monster on the battlefield and ends the turn.

After Player 1's turn ends, Player 2 takes a turn. How the game works in Player 2's turn is the same with Player 1's turn.

VI. CONCLUSION

Decision Tree has many uses in the video game industry. Learning about decision tree is required to be able to make games with high complexity, while still capable to stay consistent and bug-less. Knowledge in Decision Tree is also required in order to debug and modify games, as a little mistake in modification could ruin the entire game program.

VII. ACKNOWLEDGMENT

First of all, the author wants to express his gratitude and thanks to God, for only with His will can the author work on and finish the paper. Next, the author wants to thank his family, particularly both his mother and father, for supporting the studies and works of the author from the start until now. The author also wants to thank Bandung Institute of Technology for providing a place for the author to study and learn. Lastly, the author wants to thank his lecturer, Dra. Harlili, M.Sc alongside the entire lecturer team of course IF2120 teaching the author for six months and for giving the paper assignment, helping the author to learn more about writing a paper and the applications of the materials that are taught, particularly implementations of decision tree in different types of video games. The author also wants to point out that he is a human and is possible to make mistakes. If readers find mistakes in this paper, the author begs for pardon.

REFERENCES

[1] "Lockheed Martin Agrees to Sell Sanders Unit to BAE Systems", Anne Marie Squeo, The Wall Street Journal, July 14, 2000.
[2] "The Father of the Video Game: The Ralph Baer Prototypes and Electronic Games". Smithsonian Institution. Retrieved December 7, 2014.
[3] K.H. Rosen, *Discrete Mathematics and Its Application*, 7th ed. New York: McGraw-Hill, 2012, pp. 641-802.
[4] https://www.wrpsa.com/the-official-history-of-rock-paper-scissors/, accessed at December 4, 2019.

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.

Bandung, 6 Desember 2019

Ignatius David Partogi
13518014