Computing an Effective Strategy to Win Dark Deception Video Game Using Graph Theory

Gede Prasidha Bhawarnawa / 13520004[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*13520004@std.stei.itb.ac.id*

***Abstract*—Dark Deception is a video game where the player needs to collect shards in a maze-like environment while being chased by monsters. As it is a survival horror game, strategies are required to survive the levels. Some strategies can be extracted by using graph theory, as an undirected graph can be derived from the map of the level. The two possible strategies that will be discussed more in depth in this paper is the possibility of an Eulerian graph existing and intersections or T-junctions that are safe or high risk derived from cut-set analysis.**

***Keywords*—Cut-set, Dark Deception, Eulerian Graph, Graph**

I. Introduction

A video game, by definition according to the Cambridge Dictionary, is a game in which the player controls moving picturs on a screen by pressing buttons. In the beginning, video games were created in research labs by scientists. For example, in 1952, A.S. Douglas, a British professor, created a tic-tac-toe game as part of his doctoral dissertation at the University of Cambridge. Another example was William Higinbotham's *Tennis for Two* game on a large analog computer and connected oscilloscope screen in 1958 at the Brookhaven National Laboratory.

Nowadays, video games are the largest growing entertainment industry, especially since the COVID-19 pandemic started in early 2020. As of 2020, the global gaming market was valued at USD 173.7 billion or Rp 2.4 trillion in Indonesian currency. It is expected to reach a value of USD 314.4 billion by 2026 or almost Rp 4.5 trillion. This expectation is reasonable, as the gaming industry keeps creating new innovations both in the gaming consoles and equipments or in the games itself. Some examples of gaming consoles and equipments innovations are AR/VR technology (Augmented Reality and Virtual Reality), voice recognition, facial recognition and eye-tracking feature, and cloud gaming.

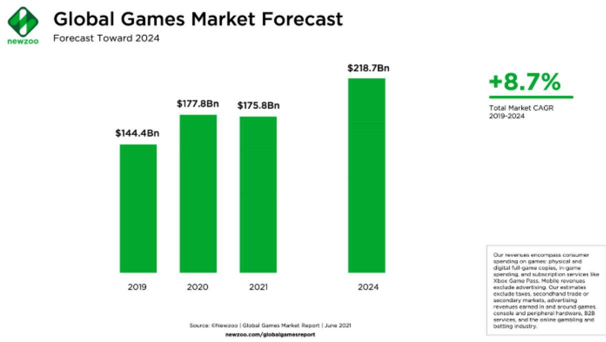

Fig. 1 Global Games Market Forecast Toward 2024 (Source : https://newzoo.com/insights/articles/newzoo-games-market-numbers-revenues-and-audience-2020-2023)

However, the main reason why gaming industry is growing so fast is the versatility of the games itself in response to the current trend. For instance, the release of Halloween-themed games on October, Christmas-themed games on December, or romance games on February. The abundance of game genres are also a key factor in the industry's success. Genres such as adventure, RPG (role-playing game), strategy and puzzle, simulation, action, and sports etc. are some genres that are popular in the gaming community worldwide. In addition, game producers are famous for mixing genres to create a diverse gaming experience for the players. One particular type of mix is survival-horror.

Survival horror is a genre in which the players are tasked to survive in a horrifying and threatening world, combining the survival and horror genre. Usually there are two types of survival horror games. The first type is where the players need to hold out for as long as they can to reach a new time record and the second type is where the players need to complete tasks and missions to complete a stage in the game, commonly with a time limit. It is typical for the players to be given limited resources to increase the difficulty of the game. The players are often accompanied with scary elements as they play such as jumpscares, ambiances, or creepily-designed antagonists, hence the survival horror genre. One game with the survival horror genre that will be throughly discussed in this paper is *Dark Deception*.

Fig. 2 Dark Deception *Video Game Cover*
(Source : https://store.steampowered.com/app/332950/Dark_Deception)

In *Steam*, a video game distribution service, *Dark Deception* is described as a "story driven first-person horror action maze game that mixes the fast-paced style of classic arcade games with fun horror game design". *Dark Deception* lets the player play as Doug Houser as he traverse through mazes of different settings to collect soul shards and unlock ring pieces while evading monsters and demons. There are 10 different levels in total, all with different types of enemies and maze patterns. There are also hidden secrets in each level, consisting of pictures, behind the scenes, and soundtracks.

In this paper, only the maze in the *Monkey Business* level will be throughly analyzed. The author will also analyze the first act in the level, consisting of three "*Murder Monkeys*" as enemies and without computing the secrets in the level. This paper will thoroughly discuss the best strategy to win the level with graph theory.

II. Basic Theory

A. Graph and Graph Types

Graphs are structures that convey how discrete objects connect and relate with each other. A graph consists of vertices and edges. A vertex or vertices in plural is an object in a graph that are or aren't connected in a graph. It is common for vertices to be symbollized with a dot or a circle in a graph. On the other hand, an edge is a connection between two vertices. Edges are commonly symbollized as lines that connect two circles/vertices in a graph.

In formal mathematics definition, we can declare a graph G as:

$$G = (V, E) \quad (1)$$

With V represents the non-empty set contains every vertice in graph G and E represents the edges set. For n members of each set, V and E may be written as:

$$V = \{v_1, v_2, v_3, \dots, v_n\} \quad (2)$$
$$E = \{e_1, e_2, e_3, \dots, e_n\} \quad (3)$$

Based on the edges on the graph, a graph may be put into two types: simple graph and unsimple graph. Simple graph is a graph that does not contain multiple edges (paralel edges) or loops. Unsimple graph, on the other hand, is a graph that either contain one or more multiple edges, one or more loops, or both. An unsimple graph can also be derived into two types: a multi-graph and a pseudo-graph. A multi-graph is a graph that contains one or more paralel edges, while a pseudo-graph is a graph that contain one or more loops.

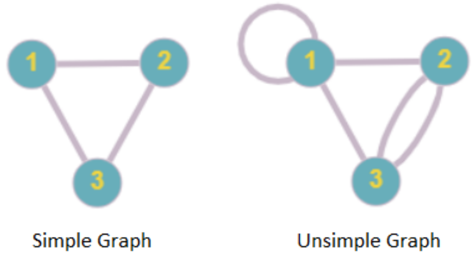

Fig. 3 Example of Simple and Unsimple Graph
(Source : Personal Library)

Based on the direction orientation on each edges, a graph may also be put into two categories: undirected graph and directed graph or digraph. An undirected graph is a graph that does not have any direction orientation for its edges. On the contrary, a directed graph or a digraph in short is a graph where every edge has their own direction orientation.

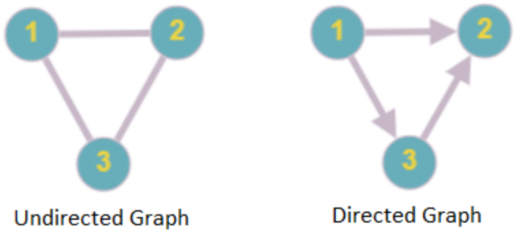

Fig. 4 Example of Undirected and Directed Graph
(Source : Personal Library)

B. Graph Terminology

In graph theory, there are some terminologies that are often used to analyze graphs. There are adjacent vertices, incidency, isolated vertex, null or empty graph, degrees, path and path length, cycles or circuits, connectedness, subgraph and its complement, connected components, spanning subgraph, cut-set graphs, and weighted graph.

In a single graph, two vertices are adjacent to each other if both are connected with each other. This can be proved with the existence of an edge that connects both vertices. In relation to adjacency, incidency is comparing an edge to the vertices that consists it. In mathematical terms, for any edge $e = (v_i, v_j)$, we can say that e is incident with the vertex v_i and v_j.

In the topic of graphs that has one or more unconnected vertices, we may identify those vertices as isolated vertices. By definition, an isolated vertex is a vertex that is not incident with any edges in the graph. And if the whole graph consists of isolated vertices, we may call that a null or an empty graph.

It is common for a single vertex to be in incidence with multiple edges. To quantify the number of incidence, a degree is used. A vertex's degree announces the number of edges that are incident with the vertex. It is important to note that a loop is counted twice and not once. A pair of paralel edges is also counted as two for the vertex's degree.

In the case of directed graph, we may split the degrees into two categories, in-degree and out-degree. An in-degree measures the quantity of edges going into the vertex, while an out-degree measures the quantity of edges going out of the vertex. Since directed graph has edges that show the connection between two vertices, it is not necessary for a vertex to have the same in-degree as it's out-degree. It is also not necessary for the total in-degree in the graph to be same amount as it's out-degree.

A path is a sequence of vertices and edges coming out in turns that notates the way from any origin vertex v_i to target vertex v_j. A path's length is measured by the quantity of edges in the path. For example, in the following directed graph, one of the possible paths from vertex *1* to vertex *4* is 1, (1,2), 2, (2,3), 3, (3,4), 4. Since there are three edges in the path, the path's length is three.

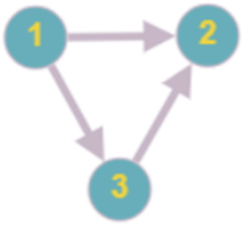

Fig. 5 Paths in a Directed Graph
(Source : Personal Library)

It is important to note that for an undirected graph, a single edge can be considered to be a two-way directed edge. Using this concept in mind, there exists more ways or paths from one vertex to another in an undirected graph compared to paths in a directed graph.

Subgraph of graph *G* is a part of a graph *G* in a way that every vertice and edges in the subgraph exists in the graph. In matemathical notation, if there exists a graph $G = (V, E)$, there is also a subgraph $G_1 = (V_1, E_1)$ if and only if $V_1 \subseteq V$ and $E_1 \subseteq E$. There also exists a subgraph complement, notated as G_2, where $G_2 = (V_2, E_2)$ so that $E_2 = E - E_1$ and V_2 are the set of all vertices incident with every edge in E_2. Lastly, there's also a spanning subgraph, or a subgraph that contains every vertex from the main graph.

In a graph, there may exists multiple vertices. Two vertices are connected if there's a path from one vertex to the other. If every vertice are connected inside a graph, we can call that graph a connected graph. Otherwise, that graph is a disconnected graph.

For a directed graph, we may categorize connectedness based on how strong the connection is for each and every vertex. Two vertices are "strongly connected" if there exists at least one directed path from the origin vertex to the end vertex and at least one more directed path from the end vertex to the origin vertex. If two vertices are connected strongly, but are still connected on the undirected graph, we may call the two vertices "weakly connected". Finally, if two vertices are not connected in either the directed graph or the undirected graph, then the two vertices are unconnected.

Cut-set is a set of edges that when removed from a graph causes the graph to be unconnected. For example, on the left graph below, one example cut-set is {(2,3),(3,4)} as it disconnects vertex 3 from the graph, as seen in the right graph. It is important to note that a cut-set is only if it does not contain a subset that is also a cut-set. For instance, {(1,2),(2,3),(2,4)} is not a cut-set as it contains {(1,2)} and {(2,3),(2,4)} which is also a cut-set.

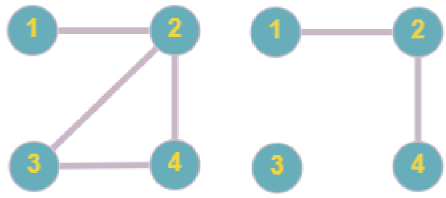

Fig. 6 Example of Graph in Cut-Set Application
(Source : Personal Library)

A weighted graph is a graph that contains a weight in each edge. This weight may represent distance, price, cost, or queue, etc. A weighted graph may be implemented in both directed and undirected graph. An example may be seen below.

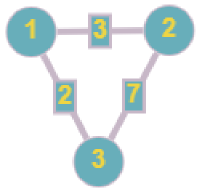

Fig. 7 Weighted Graph
(Source : Personal Library)

C. Adjacency Matrix in Graphs and Isomorphism

While we can declare a graph in a set of paired vertices and edges, it may be hard to read to understand, especially for large graphs. One of the alternative ways to represent a graph is by using an adjacency matrix. An adjacency matrix is a matrix with the size *m* x *m* with *m* being the quantity of the vertices inside the graph. For the matrix's elements, it is either zero or one. The element is one if the corresponding vertices on the row and column in the matrix is adjacent with each other and zero if not adjacent.

	A	B	C	D
A	1	1	0	1
B	1	0	1	1
C	0	1	0	0
D	1	1	0	0

Fig 8. Example of Adjacency Matrix in Graphs
(Source : Personal Library)

With the same adjacency matrix, we can create and draw multiple forms and shapes of graphs. Theses graphs have the same derivation of $G = (V, E)$. Therefore, even though the shapes and forms of the graph is different, at its core, it is still the same graph. This type of different shape but same graphs is called isomorphism.

Isomorphism in graph is when multiple graph is the same while being geometrically different. It is possible for two graphs to have different names in their vertices and their edges. Two graphs, G_1 and G_2, are isomorphic if there exists a one-on-one corespondence between each vertices and each edges on the two matrix. That way, if the graphs are represented in an adjacency matrix, they will return duplicate matrixes.

From the definition of isomorphism itself, there are three general rules for two graphs to be isomorphic. The two graphs must have the exact same amount of vertices, the same amount of edges, and for the vertexes, there need to be specific degrees for each and every one of them. However, this does not guarantee two graphs to be isomorphic. Therefore, a visual analysis needs to be taken care of.

Below are two isomorphic graph with the matrix/table in Fig. 8 as their adjacency matrix.

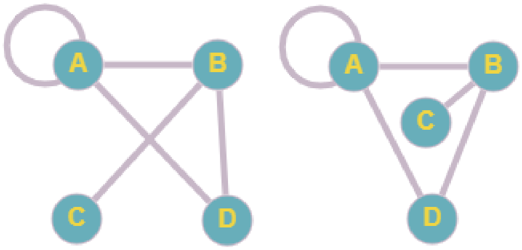

Fig. 9 Example of Two Isomorphic Graph from Fig. 7
(Source : Personal Library)

D. Eulerian and Hamiltonian Trail and Circuit

In determining whether a trail where we can trace through every edge and vertex, we may determine whether an Eulerian or a Hamiltonian trail or circuit exists or not.

An Eulerian trail is a trail where one can trace through every edge in a graph exactly once. On the other hand, an Eulerian circuit is a circuit in where every edge in a graph is traceable exactly once. This means that the main difference between an Eulerian path and an Eulerian circuit is whether the endpoint can exists at the starting point of the path. A graph that contains an Eulerian circuit is called an Eulerian graph, while a graph that only contains an Eulerian path is called a semi-Eulerian graph.

An undirected graph may be classified as an Eulerian graph if and only if the graph is connected and every vertice in it has even degrees. If that is not the case, then the undirected graph may be classified as a semi-Eulerian graph if and only if the graph is connected and have exactly two vertices with odd degrees or none at all. Note that an Eulerian graph is also a semi-Eulerian graph as circuits are paths with the same endpoint as its starting point.

For directed graphs however, there are some slight modifications on the requirements. A directed graph is an Eulerian graph if and only if the graph is connected and every vertice in it must have equal amounts of in-degree as its out-degree. If the previous requirements doesn't fit, the graph may fall into semi-Eulerian graph category, with the requirement being the graph is connected and every vertice must have equal amounts of in-degree as its out-degree with the exception of two vertices. One of the two vertices must have one more in-degree than its out-degree and the other must have the complement.

While Eulerian paths and circuits focuses on finding out whether is it porrible for someone to trace every edge on the graph exactly once, Hamiltonian paths and circuits focuses on the vertices. A Hamiltonian path is a path that passes through every vertice in the graph at least once. That being said, a Hamiltonian circuit is a Hamiltonian path, but the endpoint of the path is at the starting point. This means that in a Hamiltonian circuit, every vertice is visited once except for the starting point which is visited twice. The graphs that only have Hamiltonian path(s) are called semi-Hamiltonian graphs and the graphs that have Hamiltonian circuit(s) are called Hamiltonian graphs.

Sufficient conditions for a simple undirected graph to be a Hamiltonian graph is to have every vertice in the graph to have at least $n/2$ degrees or more with n being the quantity of the vertices and n's value at least 3 or more. However, since this is just "sufficient conditions", it is possible for a graph to not follow this theorem but still being a Hamiltonian graph.

It is worth noting that some graphs have the possibility of having combinations between Eulerian paths or circuits and Hamiltonian paths or circuits. Therefore, it is important for researches to do visual analysis and not just theorem checking to make sure a graph is classified as it's current shape and state.

IV. *DARK DECEPTION* AND *"MONKEY BUSINESS"* LEVEL

As written in *Introduction*, *Dark Deception* is a first-person horror game where the player are, according to the fan-wiki, trapped in a purgatory-like state. The player, introduced as Doug Houser later in the game, is tasked by the realm ruler, a deuteragonist named Bierce, to collect nine pieces of Riddle of Heaven, a ring said to give its bearer ultimate powers.

For the player to collect the ring piece, the player must enter nightmares from Doug's past, equipped with only a map and some abilities later on in the game, and collect soul shards while evading monsters in a maze-like environment. Soul shards, cited from the fandom-wiki online, are "the spiritual remnants of Malak's (the game's antagonist) victims, and must be collected in each nightmare to unlock the pieces of the Riddle of Heaven".

There are three types of soul shards, regular soul shards, "Enemy Reveal" shards, and "Stun Orb". The player is required to collect every regular soul shard in a level to retrieve the ring piece and exit the level. Regular soul shard are large and sharp purple crystal pieces in appearance while the "Enemy Reveal" shard look red and larger and the "Stun Orb" look like an orange sphere with glowing dots and traces of black on it. As the name suggests, an "Enemy Reveal" shard reveals all enemies in the map and a "Stun Orb" stuns all enemies. Both of these powerups only works for a limited amount of time.

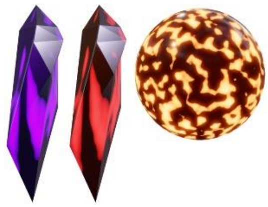

Fig. 10 Shards in Dark Deception
(Source : dark-deception-game.fandom.com/wiki/Soul_Shards)

While the main objective for the player is to evade enemies and collect all soul shard in a level, another important aspect is to reach the ring altar to collect the ring piece. Ring altar will only be unlocked after the player collects all souls shards, which means that the player need to reach the ring altar and return to the entrance portal to escape. The ring altar isn't necessarily placed in the entrance, as it differ by each level.

Fig. 11 Ring Altar
(Source : dark-deception-game.fandom.com/wiki/Soul_Shards)

In total, there are 12 levels in *Dark Deception*. The twelve levels are divided into 5 chapters by the developer, eacch with their own unique setting, environment, and gameplay. The only level of importance, as it will be explored deeper in this paper, is the only level in *Chapter 1: No Way Back*, *"Monkey Business"*.

Fig. 12 Starting Scene of Monkey Business' *Level*
(Source : dark-deception-game.fandom.com/wiki/Monkey_Business)

"Monkey Business" is the first level in the game. Placed in a hotel-like setting, the player starts the level by entering a hotel lobby and taking an elevator upstairs where the player can start collecting the soul shards and the ring piece.

On the main level, or upstairs in the game, after exiting the elevator, the player will see the ring altar right in front of the entrance (or the elevator). Then, to start the soul shard collection, the player need to break boards that separate the maze from the entrance. As previously mentioned, the player needs to collect every soul shard in the level while evading enemies, retrieve the ring piece from the ring altar, and return to the entrance of the level to escape.

Fig. 13 Starting Point of Monkey Business' *Level, with Ring Altar, Entrance/Exit Point, and Maze Entrance*
(Source : Personal Library)

One of the few things to be noted here about the level is the objectives, the level progression or Acts, and the enemies. In total, there are 289 shards (389 shards for an S rank). There are also two Acts in this level. The first Act is when the player attempts to collect all shards (the quantity depends on the rank target) while being chased by three *Murder Monkeys*. The second Act is when the player has successfully collected all shards and attempt to collect the ring piece. In this Act, the enemies are three *Chef Monkeys*, two of which can be seen guarding the ring altar while the other chases the player. The two *Chef Monkeys* that are guarding ring altar will follow and chase the player when the player is in their line-of-sight.

Fig. 14 Shard Collecting in the Maze
(Source : Personal Library)

While the player can outrun the *Monkeys*, they are always aware of the player's location and will relentlessly pursue him throughout the level. They will also take the shortest path to the player, allowing them to corner the player in some situations.

Fig. 15 Encountering a Murder Monkey
(Source : Personal Library)

Like many other games, there are also secrets hidden in rooms that can be opened by opening the correct door in the level. The secrets in this level reveal more about the plot and of the *Monkeys* themselves.

Below is the map of the *Monkey Business* level, including the maze, the elevator, the secret rooms, and the ring altar. The elevator and the ring altar are marked with the red dot while the secret rooms are marked with the yellow dot. It is worth noting that, while there are no purple dots to mark to soul shards in the map, every empty space in the map confined between two straight continous line is filled with soul shards, only separated by a tight space.

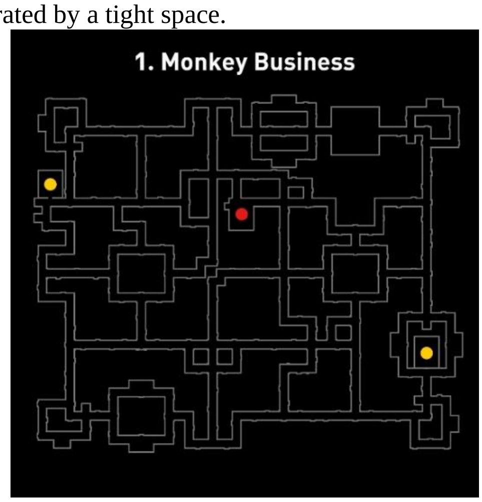

Fig. 16 Monkey Business' *Full Map*
(Source : twitter.com/daxing902)

However, to tighten the scope of this research, secret rooms and special shards such as "Enemy Reveal" shard and "Stun Orb" will be disregarded. This paper will also assume there's only Act 1 in the whole level for the entirety of the research. There will also be simplifications of the map. The map above contains irregular cavity that only functions as decoration and can be disregarded. For complicated turns and twists that have no intersections, we can also simplify this by using a straight road or line. With this in mind, the map above can be simplified into the map below. Similar to the original map, the red dot marks the ring altar and the entrance/exit.

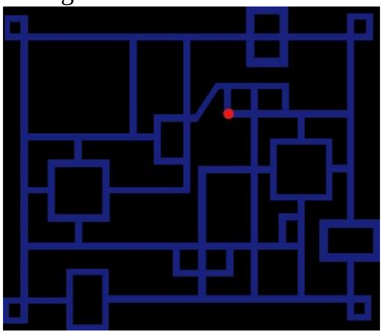

Fig. 17 Simplified Monkey Business' *Level Map*
(Source : Personal Library)

From the map above, we can derive an undirected graph by using the T-junctions or intersections as vertices and straight paths as edges. We use undirected graph as it is up to the player to decide the direction of the character from one vertex to another. It is assumed, judging by the gameplay and the screenshots provided beforehand, that all sour shards are located in the edges while vertices contain no soul shard. The graph, with the vertices numbered from 1 to 49, can be seen below.

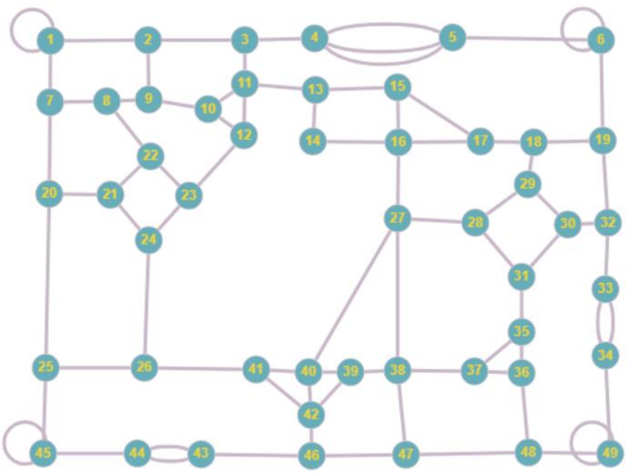

Fig. 18 Graph from Monkey Business' *Simplified Level Map*
(Source : Personal Library)

The first idea that can be considered is whether it is possible for the player to take all the soul shards in one go. This means that the player starts from the entrance, visits every edge in the maze and returns to the ring altar and exit the level without visiting the same edge. By definition, this is the term of an Eulerian graph.

The necessary condition for a graph to be Eulerian is to have every vertice in it contains even amount of degree. However, as seen above, there are multiple vertices that does not satisfy this condition, naming a few is vertex 3, 7, 8, 9, and 10. Therefore, this map, representated as the graph above, does not contain an Eulerian circuit, meaning it is not possible for the player to complete a full trip from starting point and back to the starting point again.

Even if an Eulerian circuit exists in a map, for example in later levels, it is not the most effective way to win this game. The reason being is that it is hard for a player to compute the Eulerian circuit and memorize it. Also there exists a chance that an enemy blocking the way that is in the circuit, forcing the player to take a detour.

Since it is impossible for the player to outrun every *Murder Monkeys*, collect all soul shards in one go, and return to the entrance or starting point, another viable strategy is to make sure the *Murder Monkeys* won't cut off the player from the maze. Essentially, this strategy ensures the player won't be cornered by the antagonists. We can calculate this using the cut-set terminology.

Cut-set in practice is defined as a set of edges that when "deleted" from the graph, will split the graph into two subgraphs. The idea behind this is that we can visualize two possible subgraphs from a graph, with one complementing the other. Now, we can recombine the two subgraphs, but differentiating the edges that connect them both. These graphs are elements of a sub-graph.

In the *Dark Deception* game, any edge where the antagonist is passing through is considered "cut" or "deleted". This means that the player is considered dead, even though not immediate, if the player happens to be cut off by the antagonists and ends up in one of the subgraph. Therefore, the elements of cut-set depends on the quantity of the antagonists in the level.

Since this paper only covers Act 1 of *Monkey Business*, there are three *Murder Monkeys* as antagonists. Therefore, the maximum amount of the cut-set members are three, with the minimum being one. From here, we can start by counting the amount of degree a vertex have. If the degree is less or equal to three, we can call that vertex "High Risk". This term is used to show that the player is vulnerable to being cornered while going in, being, or going out from the designated vertex. The complete list of "High Risk" vertices, as seen in Figure 17, is {1, 2, 3, 6, 7, 8, 9, 10, 12, 13, 14, 15, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 39, 41, 43, 44, 45, 46, 47, 48, 49}. In total, there exists 41 vertices that are "High Risk". This means that 83.67% of the intersections and T-junctions in the maze are dangerous and left the player vulnerable to cornering which ultimately led to their death.

It is worth noting that vertices 1, 6, 45, and 49 are also considered "High Risk" even though having a degree of four. The reason for this is that, if a *Murder Monkey* is inside the loop, for example the edge (1,1) in vertex 1, it only leaves the player to two exit routes, either edge (1,2) or (1,7), which is risky because there exists a possibility where there are two *Murder Monkeys* blocking the path.

Since there are multiple ways to create cut-sets from any combination of the "High Risk" vertices, it is not effective to list and memorize every subgraph and vertex to be avoided while being chased by *Murder Monkeys*. Since 83.67% of the map is high risk, it is more important and easier to memorize the safe points in the case when the player is chased.

As seen in Figure 18, vertices 4, 5, 11, 16, 27, 38, 40, and 42 are considered safe since they allow the player an escape route while being chased, even by all three *Murder Monkeys*. And since *Murder Monkeys* are programmed to chase the player using the shortest path possible, they would have no choice but to follow the player. This essentially makes every "High Risk" vertices a risk-free zone for the player to collect shards in a short period of time. But since there is no time limit, this strategy ensures the survivability of the player.

Even though higher levels introduce a new concept and new gameplay mechanics, the previous strategy is still practicable. By accounting the amount of enemies in the level, all the player needs to do is to find an intersection with the path going in or out the junction being at least one more than the amount of the enemies. This way, the player will not get trapped and free to explore the previously dangerous vertices and edges with all enemies behind the player for a short time.

V. Conclusion

The video game *Dark Deception* is a survival-horror maze-game where the player needs to collect "shards" in the maze, retrieve a "ring piece", and exit before the antagonists catches the player and kills him. While there exists different shards that can "buff" the player momentarily and powers that the player can use to their advantage as the game progresses, there exists a general strategy by using graph theory, specifically cut-set analysis.

The idea is for the player to count the amount of enemies in the level, and then searching for intersections (or vertices in graph) that have more exit and/or entrance routes at least one more than the amount of the enemies. This way, when one or multiple enemies are chasing the player, using the remaining escape route in one of the intersections can stack every enemy behind the player, making the player free to explore the dangerous and high risk parts of the maze.

VI. Acknowledgment

The author would like to thank Mr. Rinaldi Munir as lecturer of Discrete Mathematics Class 01 in Institut Teknologi Bandung for assigning this paper, as it is a way to find applications for the materials taught by Mr. Rinaldi in class. The author would also like to thank Mr. Ahmad Alfani Handoyo as a fellow classmate for without his help, some pictures and assets in this paper would not be possible to be presented.

References

[1] R. Johnsonbaugh, *Discrete Mathematics*. DePaul University, Chicago: Pearson, 1941.
[2] K. Rosen, *Discrete Mathematics and Its Applications Seventh Edition.* New York: McGrawHill.

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, 14 Desember 2020

Gede Prasidha Bhawarnawa - 13520004