

DEVELOPING A SPACE SYNTAX-BASED EVALUATION METHOD FOR PROCEDURALLY GENERATED GAME LEVELS

Emin Alp BIYIK, Department of Modeling and Simulation, Graduate School of Informatics, Middle East Technical University, Turkey, emin.biyik@metu.edu.tr

([id](https://orcid.org/0000-0002-2771-0489) <https://orcid.org/0000-0002-2771-0489>)

Elif SÜRER*, Department of Modeling and Simulation, Graduate School of Informatics, Middle East Technical University, Turkey, elifs@metu.edu.tr

([id](https://orcid.org/0000-0002-0738-6669) <https://orcid.org/0000-0002-0738-6669>)

Received: 20.07.2020, Accepted: 20.12.2020

Research Article

*Corresponding author

DOI: 10.22531/muglajsci.706895

Abstract

Procedural content generation (PCG) has been an essential catalyzer in the last decade with its efficiency in creating game elements such as textures, game levels, and maps. Despite being successfully applied in various studies, new reliable evaluation tools are still needed to assess the quality of the generated game content. One example limitation of procedurally generated game worlds is the lacking spatial configuration. To address this issue, in this study, an assessment method was developed to evaluate the spatial quality of procedurally generated game worlds. For this purpose, Space Syntax was used, which incorporates a set of methods to analyze spatial configurations and movement. The analyses were applied to a new game developed by the authors —the Haunted House— and the performance was evaluated in terms of integration, connectivity, and depth distance. Results show that changing the room dimensions (i.e., 15x15, 25x25, and 35x35 units) modifies the performance measures as well as game design parameters —number of the spawning points (ranging from 1 to 4), critical axes (1 to 5), to name a few. The proposed approach is a first attempt to create various improved spatial configurations and provide an evaluation tool to analyze the PCG algorithms in level design.

Keywords: Procedural Content Generation, Space Partitioning Algorithm, Space Syntax, Spatial Analysis

PROSEDÜREL OLARAK ÜRETİLMİŞ OYUN SEVİYELERİ İÇİN MEKAN DİZİLİMİ BAZLI DEĞERLENDİRME YÖNTEMİ GELİŞTİRİLMESİ

Özet

Prosedürel içerik üretimi, dokular, oyun seviyeleri ve haritalar gibi oyun unsurlarını yaratmadaki verimliliği ile son on yılda önemli bir katalizör olmuştur. Çeşitli çalışmalarda başarılı bir şekilde uygulanmasına rağmen, oluşturulan oyun içeriğinin kalitesini değerlendirmek için yeni ve güvenilir değerlendirme araçlarına hala ihtiyaç vardır. Prosedürel olarak oluşturulmuş oyun dünyalarının bir örnek sınırlaması, eksik mekansal konfigürasyondur. Bu sorunu ele almak için, bu çalışmada, prosedürel olarak oluşturulan oyun dünyalarının mekansal kalitesini değerlendirmek için bir değerlendirme yöntemi geliştirilmiştir. Bu amaçla, mekansal konfigürasyonları ve hareketi analiz etmek için bir dizi yöntem içeren Mekan Dizilimi (Space Syntax) kullanılmıştır. Analizler, yazarlar tarafından geliştirilmiş yeni bir oyun —Perili Ev— üzerinde uygulanmış ve yöntemin performansı entegrasyon, bağlantı ve derinlik mesafesi açısından değerlendirilmiştir. Sonuçlar, oda boyutlarının değiştirilmesinin (15x15, 25x25 ve 35x35 boyutlarında) performans ölçümlerinin yanı sıra oyun elementlerinin ortaya çıkma noktalarının sayısı (1 ila 4 arasında), kritik eksenler (1 ila 5) başta olmak üzere oyun tasarım parametrelerini de değiştirdiğini göstermektedir. Önerilen yaklaşım, çeşitli gelişmiş mekansal konfigürasyonlar yaratmaya yönelik bir ilk girişimdir ve seviye tasarımında prosedürel içerik algoritmalarını analiz etmek için bir değerlendirme aracı sağlamaktadır.

Anahtar Kelimeler: Prosedürel İçerik Üretimi, Mekan Bölümlenme Algoritması, Mekan Dizilimi, Mekansal Analiz

Cite

Biyık, E. A. and Sürer, E. (2020). "Developing a Space Syntax-Based Evaluation Method for Procedurally Generated Game Levels," *Mugla Journal of Science and Technology*, 6(2), 79-88.

1. Introduction

Procedural content generation (PCG) is a set of automated methods to create game content such as textures, sounds, and game levels using several algorithmic processes [1-2]. Existing PCG algorithms

have already reached an adequate pace and efficiency [3], but the lack of evaluation and optimization tools for the procedurally generated game content [4] is still an important missing feature. Previous studies show that spatial configuration problems may occur while using

PCG to create game worlds. Green et al. [5] demonstrate that constructive algorithms are powerful to create fast and various solutions for 3D game worlds, but the cellular automata algorithm has resulted in some unwanted spaces in building generation showing that algorithm-based game world generation requires an evaluation process. There are several approaches to evaluate procedurally generated game levels in the literature. For example, Liapis et al. [6] created a generic method based on three parameters; symmetry, area control, and exploration. However, their study does not focus on the spatial layouts of the game levels. To address this limitation, this study proposes to enhance the use of the Space Syntax methodology for game level assessment [7].

Space Syntax [7] is a set of theories and techniques to analyze spatial configurations, which suggests the analysis of the space by a network of choices and breaking it down into smaller components [8]. There is an increasing interest in Space Syntax among researchers working on game development in the last decade [9]. Levy et al. [10] implemented Space Syntax methods to predict the paths where game players may lose or find clues in an urban space. Their results showed that urban plans and characters of the architectural space might be helpful in predicting game players' behaviors and decisions. They also showed that players tend to move towards the points that have more visibility. Additionally, Song et al. [11] studied the connectivity parameter from Space Syntax to improve the connections and distributions of the roads. Although their study does not focus on game design but city evolution simulation, their promising results can be adapted to the game development pipeline. Cho et al. [12] used Space Syntax to explore the space recognition of online game players. Their results showed that there are important differences between the real world and virtual worlds. Therefore, developers should use alternative parameters for evaluation while creating virtual worlds. For example, they showed that online game players do not select the shortest path when they enter the game environment, but they are choosing the more axial direction, which is also an essential parameter as axiality in Space Syntax. They showed that the players have a bias towards choosing the items along a more axial line in the game world. Additionally, Tremblay et al. [13] used angular changes to control danger and threat in stealth games. Similar to the mean depth parameter, their parameters—line of sight and near-miss— suggest that angular changes can increase the challenges in games. Finally, Choi et al. [14] used the Space Syntax theory to place game objects in the game world. In video games, spawning is “the act of the live creation of game items, players and non-player characters (NPCs)” [15]. Tim Schafer's suggestion for the placement of the spawn points is summarized and defined in the study of Ballabio and Loiacono [15], where several heuristics for placing the spawn points in multiplayer shooter games are introduced, as follows: “spawn points should be positioned

in areas that are of low interest for the up-player and that are easy to leave, i.e., central hubs and dead ends are a bad choice, whereas rooms with 2 or 3 exits are usually the best option.”

In this study, by using the Space Syntax theory, game levels that are procedurally generated by a constructive algorithm—Space Partitioning algorithm— were evaluated in terms of integration, connectivity, and depth distance in a 2D dungeon-like game. For this purpose, a new game, the Haunted House, was developed by the authors. The study's primary focus is to create a systematic approach to assess the spatial quality of the completed game world, introduce spawning points based on the connectivity of the rooms, and decrease the configurational problems that may occur while generating game levels procedurally. As highlighted above, the focus of the game players is mainly on the axis line, and spawn points are better placed on the rooms having a higher number of exits (connectivity) that requires a combinatorial approach to be evaluated in terms of the design decisions on axiality and connectivity. This study is a first attempt to explore the Space Syntax methodology as a game level generation assessment tool while adapting its commonly used parameters under the game research domain, with a specific focus on room generation in dungeon-like games. The results show that adapting the Space Syntax methodology to game level design is a promising step that enables a quantitative level evaluation that guides the game designers to concretize their parameter choices.

2. Materials and Methods

In this section, the details of the 1) selected procedural content generation algorithm, 2) space partitioning, 3) Space Syntax methodology to evaluate the generated game world, and 4) a new implementation of a Haunted House-themed dungeon-like game as a case study to implement the suggested approach, are explained in detail.

2.1. Space Partitioning Algorithm

Space Partitioning algorithm is a constructive algorithm based on the division of 2D and 3D spaces into subsets in a hierarchical and recursive way [3-4]. These features allow the algorithm to create a space partitioning tree where each child's space is connected to its parent's space while preventing to create overlapping areas.

The pseudocode implemented in this study is as follows:

1. Define a base space (given width and height);
2. Divide the space by vertical and horizontal lines randomly, and add them to a tree structure;
3. Check if the new spaces can be divided;
4. Repeat step 2;
5. Repeat step 3;
6. Stop when the created spaces reach the pre-defined minimum area;

7. Create a room inside for each divided space by choosing random corner points;
8. Draw corridors between the nodes of the same parent (start from the youngest tree branches);
9. Go back to the previous layer of the tree structure and repeat step 8.

The process consists of three steps; a) space division and creating a space partitioning-tree, b) room generation, and c) corridor connections. The first step is to divide the main base into its subspaces, which is done using vertical and horizontal lines randomly (Figure 1 and Figure 2). The division process continues towards the minimum area defined in the algorithm till the minimum area threshold is reached. After the division step is completed, random corner points (mostly top-right and bottom-left nodes) are used to create space boundaries —i.e., to generate rooms.

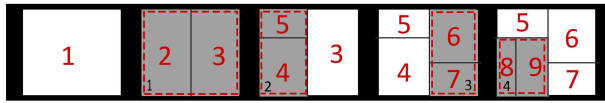


Figure 1. Space division.

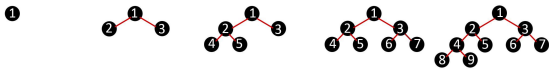


Figure 2. Binary space partitioning tree.

This process results in located rooms in each subspace (Figure 3). The final part of the algorithm is to create corridor connections between the rooms. Similar to the room creation, selected nodes are connected to create corridors, starting from the newest members of the space partitioning-tree (Figure 4).

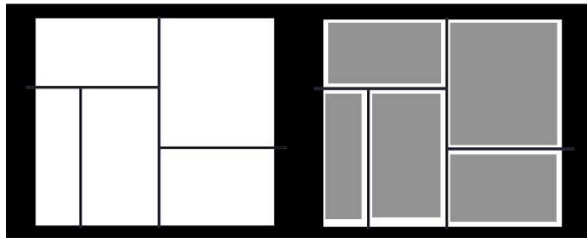


Figure 3. Room generation.

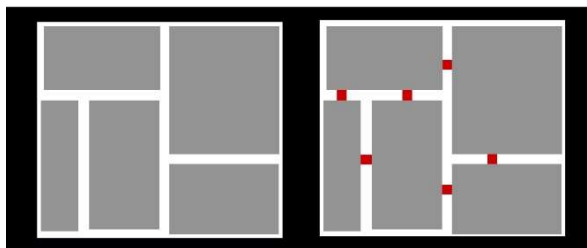


Figure 4. Corridor connection.

2.2. Space Syntax

Space Syntax is an analysis method for built environments, which are mainly represented as connected networks. The main aim of this method is to

understand possible user decisions and to retrieve the potentials of the spatial organizations. There are three basic concepts of this method, as defined in [16]:

- a) Isovist Space: the total area that is visible from axial lines and convex spaces;
- b) Axial Space: axial lines that cover all convex spaces and the connections of those spaces;
- c) Convex Space: space where no straight line is drawn between any two points.

In this study, three main concepts, integration, connectivity, and depth distance analyses, are used in order to explore the spatial potentials of the procedurally generated game levels. In these analyses, integration [17] refers to the average depth of the space with respect to all spaces and is derived from mean depth (MD) —the mean distance of all the axial lines from an axial line.

$$Dk = \frac{2 \left(K \left(\log_2 \left(\frac{k+2}{3} \right) - 1 \right) + 1 \right)}{(k-1)(k+1)} \quad (1)$$

, where K is local spaces, k is the number of lines and Dk is for relativization [18].

$$Integration = \frac{Dk}{\frac{2(MD-1)}{k-2}} \quad (2)$$

, where MD is the mean depth, k is the number of lines, and Dk is for relativization [18].

Connectivity describes measuring the number of neighbors directly connected to space, and depth distance is the number of steps needed to reach from one graph to another [17-19].

These analyses are made by using depthmapX, which is an open-source software and available on Space Syntax's official website [20].

Space Syntax methodology uses a color code to visualize analysis. In this color code, red points and lines represent higher values, while blue points and lines represent lower values [9]. The numeric values are colored in the graphs according to their quantities. However, there is not a binary measurement where red is equal to 1, and blue is equal to 0. Every set has blue-colored nodes and lines with their minimum value and red-colored nodes and lines with their maximum values.

2.3. The Haunted House Game

The Haunted House Game, which is developed by the authors, is a dungeon-like escape-the-room game where the player has to reach the exit before catching the enemies; while doing so, the player also has to take his/her children and collect the valuables to continue to play the game (Figure 5).

The game is developed by using Unity Game engine version 2019.2.12f.1. The game levels are generated

procedurally. Instead of randomly making decisions or leaving the game design process entirely to the game designer's intuition, analysis results were used as guidelines to predict the players' movements and set the pace of the levels.

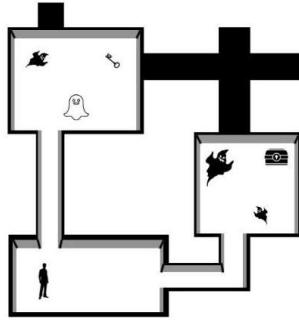


Figure 5. The initial design of the Haunted House game.

3. Results

In this study, procedurally generated game levels are evaluated using three main methods; i) visibility graph analysis (VGA), axial line analysis with three main selected parameters; ii) connectivity, integrity, mean depth; iii) spawning points and critical axes. To make these evaluations, three different test sets are prepared. The parameters for the test sets are:

1. 200x200 unit baseline with maximum (max) of 15-unit room width and max 15-unit room height;
2. 200x200 unit baseline with max 25-unit room width and max 25-unit room height;
3. 200x200 unit baseline with max 35-unit room width and max 35-unit room height.

For each set, three different procedurally generated game levels are generated using the abovementioned parameters. The graphs' representational method is based on color codes where red points have higher values and blue points have lower values in terms of connectivity, which is a viable parameter of the visibility graph analysis. In each set, three different procedurally generated game levels that have the same parameters are evaluated. Three different sets, each having three different game levels with the same parameters other than the room size (15x15, 25x25, 35x35 room size), are compared. Each test has a graph analysis and the numeric values in the tables from the graphs in connectivity and axial line graphs, and their numeric values in the tables from the graphs in integration and depth distance.

The results are shown in three different steps:

1. Connectivity analysis; based on VGA showing the possible spawning points where players, enemies, and other game elements can locate;
2. Axial line analysis; based on two different parameters; integration and mean depth,

showing where the critical axes in the game space are;

3. Quantitative analysis of the relation between spawning points and critical axes showing whether spawning points and critical axes have a connection.

As mentioned above, Space Syntax uses color codes to represent the range of values. In this study, blue areas and lines are called 'cold spaces' where interaction is expected to be limited in the game, and difficulty is lower. On the other hand, red areas and lines are expected to have more interaction in the gameplay with a higher difficulty level.

3.1. Connectivity

The connectivity results of the first set have the highest range between the minimum and maximum values (Table 1). The graph also shows that the generated levels have dark red and dark blue nodes (Figure 6).

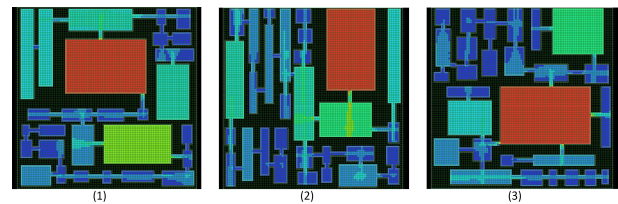


Figure 6. Visibility Graph Analysis (VGA) of the first set.

Table 1. VGA analysis of the first set — Connectivity results.

	(1)	(2)	(3)
Number of Connected Grids	6138	5809	6022
Mean	609.9	501.3	653.0
Std. Deviation	473.1	407.4	609.2
Minimum	23.0	29.0	21.0
Maximum	1434.0	1304.0	1661.0

The second set has the minimum mean in terms of connectivity (Table 2), while the difference between the minimum and maximum values is smaller than the first set and higher than the third set (Table 2). In contrast to the first set, the nodes have lighter red and light blue colors, which show that the difference between the nodes' connectivity values is smaller than the results of the first set (Figure 7).

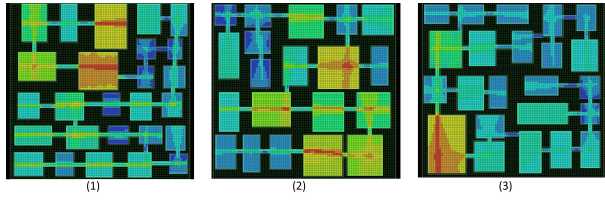


Figure 7. VGA analysis of the second set.

Table 2. VGA analysis of the second set — Connectivity results.

	(1)	(2)	(3)
Number of Connected Grids	5500	5992	5573
Mean	271.5	361.8	307.4
Std. Deviation	114.8	126.6	157.5
Minimum	74.0	131.0	60.0
Maximum	562.0	673.0	818.0

The third set has the minimum difference between the connectivity values of nodes (Table 3). Thus, the colors of the nodes do get lighter (Figure 8).

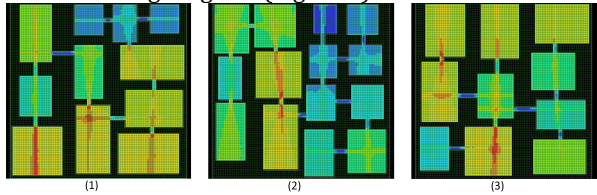


Figure 8. VGA Analysis of the third set.

3.2. Axial Line Analysis

The second analysis is based on the axial line analysis, which has two different parameters; integration and mean depth. The same test sets are used for this step. Since the integration is derived from the mean depth, the figures will have the same shapes, but their colors, where the color-coded scheme of the Space Syntax is used, will be different depending on their values. The following figures and tables show the axial lines that can be drawn between the convex spaces.

Table 3. VGA Analysis of the third set — Connectivity results.

	(1)	(2)	(3)
Number of Connected Grids	6136	6116	5931
Mean	648.1	515.6	623.9
Std. Deviation	182.6	159.9	119.8
Minimum	142.0	162.0	181.0
Maximum	999.0	966.0	946.0

The integration results of the first set show that this set has a minimum integration level. Results show that the 15x15-unit room enables more divided spaces (Figure 9). The number of axial lines created between the convex spaces is higher than the other two sets (Table 4).

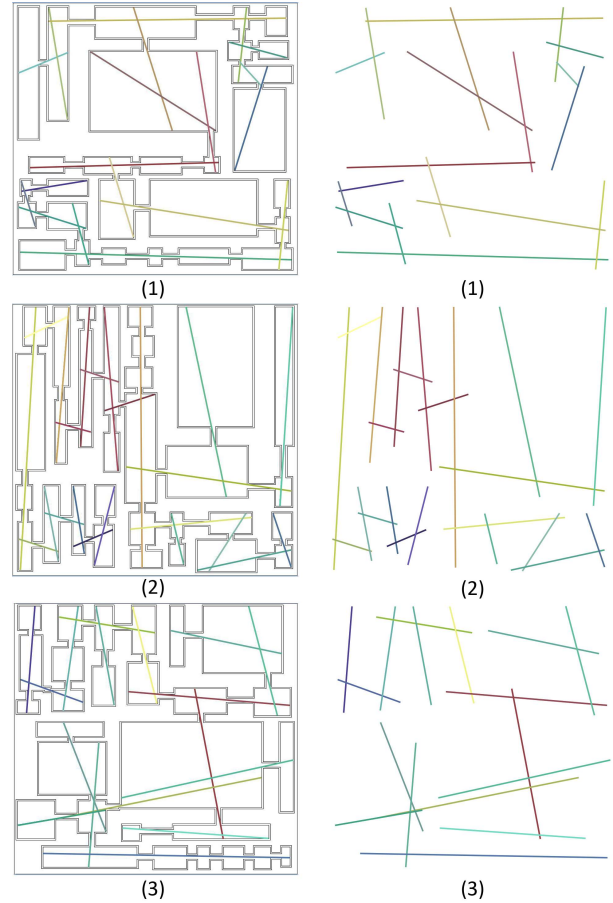


Figure 9. Axial line analysis of the first set — Integration results.

Table 4. Axial line analysis of the first set — Integration results.

	(1)	(2)	(3)
Number of Lines	359	378	339
Mean	1.3	1.2	1.7
Std. Deviation	0.3	0.3	0.4
Minimum	0.7	0.5	0.9
Maximum	1.9	1.8	2.6

The room parameters of the second set, the 25x25-unit room, result in bigger convex spaces, which means that there is less partition in the whole system (Figure 10).

This creates an increase in the maximum values of the integration results (Table 5).

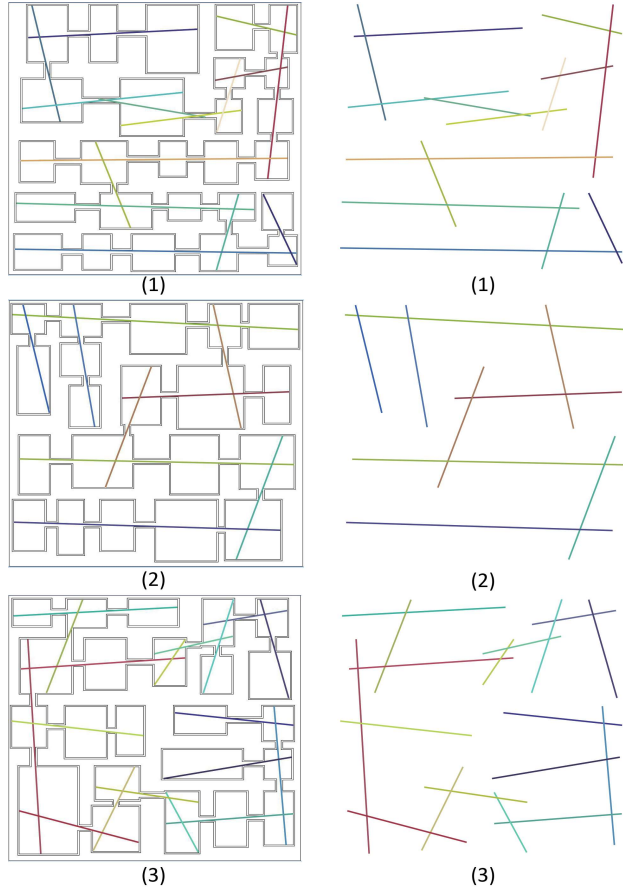


Figure 10. Axial line analysis of the second set — Integration results.

Table 5. Axial line analysis of the second set — Integration results.

	(1)	(2)	(3)
Number of Lines	294	221	236
Mean	1.6	1.8	1.2
Std. Deviation	0.4	0.4	0.3
Minimum	0.7	1.2	0.7
Maximum	2.5	2.7	1.9

The final set has the maximum room parameters (35x35-unit) and creates the largest convex spaces (rooms). This results in a decrease in the number of axial lines, the smallest number of all sets; 111, 142, and 100, respectively. This decrease in the partition of the rooms creates a more integrated spatial configuration (Figure 11)(Table 6).

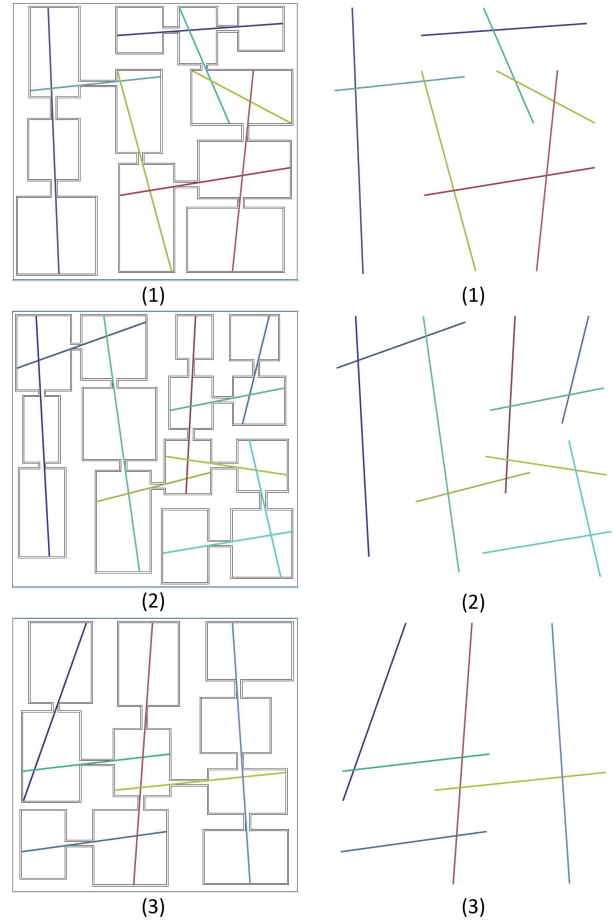


Figure 11. Axial line analysis of the third set — Integration results.

Table 6. Axial line analysis of the third set — Integration results.

	(1)	(2)	(3)
Number of Lines	111	142	100
Mean	1.8	1.8	2.7
Std. Deviation	0.5	0.5	1.0
Minimum	1.1	1.0	1.4
Maximum	2.7	3.0	5.2

The second analysis of the set shows the mean depth and the angular turns that are needed to reach from one point to the other. Similar to the integration parameters, the 15x15-unit room has resulted in more divided spaces (Figure 12). The number of axial lines is the highest in the first set; 359, 378, and 339. This set also has the highest values in terms of the depth distances (Table 7).

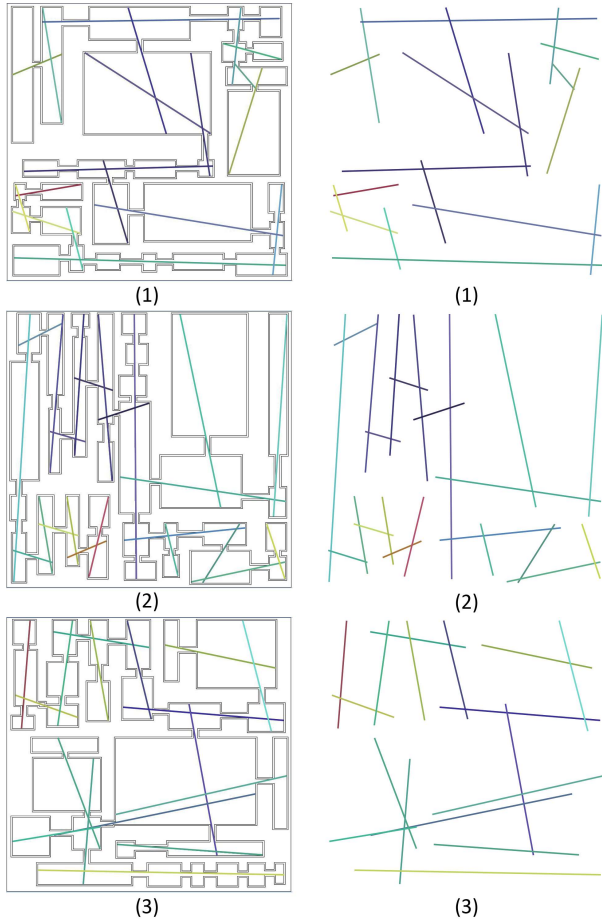


Figure 12. Axial line analysis of the first set — Mean Depth results.

Table 7. Axial line analysis of the first set — Mean Depth results.

	(1)	(2)	(3)
Number of Lines	359	378	339
Mean	5.7	6.3	4.7
Std. Deviation	1.2	1.8	0.9
Minimum	4.0	4.3	3.2
Maximum	9.4	12.8	7.1

The results of the second set also have similarities with the integration results. Due to the room size (25x25-unit room), the algorithm creates bigger convex spaces, resulting in a decrease in the number of axial lines (Figure 13). This set has higher mean depth values than the third set and lower mean depth values than the first step. Additionally, the range between the minimum and maximum values of the mean depth is getting smaller. When the convex spaces have higher dimensions, the mean depth decreases (Table 8).

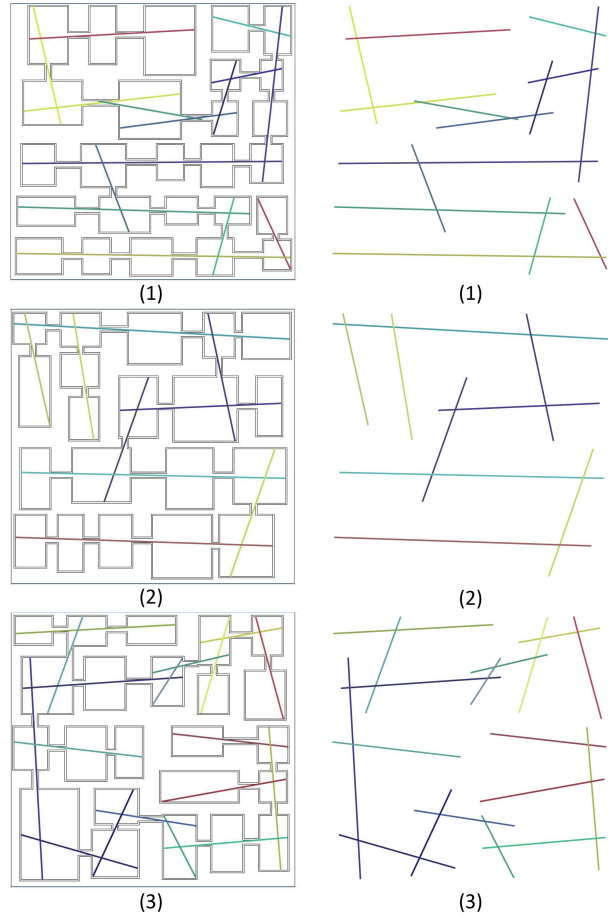


Figure 13. Axial line analysis of the second set — Mean Depth results.

Table 8. Axial line analysis of the second set — Mean Depth results.

	(1)	(2)	(3)
Number of Lines	294	221	236
Mean	5.0	4.1	5.6
Std. Deviation	1.5	0.7	1.2
Minimum	3.3	3.0	3.7
Maximum	9.4	5.3	8.7

The final test set, which has the 35x35-unit room, creates the biggest convex space. The minimum spatial partition in all the space results in fewer axial lines (Figure 14). In terms of the mean depth, this set has the lowest values amongst the three sets; thus, in order to reach from one point to the other, less angular changes are needed. The range between the maximum and minimum values of the mean depth is also getting narrower (Table 9).

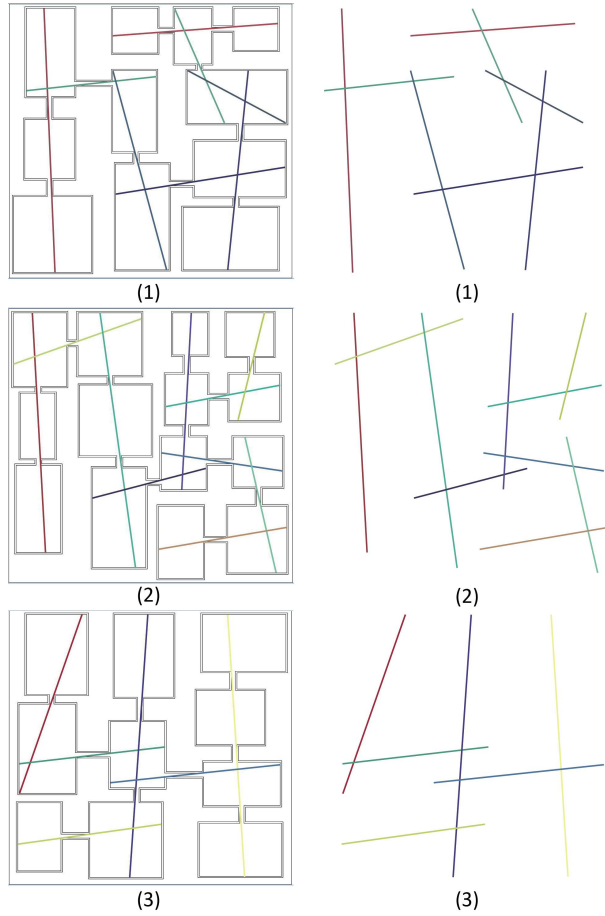


Figure 14. Axial line analysis of the third set — Mean Depth results.

Table 9. Axial line analysis of the third set — Mean Depth results.

	(1)	(2)	(3)
Number of Lines	111	142	100
Mean	3.6	3.7	2.6
Std. Deviation	0.8	0.8	0.5
Minimum	2.6	2.5	1.8
Maximum	4.9	5.7	3.9

3.3. Connection Between the Spawning Points and Critical Axes

The evaluation's last step is to check the condition between spawning points and critical axes (i.e., the axial line that has the highest value)(CA) in the game levels (Table 10). Spawning points are the red nodes and rooms in the VGA analysis, and critical axes are the red lines in the axial line analysis with the integration parameter. Spawning points (SP) are defined as the location where

continuous red nodes occur; in Space Syntax terminology, these nodes are called decision points. The generated levels have been investigated in terms of the number of SP, CA, and their connections (Table 10).

Table 10. The number of Spawning Points, Critical Axes, and Spawning Points-Critical Axes Connection.

	Spawning Points (SP)	Critical Axes (CA)	SP-CA Connection
Level 1.1	1	2	1
Level 1.2	1	5	0
Level 1.3	1	2	1
Level 2.1	2	1	0
Level 2.2	4	1	1
Level 2.3	1	3	1
Level 3.1	3	2	2
Level 3.2	3	1	0
Level 3.3	3	1	1

4. Discussion

In this study, a Space Partitioning algorithm was implemented and tested with the selected Space Syntax parameters; connectivity, integration, and mean depth. This test aims to search for a possible way to evaluate the procedurally generated game levels quantitatively. The results showed that it is possible to systematically assess the game levels' potential and create a connection between the algorithm parameters and their effects on the procedurally generated game levels. Due to the lack of a single or standard solution for the level design, instead of binary classification (successful or unsuccessful), a categorization scheme is developed. In this way, different solutions are found to the level design problem and its difficulty assessment. For example, the difficulty of the level can be arranged by using depth distance in which a larger number signifies a more difficult level because the player has to explore a wider space to finish the game. Additionally, the connectivity and integration parameters can be applied to the game objects' placements, which have central effects on the level design. In order to use the Space Syntax analysis on the Space Partitioning algorithm, a dungeon-like escape-the-room game is developed. All the parameters are applied to the game to control if the evaluation results create expected outcomes in the game environment. The first results of the tests are based on the visibility graph analysis. These tests show that increasing the room dimension in the algorithm results in different connectivity values. When the dimensions of the room are smaller, the algorithm creates more divided spaces. This partition of the rooms results in an increase in the number of convex spaces. Thus, rooms with different connectivity values are concurrently in the system

(Figure 6 and Table 1). Increasing room dimensions leads to generating rooms with similar connectivity values. An important outcome is that smaller room dimensions generate denser networks. On the other hand, bigger room dimensions create fewer dense areas with higher node connectivities (Figure 7 and Figure 8). The highly connected nodes are critical places where decision points are generated. The highly visible nodes become spawning points of the game objects. From the game design perspective, connectivity may be a complex term to evaluate. Instead of creating nodes having similar connectivity values, game developers may aim to create game levels having different connectivity values, which result in better gameplay experiences. For example, increasing the room dimension generates game levels that have similar connectivity values, which can cause repetitive gameplay and, therefore, should be avoided. Using smaller room dimensions may result in better gameplay experiences because of the high range between less connected nodes and more connected nodes. These decision points may be used as critical points in the game design where the enemies, power-ups, and checkpoints are spawned. Another advantage is providing a hierarchy between different parts of the game levels. In this way, game developers have the control to locate the game objects according to the importance of the nodes in terms of visibility.

The second test of this study is based on axial line analysis where two different parameters, integration and mean depth, are applied. The former represents how the system is integrated spatially, and the latter represents the number of angular turns that are used in order to reach from one point to another—increasing the room dimensions in the algorithm results in bigger convex spaces in the integration analysis. Bigger rooms create fewer spatial divisions, resulting in more integrated game levels (Table 6). In terms of the game design, partial divisions in game levels may become better solutions instead of creating a completely integrated game level. Therefore, this partitioning approach is potentially more useful to control the different aspects of the game.

The mean depth parameter is directly connected to the player's movement. This parameter measures the angular change needed to reach from one point to another. In the algorithm, smaller room dimensions potentially create more divided convex spaces requiring more angular changes to reach a specific point (Table 7 and Figure 12). As the room dimensions increase, the convex spaces become bigger. As a result, reaching a point requires less angular changes because axial lines between two different points become more linear and longer (Table 9 and Figure 14). The mean depth parameter can be used to control both the game difficulty and gameplay time. When the game developer aims to create a more difficult level, higher mean depth values can be used. In this way, players need to change their rotation to reach a specific point, which eventually provides longer gameplay time.

Two different testing methods of the space partitioning algorithm demonstrate that game developers can have control in analyzing the spatial quality of different game levels. Further, the game developer may tune the parameters of the selected procedural content generation algorithm. The results of this study can be summarized in two main directions; i. its impact on game level analysis, ii. gaining intuition about the parameters of the algorithm.

Visibility graphs and axial line analyses are performed as guidelines in making game design decisions. The mean depth parameter is used to select the beginning and the ending part of the game world. High mean depth implies a more complex game world due to the need for angular changes to reach the endpoint of the game. Connectivity is a visibility graph analysis parameter that is tuned to locate the enemies, door keys, and power-ups. This parameter provides a controlled way to make these decisions instead of locating the elements randomly. The integration parameter is used to divide the game world so that the spaces that have different integration values can be used to enhance the gameplay. Also, if there is a connection between spawning points and critical axes, these levels may have better gameplay due to the increase in interaction between the player and the game world.

In the newly developed video game, *The Haunted House*, the less integrated parts of the game are used as a tutorial and training section. Before starting to play more integrated sections, the player is trained in these areas. It also enables the game developer to control the learning process of the players. This method offers a structured way for the game design process, which can be quite complex while the game developer does not have measurable parameters in design decision making.

Overall, these results show that the Space Syntax methodology can efficiently evaluate procedurally generated game levels and control the game design decisions. Selected parameters enable the game developer to test and assess the levels before moving further in the game development pipeline.

5. Conclusion

In this study, Space Syntax-based evaluation method was developed to assess the quality of procedurally generated game levels in terms of connectivity, integration, and mean depth. Two different evaluation methods—visibility graph analysis and axial line analysis—were applied to analyze the game levels that were generated using a constructive algorithm—the Space Partitioning algorithm. A new game, *The Haunted House*, and three different test sets, which have different room dimensions, were generated. The results show that by using the Space Syntax approach, game developers can evaluate the procedurally generated game levels systematically while testing and controlling different parameters and enhancing the game levels. By using the proposed method, it is also possible to prevent spatial configuration-related problems while designing game levels.

6. References

- [1] Smelik, R. M., Tutenel, T., Bidarra, R., Benes, B., "A survey on procedural modelling for virtual worlds", *Computer Graphics Forum*, 33 (6), 31-50, 2014.
- [2] Barriga, N. A., "A Short Introduction to Procedural Content Generation Algorithms for Videogames", *International Journal on Artificial Intelligence Tools*, 28 (02), 1930001, 2019.
- [3] Shaker, N., Togelius, J., Nelson, M. J., *Procedural content generation in games*. Switzerland: Springer International Publishing, 2016.
- [4] Van Der Linden, R., Lopes, R., Bidarra, R., "Procedural generation of dungeons", *IEEE Transactions on Computational Intelligence and AI in Games*, 6 (1), 78-89, 2013.
- [5] Green, M. C., Salge, C., Togelius, J., "Organic building generation in minecraft", In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, 2019, 1-7.
- [6] Lianis, A., Yannakakis, G. N., Togelius, J., "Towards a generic method of evaluating game levels". In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013, 30-36.
- [7] Bafna, S., "Space syntax: A brief introduction to its logic and analytical techniques", *Environment and behavior*, 35 (1), 17-29, 2003.
- [8] Al-Sayed, K., *Space syntax methodology*, 2014.
- [9] UCL-Space Syntax Official Website, <http://otp.spacesyntax.net/>, Retrieved on July 20, 2020.
- [10] Levy, R. M., O'Brien, M. G., Aorich, A., "Predicting the methbehavior of game players-space syntax and urban planning theory as a predictive tool in game design", In *15th International Conference on Virtual Systems and Multimedia*, 2009, 203-208.
- [11] Song, A., Whitehead, J., "TownSim: Agent-based city evolution for naturalistic road network generation", In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, 2019, 81.
- [12] Cho, I. O., Kim, Y. O., "The relationship between spatial configuration and spatial behavior in online game space", In *Proceedings Sixth International Space Syntax Symposium*, 2007, 1-14.
- [13] Tremblay, J., Torres, P. A., Verbrugge, C., "Measuring risk in stealth games", In *FDG*, 2014, 1-8.
- [14] Choi, S. K., Kim, D. H., Kim, Y. O., "A study on the placement of game objects using space syntax", *Journal of Korea Game Society*, 12 (5), 2012.
- [15] Ballabio, M. and Loiacono, D., "Heuristics for Placing the Spawn Points in Multiplayer First Person Shooters", In *2019 IEEE Conference on Games (CoG)*, 2019, 1-8.
- [16] Martin, P., *Space and place as expressive categories in videogames*, 2011.
- [17] Hillier, B. and Hanson, J., *The social logic of space*, Cambridge university press, 1989.
- [18] Asami, Y., Kubat, A.S., Kitagawa, K., Iida, S.I., "Introducing the third dimension on Space Syntax: Application on the historical Istanbul", In *Proc. 6th International Space Syntax Symposium*, 2003, 48-49.
- [19] El-Agouri, F.A., *Privacy And Segregation as A Basis For Analyzing and Modelling The Urban Space Composition of The Libyan Traditional City Case Study: The City of Ghadames*, 2004.
- [20] DepthMap Tool GitHub Page, <https://github.com/varoudis/depthmapX/>, Retrieved on July 20, 2020.