



UNDERGRADUATE THESIS

APPLICATION OF GRAPH GRAMMAR METHOD TO PROCEDURAL LEVEL GENERATION WITH CONDITIONAL ACCESS MECHANISM IN 2D PLATFORMER

AKMAL ZIDAN FAHREZA
NPM 22081010163

THESIS ADVISORS

Chrystia Aji Putra, S.Kom, M.T.
Yisti Vita Via, S.ST, M.Kom.

**MINISTRY OF HIGHER EDUCATION, SCIENCE, AND TECHNOLOGY
UNIVERSITAS PEMBANGUNAN NASIONAL VETERAN JAWA TIMUR
FACULTY OF COMPUTER SCIENCE
INFORMATICS STUDY PROGRAM
SURABAYA
2026**

APPROVAL SHEET

APPLICATION OF GRAPH GRAMMAR METHOD TO PROCEDURAL LEVEL GENERATION WITH CONDITIONAL ACCESS MECHANISM IN 2D PLATFORMER GAMES

By:
AKMAL ZIDAN FAHREZA
NPM. 22081010163

It has been defended in front of and accepted by the Thesis Examiner Team of the Informatics Study Program, Faculty of Computer Science, National Development University of East Java Veterans on September 02, 2026:

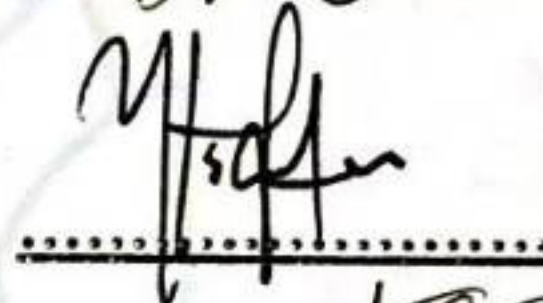
Approved,

Chrystia Aji Putra, S.Kom, M.T
NIP. 19861008 202121 1 001



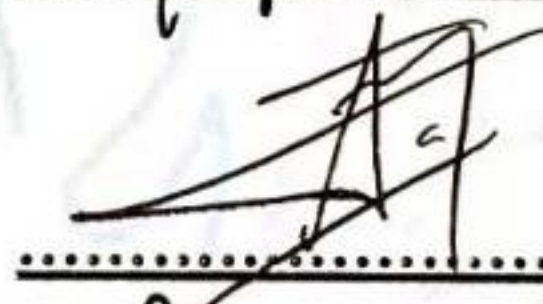
(Supervisor I)

Yisti Vita Via, S.ST, M.Kom.
NIP. 19860425 202121 2 001



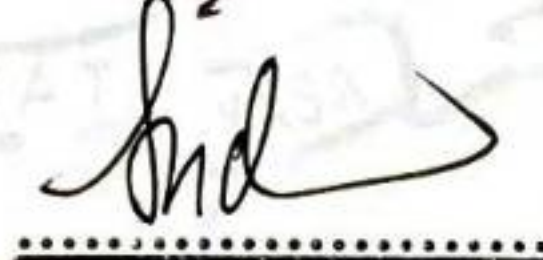
(Supervisor II)

Dr. Firza Prima Aditiawan, S.Kom, M.T.I,
M.C.F, M.O.S
NIP. 19860523 2021211 003



(Chief Examiner)

Andreas Nugroho S., S.Kom., M.Kom.
NIP. 19900412 202406 1 003



(Examiner I)

Known by,

Dean of the Faculty of Computer Science



Prof. Dr. Ir. Novirina Hendrasarie, MT.
NIP. 19681126 199403 2 001

APPROVAL SHEET

APPLICATION OF GRAPH GRAMMAR METHOD TO PROCEDURAL LEVEL GENERATION WITH CONDITIONAL ACCESS MECHANISM IN 2D PLATFORMER COVER GAME

By:
AKMAL ZIDAN FAHREZA
NPM. 22081010163

Approved to proceed to the Thesis Exam

Approved by,

Informatics Study Program Coordinator
Faculty of Computer Science



Dr. Intan Yuniar Purbasari, S.Kom. MSc.

NIP. 19800602 202521 2 029

STATEMENT OF ORIGINALITY

I am the undersigned:

Student Name : Akmal Zidan Fahreza
NPM : 22081010163
Degree Program : Bachelor (S1)
Study Program : Informatics
Faculty : Faculty of Computer Science

Hereby declares that this undergraduate thesis contains no part of any other scientific work that has been submitted to obtain an academic degree at any higher education institution. Furthermore, it does not contain any work or opinions previously written or published by others, except for those which are explicitly cited in this thesis and listed completely in references.

And I declare that this scientific document is free from elements of plagiarism. If in the future indications of plagiarism are found in this Thesis, I am willing to accept sanctions in accordance with the applicable laws and regulations.

Thus, I made this statement without any coercion from anyone and to be used as it should.



Surabaya, September 03 2026

Who made the statement,



AKMAL ZIDAN FAHREZA
NPM. 22081010163

ABSTRACT

Student Name / NPM : Akmal Zidan Fahreza / 22081010163

Thesis Title : APPLICATION OF GRAPH GRAMMAR METHOD
TO PROCEDURAL LEVEL GENERATION WITH
CONDITIONAL ACCESS MECHANISM IN 2D
PLATFORMER GAME

Supervisor : 1. Chrystia Aji Putra, S.Kom, M.T
2. Yisti Vita Via, S.ST, M.Kom.

Procedural Content Generation (PCG) is an approach that can be used to generate game content automatically, including game levels. One challenge in procedural level generation is producing a consistent gameplay flow that can be translated into a playable physical layout. This research aims to implement the Graph Grammar method to produce a logical structure for a 2D platformer game, translate the graph structure into chunk-based physical levels, and measure playability using Breadth-First Search (BFS) validation. The Graph Grammar method is applied through three generation rules: Linear Expansion, Branching, and Key-Lock Mechanism. The resulting graph structure is converted into a physical level by mapping each node into a chunk and each relationship into a connection between spaces. Level validation is based on two criteria: the successful establishment of the Key-Lock mechanism and BFS successfully finding a path from the Start node to the Goal node. Testing was conducted using four scenarios of 10, 20, 30, and 50 iterations. The results showed validity levels of 95% for 10 iterations, 99% for 20 iterations, and 100% for 30 and 50 iterations. Failures were mainly caused by the Key-Lock mechanism being unable to form due to limited free space in dense level structures. Overall, the results demonstrate that the Graph Grammar method can generate consistent 2D platformer levels, translate graph structures into chunk-based levels, and achieve high playability based on the applied validation criteria.

Keywords: Procedural Content Generation, Graph Grammar, Level Structure, 2D Platformer, Breadth-First Search

ACKNOWLEDGEMENTS

Praise be to Allah SWT for all His graces, guidance and gifts to the author so that the thesis with the title **"THE APPLICATION OF THE GRAPH GRAMMAR METHOD IN THE GENERATION OF PROCEDURAL LEVELS WITH CONDITIONAL ACCESS MECHANISMS IN 2D PLATFORMER GAMES"** can be completed properly.

The author would like to thank Mr. Chrystia Aji Putra, S.Kom, M.T as the main Supervisor and Mrs. Yisti Vita Via, S.ST, M.Kom. who have taken the time to provide guidance, advice and motivation to the author. The author also received a lot of help from various parties, be it in the form of moral, spiritual or material. For this the author would like to thank the following:

1. Prof. Dr. Ir. Novirina Hendrasarie, MT., Dean of the Faculty of Computer Science at the “Veteran” National Development University of East Java.
2. Dr. Intan Yuniar Purbasari, S.Kom., M.Sc., Chair of the Informatics Study Program at the Faculty of Computer Science, “Veteran” National Development University of East Java.
3. Dr. Firza Prima Aditiawan, S.Kom., M.T.I., M.C., and Andreas Nugroho Sihananto, S.Kom., M.Kom., as examiners who provided constructive criticism and suggestions to the author.
4. Chrystia Aji Putra, S.Kom., M.T., and Yisti Vita Via, S.ST., M.Kom., as the academic advisors who dedicated their time and expertise to guiding the author throughout this research.
5. All faculty members of the Computer Science Program, for their efforts in imparting knowledge, academic ethics, and an education that has been invaluable to the author.
6. The author’s parents and extended family, who have provided all kinds of support and prayers, enabling the author to successfully complete this research and studies.
7. The author’s dear friends, Muhammad Ridho Habibie, Maulana Muzakki Bhaswara, Alfianita Putri Syafnadina, and Niken Febrinikmah Siharta, who consistently provided unwavering support and encouragement from the beginning to the end of the author’s studies.

8. To all my classmates in the Computer Science Department, Class of 2022, for your extraordinary friendship, the struggles we've gone through together, our mutual support, and the constant motivation you've given me from the very beginning

The author realizes that in the preparation of this thesis there are many shortcomings. For this reason, constructive criticism and suggestions from all parties are highly expected for the perfection of writing this thesis. Finally, with all the limitations that the author has, hopefully this report can be useful for all parties in general and the author in particular.

Surabaya, 02 September, 2026

Author

TABLE OF CONTENTS

APPROVAL SHEET	iii
APPROVAL SHEET	v
STATEMENT OF ORIGINALITY	vii
ABSTRACT	ix
ACKNOWLEDGEMENTS.....	xi
TABLE OF CONTENTS.....	xiii
IMAGE LIST.....	xvii
TABLE LIST	xxi
CODE LIST	xxiii
CHAPTER I INTRODUCTIONS	1
1.1 Background	1
1.2 Problem Formulation	4
1.3 Research Objectives	4
1.4 Research Benefits.....	4
CHAPTER II LITERATURE REVIEW	7
2.1 Previous Research	7
2.2 Theoretical Foundations.....	9
2.2.1 Game Concept.....	9
2.2.2 Genre 2D Platformer	9
2.2.3 Chunk	10
2.2.4 Procedural Content Generation (PCG).....	11
2.2.5 Traditional Method.....	11
2.2.6 Search-Based Method	11
2.2.7 Evaluation of the Procedural Content Generation Algorithm.....	12
2.2.8 Graph Theory	13
2.2.9 Undirected Graph	13
2.2.10 Labeled Graphs	14
2.2.11 Graph Grammar	15
2.2.12 Production Rules	15

2.2.13	Mission Graph & Space Graph.....	16
2.2.14	Breadth-First Search.....	17
2.2.15	Unity Engine.....	17
2.2.16	Game Asset.....	18
CHAPTER III RESEARCH METHODOLOGY.....		19
3.1	Research Stages	19
3.2	Literature Studies.....	20
3.3	Graphics Design and Game Environment	20
3.3.1	Graph Planning	20
3.3.2	Game Environment Design	22
3.4	Planning Rules Graph Grammar	24
3.4.1	Linear Expansion.....	24
3.4.2	Branching	25
3.4.3	Key-Lock Mechanism	26
3.5	Graph Structure Generation and Validator Implementation (BFS).....	26
3.5.1	Graph Generation Algorithm.....	27
3.5.2	Validation Algorithm.....	34
3.5.3	Convert to Game Level.....	37
3.6	Trial Scenario	38
3.6.1	Testing Environment	39
3.6.2	Test Parameters.....	39
3.6.3	Testing Procedure	40
3.6.4	User Acceptance Test (UAT)	41
3.7	Results Analysis	43
CHAPTER IV RESULT AND DISCUSSION		45
4.1	System Implementation	45
4.1.1	Development of PCG Modular Units (Chunk).....	45
4.1.2	Inter-Spatial Connector System.....	55
4.1.3	Mapping Each Node on the Chunk.....	59
4.2	Implementation of Graph Grammar Rules	61
4.2.1	Linear Expansion Rule	61
4.2.2	Branching Rule	65

4.2.3	Key-Lock Mechanism Rules.....	67
4.3	Level Structure Generation	69
4.3.1	Graph Structure Initiation	69
4.3.2	Graph Grammar Rules Execution	70
4.3.3	Graph Structure Instances to Game Level	71
4.3.4	BFS Implementation as Level Structure Validation	73
4.4	System Testing.....	78
4.4.1	Testing with 10 Iteration Scenarios	78
4.4.2	Testing with 20 Iteration Scenarios	79
4.4.3	Testing with 30 Iteration Scenarios	80
4.4.4	Testing with 50 Iteration Scenarios	82
4.5	User Acceptance Test (UAT).....	83
4.6	Analysis of Test Results.....	86
CHAPTER V CONCLUSIONS AND RECOMENDATIONS		89
5.1	Conclusions	89
5.2	Recommendations	90
BIBLIOGRAPHY		91
ATTACHMENT.....		95

IMAGE LIST

Image 2.1 2D Illustration Platformer based grid.....	10
Image 2.2 System Chunk on Games Spelunky	10
Image 2.3 Directionless Graphs	14
Image 2.4 Labeled Graphs	14
Image 2.5 Example Mission Graph.....	16
Image 2.6 Space Graph Example	17
Image 2.7 Creators of Tileset Used.....	18
Image 3.1 Research Flow Diagram	19
Image 3.2 Graph Data Structure.....	21
Image 3.3 Attributes On Node	21
Image 3.4 Representation Node As Chunk	22
Image 3.5 Chunk in Grid.....	22
Image 3.6 Platformer Environment.....	23
Image 3.7 Standardization of Connection Points Chunk	23
Image 3.8 Illustration of the Operation Rules Linear Expansion.....	24
Image 3.9 Illustration of the Operation Branching	25
Image 3.10 Illustration Rules Key-Lock.....	26
Image 3.11 Graph Generation Flowchart and BFS Implementation.....	27
Image 3.12 Simple Graph Initiation.....	28
Image 3.13 Selection Flowchart Rules.....	29
Image 3.14 Flowchart Rule Evaluation.....	30
Image 3.15 Linear Expansion Evaluation	31
Image 3.16 Branching Evaluation.....	32
Image 3.17 Key-Lock Evaluation	32
Image 3.18 Example of Graph Transformation Results with 10 Iterations	33
Image 3.19 BFS Illustration in Visiting Graph	34
Image 3.20 Graph Testing Flowchart	36
Image 3.21 Valid and Invalid Structures.....	37
Image 3.22 In-game level conversions	38
Image 3.23 Test Flow Diagram.....	39

Image 4.1 Design Chunk with Type Home	46
Image 4.2 "GoalHeart" Item That Indicates Level Completion	48
Image 4.3 Player Character Picking Up "GoalHeart" Item	48
Image 4.4 Chunk Design with Goal Type	49
Image 4.5 Design Chunk with Normal Type.....	50
Image 4.6 Design Chunk with Treasure Type.....	51
Image 4.7 Key As a Functional Item to Open the LockedDoor	52
Image 4.8 Player Character Retrieves the "KeyPrefab" Item.....	53
Image 4.9 Design Chunk with Key Type	53
Image 4.10 Player Character Finds a Locked Door.....	54
Image 4.11 Player Character Successfully Unlocks the Door with Key Items	55
Image 4.12 Chunk Design with Lock Type.....	55
Image 4.13 Space connectors on all four sides of the chunk.....	56
Image 4.14 Example of Level That Is Not Connected Between Chunks	58
Image 4.15 Example of a Level That Has Been Connected Between Chunks.....	59
Image 4.16 Linear Expansion Expands Levels Sideways	63
Image 4.17 Linear Expansion Expands Levels Downwards	63
Image 4.18 Levels with Execution Linear Expansion Rule 5 times.....	64
Image 4.19 Linear Expansion Execution Log	64
Image 4.20 Initial Conditions Before Branching Rules Execution	66
Image 4.21 Branching Rule Execution Results Node Treasure	66
Image 4.22 Branching Rule Execution Log	67
Image 4.23 Normal Node Pairs for Key-Lock Candidates.....	68
Image 4.24 Key-Lock Execution Results in Chunk Keys and Locks	68
Image 4.25 Key-Lock Mechanism Execution Log.....	69
Image 4.26 Visualization of the Initial Structure of the Level with Two Node	70
Image 4.27 Sample Graph with 10 Iterations	72
Image 4.28 Sample Results of Graph Conversion to Game Layout.....	72
Image 4.29 Sample Level with 10 Iterations That Will Be Subject to BFS.....	74
Image 4.30 BFS Search in Tree Diagram.....	77
Image 4.31 10 Iteration Testing Documentation	79
Image 4.32 20 Iteration Test Documentation	80

Image 4.33 30 Iteration Testing Documentation	82
Image 4.34 50 Iteration Test Documentation.....	83

TABLE LIST

Table 3.1 Weighting Rules.....	29
Table 3.2 Mapping Node in the Physical Room	38
Table 3.3 User Acceptance Test (UAT) Question Instrument.....	42
Table 3.4 User Acceptance Test (UAT) Assessment Scale	42
Table 4.1 Mapping Node other Chunk.....	60
Table 4.2 Graph Generation Execution Log with 10 Iterations	71
Table 4.3 Graph Search Logs with BFS	75
Table 4.4 10 Iteration Scenario Testing Data.....	78
Table 4.5 20 Iteration Scenario Testing Data.....	80
Table 4.6 30 Iteration Scenario Testing Data.....	81
Table 4.7 50 Iteration Scenario Testing Data.....	82
Table 4.8 UAT Result	84
Table 4.9 Test Performance from Four Scenarios	86

CODE LIST

Code 4.1 Codes for Player Character SpawnPoint	46
Code 4.2 GoalHeart Logic As an Item to Complete Levels	47
Code 4.3 Logic Obstacle with Player Respawn	49
Code 4.4 Random Item Drop Logic on Treasure	50
Code 4.5 Key Item Retrieval Logic on Chunk Key	52
Code 4.6 Logic When Players Open Locked Doors	54
Code 4.7 Determining the Direction of Neighbors	56
Code 4.8 RemoveWallSection for Punching Hole in the Side of the Room	57
Code 4.9 Finding Nodes Connected to the Goal Node	62
Code 4.10 Random Value to Determine the Direction of Linear Expansion.....	62
Code 4.11 Breaking Off Old and Create New Relationship Node Goal.....	63
Code 4.12 Left-Hand-Side of the Branching Rule.....	65
Code 4.13 Random Selection of Valid Nodes and Placement of Treasure Nodes	65
Code 4.14 Left-Hand-Side Key-Lock Rules	67
Code 4.15 Code Implementation for Graph Initiation	69
Code 4.16 Validation Implementation with BFS Algorithm	73