

Universiti Teknologi MARA

**Mobile First-Person Shooter (FPS)
Game Using Basic Theta* Algorithm**

Muhammad Syurahbil Bin Abd Rohaman

**Thesis submitted in fulfilment of the requirements
for Bachelor of Computer Science (Hons.)
Faculty of Computer and Mathematical Sciences**

August 2020

ACKNOWLEDGEMENT

Alhamdulillah, praises and thanks to Allah because of His Almighty and His utmost blessings, I was able to finish this research within the time duration given.

I would like to express my gratefulness to my supervisor, En. Zawawi Ismail @ Abdul Wahab for his incredible contribution in shaping the foundation of my research. His advice, constructive criticism, wisdom and encouragement have helped me greatly in making this research possible. It was a great honour to work and study under his guidance.

I am also thankful to Prof. Madya Dr. Hamidah Binti Jantan for guiding me in every stage of this research. Without her support it would have been difficult for me to properly conduct this research.

Special appreciation goes to my beloved parents, Abd Rohaman Bin Mohamed and Norhayati Binti Che Nik for their constant moral support throughout my life. They are my source of strength and motivation for me to finish this study.

Last but not least, I would like to give my gratitude to all my friends for always being there for me whenever I need helps or guidance. I cannot thank you guys enough.

ABSTRACT

Video games has risen in popularity due to the increasing number of mobile phone sales. Its portability attracts a lot of people to play mobile games on the go anywhere. Video game industry see this opportunity to further increase the video game engagement by implementing a lot of Artificial Intelligence area such as pathfinding. The most common pathfinding algorithm used in video game is A*. Even though A* is the most popular algorithm amongst game developers, it's hardly the best algorithm in finding the true shortest path. This study explores one of the A* variants called Basic Theta*. A mobile game app is developed using Unity Engine. Basic Theta* is implemented in the game using grid-based map representation. Unity Engine provide its own pathfinding solution using A* algorithm with Navigation Mesh (NavMesh) map representation. A comparison is made between these two algorithms to test their performance. The test is done in three different map sizes, 50x50, 100x100 and 200x200. Each map size is tested three times with different starting and ending point. This study shows that Basic Theta* is able to produce the true shortest path compared to A* which produced slightly longer path. The downside is the computation time on Basic Theta* using grid is way higher compared to A* using NavMesh. On a smaller game map, the algorithm still takes an acceptable computing time but on a larger size map, the computation time suffers a lot. It can be concluded that Basic Theta* can produce a shorter path compared to A* but with the price of having a higher computation time. Further optimization can be made to reduce the computation time.

TABLE OF CONTENTS

CONTENT	PAGE
SUPERVISOR APPROVAL	ii
STUDENT DECLARATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	ix
LIST OF TABLES	xi
LIST OF ABBREVIATIONS	xii
CHAPTER ONE: INTRODUCTION	
1.1 Background of Study.....	1
1.2 Problem Statement.....	4
1.3 Objectives.....	5
1.4 Scope.....	5
1.5 Significance of Study.....	6
1.6 Summary.....	6
CHAPTER TWO: LITERATURE REVIEW	
2.1 Introduction.....	8
2.2 Mobile Applications.....	8
2.2.1 Type of Apps.....	9
2.2.2 Categories of Apps.....	10
2.2.3 Related Issues/Problems in Apps.....	12
2.3 Pathfinding.....	14
2.3.1 Pathfinding Algorithms.....	14
2.4 Basic Theta* Algorithm.....	17
2.4.1 Fundamental of Basic Theta*.....	18
2.5 Related Works.....	18

2.5.1	Discussion	20
2.6	Summary	21
CHAPTER THREE: METHODOLOGY		
3.1	Introduction	22
3.2	Research Methodology Framework	22
3.3	Preliminary Study	24
3.4	System Analysis	24
3.5	System Design	25
3.5.1	Logical Design	25
3.5.2	Basic Theta* Algorithm Pseudocode	27
3.5.3	System Layout	29
3.6	Implementation Phase	30
3.6.1	Hardware and Software Requirements	30
3.6.2	Concepting	31
3.6.3	Creating 3D Models	32
3.6.4	Coding/Scripting	35
3.7	Evaluation Phase	36
3.7.1	Statistical Test	36
3.7.2	Functionality Test	36
3.8	Documentation	37
3.9	Gantt Chart	37
CHAPTER FOUR: RESULTS AND FINDINGS		
4.1	Introduction	38
4.2	Zombie Survival Conceptual Framework	38
4.2.1	Initialization Process	39
4.2.2	Input	40
4.2.3	Process	40