**Chapter I**

**The Problem and Its Background**

1. **Introduction**

First person shooter games have been the craze of gamers on recent years. The genre has made a name of itself by attracting gamers of various ages to its competitive and skill based gameplay. Although the competitiveness of the games is derived from its multiplayer oriented function. The charm of the genre is still existent on the single-player part as it delivers a richer experience with its impeccable scenery and story-telling.

In a single-player play-through, players are tasked to accomplish given sets of objectives as the journey throughout the environment of the game. Most of the times, they are accompanied with an NPC (Non-Playable Character) to accompany them and fulfil their missions. Enemies encountered by the players are also considered NPCs

AI (Artificial Intelligence) is used to give actions on these NPCs. It matches its actions based on the course of actions the player uses. Because of this, human players have a certain expectation on their NPC counter parts.

Pathfinding algorithms are used to give the AIs a certain distinction and understanding on their surroundings. It lets them choose a path that will reach the destination with the least distance covered.

A\* is a [computer algorithm](https://en.wikipedia.org/wiki/Computer_algorithm) that is widely used in [pathfinding](https://en.wikipedia.org/wiki/Pathfinding) and [graph traversal](https://en.wikipedia.org/wiki/Graph_traversal), the process of plotting an efficiently traversable path between multiple points, called nodes. Noted for its [performance](https://en.wikipedia.org/wiki/Computer_performance) and accuracy, it enjoys widespread use. However, in practical travel-routing systems, it is generally outperformed by algorithms which can pre-process the graph to attain better performance, although other work has found A\* to be superior to other approaches.

AssaultCube is an open-source first person shooter game that takes place in realistic environments, with fast arcade gameplay. Although the main focus of AssaultCube is multiplayer online gaming, a single-player mode is also available which consists of computer-controlled NPCs. The game started its development from July 2004 and was first officially released on November 2006. Despite its simplistic graphics and gameplay, the game maintains a consistent player base of hundreds of players, players which are part of the community that help improve the game.

**Existing Algorithm**

The Non-Playable Characters in the game use A\* as a pathfinding algorithm. But, the way the algorithm is implemented to the application lacks in quality which then influences the performance of the AI. The algorithm searches for the location of an enemy, it will then search its immediate surrounding to figure out the path to its designated location. This results in the subpar performance of the AI.

1. Add the starting node to the open list.  
2. Repeat the following steps:   
 a. Look for the node which has the lowest f on the open list. Refer to this node  
 as the current node.  
 b. Switch it to the closed list.   
 c. For each reachable node from the current node   
 i. If it is on the closed list, ignore it.   
 ii. If it isn’t on the open list, add it to the open list. Make the current   
 node the parent of this node. Record the f, g, and h value of this node.   
 iii. If it is on the open list already, check to see if this is a better path. If   
 so, change its parent to the current node, and recalculate the f and g  
 value.   
 d. Stop when   
 i. Add the target node to the closed list.   
 ii. Fail to find the target node, and the open list is empty.   
3. Tracing backwards from the target node to the starting node. That is your path.

**Pseudocode of A\***

**Start**

1. Checks list for “alive” opponents.
2. Designates the position of opponent as primary destination.
3. Scans surrounding area for valuable items.
4. If opponent dies before the bot reaches them, return to 1.
5. If bot dies before reaching the opponent, return to 1.
6. If bot kills the opponent, return to 1.
7. **Statement of Problems**

The following problems have been observed in the given and current technology:

1. **The Pathfinding Algorithm is unable to find the best path.**

In combat situations, it’s important to be able to move around from cover to cover quickly and effectively as a player would. However, in most cases the shortest path isn’t the one you want. The AI controlling the enemies may decide that taking the shortest path will be the best decision, even if that path is right in front of where you are. Various waypoints are used by the AI to identify which path it should follow. Figure 1 displays the action of two NPCs as they run toward walls, indicating the problem within the Pathfinding Algorithm.



1. **The Algorithm takes too much in consideration that it reduces the reasoning of the AI.**

To make good decisions in combat, you need good quality information taken from annotations, terrain analysis, pathfinding, line of sight, and many other systems. This can quickly get inefficient though, as the need to evaluate many different options to find combat positions reliably are required. Figure 2 shows an NPC’s lack of reason for jumping without any valid reason.

**

1. **The Algorithm limits the action of the AI when finding the best path.**

Most games today have the AI dictating to the animation what should be done, for example when following a path. While this works most of the time, it can result in low quality movement — for example when taking the shortest path means the animation isn't as smooth as going the longer way around. Having the AI in complete control also causes bugs when there's a mismatch between what the AI wants and what the animation can provide.

1. **Objectives of the Study**

The objective of the study is to help improve the Artificial Intelligence currently used by modern games such as First Person Shooters. The use of pathfinding algorithms will help the AIs to make the best decision that will give the players a better experience in playing the game.

Specific Objectives

1. Pathfinding performance problems are often resolved using hierarchies or better heuristics. Application of A\* algorithm will be necessary to calculate a more appropriate route.
2. In improving the AI systems that reason with large quantities of information, the AI reasoning should be able to deal with this, carefully making requests for information which will prune its option down quickly, but also batching up queries efficiently.
3. Using the A\* algorithm, it will help the AI to distinguish which action the characters must do in following an appropriate path, with this, the movement of the NPCs will not deteriorate in terms of its performance.
4. **Importance of the Study**

The study seeks to benefit the following people:

1. To the ***game developers,*** this study will help them know the current issues of pathfinding Artificial Intelligence on first-person shooters and will help them advance and improve the quality of the AI.
2. To the ***gamers,*** the study will let them know the existing problems found on FPS games which will help them identify the problem.
3. To the ***students,*** that the study will let them be interested to games, as well as game development and help them expand their choices on their career.
4. To the ***other developers,*** that the study will be useful for them on understanding the problems of pathfinding algorithms and knowing the solutions for it.
5. Lastly, to the ***future researchers,*** that the study will help them on their own study and may help on furthering the study through finding out better solutions on recent game AI problems.
6. **Scope and Limitations**

Scope

The study covers the understanding on how an AI works in AssaultCube, a First Person shooter game. To improve the AI’s ability on how it performs, how it acts based on its surroundings, and identifying the path it should take.

Limitations

The study will not cover on how an AI works nor what an AI is. The study will only focus on how AI is used on AssaultCube. Other First person shooter games will not be a part of the study, such as: “Medal of Honor”, “Battlefield”, “Halo”, “Half-Life”, etc. Though the idea or logic on how pathfinding algorithms are used on mentioned games will be used by the researchers for reference purposes only. The study will not include how pathfinding is used on outside systems such as robotics.

1. **Definition of Terms**

**A\*.** A [computer algorithm](https://en.wikipedia.org/wiki/Computer_algorithm) that is widely used in [pathfinding](https://en.wikipedia.org/wiki/Pathfinding) and [graph traversal](https://en.wikipedia.org/wiki/Graph_traversal), the process of plotting an efficiently traversable path between multiple points.

**Algorithm.** A self-contained step-by-step set of operations to be performed.

**Artificial Intelligence.** Used to generate [intelligent](https://en.wikipedia.org/wiki/Intelligence_(trait)) behaviors primarily in [non-player characters](https://en.wikipedia.org/wiki/Non-player_character) (NPCs), often [simulating](https://en.wikipedia.org/wiki/Simulating) human-like intelligence.

**Bugs.** An error, flaw, failure or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways.

**Combat Reasoning.** The ability of an AI to identify the most suitable position it can take during an encounter with the player.

**First-person shooter (games).** A [video game genre](https://en.wikipedia.org/wiki/Video_game_genre) centered on gun and projectile weapon-based combat through a [first-person perspective](https://en.wikipedia.org/wiki/First_person_(video_games)).

**Nodes.** Multiple points used to traverse between paths.

**Non-playable characters.** Characters in a game that cannot be controlled by a player.

**Open-source.** Computer software with its source code made available with a license in which the copyright holder provides the rights to study, change, and distribute the software to anyone and for any purpose.

**Pathfinding.** The plotting, by a computer application, of the shortest route between two points.

**Player.** The person the controls the character in a game.