Neil Patil

Computer science project

AQA A Level Computer Science NEA

Contents

[Analysis 1](#_Toc469380628)

[The Problem 1](#_Toc469380629)

[Survival Mode 1](#_Toc469380630)

[Role Playing Games 2](#_Toc469380631)

[The Solution 3](#_Toc469380632)

[List of Objectives 4](#_Toc469380633)

[Design 5](#_Toc469380634)

[Algorithms 5](#_Toc469380635)

[A\* Algorithm 5](#_Toc469380636)

[Data Structures 7](#_Toc469380637)

[File Structure 7](#_Toc469380638)

[Database Design 8](#_Toc469380639)

[Queries 9](#_Toc469380640)

[Human Computer Interaction 9](#_Toc469380641)

# Analysis

## The Problem

### Survival Mode

One of the most popular game subgenres are horde style survival shooters. Horde style means that the players are trapped in a finite arena and must defeat waves of progressively stronger enemies until all players are defeated. Popular games with survival modes include Call of Duty, Gears of War and Halo. However, all these games are primarily 1st or 3rd person shooters, which have more focus on the campaign and/or player-vs-player multiplayer, causing a lack of content and replayability for the more interested players.

#### Call of Duty

To analyse the problem more in-depth, the Call of Duty series can be looked at. The original horde mode in the CoD series was in the WWII shooter Call of Duty World at War. This game is very notable, as it first introduced the gaming world with the Zombies mode. Originally it released on a simple small map. This caused a huge interest in the gaming community, and so Activision launched DLCs for more maps, which were received even better. Zombies then made a return in Call of Duty Black Ops, using the combination of the original WWII setting and the Cold War era in which the game’s story was set to captivate players. It was around this time it peaked in popularity. In the 5 games since, the variations of the game mode have been released, but especially recently the interest in the game mode has died down.

The major factors resulting in this are to do with the lack of content in the newer releases. The majority of gaming companies seem more focused on offering DLC and microtransactions which negatively impact the base game. With only one map available per launch, users have often been turned away by the required £40 on top of the base game to play the horde mode to its fullest.

Another issue is to do with the complexity of the game. Initially, the objective was to shoot the zombies to earn money, spend the money on weapons and survive as long as possible. As the franchise progressed, crafting and hidden Easter eggs became prevalent, to the point that the game has become a linear story instead of a simple survival mode. This impacted the overall game negatively, as people became less concerned with getting a high score and more focused on actually completing the Easter egg and ending the game. This massively reduces the replayability of the game and the appeal of it to players who do want to simply survive.

### Role Playing Games

One game genre in particular that counters some of the main problems with the horde mode is Role Playing Games. These types of games are some of the most popular due to the progression systems, large variety of characters and/or items and the ability to restart the game and have an entirely different experience. The greatest titles in this genre include The Elder Scrolls, Fallout, Pokémon and Destiny. These games are widely encompassing, some designed as pure single player experiences, others as competitive player-vs-player battling simulators and even some as Massively Multiplayer Online games. The elements that can be taken from these games in order to improve the horde mode are best seen in Pokémon, Fallout and Destiny.

#### Pokémon

The Pokémon games are different to many other RPGs, but have certain characteristics that make them unforgettable to most. Pokémon boasts almost 800 unique and different battlers which are all given a large amount of thought when developing. This makes it much more varied, as there are almost an infinite number of different teams and strategies that can be used, meaning the game retains it replayability for years. Even at 20 years old, the original Game Boy games some of the most popular games used on the Nintendo 3DS. This shows how much even 151 Pokémon can retain their users. Also, Pokémon has a multiplayer system in all of its games, allowing for players to play together, even across large distances.

#### Fallout

The Fallout series has actually had two renditions. The first 2 main games used a top-down tactical-based system, whereas Fallout 3, New Vegas and 4 are 1st person shooters. The intended project is more similar to the first iteration, but the series has some core aspects which are optimal. Each game has a set of companions with distinct personalities and stats that, similarly to Pokémon, completely impact the way the game is played. Additionally, Fallout has the S.P.E.C.I.A.L. system which is the stat and perk allocation system for the game. This affects not only the player’s abilities, but also the choices they can make in the game. This makes the player more in control of the game, giving a more personal feel.

#### Destiny

A very different addition to RPGs, Destiny is more like a First Person Shooter than a traditional RPG. The main concept that it would add is a good reward system as well as great and memorable design throughout the game. As a loot based shooter, Destiny rewards the player with different armour and weapons exclusive to different activities. From endgame activities, the player is guaranteed loot that is personalised to the activity (Competitive multiplayer rewards the player with guns with stats designed around player vs player combat, whereas the cooperative Raid gives the player weaponry and gear optimised for certain boss encounters), which will help the player progress. Also, for less skilled players, non-endgame gear can be obtained randomly just for participating in activities. This incentivises players get the loot they want, regardless of ability. Another major aspect of Destiny is the design element. The rocket launcher ‘Gjallarhorn’ has become legendary within the gaming community for its aesthetics, power and rarity, despite not being advertised at all by the creators any more than any other gun. By having such well-designed components, the game has really succeeded at attracting players to the game and incentivising players to play as much as possible to obtain weapons like it.

## The Solution

The final game is an amalgamation of the classic arcade horde mode style and a progression based RPG. In order to keep the game fresh, the maps will be numerous and easily made, so players will be retained over longer periods of time. The theme of the game will be based on the Star Wars Clone Wars television show. This is because it has many recognisable characters and widens the age range considerably compared to the violent Zombies in CoD. Also, there are a number of different Clone Troopers that the player can use, once again, keeping content levels high. Furthermore, all graphics will be personally designed to be appealing to the player. This will attract players and make it a better, more immersive experience.

To make the game more worth playing in shorter play sessions, the character’s progression will be saved after every round. This means that a player will feel more accomplished, as they will be rewarded with more than just a high score. The gameplay of the game will be styled as a top down shooter with a grid based movement system.

The control system will be as simple as possible to make it easy to learn. The WASD keys will be used for movement. Left shift will be used for sprinting to quickly cover distances. Q will allow the player to throw a grenade, and E will make the player perform a melee attack.

The challenge of the game will be in the form of the AI, which will be programmed to intelligently follow and attack the player. The aggressiveness of the enemies will also increase as the number of rounds the player survives for increases. The AI will also have pre-set difficulty levels that can be manually altered before every game so players of all skill levels will be able to enjoy the game. However, the rewards will be scaled by difficulty, so more accomplished players will obtain the best rewards. This keeps the player incentivised to challenge themselves as much as possible.

The player themselves will choose from premade Clones all with individual stats and weapons. All but the initial default Clone Trooper will be locked behind either an XP requirement or a challenge. The player will get XP from destroying droids and from completing rounds. This will be recorded and saved every time the player completes a round or is defeated. Challenges will require the player to either get to a certain round on a map, destroy a boss or get a certain number of kills with a specific weapon or character. This gives the player objectives outside of getting a high score. Also, each Clone can be equipped with weapon and armour modifiers that will either change the character’s stats or give them additional perks which will largely change their play style. This gives the player multiple avenues of customisation they can take.

Finally, the game will be aimed to be able to be played by anyone, independent of skill or experience. However, the game’s theme and style will be aimed more at teenagers and children, so the third parties used to test and obtain feedback will be around 17-18 years old.

### List of Objectives

1. The game will run using a text based map
   1. Each character on the map will correspond to a block with individual characteristics
   2. The map will be able to be edited in any text editor
   3. Players will be able to submit their own maps to be used
2. All files relating to the game will be stored on the internet
   1. The player will require internet connection to play, but will have automatic access to updates for characters, maps etc.
   2. The user will only need the .jar file to play the game
   3. The player can upload maps and units
3. A database will be used for storing all data relating to the game
   1. There will be a database for:
      1. Player characters
      2. Weapons
      3. Grenades
      4. Melees
      5. Enemies
      6. Modifiers
      7. Blocks
   2. The database will be accessed using JDBC for SQL statements
   3. Several tables will be linked together for maximum efficiency and normalisation
4. The controls will be intuitive and responsive
   1. The menu system will be minimal in content so the user can easily navigate without requiring help
   2. The game should have a high FPS, so inputs have minimal delay
   3. The code will be as efficient and as less CPU intensive as possible to keep the program running smoothly
   4. The game should be playable on low and high-end machines with Java to allow as many people as possible to play the game
5. The design will be authentic and immersive
   1. The game’s User Interface will be designed as aesthetically as possible
   2. The game itself will look professionally designed
6. The player is given the option to play solo, locally with a friend or over the internet with a friend
   1. The game will have a socket system with one instance acting as a host and another as a client
   2. The games will have to be on the same version to play together
   3. A guest account can be used for local play, so the second user will not require login credentials
7. The Artificial Intelligence of the enemies will challenge the player
   1. The enemies will path find to the player when not in line of sight by navigating obstacles and patrolling set areas using the A\* Algorithm
   2. The enemies will work together to attack the player
   3. As the player increases in level, the enemies will become more aggressive i.e. reacting faster to spotting and acting together more
   4. The enemies will adjust their approach to attacking the player based on what weapons and perks the player has
8. The player will have access to large amounts of customisation
   1. The player will have a wide range of player characters to unlock and choose from
   2. The player will have perks that can be selected in numerous combinations to personalise the experience
   3. The player can unlock gear as they play to boost their character’s stats
   4. The player can unlock unique perks for each character by completing challenges for them, which enhance their abilities drastically.
9. The game will provide the ability for players to create maps and characters
   1. The player will be able to use an in-built UI to create maps which can be accessed by anyone
   2. The player can upload self-created images and character stats which can be approved and added to the game database

# Design

## Algorithms

### A\* Algorithm

The A\* algorithm is a complex mathematical algorithm which enables a path to be made between two nodes as efficiently as possible by using heuristics. The nodes in the algorithm are represented in game as each block on the main grid visible when playing the game. Each block is assigned its own ‘cost’ which is effectively how expensive it is to move through that medium.

First, the starting and target blocks are chosen. In game, the starting block would be the current position of the enemy, and the target block would either be the intended patrol path or the location that would give the enemy direct line of sight to the player.

The starting block is then added to the Open list. This list will contain all blocks that are candidates for examination, sorted by their current F value (this will be explained later). The Open list is in effect the frontier of the area that has been scanned by the algorithm, with the corresponding Closed list formed of all nodes that have been checked but not requiring expansion. Initially, the starting block is given a G value of 0. This is the cost of getting to the current block from the starting block. Then the H value for the block is set. The formula for getting a H value for two blocks is:

This is the heuristic value determined for a pair of blocks. It is based on the Manhattan distance between the two points, added to the lowest possible cost of moving between blocks. Manhattan distance is used instead of the usual Euclidean distance because it is much, much faster for the computer to process multiplication, which is an inbuilt operation, than square rooting, which requires external libraries as it is a process which is complex enough to require multiple clock cycles for the CPU to perform it. This causes it to be much slower to perform, especially in a real-time game where FPS is high priority.

When stored, the H value is also multiplied by 1.001. This is used to resolve tie-breaks, i.e. where the algorithm can choose multiple blocks to look at. This scaling multiplier changes the H value so slightly that it will not change the result of the algorithm, but will shift the algorithm away from checking nodes close to the starting node, towards expanding nodes closer to the target node.

Once the heuristic value has been calculated, the algorithm checks whether the current node is the target node. If it is, it breaks out of the loop. If not, it proceeds.

Now, the algorithm passes through each of the neighbouring nodes to the current node. These are called successors. It works out the what the G value would be between the successor and the current node and saves it as the potential G. The G value is worked out as:

Where,

Then, a series of checks are performed on the successors to determine if they are optimal for expansion:

* The successor is compared with all the nodes in Open. If it is contained within Open, its current G value is compared with the potential G, and if the potential G is smaller, the node is skipped. If it is larger, its G value is set as the potential G and it is given a precursor of the current node.
* If the successor is not in Open but is contained within Closed, a similar check is done on its G value. If it is lower than the potential G, the node is skipped. If it is larger, then the node is removed from Closed and added to Open. The G value is then set as the potential G and it is given a precursor of the current node.
* If it is not contained within any list, the successor is added to the open list, and its H value is calculated and set. The G value is then set as the potential G and it is given a precursor of the current node.

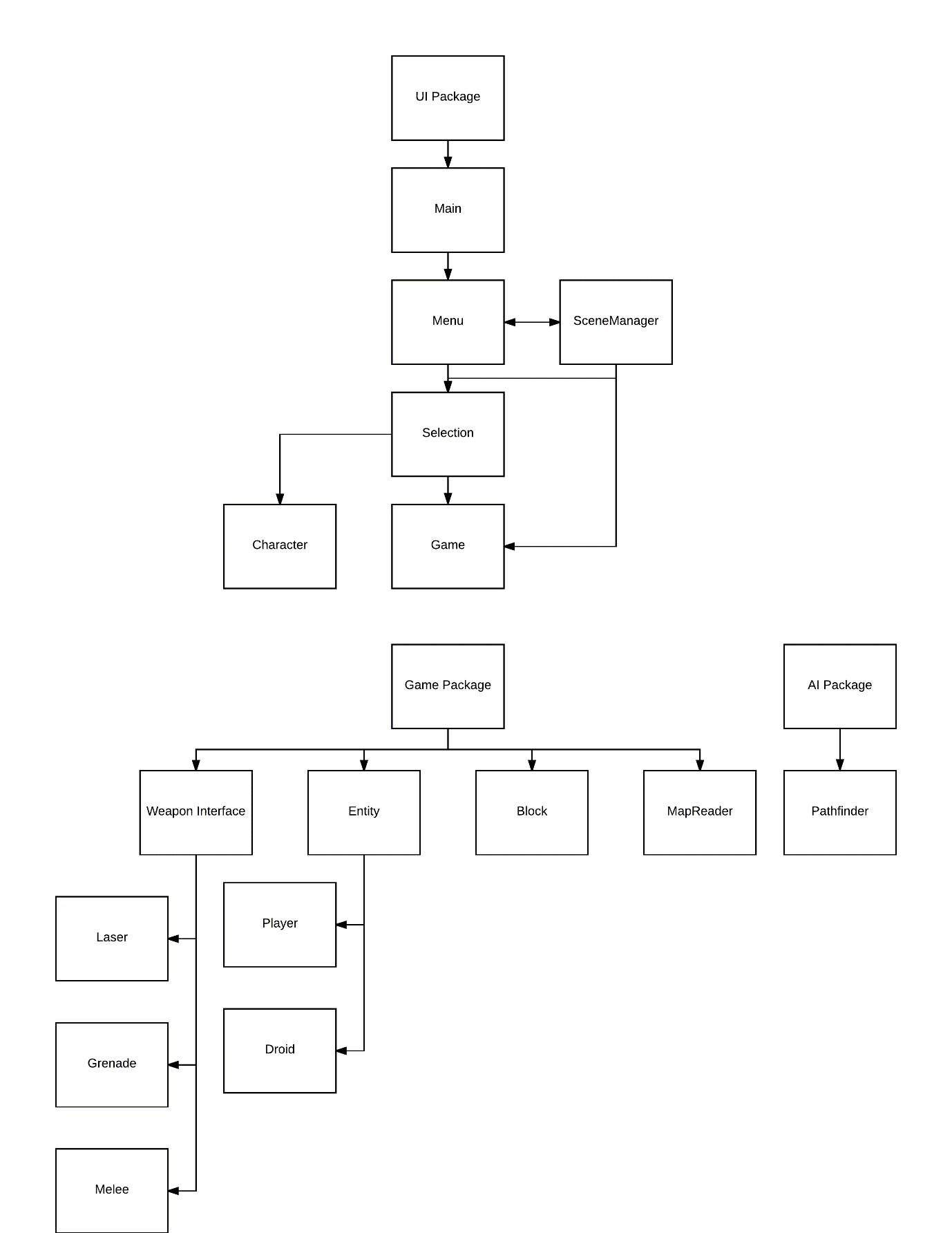
Then, the current node is added to the Closed list, and the loop repeats, this time choosing the block from Open which has the lowest F value.

This process will continue until the expanded node is the target node, where the loop will break.

The program then creates the path by backtracking through the blocks and adding each precursor to the Path list. The Path list contains all blocks in the path required to make the most optimal journey from the starting block to the target in reverse order. The algorithm then returns the Path. This Path can then be used by the program to make the AI controlled enemies move as intended.

Here is a flowchart representation of the algorithm:

## Data Structures

Here is a hierarchal diagram of the classes used in the program:

## File Structure

The only files the program uses when running are for the maps. The maps are made from simple text files which contain characters. Each of these characters corresponds to a block in the game. This allows for anybody to have the ability to edit and make maps for the game without requiring any more information than a simple key. For example:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

g g g g g

g g g g w

g g g g w 🡪

g g g w w

w w w w w

## Database Design

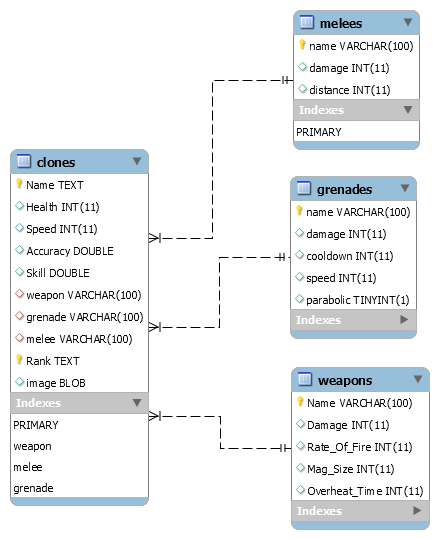
The game is very reliant on databases, as all of the data is stored on them. A MySQL 5.7 Server is used, and must be running for the game to be playable, as the information and images are stored on it. The server’s schema is called battleground, and is accessed by the game by using JDBC. This means the user does not have to do anything, as JDBC will automatically connect the player to the server. In the event that the server is not reachable, the game will terminate and the user will be told that the server must be connected to in order to play the game.

The schema itself contains 7 tables. The first table, Clones is used to access all of the player’s characters. It contains data about all the stats and images of the characters, and only needs to be queried when first starting a game. The information about the characters is stored in an Array of classes once the initial set of data is retrieved.

The weapons table is used similarly to the Clones table, as it is used when the player character is selecting their player. It is also used when initialising the enemies, as their weapons are stored on the weapons table. The weapons table is also stored within the game after it is first accessed, as classes. However, as weapons share properties with melees and grenades, an interface is used to keep the classes as abstract as possible.

The grenades table is identical in properties to the weapons table, but has fewer attributes. The same applies for the melees table.

Here is the Entity Relationship Diagram of the tables so far:



## Queries

When the player is selecting their character, a query is sent to retrieve the list of clones. However, the list of clones is ordered so the more basic, more commonly unlocked characters appear first, with the more potent, harder to unlock characters are last in the list. To do this, the select statement is sorted

This makes it so the clones are given an average rating based on their stats percentages, and ordered by it, lowest first.

## Human Computer Interaction

To allow all types of player easy access to the game, the menu GUIs have been designed as simplistically as possible. The minimalist design is easy to follow and also aesthetic. The buttons along the bottom, in order, are Start, Stats, Options and Exit. The logo of the game appears in the middle, and when implemented, will feature a fully animated battle in the background. This will immerse the player in the game before even playing.

When the player selects the Start button, they will be transferred to a screen showing the characters and their stats. The player can use buttons to change the selected character. The stats appear on the side of the screen. They are represented as bars with varying length depending on the amount of each stat the character has. The decision to show the stats as bars instead of the number values that are handled in-game was made to keep the game as simple looking as possible, reducing the chance that a potential player could be turned away with a large amount of confusing numbers. Additionally, it is much more aesthetic to show the stats in this way. This screen also has a Back and Continue button. The Back button will return the player to the home screen, and the Continue button will move the player on to another customisation menu.