

School of Computing, Electronics and Mathematics

PRCO304

Final Stage Computing Project

2015/2016

BSc (Hons) Computing & Games Development

Daniel Bendell

<*Your Registration Number*>

<*Your* *Project Title*>

Supervisor: <your supervisor’s name>

Second marker: <your second marker’s name>

Table of Contents

[Abstract 3](#_Toc450055270)

[Introduction 3](#_Toc450055271)

[Background and objectives 3](#_Toc450055272)

[Approach 4](#_Toc450055273)

[Unity 5](#_Toc450055274)

[Visual Studio & Resharper 5](#_Toc450055275)

[Game Design Document 5](#_Toc450055276)

[Illustrator 6](#_Toc450055277)

[Blender 6](#_Toc450055278)

[Competition Comparison 6](#_Toc450055279)

[Bayesian Probability 7](#_Toc450055280)

[Platform 7](#_Toc450055281)

[Mobile IOS & Android 7](#_Toc450055282)

[Consoles Xbox & PlayStation 7](#_Toc450055283)

[Web & PC 7](#_Toc450055284)

[Outcome 8](#_Toc450055285)

[Objective completion 8](#_Toc450055286)

[Game Design 9](#_Toc450055287)

[Overview 9](#_Toc450055288)

[Game Flow 9](#_Toc450055289)

[Core mechanics 10](#_Toc450055290)

[Bayesian Probability Adaptation 13](#_Toc450055291)

[Architecture 16](#_Toc450055292)

[Modelling and User Interface 16](#_Toc450055293)

[Project Management 16](#_Toc450055294)

[Legal social ethical and professional 16](#_Toc450055295)

[Reflection 16](#_Toc450055296)

[Communication 16](#_Toc450055297)

Abstract

Save for the end of the report. It covers the report as a whole so impossible to write before the report is complete.

# Introduction

With the forever growing popularity of computer based games, it is becoming essential to create unique, engaging and powerful artificial intelligence (AI) systems.

The aim of this project was to create a turn based strategy game that can be played against the computer in the form of AI.

This report presents the research, design and production of the turn based strategy game, with a high focus on the artificial intelligence, the main aim being a smart AI system that’s able to assess and formulate a response to a range of different situations.

Main Body –

# Background and objectives

The initial plan for this project was a multiplayer real time strategy game, where two players would compete against one another. The players carefully select from a set of predefined characters, position them in any strategical arrangement, then taking turns to move the chosen characters around a fixed map or environment to out manoeuvre their opposition, with an end goal of defeating the opposition in tactical combat.

This initial project idea was carefully broken down into more detail in the form of a project initiation document (PID). After further discussions with my supervisor it soon became clear that this was a suitable project to undergo, however it lacked slightly in terms of a challenge. This level of challenge was expressed to give the project something more exciting than the initial plan and would be a welcome adjustment to the proposed project.

PID – “3. Project objectives, 4. Scope out development miles stones, what is going to be key to make a minimal viable product, increase the focus on areas that are going to be most important”

After further discussion with my supervisor it was suggested that instead of making a game that focused on a player vs player environment that it shifted towards some form of artificial intelligence (AI). Creating a game that was playable verse the computer would require more complex coding methods and research into a number of different approaches to solve this new problem. Using AI in the project would introduce a new area of development to me as I had not programmed any form of AI in the past.

As this was the only additional suggestion made towards the proposed project as well as the PID being approved it in original state. I was confident that it would be suitable project to undergo.

After assessing the project objectives within the PID and comparing them to the Specific, Measurable, Achievable, Relevant, Time-bound (SMART) principle I could see some room for improvement. Although the vast majority of the objectives were specific, giving clear and precise goals and outcomes, I believe that some of them (Objective 4, 5) could have been broken down further to give a greater understanding of the original objectives.

In terms of being measurable all bar the last objective have been formulated to give a clear indication of being completed or not. As for the last object (6. To further enhance my programming and game development skills with a fully functional game) it is more of a personal objective. Although the objective clearly states that it will be measured by the outcome of a fully functional game. The completion of said game could be very basic and not have challenged my programming ability, because of this I feel that it could be rephrased to make it more applicable, emphasizing the need to create a fully functional game that introduces new and challenging elements.

Creating achievable objectives was very important as it would give me a sense of accomplishment throughout the research and development of the project. The moral boost that comes with completing a task is very important in my development process. This is why I have broken down the entire project into incremental steps (available Trello link). Each one of the objectives has resources available to show that they were indeed achievable.

Relevance is an area in which I think I could have improved or altered the objectives. Although the objectives are relevant in terms of the entire research and development process I should have been more specific when writing objectives (5 + 6). I have clearly scoped out appropriate objectives for the research part of the project, however point 3 isn’t as focused as it should be. This point targets the environment I chose to use when developing the project opposed to focusing on the project itself.

Time-bound is an interesting element to consider as the overall project was broken down into project plan which specifies the goals and time frames in which I had hoped to complete each task. Although the project plan doesn’t cover all of the objectives it does break it down into more reasonable tasks to be achieved throughout the research and development stages. Each one of the objectives has been accounted for within the plan and progress was documented through weekly highlight reports. These reports were a healthy reminder of the initial project plan and how each weeks work had fit in with the corresponding time frame.

# Approach

As the project and its initial ideas began to unfold it was made clear it was a sensible plan with scope for expansion if necessary. As stated in the background and objectives there were little changes to the original project, these changes were made to make the game more challenging to design, develop and test. Although the changes were not set in stone it was important to evaluate them as this would ensure a better overall project.

“3. Analyse development environments, which one is going to be most beneficial, boost and aid development.”

## Unity

One of the first hurdles I needed to overcome before the development started was to assess the different technologies available and how each one was going to benefit me during the developing stages. To give me a better understanding of my options a research paper was completed in which I assessed several game engines and their strengths, weaknesses, opportunities and treats (SWOT). Not only did this paper help me understand which game engine was going to help me develop a top quality project it was able to enlighten the areas of the project that could prove to be more difficult that first anticipated. The end result of the paper had me deciding to use Unity as the game engine. Some of the deciding factors were the past experience I have with the environment, the rapid development possible with the built in level design and the access to its own asset store. These are only some of the factors that played a part in the decision but I felt where most important knowing that the application would have to be developed swiftly and there was a massive need for external assets, mainly 3D models.

Using Unity allowed me to play to my strengths as a programmer. I was able to use C#, which is the programming language I am most comfortable with. There was less emphasis and pressure on level design as Unity has its own built in level editor. Unity can be debugged extremely easily during runtime, giving me the option to edit variables and values and seeing the effects as the game unfolds. Unity makes it effortless to deploy to multiple platforms, this means that if I decided to change the direction of the project and move it on to mobile it would require little change if any at all.

## Visual Studio & Resharper

One of my main concerns with using Unity was the integrated development environment (IDE) Mono Develop. Although this may not strike many people as a real issue, but I personally feel that programming on a poor IDE can lead to sloppy coding. Luckily after only a small amount of research I found out that Unity has been integrated with Microsoft Visual Studio. The Visual Studio plug in offers full syntactic support along with simple debugging options, everything I needed to get the ball rolling. Another bonus of using Visual Studio was that I could also link in Resharper. Resharper is like a second pair of eyes for your code, it will ensure you code in a uniform format, keeping the variables consistent throughout that project and offering alternatives to certain coding approaches. For example using a LINQ statement instead of iterating through a list of items in a for loop.

These are very small enhancements for such a large project but the payoff can be immense when passing on a code base to another programmer or simply ensuring your writing clean code.

## Game Design Document

After deciding on a game engine to use I created a game design document (GDD). The aim of a GDD is to be a more in-depth, descriptive document, outlining the story, characters, level, gameplay, art, sound, music, user interface, game controls. All of these sections are vital when developing a successful game and all areas in which I had to assess before starting the development of the project. Traditionally as the development of a game grows so does the size, structure and information stored in the GDD however creating the skeleton of this document was important before the development started. It added direction towards all of the important areas required. The GDD is a very visual document expressing ideas with a variety of different media, whether that be links to videos or music, pictures from existing games, concept art scanned in from a scrap book. With this mixture of resources I was able to better visualise the intended game.

When first deciding to create a GDD I was unsure if I should do it in a form of a blog as this would allow me to reach out to an audience, granting me feedback on my development and providing external advice to a very remote project. The decision went against this method as a thriving GDD seemed to be a better approach and resource.

After the creation of the GDD I realised there were several different areas I needed to explore before I was comfortable starting any of the programming.

## Illustrator

First of which was which resource I was going to use to create any 2D and or 3D assets. At this point I was aware that I was going to need both of these but knowing whether or not if I needed to create them or find external resources was important.

I knew that I was going to need 2D illustrations to act as a user interface otherwise the game was not going to look very professional. As this was not as big as finding a game engine to use, I didn’t feel the need to create an external document that went into as much detail, however I did go through some of the more popular options e.g. Illustrator, Photoshop, Inkscape and created basic assets, getting used to the options and finding which one I was going to be able to use quickest and easiest. It soon became apparent that Illustrator was going to quickest way of creating simple assets and was the chosen application. This being said a variety of the mentioned applications were used at some point during the design of the game.

## Blender

Continuing from the need for assets brought the research to 3D modelling. 3D modelling is extremely complex and takes years to master, however the need for outstanding models with beautiful texturing was not one of concern. It would be ideal for a full release as it increases the engagement but this was not the case. Knowing very little about 3D modelling made it important to find a tool that was going to be easy, yet efficient to use. This is where Blender came to shine. Free to use, with thousands of tutorials online to help with the basic requirements, it was the software used throughout the designing phase.

## Competition Comparison

Secondly I thought it would be beneficial to go through some existing turn based strategy games to gain a little inspiration. This research allowed me to scope out potential competition, whilst giving me a better understanding of the initial project objectives. It was important to loosely compare my idea against others as they will have been completed by a full team opposed to one person. I didn’t want to look at these games and over scope my project thinking that I would be capable of completing a team’s worth of work.

## Bayesian Probability

Lastly was the new focus of the project, the AI. I knew before even starting any research that this was going to be a vital part of the project seeing as I hadn’t completed any AI based programming in the past. It was key to understand how different AI techniques could be used to really bring out the true potential of the game. I was given a little direction into which AI techniques could be appropriate. From this information and additional research I categorised their possible uses to help me narrow down which choices would benefit me greatest. The end result showed an adaptation of the Bayesian probability technique, in which the AI will assess its given situation, calculate the probability of a series of different outcomes and execute an appropriate response.

## Platform

Upon deciding on all the above, the need to focus the game towards a specific platform was still undecided. Due to picking Unity as the desired game engine this opened up the options to target the project to several different platforms. Unity can easily build the application into Android, IOS, console (Xbox, PlayStation), web or standalone PC. Although it would be ideal to have the game available for each and every option it needed to be narrowed down.

### Mobile IOS & Android

There has been a massive spike in popularity for mobile based applications and games. These are normally tailored towards people on the go, being intuitive to use, simple to maintain and engaging. Both IOS and Android are worthy candidates. However they might not produce the system power required to run the game. Knowing that Android alone has thousands of different devices all varying in sizes and specs it was enough to sway the decision against the mobile platform.

### Consoles Xbox & PlayStation

Next were the consoles. Both Xbox and PlayStation offer a lot more power than a mobile device and would comfortably be able to run any game developed through Unity. Despite their extended power the need to develop a game in such a way that is passes the criteria required by either Microsoft and or Sony proved problematic. Not only would it be mandatory to abide by their strict rules the game would need to be finished to a professional standard and the time required to do so was not available.

### Web & PC

Targeting the game towards the web would not have been a bad idea. It can be hosted on game sites, have a large audience ready and waiting, allow for instant exposure. However the need to play the game online is a large obstruction. Ideally the game will be able to be played offline as it maximises the opportunity to be played. This brought around the final choice of a stand along PC game. Offering the options of online and offline play, being run on a powerful machine and with no need to meet the expectations of an external source. Building a standalone PC game was seen to be the best option for the proposed game.

# Outcome

## Objective completion

Were the project objectives met? Were they demonstrated to have been met?

1. To analyse existing turn based strategy games, note pros and cons of similar indie developed games.

During the project approach phase one of the biggest aids was analysing the potential competition. Not only did it give a larger insight into how successful turn based strategy (TBS) games have been developed and marketed, it acted as inspiration.

This research showed that the level of detailed needed to immerse the user into a strong storyline was extremely difficult to achieve. TBS games are traditionally very slow paced yet keep the user involved, this slightly deters users from a storyline as they want to return to the interactive gameplay opposed to the listening through a series of dialog. Due to this reason it is thought to be why many TBS games remove themselves from in-depth stories.

This resulted in not adding a storyline to the final project. Instead the focus was more towards immersing the players into the core gameplay. The game needed to have a simple to follow environment, smooth flowing game mechanics and intuitive user interface to avoid obstructing the player with too much information. This allows the player to concentrate more on the game and out manoeuvring their opposition.

2. Analyse the areas of the projects I am less skilled in (Art, Modelling, Sound), formulate solutions to issues.

Assessing the weaker areas of the project was crucial to ensure that the end result was going to be the best it could be. The developer noted that their weaker points and the areas in which they felt could jeopardise the final result. However this was carefully evaluated prior to development and smart solutions arose.

When the comparison of competition was completed the worry of needing top grade 3D models was extinguished. A lot of indie TBS games focus more on the intuitive user interface (UI) rather that the models within the game. As a result of this the need for appealing UI increased but this was more achievable that modelling and texturing quality assets.

… More about overcoming the issues?

1. Analyse development environments, which one is going to be most beneficial, boost and aid development.

This was covered in the Approach!!

4. Scope out development mile stones, what is going to be key to make a minimal viable product, increase the focus on areas that are going to be most important.

During the beginning of the development stages the milestones created and maintained were placed within the GDD.

An overview of those goals are…

- Milestone one

- Milestone one

- Milestone one

- Milestone one

- Milestone one

- Milestone one

Although the milestones themselves were not altered in anyway, their progress and outcomes were maintained throughout. Each one of the milestones has been broken down into more detail further on in this report. Section Milestones

5. To create an interactive TBS game that immerses the player in a strategy based platform.

In order to create an interactive game that immersed the player, there was a need for a strong, intelligent AI system to play against. An AI system that is seen to be intelligent is crucial. Otherwise the game will become too easy to play not creating a high enough challenge for the user.

6. To further enhance my programming and game development skills with a fully functional game.

## Game Design

### Overview

The title “Final Adventure” was the result of this project, in which one player will compete against the created AI system. The goal of the game is to strategically move your chosen characters around a ten by ten map forcing your opponent into difficult to fight scenarios. The player will carefully select five characters of either different classes or the same and place them into starting positions. Each class type boasts advantages and disadvantages within the battle field, so selecting a worthy combination is the first step to victory. As stated previously the players manoeuvres their characters around the map and is allowed to execute one action per turn. This may be a simple attack or a more complex ability or spell. Once a characters health points (HP) has been reduced to zero they are removed from the game given the advantage over to the opposition. The game will end once the player or the AI has run out of characters.

A lot like chess the game focuses on a simple terrain, yet the understanding of each different characters and their varying strengths in battle is what brings victory to a player.

### Game Flow

Upon loading the game the player is presented with the main menu. There is not much in terms of options however they are able to continue towards the game or they are able to familiarise themselves with the controls and aims of the game before starting.

Once the game has begun the player will be navigated towards the character select scene. This is where the player is able to explore the different types of character classes ranging from wizard to warrior. This scene shows off the given stats for each class e.g. strength, defence, health and more, as well as their available abilities and or spells. Understanding how these stats effect the gameplay are important however it is also something that the player will need to experience in game to fully understand. The player is also able to read an overview of spells and abilities available to give them more insight to the class choices.

Once the player has selected their five characters and placed them within the designated positions the game will begin. Starting with the player they will be given the option to move each one of their characters in turn as well as the option to complete an action. Once a characters has moved and completed an action their turn is over, however if they choose not to move or not to complete an action they are able to wait. This will skip their turn and move on to the next character. The back and forth between the player and AI will continue till a victor has emerged. The game will then return to the main menu allowing for another go.

### Core mechanics

#### Selecting a team of characters

This is the first interaction for the player, increasing the need for it to be totally immersive. This scene includes a lot of animating between actions e.g. spinning the carousel of character types and moving the characters into position. These subtle actions are enough to draw in the players attention getting them hooked as soon as possible. This core mechanic allows the user to explore the characters for the first time, granting them additional knowledge of the overall look and feel. Using the left and right arrow keys rotates the carousel displaying either the next or previous character. Each character in the carousel can be rotated on its Z axis with the D key rotating them right and the A key rotating them left. This simple feature allows the user to better familiarise themselves with the character models to avoid confusion once the game has started. Finally the player is able to use the up and down arrow keys to move through the menu. The menu doesn’t offer much but it does allow for the current character to be selected and places into position or for addition information on the current character. This additional information better explains their strengths, weaknesses plus their abilities and spells.

#### Placing characters

#### Moving the pointer

Once the game begins the player is able to start their first move. As long as the game is user the players control they are able to explore the map and investigate the random opposition. Do to this the player can use the in game pointer. Resembled as a purple tinted square, this indicates where the current game focus is. Moving the pointer around the map will show both ally and enemy information e.g. Health, Mana, Class. The pointer is controlled using the W, A, S and D keys, moving it up, left, down and right. When the pointer is manipulated around the map the current active character will rotate to face its direction. This was a simple, yet effective minor mechanic that helps bring the game alive. The pointer is used to aid the selection of movement and character selecting when completing an action.

#### Using the UI

The UI has been made to be extremely simplistic yet intuitive to use. The majority of the UI throughout the game uses the same scrolling menu on the side of the screen. The menus are created dynamically from a list of options. This has been done so it can easily be altered for each character as they all have different options. The up and down arrows are used to cycle through the menu options while the return key (carriage return) is used to select an option within the menu. The menu only ever shows a maximum of four items at one given time, so by cycling down the items the menu scrolls down to reveal the entirety of the list. As the player is only able to complete one movement and one action per turn, each item in the menu can be disabled to visually show that the action has been completed. The currently selected item is also highlighted in a light blue and animated to stretch out beyond the alternate options.

These menus use a method of main and sub menus to traverse through the cascading options e.g. selecting the ‘Spells’ menu item will hide the main menu, and reveal the sub mean containing the available spells. When in a sub menu or while finishing an action the player can press the right shift key to return to the previous menu.

#### Moving characters

To move the character during their turn the player must first select ‘Move’ from the menu. Once selected the floor will highlight with the available tiles to move to, these tiles are highlighted in a red tint. Each character has a speed stat which determines how many tiles they are able to travel each turn. The player may decide to move the character to any one of the surrounding tiles assuming that it is within their speed limit and it is not taken by another character. The user is able to decide which tile they wish to move to using the game pointer mentioned previously.

When an appropriate tile is selected using the enter key (carriage return), the game will use the A star path finding algorithm to find the best route to the chosen destination. This algorithm returns a list of coordinates that navigate from start to finish. The game then moves the character from tile to tile whilst rotating them to face the correct direction.

#### Completing an action

Much like moving the character, to complete an action the player must use the menu and sub menus to narrow down the options to just one. The actions have been broken down into more defined sections explaining their purpose in a little more detail. However as an overview, the actions are used to bring the game state in favour of the characters team. Generally this is in terms of an offensive action, tailored to bringing down the HP of the opposition.

**Attack**

Every class has access to the attack action as it is the fundamental element of damaging the opposition. An attack has no prerequisites bar that character having more than one health point. Even though every class can attack an opposition they all have varying attack ranges. The attack range has been used to bring another level of diversity towards the different classes. Traditionally the attack range is one, meaning the characters have to be next to each other. However an example of a more complex attacking action is the archer. Seeing as their role includes using a bow and arrow they are not able to attack characters within a close proximity, but are able to attack further than any other class. Knowing that every class type can attack is it not always the best option as it is based off of their strength statistic, meaning some classes are better at attack than other.

**Spells & Abilities**

Spells and abilities are more tailored towards the given class and not necessarily always offensive. Most characters have access to the heal spell which simply heals a targeted character but the heal amount is based off of the classes magic statistic. This brings another level of strategy to picking the perfect team composition. It means that a strong team doesn’t rely on a set of offensive character choices, yet a combination offensive and defensive. There are a range of other damaging spells available for use and have elegant partial effects to really bring them to life.

Abilities are similar to spells however they are aimed towards manipulating statistics and damaging the foe. E.g. boosting accuracy to ensure the attack hits, or charging up power to unload a devastating attack but with reduced chance of success than a normal attack.

Both spells and abilities consume mana when cast, meaning they are to be used situationally opposed to every turn. This again brings more strategic thinking towards each and every turn, getting the user to think about every action and how it is going to affect them in future turns.

**Wait**

Waiting is the last option the player has for each characters turn. This will end their turn and move on to the next character. Waiting is used when a character wishes to attack an enemy yet stay in the same position. Or run away from the danger and not complete an action.

#### Statistical formulas (Damage, Abilities, Hit/Miss, Critical hit)

Every action available in the game except waiting needed a tailor made formula to calculate the changes in values. This could be calculating the damage of an attack or the amount of health received from healing. A lot of these formulas are similar in structure but have been manipulated to produce the required outcomes. Here is the formula used to work out the physical damage produced for a normal attack.

D = (((2 \* LV + 10) / 250) \* (SV / DV) \* B + 2) \* Mod

D = Damage

LV = Character Level

SV = Attackers Strength Value

DV = Defenders Defence Value

B = Base Damage Value

Mod = Additional Modifiers

This formula allows the damage to scale on several different aspects. Firstly is the characters level. The character level isn’t changed throughout the game, but was included in case of expansion. In its current state each character is set to level one with the hopes of levelling up if that character successfully aids in a victorious battle. Secondly the damage is scaled based off of the class’s strength statistic. Traditionally the warrior type class has boosted strength stats as they are the head of a fight, opposed to a wizard who should keep their distance. Thirdly is the base damage value. In a normal attack action the character strength statistic is used, however this is changed for certain abilities like the focus or slash ability. Lastly is the modifier value, which mainly consists of the critical hit chance. This is where the characters critical hit value is used against a random number between zero and one hundred to determine if the hit should deliver twice the intended damage. The modifier value has been created to introduce weakness and advantage attacks however this has not been included in the final product as not enough character classes are available to create a healthy balance.

When a physical attack is being carried out the game also takes into consideration the chance of successfully hitting. The formula used to calculate each hit is not very complex when compared to working out damage values.

HC = HR \* (AC / EV)

HC = Hit Chance

HR = Hit Rate

AC = Attackers Accuracy Value

EV = Defenders Evasion Value

Hit rate is a class specific value based off of the class attack range. This means that archers have a slightly smaller hit rate as they are able to attack from several tiles away, whereas a warrior has a higher hit rate because their attack range is very small. This formula will result in a value relatively close or marginally above one. Any result over one is an automatic hit, where any hit chance under one will be compared to a random number between zero and one hundred. If the random number is higher than the hit chance multiplied by one hundred the attack will miss, otherwise it will hit.

All spells are directly cast upon the chosen recipient therefore there isn’t the need to calculate a hit chance for them.

### Bayesian Probability Adaptation

#### Summary

Bayesian probability is used to produce statistical probabilities for an unknown event based on information available either after or prior to the action. An adaptation of this method has been used to create the probability values for each human based character in terms of the chance to attack, move and heal. These are the three main options a player has when completing a turn so are the bread and butter for all the probability calculations.

#### Calculating probabilities

At the start of every AI controlled characters turn the game will run the current situation through the probability calculator. Every AI character will assess every human controlled character comparing stats like surrounding ally and opposition count, current health percentages and other variables. All of these and more have been used to carefully produce a set of values deemed viable as accurate predictions.

Ensuring that the probabilities for the three options attack, move and heal are calculated to benefit the AI’s decision making is crucial. Without this the AI might seem to make rash or in other terms stupid decisions going against the goal of an intelligent AI system.

There is no right way of correctly calculating the probability values but the formulas used have gone through several iterations to help make them more precise and meaningful.

**Attack Probability**

AP = ((ANT / ONT) \* ((TPH / OTH) \* (SV / DV))) \* Mod

AP = Attack Probability

ANT = Allies Near Target

OPT = Opposition Near Target

TPH = Target Percent Health

OTH = Opposition Percent Health

SV = Strength Value

DV = Defence Value

Mod = Additional Modifiers

This formula was created whilst playing other TBS games, by carefully analysing the different components in a player’s decision. Some of the reasoning behind a players choice may not always be cognitively assessed, instead just a habit from playing the game or similar TBS titles. This habit or instinct is comprised of so many different variables it is extremely difficult to focus them down into one formula. The above attack probability formula uses a variety of elements seen to be the most impactful in a player’s decision.

To breakdown the implemented attack probability falls into four main sections. Firstly is the character’s surrounding the target. Normally in warfare there is a great power in numbers and generally seen as a reason to attack or retreat. The number of allies is compared to the number of opposition characters but only if they are in movement and attack of the target. Any character out of range from the target is disregarded as they will not be able to directly impact the target in the given game situation.

Secondly is the remaining health of the target character verse the active AI character as a percent. These are compared to adjust the aggressiveness of the characters decision. Typically if a character is low on health they will become less aggressive and more defensive, this is also amplified when the targeting character has a greater health percentage.

Thirdly are the targets strength and the AI’s defence stats. These resemble the chances of the attack being an impactful action, swaying the decision to attack against a strong foe but enhancing it if they are weak.

Lastly is the modifier at the end of the formula, much like the attack damage calculation this acts as a final buffer towards the value. In the attack probability it is very simple, either increasing or decreasing the overall probability based on the targets class type. Assuming classes such as a warrior are going to be used more aggressively that a wizard.

**Heal Probability**

HP = (DT (DT\*JM, …, …)) \* (OTH / ATH)

HP = Heal Probability

DT = Damage Taken

JM = Job Modifier

OTH = Opposition Total Health

ATH = Ally Total Health

The heal probability is certainly the most interesting formula to be created for the game. It has a reoccurring calculation for each and every ally surrounding the current target that exponentially increases the chance of the heal action given the targets health lost as a percent and their class type.

To break this down a little further each human controlled character assesses their surrounding allies. The assessment is manipulated to increase the heal chance if one of the following situations is correct. An ally surrounding the target has very low health, the majority of the surrounding allies have lost a portion of their maximum health or the target assessing their allies are of class type that would deliver the greatest heal values e.g. a wizard.

Even after this calculation there is the chance that the heal probability could be altered. The sum of all the surrounding AI characters health is compared to the sum of the surrounding human controlled characters and used to amplify the probability. This ensures that the heal action being completed couldn’t be used more effectively as an attack due to the health differences or vice versa.

**Movement** **Probability**

Probably the least important of the three options, but still an important element to take into account. Calculating the movement probability of a character is by far the most difficult to achieve as it can be done either offensively, defensively, tactically or sporadically.

Formula needs adjusting!!

#### Assessing probabilities

At the end of the probability calculating each and every human controlled character will have a value ranging from zero to one for each of one of the three assessed options. Each one of the options is used to finalize the actions completed by the AI. The vast majority of the time the actions completed by the AI are rather predictable, but this is a great sign. This shows the AI system is resulting in actions a human player would cognitively decide to do oppose to something seen as random.

#### Decision making

Ensuring the AI make intuitive decisions was one of the biggest back and fourths in terms of development. Even though the final outcome is actually rather simple it was important to test a variety of different methods to truly understand the underlining elements needed. The decision is broken down into two stages.

The first stage is finding a target on the opposite team. To do this the game goes through all the human controlled characters that are in attack range and applies weighted modifiers to the probabilities created in the previous step. The three weighted modifiers hold different values but starting with the most impactful is the amount of health missing from the target character, next is the chance that character is going to heal another member of their team, lastly is their chance of attacking a fellow AI character. These weightings have been associated with the given attributes because they are seen to be the most impactful when deciding which character to target.

To give a better understanding of the approach the following has been produced.

* Targets missing health produces a value between zero and three
* Chance of the target healing another produces a value between zero and two
* Chance of target attack another unity produces a value between zero and one

The sum of the three values is assigned to each assessed target as a priority level.

The target with the highest priority level will be selected as a target. However there might not be any targets in attack range to evaluate in the first place. If this is true all of human controlled targets out of attack range are then assessed with the same criteria. It is important to assess the targets in attack range first as they are the only characters that can directly impact the AI unit.

This first step will always produce a target, even if their priority level is extremely low. The second step is for the AI unit to evaluate it’s in range allies for healing. No were near as complex as the targeting system, the healing target finder compares all the ally units within movement and heal range, then orders them descending by the percent health remaining. If the target at the top of the list has a health percent lower than the provided bound the AI will overwrite any human controlled target with its ally and prepare for healing. The bound used for comparison is altered based on the AI’s class type e.g. increasing if the current AI of a Wizard class type. This way the AI units are going to act more specifically to their given role.

If the AI does not need to heal one of its fellow team members it will resume attacking the opposition target. At this point the game will pick a random offensive spell or ability from the list of available options linked to that class type. Assuming the current AI character has enough mana to cast or complete the action they will do so, otherwise they will fall back to the standard attack option.

#### Storing data

Part of the implemented AI system includes a level of learning, this enables the AI to adapt over time making its decisions less programmatic and more assessment based. In order for the AI to begin its learning there needed to be a constant source of data being fed to it. This data is comprised of several different variables but the end result is taken from the human players and what they decided to do at the end of each given turn.

To store the required data a simple web server, application program interface (API) and database were created. The focus isn’t towards the web production of the product but more details can be found ….

The API has two simple calls available. The one in regards to storing data requires each one of the following fields to be filled.

* Number of surrounding allies
* Number of surrounding AI
* Total number of ally characters
* Total number of AI characters
* Class type
* Target health percent
* Target Mana percent
* Movement (True/False)
* Action completed

The variables are taken at the beginning of the characters turn, and sent off to the database at the end of their turn.

#### Learning

One of the more important areas of the AI is the learning system in place. Although it isn’t the most intricate in design or uses it proves to be a vital aspect of the overall behaviour. At the beginning of the AI’s turn it will assess the opposition and retrieve all the corresponding data to their current in game scenario. This will return all of the data entries matching their current situation and is used to aid the decision making process. The returned entrees are first grouped by the action value, and then ordered descending by the count of each group type. This will place the most common completed action at the top of the list. The game takes into account that the information gained from the database might not always be useful. If only a few entrees are returned it is ignored completely. This stops the AI from making decisions based off of little background knowledge. When enough data is returned and organised it plays a small part in making an overall decision, if this wasn’t the case the game could be played and manipulated in such a way to jeopardise the AI.

#### Completing an action

### Architecture

## Modelling and User Interface

## Project Management

## Legal social ethical and professional

## Reflection

## Communication