# Technical Documentation

**Primary Algorithms:**

* **A\* Search Algorithm:** This algorithm will be helpful whenever generating the game’s map, more specifically to find the most suitable, shortest path between two locations on the map to create a waypoint system. This feature will benefit the player as it will help them to navigate an open world environment at ease.
* **Sorting Algorithm**: This algorithm could be useful for things such as the shop that player can buy items from and could sort the items into the correct alphabetical order, high to low (price) and vice versa etc, to enhance the player’s experience.
* **[ARTIFICIAL INTELLIGENCE] Algorithms for Steering Behaviours:** Behaviours such as fleeing, following and arriving at waypoints will be useful for certain AI in our game to both make NPCs seem more human-like and to also contribute to the game world (i.e. Witch Doctor fleeing from player until he arrives at certain point on map and comes to a halt). This can be programmed by using basic object transformations and a set of switch/if else statements.
* **Finite State Machines:** A finite state machine is required for AI to enter different states (i.e. *Enemies:* Attacking the player, Idle, Wandering etc). This can be via general behaviours or the different states of animations of the player character and NPCs.

Player is close enough for the NPC to attack and take health from player

Player enters the NPC’s line of sight at a relatively close distance

Player approaching the area of NPC

Attack Player

Idle

Wandering

Chase Player

Player leaves the NPC’s line of sight at a relatively far distance

Player leaves the area of NPC

Player is not close enough for the NPC to attack and take health from player

An example of a finite state machine for a random NPC which spawns whilst the player is exploring the land (i.e. one of Malum’s men)

**Game Cheats for Testing:**

*Cheats should be used by the testers by entering a secret cheat code which will be supplied to them by the programmer.*

* Cheat which allows the user to skip to different levels in the game.
* Cheat for teleporting throughout the game’s map on the input of x and y values
* Cheat to skip to time of day by inputting the time in the 24-hour time format (i.e. 13:00 as opposed to 1:00pm)
* Cheat for increasing/decreasing the money player has
* Cheat for increasing/decreasing player reputation
* Cheat to increase/decrease health state
* Cheat to have unlimited money/health
* Cheat to add items to inventory
* Cheat to navigate the map via flying

\* Some of these cheats may also be used by the player in “New Game +” to enhance user experience and the reward for completing the game.

**Hardware Requirements:**

* PC equipped with mouse and keyboard
* Minimum system requirement of:

OS: Windows 10 / MacOS Catalina

Processor: Intel Core i5 / Ryzen 5

Memory: 8 GB RAM

Graphics: GeForce GTX 660 / Radeon R9 270

Storage: 4 GB available space

Sound Card: DirectX compatible

**Game Engine:**

The game engine the team have agreed to use is ***Unity.*** This is because all team members have had prior experience with this game engine. It is also widely used and therefore a vast amount of information and solutions can be easily found on perhaps, online forums, if we were to encounter any critical errors during the development process. Unity is also considered rather easy to use in comparison to many other game engines, and therefore will be a lot easier for our small team to handle and navigate.

Overall, the main reason why we have chosen to use Unity as the game engine to develop The Cure on is due to the fact of it being free to use (as long as a significant profit is not made), and will suit the restrictions we have budget-wise with the project.

**Good Will:**

A method will be required to set and store the percentage of good will the player has. Good will refers to the reputation that the player has with the NPCs in the game. The player can gain good will by doing tasks for NPCs or interacting with them in the correct way. This will essentially be the main point system within the game and will determine the quality of the ending the player receives; it works as follows:

Criteria for the percentage of good will required for each ending:

Possible good will which can be gained by the player at certain points in game:

**Ending 1 (“Bad” Ending):** <=59%

**Ending 2 (“Normal” Ending):** <=94%

**Ending 3 (“Golden” Ending):** >=95%

**Tutorial Level**: 12%

**Witch Doctor:** 13%

**City of Haven:** 40%

**Healer Level:** 18%

**Ingredients Level:** 18% / 45% **(dependant on success of previous level**)

This method will be comprised of constantly checking if the Good Will is increased/decreased and running through perhaps an if statement to compare the current good will percentage to the criteria whenever an ending is required to be determined.

# MVP Systems & Features

**Multiple Endings**

If the product were to be developed to a bare minimum the team have agreed that it would still be suitable to keep the multiple endings feature. This is due to it being one of our major selling points and we would not like to reduce the amount of current selling points.

However, if we were to produce the project to a minimum, the game would consist of 2 core endings as opposed to 3. This is because it would be a lot more time consuming to produce less endings, more computationally efficient and more cost efficient. It would be more cost efficient due to less voice lines needing to be produced, which would allow the team to pay the voice actors less. It would also require less production of cutscenes and other relevant and costly assets.

The endings which we intend on producing in the minimum viable product is Ending 1 (The “Bad” Ending) and Ending 3 (The “Golden” Ending). This is because we believe that these endings are the most diverse in comparison to one another. Also, allowing the player to receive the most dissatisfying ending will give them the most incentive to replay the game to receive the most satisfying ending.

**Good Will**

To determine the multiple endings, we will need to keep the concept of good will in the minimum viable product. Also, the algorithms and methods which are associated with the good will system should be simple to implement. However, the tasks that need to be completed to achieve a high percentage good will can be lessened.

Such tasks include tasks that do not have a direct impact on the general story, such as delivering presents to in-laws or helping the shepherd find his sheep. (i.e. “Short term goals” listed on the Player Objectives/Progression page.

**Technical Specifications**

* Minimum system requirement of:

OS: Windows 10

Processor: Intel Core i5 / Ryzen 5

Memory: 8 GB RAM

Graphics: GeForce GTX 660 / Radeon R9 270

Storage: 4 GB available space

Sound Card: DirectX compatible

If the user is intending on playing on a PC which does not meet or exceed these specifications, they should not run it. We expect most standard laptops that are owned by casual PC gamers (who are of the student demographic) to meet/exceed these specifications.

At minimum, we would like to release The Cure on Windows operating systems and exclude Mac operating systems. This is because the majority of the development team own Windows computers and programming/testing/releasing in two different operating systems could be time consuming. Also, we believe that most PC gamers own computers with the Windows operating system, so we would be catering to the majority.

Users that do not understand these specifications will be advised to check this using an online tool such as <https://www.systemrequirementslab.com/cyri> .

**Time System**

The game will have time system implemented to increase realism and player immersion. This time system will affect the tasks the player can perform (i.e. Only talking to the witch doctor at night) and the environment around them (i.e. Lighting and skybox changes).

The time system will run off the idea that 1 minute in real-time equates to 20 minutes in the game time; This essentially means the player will have reached the full game cycle after 1 hour and 12 minutes of real-time. This time system is used in The Elder Scrolls V: Skyrim and has proven to be a successful time mechanic.

The time system can be implemented by essentially assigning time to a hidden game object, with a set of conditions that must be met at each given time when reached. (i.e. Villager NPC follows path to into their home at 18:00)

The player can skip a portion of the day to prevent them waiting with no tasks available, but to prevent this making the game not challenging enough, the player can only sleep if they have access to their bed, in their home in Peperit. This sleep function will require the user to input the time they would like to sleep until.

# New Game +

An option for New Game + is also necessary to produce the full version of *The Cure*. This is a necessity as it allows the user to replay the game with extra features to keep gameplay fresh and interesting to the player. This will further give the player an incentive to replay the game to achieve a better ending if they were dissatisfied in their first playthrough. A better ending received will hopefully increase the player’s opinion of the game as a whole- leading to *The Cure* to be promoted via word of mouth.

Some of the features which *The Cure*’sNew Game + entails include:

|  |  |
| --- | --- |
| **Feature Title** | **Feature Explanation** |
| An Animal Companion | Whenever the player enters a New Game + playthrough, they will receive an animal companion they would like to be present with Ordinious on his travels. This companion is not present during a player’s first playthrough of the game, so it acts as a motivation to play through it once, then once more.  The companion is named Arthur, a white horse who provides the player with speed whenever the player is on it. The speed the horse provides is approximately 50% increased from the normal running speed and 70% faster than the normal walking speed.  Whenever the player is not on the horse, it will follow behind them.  This type of bonus will also improve the player’s experience as they can get from and to locations much faster and therefore, the game will seem much less slow paced. |
| Cheats | The player will be offered to use a selection of cheats. The ability to use cheats in game will have already been implemented in the developer’s version of the game, therefore these should be simple to implement in New Game +.  The cheats can be used using cheat codes in menu the player can access by pressing a certain key. This menu will include different sliders (for Enable/Disable) which the player can amend to their liking.  The cheats can prove to make the game more fun for the player whilst preventing them from being required to spend large amounts of time on their second playthrough. |
| Extra Good-Will | To reward the player for playing and completing the game once, the player will automatically begin the game with 20 additional good-will points, whereas before they would have begun with 0. This will give players a better chance of achieving the “golden ending” and will leave them feeling much more satisfied, leading to potentially a better outlook on the overall game. |
| Visible Good Will | There will also be a visible good-will status bar, which was not previously present in the first playthrough. This allows the player to view which interactions/choices have a direct negative/positive effect on the good will meter as they happen. It also gives them incentive to perform non-critical tasks to boost their good-will points. |
| Keeping existing relationships | The player can keep existing relationships which some major NPCs in New Game + as a reward. This will allow interactions and choices that would initially be negative, to be less so, giving the player a much better chance to receive the “golden ending”. |
| Keeping some items in their inventory | In order to save the player from re-collecting certain items by performing more insignificant, tedious tasks, they will be able to keep the previously collected items in New Game +. For example, the player will begin the game with the Sana Lilies that they collected in the first playthrough and this eliminates the need to visit the gardener. |