Isaac Dunn

Generating Character Narrative AI

Contents

[Abstract 2](#_Toc102125731)

[Literature Review 3](#_Toc102125732)

[Research methodology 3](#_Toc102125733)

[What are methodologies 3](#_Toc102125734)

[Waterfall 3](#_Toc102125735)

[Agile 4](#_Toc102125736)

[Which is better for the research project 4](#_Toc102125737)

[Introduction 5](#_Toc102125738)

[What is narrative AI 5](#_Toc102125739)

[History of Narrative AI 5](#_Toc102125740)

[How narrative AI is used 6](#_Toc102125741)

[Methods of generating narrative AI 8](#_Toc102125742)

[Behaviour Trees 8](#_Toc102125743)

[HTN plans 9](#_Toc102125744)

[Recursive narrative scaffolding 11](#_Toc102125745)

[Natural language processing 12](#_Toc102125746)

[Emotive programming 13](#_Toc102125747)

[Conclusion 14](#_Toc102125748)

[Artifact – Design 16](#_Toc102125749)

[Project requirements 16](#_Toc102125750)

[Wireframe for artifact 16](#_Toc102125751)

[How the artifact will be set up 17](#_Toc102125752)

[Artifact – Implementation 18](#_Toc102125753)

[Further improvements 19](#_Toc102125754)

[Reference list – may be useful 20](#_Toc102125755)

[Bibliography 21](#_Toc102125756)

# Abstract

Many developers have attempted to procedurally generate narrative using AI but this is seen in very few games with common examples being Rimworld (2016) and Dwarf Fortress (2006) which have managed to develop a way for randomly generated characters to interact with the environment and each other to create a narrative which the player is invested in. in this study there will be a discussion of different methods to create a narrative AI for characters to interact with each other, develop relations and to call back on previous events for those interactions. Overall, in this project many methods were used including {add methods here} before it was found that the {best method} was ideal for this project.

# Literature Review

## Research methodology

### What are methodologies

Whilst there are many programming methodologies each are used for certain situations and deciding which to use can be very important for any project. Here the discussion will be limited to just waterfall and scrum methodologies, and which one is more suitable for a research project.

### Waterfall

Waterfall according to (M. Mahalakshmi, 2013) is a methodology where a project is developed on a set path with no deviation. The methodology is made up of 5 stages first being the requirements stage where all of the project features, software and hardware requirements are assessed this is done to aid the next step. the design stage is where the development of the features is planned out often with pseudocode and diagrams to help developers understand how the project will work. next is the implementation stage where the designs are implemented, in terms of programming this is taking the pseudocode and the diagrams from the design phase and physically programming it to develop the project, next is the verification stage where the project is tested to see if it meets the requirements from the first stage and is it fails the verification then the project would be repaired to meet the requirements. The final stage is the maintenance stage where the project is released and maintained to ensure it still works as the original requirements asked for.

This model is very useful for developing a project with that can or should be fully planned out as the advantages for a project created using the waterfall methodology as stated by (M. Mahalakshmi, 2013) are that because it is a sequential model with no deviations in its development means that development can be comparatively fast compared to other models like scrum as all of the features are planned out and just need to be applied stopping the scope of the project from getting larger than expected bringing development times down. However, there are also some disadvantages to using this model which are that because it is sequential once one stage of the methodology is complete you can not go back to a previous stage meaning any issues that are found from one stage of the project can not be changed resulting in poor quality project unless plenty of time and resources are given to ensure that the project has no or at least few mistakes. This is also an issue as if a client or project manager wants any additional features halfway through development there is no mechanism within the waterfall methodology that would allow for those features to be added.

### Agile

According to (Khalil & Kotaiah, 2017)the Agile methodology is an adaption of waterfall to help avoid the issues of the Waterfall methodology as mentioned above, while the stages on an Agile project are similar there is a major difference with the Agile methodology is that previous stages can be revisited to add additional features allowing any project to be expandable resulting in a better project overall this can also be backed up by (Khalil & Kotaiah, 2017)showing the total number of failed projects using the waterfall and agile methodology as well as challenged projects i.e. incomplete project and a successful project with all required features added and as seen in appendix 1.1 the average fail rate for agile(9%) is lower than waterfall (20%) with success rates being 54% for Agile and 35% for Waterfall showing how Agile can be have better odds of being either successful or at least partially successful project. However according to (Sheetal Sharma, 2012) the disadvantages of an agile project are that the time and cost requirements can go greatly over the target if managed poorly as the project is poorly planned as the scope is increasing causing unpredictable time and development costs which can cause the project to fail. Also due to the constantly changing project it may be harder to document the project as the documentation is also constantly being changed and expanded making maintenance of the project harder.

### Which is better for the research project

Choosing a methodology for this research project is crucial as a bad choice may cause this project to fail as according to (Kaushal Chari, 2017) many projects fail due to not choosing a methodology compatible with the type of project being made. As this is a research project waterfall could be useful as the research for the project could be for its design elements however there is a risk in utilizing this methodology as mentioned before there is no back-tracking allowed on a Waterfall project, if the project fails then there is no more planning whereas Agile allows for a failed project to be reiterated on although at the cost of extra time. Another reason to use an Agile framework for this project is that extra features and discussions can be had if time is available. However Agile is a whole group of methodologies and picking a specific methodology should be considered. According to (Igor Ribeiro Lima, 2012) Scrum is a methodology that could be useful in research setting as it allows a project to be developed in a series of sprints to get individual tasks done. This makes it useful as a research project is difficult to define the scope of early on and having this be the methodology for researching and developing the artifact means that the project can be expanded for as long as there is time. An important feature of the scrum methodology would have to be changed however as according to (Marchesi, Mannaro, Uras, & Locci, 2007)Scrum works with teams for example a Scrum master, product owner and individual team members however this is a research project with only one member of the team meaning that many of these roles are redundant and are not required is the planned implementation of the Scrum.

Overall Agile should be used in the research project as it allows flexibility in a field currently being studied and likely to need reiteration and redesigns to create the project successfully as well as further discuss potential features that could be added to the project or alternate ways for the project to be implemented. The specific Agile methodology chosen shall be Scrum as it provides the ability to sprint through features that can be expanded on as the research expands giving more liberty to research to add to the project.

## Introduction

### What is narrative AI

Narrative AI is the use of artificial intelligence techniques to generate stories this could be a whole story generator as seen with (Deutsch, 2016)’s deep learning algorithm which generates entire stories from a top-down approach or (Bay12Games, 2021)’s Dwarf Fortress multiple agents and ai systems that create an emerging narrative i.e. each character have likes and dislikes which decides what they do or a personality that affects relations with other characters.

### History of Narrative AI

While it may be hard to define narrative AI so dating the first use of it may be difficult but, according to (procedural story telling book ref ) The history of narrative generation in games could be dated back as far as 1949 with the board game Cluedo and some of the earliest games to utilize narrative ai was the EA developed, Murder on the Zinderneuf which was a murder mystery game that generated a new mystery with each playthrough having a variety of characters any of which could be the murderer which the player had to deduce with in game clues that they could find by exploring and while being a very basic story generator with the only features changing in each playthrough being who murdered someone, the name of the victim and the clues for them, the game had to deal with the extreme computational limits of the time having only 48 kilobytes of memory which is minuscule compared to average gaming memory nowadays so since then narrative AI has been able to encompass more of the narrative in certain games like Dwarf Fortress (Bay12Games, 2021). which released in 2006 which heavily utilized narrative AI to generate an entire world with whole characters, nations and history being generated to create a unique and immersive world which has inspired many more games such as Rimworld in 2018 (Ludeon Studios, 2018). which focuses more on individual relationships between the generated characters.

### How narrative AI is used

As mentioned before narrative AI can be used to create whole stories as seen with Jamie Brew’s predictive text program (Hudson, 2016). which was used to generate a story feeding stories like Harry Potter to generate its own telling of the story. It worked by using a long-short term memory (LSTM) neural network which allowed the program to generate the story using the books as a reference and estimate what words would be used to create the story.

Other ways narrative AI can generate stories is through having individual AI agents and systems which can create emergent behaviours as seen in the previously mentioned games like Rimworld and Dwarf Fortress which generate a narrative using multiple methods with both games recording events that happened in the game and then reusing them for other things for example both games a an art creation system where a character makes a piece of art and its description is generated using those events to create art unique to the playthrough as seen in fig 1. Rimworld also has dialog between characters where they can talk about a number of things, flirt or insult each other which can raise or lower their reputation with each other fig 2. Another system in the storytelling AI in Rimworld that uses the value of the settlement you built as a heuristic to decide what events to throw at you with the more money you have the worse the events the AI can throw at i.e. large raids of hostile characters that could kill characters fig 3. All of these events together create the emerging story for the purpose of getting the player more invested.



Figure : art description from Rimworld generated using characters from the playthrough i.e. Boomrat and the Muffalo. this also show events being used with a date and the event of Boomrat catching fire

## Methods of generating narrative AI

There are many methods to creating narrative AI each being useful for different mechanics such as character behaviour as well as dialog and other text generation needs. Each method will be discussed to see how they work and how they can be utilized in this project.

### Behaviour Trees

To create behaviours for characters in games there many methods and a popular one to look at are behaviour trees. Behaviour trees according to (Anurag Sarkar, 2021). as the name implies are a tree of nodes which run behaviours with several transition methods to control how the AI agent flows between the nodes. These transition methods are sequence which will sequentially run through each child node and run the behaviour for them however it only succeeds if all child nodes were successful is running their behaviours. Parallel which works by simultaneously running all child behaviours at once and only succeeds its tasks if a given number of child nodes were successful at running their behaviours. Selector is like sequence as it will sequentially run all child nodes however it will be successful when any child is successful ignoring the rest however the selector transition type will fail if all child nodes fail to run their behaviours. According to (Simpson, 2014), the managing director of the game Project Zomboid. all previously discussed transition types are known as composite nodes as they require one or more children however decorator nodes can only have one child these include inverter nodes which returns a success or failure opposite to what the child node returns, a succeeder that always returns success and a repeater which will repeat running its child node a given number of times before continuing but there is also a version that can repeat until the child returns a fail known as a repeat until fail node.

Behaviour trees are popular for modelling AI behaviours because according to (Anurag Sarkar, 2021). they allow for relatively complex reactive AI as the tree structure con be made more complex to suit a number of emerging characteristics from few behaviours which could be useful for making an AI agent decide what to do for example throughout the day an AI agent has a number of items on their to-do list with the priority being eating breakfast before going to work then going out with friends later in the day. This allows for priority tasks to be completed first but could transition between behaviours when required as if the AI agent’s car breaks down, they may stop to call a mechanic and wait before heading to work as the behaviour tree unsuccessfully attempt to get to work the behaviour would change. This can also have extra behaviours to be added on without adding much complexity as all the behaviours are modular and can just be added onto the tree in a clear order for what behaviours will trigger making ideal for relatively complex AI behaviours. Overall behaviour trees allow characters to establish their behaviours and run them for each character making it ideal for individual bottom-up approach for games like Rimworld where there are many characters with their own goals.

Diagram

Description automatically generated

Figure 2: graph showing behaviour trees with different node types (Simpson, 2014)

### HTN plans

According to (Humphreys, 2021). a HTN which stands for Hierarchical Task Networks are type of behaviour tree that are often used for planning in AI agents and is a behaviour tree made up of either primitive tasks which are simple tasks like opening a door or compound tasks which are a set of primitive tasks which have multiple solutions so for example a locked door may require a key so the AI agent would have to first find a key, then unlocking the door with the key before finally opening it or an alternative solution could be to bash the door down to open a door. This makes HTNs very similar to behaviour trees however HTNs differ by essentially generating its own tree from a pool of tasks known as the search space. The search space is a set of primitive or composite tasks which are then used by the HTN planner by taking all the primitive tasks and integrating them into the HTN’s behaviour tree and then deconstructing the composite tasks into primitive tasks and implementing those into the behaviour tree all these primitive tasks are then decomposed into a planned goal allowing a given AI agent to develop a plan. This could be for example an enemy AI soldier attacking the player, they will shoot, roll into cover and throw grenades however if a player gets to close new tasks will present a new way for the AI to achieve its goal of killing a player i.e. melee attack. According to (Ilche Georgievski, 2014). this makes HTNs useful for creating interactive stories as each character in a given story could be given a goal which they can make a plan for using the search space to gather tasks to achieve that goal with for example a exiled king character in charge of a small county could have a number of actions available to them to either to do nothing however it may have a goal to retake their land so they may focus on tasks that build them an army to take back that land driving a narrative. While this is useful to generate an overarching narrative between all characters this however might not be ideal as it undermines the individuality of the characters as it forces the behaviours of multiple characters to work for one goal rather than to work for their own which behaviour trees are more suitable for as for the overarching plot while it may be less capable at making reactive AI as a group of behaviour trees are preferable for this project.

Diagram

Description automatically generated

Figure 3: Graph of a HTN applied to an individual (Marc Cavazza, 2003).

### Recursive narrative scaffolding

Recursive narrative scaffolding is a term given by (Garbe, 2018). to describe the use of events in a narrative AI being used to generate a history which can be referred to. This comes from an idea by the works of Russian psychologist Lev Vygotsky you wrote about the psychology of narrative scaffolding where history can be codified into events which eventually develop into goals.

This takes into consideration the use of narrative scaffolding for use in world building where for a character could be born into a noble family and that family could have a rival noble family which may have wronged the character’s family in the past i.e., sided against the king in a civil war. This lays the groundwork for this character as when the story for the character starts there is a predefined history for the character that makes other characters in their family, or the rival family have history with them and drives a narrative. Now this can be used is a procedural generation model as mentioned by (Jason Grinblat, 2017). where events are generated before the character exists which defines how they will subsequently interact with the world.

How recursive narrative scaffolding could be implemented is by applying this model to all generated characters in a similar manner to projects like (Max Kreminski, 2020). Where the scaffolding of the narrative is built off previous events and individual character’s behaviours with each character in the project being generated with a series of traits and personal quirks with predetermined behaviours assigned to them with those traits also affecting the character’s history which will then be generated. This has the benefit of creating stories that are both guided and customisable for the player with the scaffolding creating stories and the player choosing the traits that then generate the character’s history. However, unlike methods like natural language processing as narrative scaffolding will require traits and events to be determined and programmed in which may take time to set and requires knowledge of good story crafting as well as balancing how events that will cause narrative scaffolding to take place will transpire.

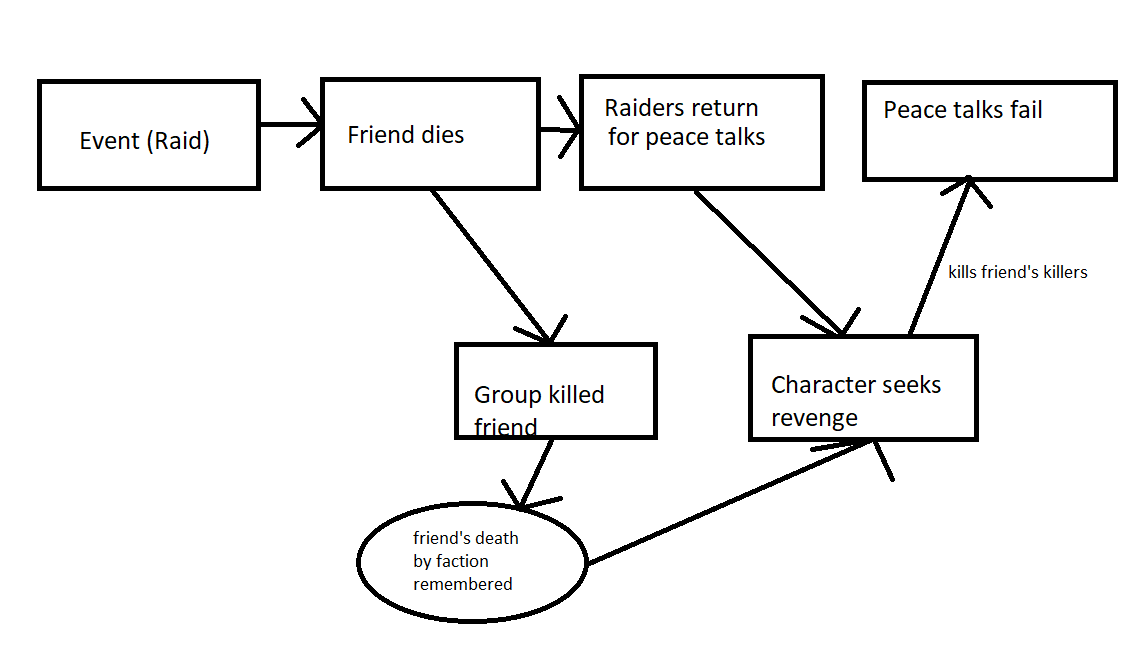


Figure 4: diagram showing reverse narrative scaffolding playing out through events and characters reactions to those events

### Natural language processing

According to (IBM, 2020). Natural language processing or NLP is the process of using a computer to process and understand language using statistical and machine learning principals. NLPs like are able to recognize speech and the intent behind its content for example the word catch could mean the verb ‘to catch’ but an AI may not be able to understand the saying “what’s the catch” as catch here means a downside of something rather than the typical meaning of the word with speech tagging through machine learning and/ or statistical analysis an NLP can be taught the difference between these to saying to add context and understanding of the content given. Models like GPT-2 (Alec Radford, 2019). an AI agent could be able to talk and understand dialog between different characters using GPT-2’s NLP to use the sentences given by other characters and process a response to them and this response could even implement the characters background and relation with other characters as a way to guide how a given character would respond to a sentence for example a character (Bob) talks about cats to another character (Dave) however Dave hates so the response could be tagged as an angry response about how the character hates cats and the NLP will attempt to generate the response and if Bob likes cats the NLP could use that fact as a tag to attempt to argue back to Dave. This could massively improve any narrative AI however NLPs face many flaws that may make it unusable in this project as according to (Vincent, 2020). many NLPs like GTP-2 and its successor GTP-3 still have many issues with speech recognition and intent with one example of an AI with GTP-3 being asked “which is heavier a pencil or a toaster” with the NLP simply responding “a pencil is heavier than a toaster” so while NLPs like GTP-2/3 are very capable of understanding language and context they often can have issues and require fine tuning with its complexity being very difficult to work with. This complexity can also cause issues with speed as implemented an NLP into a real-time game into even a single character could greatly affect performance let alone multiple characters talking with each other. These performance issues can be seen in games like the text-based dungeon crawler, AI dungeon (Latitude, 2022). which can take up to a few seconds to generate a line of dialog making it very difficult to implement into a real-time game without any major slowdown issues or major optimisation which may be too large in scope for this project to handle. Another issue with the text generation in NLP is how they can be unpredictable in what they will say, and game designers and writers may want to make sure that the content they are producing is suitable for what they want in the game and NLPs are naturally unpredictable as previously seen. As stated by (Emily Sheng, 2019). in the case of GTP-2 natural bias in the AI could contribute to undesired stereotypes in what characters would say an example stated in the paper shows the AI being asked a number of statements and were judged on what someone from a demographic is and the results showed a net negative for certain groups such as black and gay people which could create unwanted bias in dialog created by NLPs which may require further tweaking to solve.

Overall while it is an effective strategy to utilize an NLP as a text generator its unwieldy, unpredictable and performance intensive problems make it unusable in this project.

### Emotive programming

According to (Marc Cavazza, 2003). there is a need for an emotional layer to inform the narrative layer on how to act as the narrative layer uses emotional responses and goals to create events this can be developed to a similar manner as discussed by (Alvim & Cruz, 2008). where a fuzzy state machine is applied to simulate emotions in the AI agent which can change depending on what happens to a given character making happy or sad when something that is good for them happens this could be something like a gift which makes them happy or someone they liked dying driving an emotional response from the emotional layer to the narrative layer this can also be further expanded with the addition of a relations emotional response where peoples actions change depending on how much they think of each different character. Each of these emotions can be added just by a variable to store the value of the emotion and the rules relating to that emotion as seen with a similar system used by (Li & Campbell, 2010). where each emotion is valued from -1 for negative values i.e., sad and +1 for positive values happy however this model is more used for player vs non-player interactions and for emotions like relation i.e. admiration or hatred for a character a list of all relations for each character the character has interacted with will be required. This overall adds a dynamic system to simulate emotional states in the AI to control behaviours.

A picture containing diagram

Description automatically generated

Figure 5: Image showing a diagram of how the emotions can be shown (Li & Campbell, 2010)..

Table

Description automatically generated

Figure 6: diagram showing how events can change a character’s emotion (Li & Campbell, 2010)

## Conclusion

Overall the decision of what behaviour system would be better for characters to use depends on how the narrative will be utilized and as the project will have more individualised characters, behaviour trees will be preferred to HTNs as an adaptive behaviour tree can be applied to each character with each decision the character making being for their own set traits making the narrative overall more individual in scope whereas HTNs can provide a more greater scope narrative with all characters working together following a combined goal overall either system could work for creating narrative in the artifact with character relationships and drama between individuals whereas HTNs may be better for generating narrative through wars and factional drama but as the scope of the project is focusing on smaller groups of around a dozen or so characters it may be more suitable for the characters to have behaviour trees instead of HTNs for that reason however for greater numbers of characters making their narrative harder to keep track of and thus less important it may be better to focus on the narrative of the society the characters live in only following major characters controlling less individual characters. As for the generation of narrative text, the use of reverse narrative scaffolding methods to develop text based on the simulated traits and attributes of a character or entity will be used instead of natural language processing due to the issues of language processing having with efficiency and general ability to create comprehensive text due to the experimental nature of the technology making it not useful for this project with reverse narrative scaffolding being able to build up a history of events and reuse those events as context for new decisions or text for the character to utilize.

## Artifact – Design

After researching for the project its time to recreate the research and test if it helps to generate narrative. First step for this is to plan out how this project will work with the software requirements, wireframe of how the artifact will look and a discussing on how the research will be applied to the project.

### Project requirements

The project will utilize the Unity game engine to develop the project. This was chosen instead of an inhouse crafted game engine or an engine such as Unreal Engine 4 as the engine will not affect any parts of this project as the performance of the project is not expected to make the choice of engine crucial with prior experience in using Unity and not as much in Unreal Engine 4. As for not using the inhouse engine and instead using Unity may even benefit them with ease-of-use features and the time saved from the engine being already developed. Overall, the small scope and need for computation resources of the project mixed with the time constraints to finish the project means that prebuilt engines are going to be the best option for this project and as the choice of engine will have minimal effect on this project Unity will be chosen due to prior experience.

### Wireframe for artifact

To help visualise how this project will be set up a wireframe for the artifact has been made with each feature shown in the image below having a number assigned to detail what parts of the project mock-up ui represent (underlined in red).

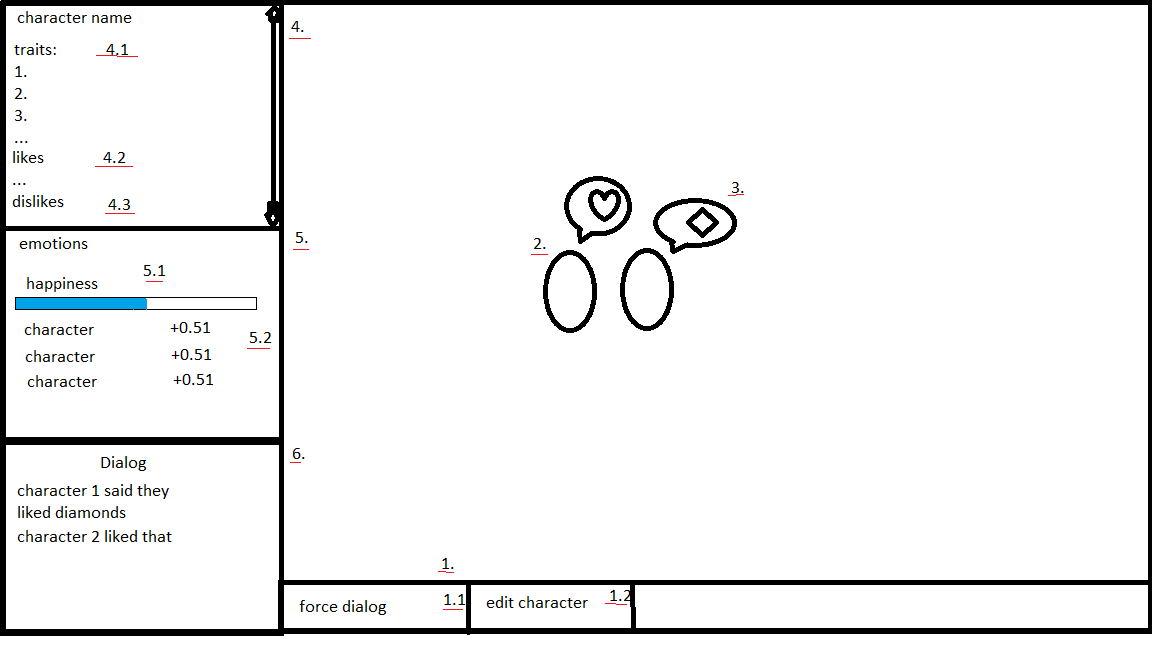


Figure : image for the wireframe

1. This represent the testing buttons for the project giving the user the ability to affect the artifact this is not limits to forcing dialog (1.1) and editing character to change their traits, likes, dislikes and emotional state of the character.
2. Represents the character on the screen (ovals) which will move around carrying out basic actions and interacting with other characters.
3. To aid in understanding what is going on between characters without having to look at the dialog box, speech bubbles will be added to the artifact which symbolize what the character is doing be it chatting (and what about i.e. a dog symbol could represent a conversation about pets), flirting or fighting.
4. To understand what a character’s traits, likes and dislikes are with those details just being listed.
5. This represents the emotional state of the character with meters showing the happiness level of the character as well as a list of relations with other characters in the artifact.
6. The dialog box will list all dialog between characters in the artifact allowing the user to determine how well the narrative system is working with events also being shown i.e. a characters declaring that they are in love/ friends with someone or someone swearing vengeance on a attacking party.

### How the artifact will be set up

The plan behind this project is to mimic much of the emerging narrative systems in Rimworld and dwarf fortress with a small settlement filled with auto generated people each with different personalities, like and dislikes which change how they interact with each other some could have an abrasive personality that might make them more likely to insult someone if they talk about something they do not like. As mentioned in the research each character is given a set of relation scores which determine how friendly or intimate, they are with each character which can change with the dialog they have with each other. Other features can be added in the future such as the art feature like Rimworld where actions are recorded and can be referred again by the characters to create a simple art piece with a description on it showing what inspired the art piece.

#### Environment of the project

First part of the artifact to set up is the basic environment for the artifact this will be the UI as discussed in the wireframe to aid in interactivity for the scene. The next feature will be the resources for the settlement which can be used to establish the characters place in the world i.e. soil can be turned into farmland so that a character that is a farmer can farm or rocks for a miner to mine as well as basic houses for characters to live in. another feature is animals which have basic behaviours like roam, flee and attack but could also be tamed. Finally, a simple faction system where factions’ leaders can declare wars against the user for added narrative.

#### Character systems

First the characters will need basic systems implemented such as movement which can be either a basic move to position or an A\* implementation as well as a health system so that a character can be killed off or simply injured giving the environment the ability to affect the characters another feature is a simple job system so a farmer if they are not doing anything can tend to a farm building providing food.

Next feature to add will be the character behaviour tree so that the character can decide on how to interact with the game world. This will update what the character can do whenever an event is triggered so that if an event that changes the behaviour occurs then the character with the behaviour tree can act in the appropriate way for that character’s traits.

The character narrative layer will be set up with a system which picks up events in the game world and utilizes them to populate a memory system which then in turns affects the emotional layer. Each event will have to be pre-written for example an event could be “friend murdered” which will be triggered if a character is killed with high relation with other characters. This event will have a context (the person who died) a recipient (the character’s close friends) and a cause (the character’s killer) each of these events will have an effect on the emotional layer using the emotional layer’s lookup system to find the effect the event will have on the player.

The emotional layer will have a system for record the emotional status of the character with as discussed in the research a system to record the emotional values with a range from +1 to -1 representing the positive and negative version of that emotion respectively i.e. +1 being happy and -1 being sad. This will then be extended into the relations with a list of relation values to show how friendly or unfriendly a character is with another given character. To affect this layer the narrative layer will give data about certain events that happened to them which are then compared to their traits which has a list of effects on the character so for example is a character has a trait like aggressive then they will when an event that is tagged as being angry happens, the character will be more likely to start getting aggressive at whoever caused the event like if a given character (Dave) shouts at another character’s (John) dog and john had the aggressive trait then John may fight Dave. Other effects from personality traits can be added like edgy which can make a character more to say something that loses relation or emotionally sensitive which makes gains and losses of emotion and relation values have a increased modifier.

#### History System

The likes/ dislikes and the traits of the character need to be determined on initialisation so the plan would be to develop a system which would develop multiple characters at once initialising their history together. Each character in the predefined list would be put though a number of events with the reverse narrative scaffolding system to predefine their history. This would be a system that would pick from a number of events for different moments in their life for example a child could have a “rough upbringing” event that would give them various traits based on that moment some that could be negative but later on in the history generation there could be a follow up event that rectifies that previous event removing the features applied to them or replacing them with different features.

This history could be given supplementary text which then makes up a history log which can then show the history of the character overall showing how the character has developed.

#### UML

#### Sequence Diagrams

## Artifact – Implementation

### Project development

Its important here to note how the project has been implemented logging any changes from the previous design overall the project was completed with some changes from the original design as seen below with…

#### Basic artifact design

For the basic artifact design what will be covered is the basic components of the artifact such as UI or project set up. The project was set up in unity and had the UI developed in a similar manner to the original design with the specified features being added such as…

* A character info section (top left) which shows information about each character for their names, likes/dislikes and their traits. The character assigned to the character info box can be changed by simply clicking with a mouse to a different character which uses Unity’s 2D ray cast system to identify and select a new character.
* The emotional box (middle left) which records the selected characters emotional and relationship data. This shows a number of values for the character with the current values of the emotional behaviours being shown first for both happiness and anger. This emotional box UI then gets the relation ship list to display the selected characters current relation with each character.
* The dialog box (bottom left) simply shows a list of dialog provided by the character’s dialog system. This box shows which shows the list uses a scrollbar along with Unity’s mask system to hide the dialog text that is not required to be showing keeping the text confined to the dialog box while making it easier to read than to simply display all at once.
* The action bar (bottom) has been added from the initial design to allow for testing and various actions for the project. The initial design only called for the force dialog system and an edit character button which the force dialog button has been fully implement which simply forces a random character to change its behaviour to the “talk” state although the edit character was removed as the unity editor worked fine as a alternative to change each characters actions. Further additions were made from the design that were not thought of at the time to improve the testing of the project. This includes an “add character” button which initialises a new character and places them in the artifact similar to the starting 3 characters. Another addition is the start event button which can have force an event to be added to a given character for example the “murder” event where one character will move towards another character with the “kill” behaviour state and “kill” them which will then follow into the rest of the memory creation system and so on. This could be changed for any event and any member of the cast in the artifact.
* An addition UI element was also added for the history system which is seen in the “?” button in the character info box which then opens up a screen which takes the history script provided by the history system and simply show it as text in the centre of the screen (see fig 9) so the user is able to see the selected characters history.

A screenshot of a computer

Description automatically generated with medium confidence

Figure : finished UI for the artifact

Graphical user interface

Description automatically generated

Figure : image showing the history script in UI

#### Removed features

Some removed features in the artifact from the original design is the animal system which while would be simple to build as it only requires a basic behaviour tree for walking around random positions and being interacted with by the characters. It was a feature that is not required to show the emotional behaviour aspects of each character and therefore can be removed to limit the scope of the project.

Another feature removed is the art system which while would be an extension of the history system would simply unnecessary for showing the recursive narrative system so was removed.

#### Behaviour tree to state machine

During the development of the behaviour tree as wanted through the design there were a number of problems during its development which caused errors when the behaviour was updated as the design required a behaviour tree that was dynamic and allowed for new behaviours to be made available for evaluation when needed. The errors that resulted during its creation made the system too time intensive to implement so it was instead replaced with a state machine.

What are state machines

Why its not an issue i.e. behaviour trees and state machines are similar enough

#### Characters

#### Emotional AI

#### Narrative layer

#### Event scaffold system

### Testing

## Further improvements

## Conclusion

# Reference list – may be useful

use of htp systems to create actions the user takes and for other character to interrupt task of the player to help generate narrative

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1024747>

HTN with emergent ai creating stories

[ResearchGate](https://www.researchgate.net/publication/337325308_Evaluating_AI-Based_Games_through_Retellings/link/5dd283bd299bf1b74b4b86ac/download)

Shows how the project could be tested with player retelling to deal with complex and difficult to imperially answer questions about the project

<https://www.researchgate.net/publication/274290528_Scaffolding_and_Concept_Formation_in_Narrative_Therapy_A_Qualitative_Research_Report>

[Why Are We Like This?: Exploring Writing Mechanics for an AI-Augmented Storytelling Game (ucf.edu)](https://stars.library.ucf.edu/cgi/viewcontent.cgi?article=1160&context=elo2020)

[(PDF) A Personalized, narrative and interactive simulation based on a rules-engine system designed to confront informal caregivers with personalized virtual Alzheimer's patients and to train their communicative coping strategy skills (researchgate.net)](https://www.researchgate.net/publication/345241095_A_Personalized_narrative_and_interactive_simulation_based_on_a_rules-engine_system_designed_to_confront_informal_caregivers_with_personalized_virtual_Alzheimer's_patients_and_to_train_their_communicat)

Use of narrative AI beyond games but to help Alzheimer’s

<http://www.journalssystem.com/shagh/Reactive-games-as-an-example-of-extensive-use-of-evocative-narrative-elements-in,132785,0,2.html>

needs reviewing – discussion of how dwarf fortress uses narrative through events although this more through a narrative perspective rather than a programming method

[Subverting Historical Cause & Effect: Generation of Mythic Biographies in Caves of Qud (acm.org)](https://dl.acm.org/doi/pdf/10.1145/3102071.3110574)

Talks about the text generation system from caves of qud a dwarf fortress inspired game and how events can change the relationships between characters and the game world through generative methods

<https://ieeexplore.ieee.org/abstract/document/7439785>

story generator

<http://dspace.mit.edu/handle/1721.1/54502#files-area> really bloody long but might be good read

# Bibliography

Alec Radford, J. W. (2019, 2 14). *Better Language Models*. Retrieved from OpenAI: https://openai.com/blog/better-language-models/#sample3

Alvim, L. G., & Cruz, A. J. (2008, 6 6). *A Fuzzy State Machine applied to an emotion model for electronic game characters.* Retrieved from IEEE explore: https://ieeexplore.ieee.org/document/4630637

Anurag Sarkar, S. C. (2021, 10 8). *Procedural Content Generation using Behavior Trees*. Retrieved from arxiv: https://arxiv.org/pdf/2107.06638.pdf

Bay12Games. (2021, 12 1). *Dwarf Fortress*. Retrieved from Bay12Games: http://www.bay12games.com/dwarves/

Deutsch, M. (2016, 7 8). *Harry Potter: Written by Artificial Intelligence*. Retrieved from Medium: https://medium.com/deep-writing/harry-potter-written-by-artificial-intelligence-8a9431803da6

Emily Sheng, K.-W. C. (2019, 10 23). *The Woman Worked as a Babysitter: On Biases in Language Generation*. Retrieved from Arxiv: https://arxiv.org/pdf/1909.01326.pdf

Garbe, J. (2018, 2 18). *Simulation of History and Recursive Narrative Scaffolding*. Retrieved from Delve: http://project.jacobgarbe.com/simulation-of-history-and-recursive-narrative-scaffolding/

Hudson, L. (2016, 9 28). *Some Like It Bot*. Retrieved from FiveThirtyEight: https://fivethirtyeight.com/features/some-like-it-bot/

Humphreys, T. (2021, 12 1). *Chapter 12 Exploring HTN Planners*. Retrieved from gameaipro: https://www.gameaipro.com/GameAIPro/GameAIPro\_Chapter12\_Exploring\_HTN\_Planners\_through\_Example.pdf

IBM. (2020, 7 2). *Natural Language Processing (NLP)*. Retrieved from IBM: https://www.ibm.com/cloud/learn/natural-language-processing

Igor Ribeiro Lima, T. d. (2012, 1 1). *Adapting and Using Scrum in a Software .* Retrieved from fsma.edu.br: http://fsma.edu.br/si/edicao9/FSMA\_SI\_2012\_1\_Principal\_2\_en.pdf

Ilche Georgievski, M. A. (2014, 3 28). *An Overview of Hierarchical Task Network Planning*. Retrieved from Arxiv: https://arxiv.org/pdf/1403.7426.pdf

Jason Grinblat, C. B. (2017, 8 14). *Subverting historical cause & effect: generation of mythic biographies in Caves of Qud*. Retrieved from ACM: https://dl.acm.org/doi/pdf/10.1145/3102071.3110574

Kaushal Chari, M. A. (2017, 4 22). *Impact of incorrect and new requirements on waterfall software project outcomes.* Retrieved from Research Gate: https://www.researchgate.net/publication/316362583\_Impact\_of\_incorrect\_and\_new\_requirements\_on\_waterfall\_software\_project\_outcomes

Khalil, M. A., & Kotaiah, B. (2017, 8 1). *Implementation of agile methodology based on SCRUM tool*. Retrieved from IEEE Xplore: https://ieeexplore.ieee.org/abstract/document/8389872

Latitude. (2022, 1 1). *AI Dungeon*. Retrieved from AI Dungeon: https://play.aidungeon.io/

Li, L., & Campbell, J. (2010, 1 1). *Emotion Modeling and Interaction of NPCS in Virtual Simulation and Games.* Retrieved from International Journal of Virtual Reality: https://ijvr.eu/article/view/2784/8842

Ludeon Studios. (2018, 10 17). *Rimworld*. Retrieved from Steam: https://store.steampowered.com/app/294100/RimWorld/

M. Mahalakshmi, D. M. (2013, 6 6). *Traditional SDLC Vs Scrum Methodology – A Comparative* . Retrieved from citeseerx: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.413.2992&rep=rep1&type=pdf

Marc Cavazza, F. C. (2003, 1 1). *Interactive storytelling: from AI experiment to new media.* Retrieved from ResearchGate: https://www.researchgate.net/profile/Marc-Cavazza/publication/220851669\_Interactive\_storytelling\_from\_AI\_experiment\_to\_new\_media/links/580f3d0308aef2ef97afbf4b/Interactive-storytelling-from-AI-experiment-to-new-media.pdf

Marchesi, M., Mannaro, K., Uras, S., & Locci, M. F. (2007, 6 22). *Distributed Scrum in Research Project Management*. Retrieved from ResearchGate: https://www.researchgate.net/publication/221592443\_Distributed\_Scrum\_in\_Research\_Project\_Management

Max Kreminski, M. D.-F. (2020, 1 1). *Why Are We Like This?: Exploring Writing Mechanics for an AI-Augmented Storytelling Game.* Retrieved from ucsc.edu: https://stars.library.ucf.edu/cgi/viewcontent.cgi?article=1160&context=elo2020

Sheetal Sharma, D. S. (2012, 5 5). *Agile Processes and Methodologies: A* . Retrieved from yashada: Microsoft Word - IJCSE12-04-05-186.doc (yashada.org)

Simpson, C. (2014, 7 18). *Behavior trees for AI: How they work*. Retrieved from GameDeveloper: https://www.gamedeveloper.com/programming/behavior-trees-for-ai-how-they-work

Vincent, J. (2020, 7 30). *OPENAI’S LATEST BREAKTHROUGH IS ASTONISHINGLY POWERFUL, BUT STILL FIGHTING ITS FLAWS*. Retrieved from The Verge: https://www.theverge.com/21346343/gpt-3-explainer-openai-examples-errors-agi-potential