Aalborg University Copenhagen

Aalborg University Copenhagen

Frederikskaj 12,

DK-2450 Copenhagen SV

Semester Coordinator: Stefania Serafin

Secretary: Lisbeth Nykjær

Semester: MED10 - Spring 2020

Title: Embedded Narratives in Procedurally Generated Environments

Project Period: February 3rd – May 28th

Semester Theme: Master Thesis

Supervisor(s):
George Palamas
Henrik Schønau-Fog

Project group no.: N/A

Members:
Eoin Ivan Rafferty

Thomas Lund Nielsen

Abstract:
Procedural Content Generation (PCG) is a widely used tool in game development. It can enhance the creativity of developers and reduce many aspects of their workload. One popular use of PCG is the generation of virtual environments (VE) for games. On the surface, this is incompatible with another popular trend in video games: environmental storytelling (ES).

This thesis aims to reconcile these two fields and explores a method in which PCG can be used to create VEs with embedded narratives. This is done using a Space-Time Drama Manager (STDM) to dynamically place narrative elements in the VE.
The Narrative Experience Measurement tool (NEM) is proposed as a combination of methods to evaluate players' experiences with these narratives in terms of narrative engagement, internal coherency, and external consistency.

A prototype game was developed and evaluated with NEM in a between-groups test (n=69). Participants in the control condition all experienced the same environment, while those in the experimental condition each experienced a different environment. The aim of this evaluation was to determine if there was a difference in the NEM measures between conditions. No significant difference was found for any of the measures.

The lack of significant difference between conditions indicates that it is possible to convey an embedded narrative consistently to players, even when the environment which contains the narrative elements is procedurally generated, yielding a different experience for each participant. This thesis can therefore be used as a foundation for further research within the field combining PCG and Environmental Storytelling.

Embedded Narratives in Procedurally Generated Environments

Eoin Ivan Rafferty &
Thomas Lund Nielsen

Supervisors
George Palamas & Henrik Schønau-Fog

Master of Science in Medialogy

Department of Architecture, Design, and Media Technology
Aalborg University Copenhagen

Preface

This is a Medialogy Master Thesis written by students at Aalborg University Copenhagen.

Reading Guide

The structure of this thesis is shown in the table of contents below along with a complete list of figures and tables. In addition, abbreviations are used throughout the body of text. These are all explained in the table below. Important appendices will be included in the end of the thesis, and additional ones can be found online. These are referred to as “Electronic Appendices”.

Additional Content

Some content has been too excessive or impossible to include in this thesis. This includes the aforementioned Electronic Appendices, the complete developed prototype, the GitHub repository, and an audio-visual production providing an overview of the thesis. At this website all these parts are collected:

`eoinraff.github.io/ThesisRepo/`

List of Abbreviations

Abbreviation	**Meaning**
AAD	Author-Audience Distance
ES	Environmental Storytelling
fBm	Fractal/Fractional Brownian Motion
PCG	Procedural Content Generation
SA	Staged Area
STDM	Space-Time Drama Manager
VE	Virtual Environment

Contents

1 Introduction **1**
1.1 Motivation . . . 3
1.2 Methodology . . . 3

2 Analysis **4**
2.1 Procedural Content Generation . . . 4
2.1.1 Taxonomy of PCG . . . 4
2.1.2 Types of PCG Systems . . . 6
2.1.3 Approaches to PCG . . . 6
2.1.4 Delimitation of PCG . . . 9
2.1.5 Generating Virtual Environments . . . 9
2.1.6 Evaluating PCG Systems . . . 11
2.2 Narratology . . . 14
2.2.1 Defining Narratives . . . 15
2.2.2 Space and Time in Narratives . . . 16
2.2.3 Diegesis . . . 17
2.2.4 Plot and Causality . . . 17
2.2.5 Linearity and Suspense in Narratives . . . 17
2.2.6 Narratvity as Simulation . . . 18
2.2.7 Summary of Narratology . . . 19
2.3 Environmental Storytelling . . . 20
2.3.1 Embedded Narratives . . . 21
2.3.2 Interpretation of Embedded Narratives . . . 22
2.3.3 Staged Areas . . . 22
2.3.4 Delimitation on Environmental Storytelling . . . 23
2.4 Designing for Environmental Storytelling . . . 24
2.4.1 Space and Time of Narratives in Games . . . 24
2.4.2 Player Guidance in Games . . . 26
2.4.3 Delimitation of Design for Environmental Storytelling . . . 27
2.5 Narrative Experience Measurement . . . 27
2.5.1 Overview and Purpose of the NEM tool . . . 28
2.5.2 Narrative Engagement . . . 28
2.5.3 Narrative Coherency . . . 29
2.5.4 Delimitation of Narrative Engagement Measurement . . . 31
2.6 Analysis Conclusion . . . 32
2.6.1 Research Question . . . 33

3 Methods **35**
3.1 Sampling & Target Group . . . 35
3.2 Initial Evaluation of Topography . . . 36
3.3 Formative and Summative Evaluations . . . 36
3.4 Outcome Evaluation of Narrative Experience . . . 37
3.4.1 Experimental Setup . . . 37
3.4.2 Data Analysis . . . 39

3.5 Design Requirements . . . 41

4 Design **43**
4.1 Narrative Design . . . 43
4.1.1 Designing for Embedded Narratives . . . 43
4.1.2 Backstory . . . 44
4.1.3 Specific Narrative . . . 44
4.1.4 Key Narrative Information . . . 45
4.2 Staged Areas . . . 46
4.2.1 Design of Staged Areas . . . 46
4.3 Player Guidance . . . 48
4.4 Audio . . . 48
4.5 Design of Procedural System . . . 49
4.5.1 Terrain . . . 49
4.5.2 Evaluation of Terrains . . . 50
4.5.3 Textures . . . 52
4.5.4 Foliage . . . 53
4.6 Design Conclusion . . . 54

5 Implementation **56**
5.1 Terrain Generation . . . 56
5.1.1 Basic Topography . . . 56
5.1.2 Fulfilling Designer Constraints . . . 57
5.2 Space-Time Drama Manager . . . 63
5.3 Linking Systems with an Observer Pattern . . . 64

6 Evaluation **68**
6.1 Test Description . . . 68
6.2 Independent Variable . . . 68
6.3 Dependent Variables . . . 69
6.4 Test Procedure . . . 70

7 Results **71**
7.1 Narrative Engagement . . . 71
7.2 Narrative Tension . . . 73
7.3 Narrative Coherency . . . 74
7.4 PCG . . . 74
7.5 Player Types . . . 75
7.6 Findings . . . 78

8 Discussion **81**
8.1 Interpreting Results . . . 81
8.2 Setup, Procedure, and Test Specifics . . . 83
8.2.1 Technical Issues . . . 84
8.2.2 Validity . . . 85
8.2.3 Reliability . . . 86

9 Conclusion **87**

10 Future Works **88**

References **90**

Appendices **98**
A Questionnaire Items for Narrative Experience Evaluation 98
B Formative Evaluation of System Performance 101
C Initial Prototyping Overview 102
D Visualising the Expressive Range of the Terrain Generator 111
E Backstory for Narrative 114
F Script for Narrative 116

List of Figures

1 Images from Games Using PCG 2
2 Illustration of an L-System 7
3 Heightmap Examples 10
4 2D Noise 11
5 Perlin Noise with Octaves 12
6 Diamond-Square 13
7 Example of Expressive Range Visualisation 14
8 Embedded Narratives in Portal 21
9 Author-Audience Distance 23
10 Example of Paths in Uncharted 26
11 Examples of Wienies in Uncharted 27
12 Player Types 39
13 Symbols for Narrative 45
14 Overview of Staged Areas 46
15 Staged Areas for Prototype 47
16 Wienies Used in Prototype 49
17 Images of Three Different Terrain Angles 52
18 Examples of Assets Used in Prototype 54
19 Final Design of Prototype 55
20 Voronoi Diagram Example 57
21 Different Falloff Curves 58
22 Terrains Generated With Different Techniques 58
23 Combination of Terrain Techniques 59
24 River Erosion 60
25 Landslide Erosion Graph 61
26 Landslide Erosion Example 61
27 Tidal Erosion 61
28 Example of Texture Splatting 62
29 Screenshot of Final Prototype 66

30 Distribution and Box Plot of Narrative Understanding 71
31 Distribution and Box Plot of Attentional Focus 72
32 Distribution and Box Plot of Presence 73
33 Distribution and Box Plot of Narrative Engagement 74
34 Box Plot of Players"Desire to Play the Game Again 74
35 Distribution and Box Plot of Narrative Tension 75
36 Distribution and Box Plot of Narrative Coherency 76
37 Distribution and Box Plot of Usability 76
38 Distribution and Box Plot of Suitability 77
39 Distribution and Box Plot of Navigability 77
40 Distribution and Box Plot of Interest in the Environment 77
41 Distribution and Box Plot of Narrative Engagement Based on Player Types 78
42 GAN Heightmap Output 102
43 Seams Between Terrain Chunks 103
44 LOD Meshes 105
45 LOD Mesh Borders 105
46 High Detail Borders 106
47 Fast-Poisson Disk Sampling Trees 109
48 Illustration of the Box-Counting Method 112
49 Segmented Heightmap Clusters 112
50 Fractal Dimension Graphs 113
51 t-SNE Plots of Fractal Dimension 113

List of Tables

1 Results From Evaluation of PCG Terrains 51
2 Codes From Terrain Evaluation 52
3 Fixed-Space Constraints 64
4 Ranking of Answers for Narrative Coherency 69
5 Descriptive Statistics of Narrative Understanding 71
6 Descriptive Statistics of Attentional Focus 72
7 Descriptive Statistics of Presence 72
8 Descriptive Statistics of Narrative Engagement 73
9 Descriptive Statistics of "Play Again" Data 73
10 Descriptive Statistics of Narrative Tension 75
11 Descriptive Statistics of Narrative Coherency 76
12 Descriptive Statistics of Environment Usability 78
13 Descriptive Statistics of Environment Suitability 78
14 Descriptive Statistics of Environment Navigability 79
15 Descriptive Statistics of Interest in the Environment 79
16 Descriptive Statistics based on Player Type 80

1 Introduction

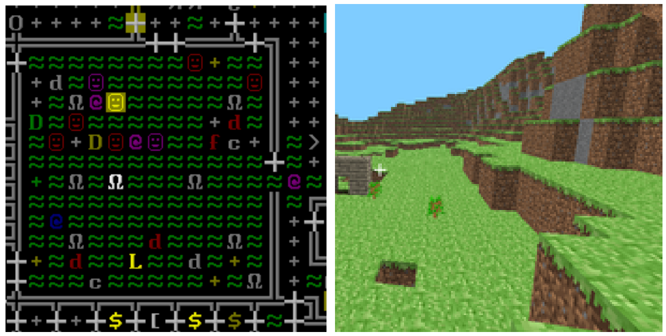
Procedural content generation (PCG) is becoming a bigger and bigger part of the Games industry. It can be used by both large and small development teams, and has many reasons to be used. One powerful use for PCG is to augment the workflow and creativity of artists and designers. Many games use PCG for creating and distributing vegetation, which is time-consuming manual labour if done by hand. In the case of *Marvel's Spiderman* (Insomniac Games, 2018), the developers went beyond looking just at vegetation, and relied heavily on a PCG pipeline for creating many aspects of their virtual Manhattan. PCG was used in conjunction with designers to create the layout of the city's streets, placement of props and events, assignment of materials and textures, audio systems (Santiago, 2019), and placement of lights and light probes (Halperin, 2019).

In some cases, rather than aiding and augmenting the workflow of artists and designers, PCG can remove the need for them in the development pipeline entirely. This could make the production of games "faster and cheaper" (Shaker et al., 2016, Chapter 1). Using PCG to create game assets and levels has allowed small development teams to produce successful, ambitious and content-rich games. For example, game developer Notch used PCG to develop *Minecraft* (Mojang, 2011) as a single-man team. Bay 12 Games, a two man development team consisting of brothers Tarn and Zach Adams, heavily used PCG in their ambitious ASCII-art world simulator, *Dwarf Fortress* (Bay 12 Games, 2006) (see Figure 1).

However, creating game environments alone is often not enough, those environments need to contain engaging content. *No Man's Sky* (Hello Games, 2016), for example, is a game with a vast procedurally generated universe, but was criticised on release for a lack of interesting content (though much of the criticism in fact was about unfulfilled expectations (MacDonald, 2018)). Furthermore, it can be argued that hand-crafted content from artists and designers will almost always be superior to procedurally-crafted content, as the quasi-randomness of PCG removes some control from the designers that they would normally be able to use to finely tune and tailor the game to their desired experience.

In another way, PCG has the potential to create powerful, engaging worlds for players. While hand-authored content can in many ways be superior to computer-generated content, it is relatively static. Generally it is a finely crafted experience which is repeatable, and will be experienced the same way over and over. PCG offers the chance to increase variability and replayabilty to a game. This can be seen in the procedurally generated world maps of Civilisation VI (Firaxis Games, 2016), and the procedural dungeons in games such as Spelunky (Mossmouth, 2008) and Diablo III (Blizzard Entertainment, 2012).

Going beyond replayability, algorithmically generating content opens up exciting possibilities in creating adaptive games, where the new content which is generated can be tailored to the player. This could be based on basic telemetry or in-game metrics, or even more complicated models and measures of player enjoyment or engagement (Shaker et al., 2016, Chapter 10).



(a) *(b)*

Figure 1: *Both (a) Dwarf Fortress (Bay 12 Games, 2006) and (b) Minecraft (Mojang, 2011) use PCG to generate the game's environment*

One aspect which has been overlooked in this discussion of PCG for games is that of a game's narrative. Most games have some form of narrative, whether it is a simple short background, or an integral part of the game experience. Even in non-story driven games, for example simulation games like The Sims (Maxis, 2014), a narrative can emerge. That is to say, a game's narrative does not need to be explicitly told or even planned and written (Aylett and Louchart, 2007). Whereas the conflict over authorship of content has been between the designer and the algorithm in the discussion thus far, when considering a narrative, the player is introduced as another creative force with a claim to the authorship. This is due to the fact that players create their own meaning and narrative from the information provided to them by the game. This information can be provided explicitly or implicitly to the player.

Explicit methods, such as text and dialogue, are often used to tell a story in a game. These methods are generally unambiguous and lead to the player interpreting the story as per the author's intention. Game stories told this way are often supported by the environment, which is generally hand-crafted to support that narrative. Many story-focused AAA games, such as the *Uncharted* series (Naughty Dog, 2007-2017) use this approach. Even *Marvel's Spider-Man* (Insomniac Games, 2018), which as mentioned above uses many procedural tools, relies on hand-crafted environments for story focused elements of the game (Santiago, 2019).

On the other end of the scale there are more implicit methods of storytelling, such as environmental storytelling (ES) (Jenkins, 2004). Historically, storytelling has mostly been viewed in terms of time i.e. the sequence of events in the story (Ricoeur, 1980). However in games, particularly those which use ES, space plays an arguably more important role in the storytelling (Jenkins, 2004). Due to the interactive nature of games, and often the ability for players to explore the environment they are in, this environment can cease to be a mere container for the narrative, but instead can directly enhance it, or even play the role of a main actor in the storytelling process (Baynham, 2003; Fina, 2003; Zakowski, 2016).

The use of PCG environments seems to clash on a fairly fundamental level with the use of ES. Jenkins (2004) argues that space plays a major role in storytelling in games, and using PCG to create a game environment takes away a lot of control over that space which authors and developers generally desire. It is perhaps therefore that not many narrative-focused games rely heavily on PCG (Portnow and Floyd, 2015).

1.1 Motivation

Game developers and studios of all sizes have begun to use PCG as a key tool in their workflow. For small and indie developers in particular, PCG allows them to create huge amounts of content with limited resources. At the same time players are coming to expect an engaging narrative from their games. Combining PCG with ES may provide a means for game productions with limited resources to created vast worlds, while keeping them content-rich, and still being able to convey a meaningful, engaging narrative to their audience. Furthermore, this combination could increase the replayability of narrative-focused games, allowing players to experience the narrative in a new way every time they play.

However, not much research has been conducted addressing how an ES based narrative is affected by changing the environment in general, let alone by removing the control over the design of the environment from the game developers. Narratives are becoming such an important part of games and exploring this relationship is called for as PCG can prove to be one of the most important tools for game developers in years to come. With the parallel rise in narrative driven games, and their use of ES, this will eventually become a problem many game developers will need to overcome.

To stay ahead of the curve, this thesis will explore what effect PCG environments have on ES, and search for a solution which allows the power of PCG to be harnessed while still conveying a narrative.

1.2 Methodology

To explore this we will approach this with realism as the philosophical stance to explore this field of research further. It will be a deductive approach utilising an experimental strategy. As so, it will use exploratory sequential mixed methods where the first part will be a qualitative analysis of existing research to arrive at a research question (see Section 2.6.1) which in turn will be answered using a mix of qualitative and quantitative methods in a cross-sectional study of the relation between PCG and narratives using ES (Saunders et al., 2007). In addition, for this study a prototype will be developed which has the required functionality and will be an integral part of the study. In Section 3 the specific techniques and methods are explained in further detail before Sections 4 and 5 will specify how the prototype is developed to combine PCG and ES. This will then be used to answer and discuss the research question in Sections 6 through 9.

2 Analysis

To further investigate the field between procedural content generation (PCG) and environmental storytelling (ES) this section will research both fields separately and propose a methodology for evaluating artefacts that incorporate these two techniques in conjunction. Firstly, this section will present some background and theory about PCG and examples of applications of this technology. Secondly, it will present important considerations when designing a narrative experience and especially for ES. As the prototype in the end is a game, there will be important lessons from game design presented before lastly, the needed theory about evaluation methods of narrative experiences will be discussed. This will ultimately lead into the creation of our own methodology that will be discussed before being utilised to evaluate a developed prototype that should be able to answer our research question.

2.1 Procedural Content Generation

As indicated in Section 1, there are many different applications for PCG. With so many applications, there are naturally many different interpretations and definitions. Togelius et al. (2011) describe the concept of PCG as having "fuzzy and unclear boundaries". In order to try and clear these boundaries, it is firstly important to clarify what constitutes *content* when talking about PCG:

> *Game content* refers to all aspects of a game that affect gameplay but are not nonplayer character behavior or the game engine itself. This definition includes such aspects as terrain, maps, levels, stories, dialogue, quests, characters, rulesets, camera profiles, dynamics, music, and weapons.
>
> (Togelius et al., 2010)

While they admit that there is no universal and precise definition, Togelius et al. (2011) clarify the concept and tenuously define PCG as "as the algorithmical creation of game content with limited or indirect user input".

This section will begin with a broad, high-level overview of the field of PCG. Section 2.1.1 will explain some of the specific taxonomy of this field. Section 2.1.2 describes in broad strokes some of the different types of PCG systems, and what their core functionality is. Section 2.1.3 will maintain this high-level overview, but start to look at different generative approaches that can be applied by various systems. After this, the focus will narrow towards PCG for virtual environments, as Section 2.1.5 will examine how landscapes are represented and generated for virtual environments.

2.1.1 Taxonomy of PCG

Online v Offline Online PCG systems generate content while the game is being played. Offline PCG systems are generally used during the development process. Online PCG facilitates player-adaptive systems, while offline systems are useful for generating complex content such as worlds and environments.

Necessary v Optional PCG can be used to develop *necessary* content, such as game worlds and levels, like in *Minecraft* (Mojang, 2011) and *No Man's Sky* (Hello Games, 2016). Other games use PCG for *optional* content, such as weapon upgrades in the Borderlands series (Gearbox Software, 2009-2019; Hellquist, N.D.). Necessary content requires a high level of "correctness" as it is a key element of the experience. Optional content generally has more relaxed requirements as it can be replaced or discarded by the user.

Degree and Dimensions of Control PCG systems can be controlled in various ways. A *random seed* is often used to ensure that the same environment can be generated multiple times. This is useful for example in multiplayer games such as *Minecraft* (Mojang, 2011) and *No Man's Sky* (Hello Games, 2016), where multiple players need to generate the same environment on different machines. A feature vector can also be used to ensure the generated content satisfies a number of specifications.

Generic v Adaptive Generic PCG does not take player behaviour into account, while Adaptive tailors content generation to the player. Generic is the most common paradigm, though there are some notable examples of adaptive PCG, such as the "AI Director" in *Left 4 Dead* (Valve, 2008), which controls pacing and intensity based on player behaviour. While it is not very common in the AAA games industry, adaptive PCG has seen some activity as a research field (Yannakakis and Togelius, 2011).

Stochastic vs Deterministic Deterministic PCG allows for the regeneration of content, such as the examples using random seeds mentioned above. Stochastic PCG generally does not allow for content regeneration, as one input will not result in the same output consistently.

Constructive vs Generate-and-test Constructive PCG generates all of its content in a single pass. Generate-and-test, as the name suggests, evaluates content and reiterates the generation process until the generated content is satisfactory.

Automatic Generation v Mixed Authorship Many games use automatic generation, allowing the PCG system main authorship over the content. More recently Mixed Authorship, where the player or designer works in collaboration with the system to generate content, has become both more common and more viable. *Marvel's Spider-Man* (Insomniac Games, 2018) used mixed authorship extensively during its development (Santiago, 2019).

2.1.2 Types of PCG Systems

As mentioned multiple times so far, there are many different applications of PCG in games. The differences in approaches and use cases for different PCG systems is substantial, and it is important to understand the various types of systems. Khaled et al. (2013) identify four types of systems, *tools*, *materials*, *designers*, and *experts*. These design metaphors are briefly expanded upon below.

Tools PCG is used, perhaps most often, as a tool to aid designers and developers by augmenting their workflow, or alleviating their workload. Most of the systems that Santiago (2019) describes from the development of *Marvel's Spider-Man* are PCG tools used at various stages of the development process.

Materials PCG can be used to generatively create materials which designers can then in turn use to create content. The most ubiquitous example of this is *SpeedTree* (Interactive Data Visualisation, 2002-2020), a commercial software which uses PCG to create similar but distinct trees to populate environments. This tool has been licensed by many AAA and indie games alike. While this could be seen as a tool to populate landscapes, Shaker et al. (2016, Chapter 1) explain that it functions as a generative material, since "the designer can click and drag [the trees] around, add and remove branches, etc., and they always look like a tree, because the trees are procedurally generated in real time as the designer manipulates them".

Designer PCG designers aim to generate content without the aid of human designers. This is often used for some aspects of the game, though an extreme case is "automatic game design", in which the system would generate a complete game. This would not only consist of content for the game, but the rules themselves. For example, *Yavalath*(Browne, 2014) is a board game which was designed algorithmically (Shaker et al., 2016; Smith et al., 2015). Yavalath itself does not use PCG materials, but the system which created it was a *Designer* PCG system.

Domain Expert Some PCG systems are experts which can critique and modify generated content based on their knowledge of a certain area. For example in an educational game, the PCG system might be an expert in the subject area and modify content based on pedagogical goals. Adaptive games may use a system which is an expert in player behaviour and modify content to respond based on that.

2.1.3 Approaches to PCG

Now that there has been a broad, high-level overview of types of PCG systems, this section will look at different methods that various PCG systems employ in order to generate content. Some of these methods may be applicable to multiple types of systems, though since this thesis is focused on procedural generation of game environments, methods useful for procedural tools and materials will be the focus of this section.

Constructive Methods Constructive methods generate content from a series of modular building blocks. Compton (2017) refers to "Tiles" when talking about constructive methods. Tiles are highlighted as being useful when generating something that can be broken into distinct regions, where tile-to-tile placement doesn't need to be constrained. The procedural placement of these tiles can still produce emergence. Spelunky (Mossmouth, 2008), for example, uses this approach to PCG for its level generation, as smaller segments are combined to generate the larger level.

These methods tend to be algortihmically lightweight, resulting in fast generation, and are therefore well suited to online generation. However these systems tend to not be very extensible, and put a lot of responsibility on designers, who need to create these individual blocks, and ensure they fit together. Furthermore, players will often begin to recognise patterns and recurring sections if they encounter the same tiles with enough frequency (Smith et al., 2015).

Grammars A grammar is a set of *production rules* for changing one string into another. The production rules are applied to the *axiom*, (i.e. the initial state) for a number of iterations or generations. *L-Systems* are a type of grammar often used in PCG. They are particularly well suited to generating plants and vegetation and other natural systems (Hørdum Jæger, 2018). For example the rule set $A \rightarrow AB, B \rightarrow A$ is a famous L-system designed to model yeast growth. It grows from the axiom A over 7 generations to $ABAABABAABAABABAABABA$. L-Systems can also be applied to sketches, or even 3D models. Figure 2 shows the first five generations of an L-System designed to draw a 2D plant.

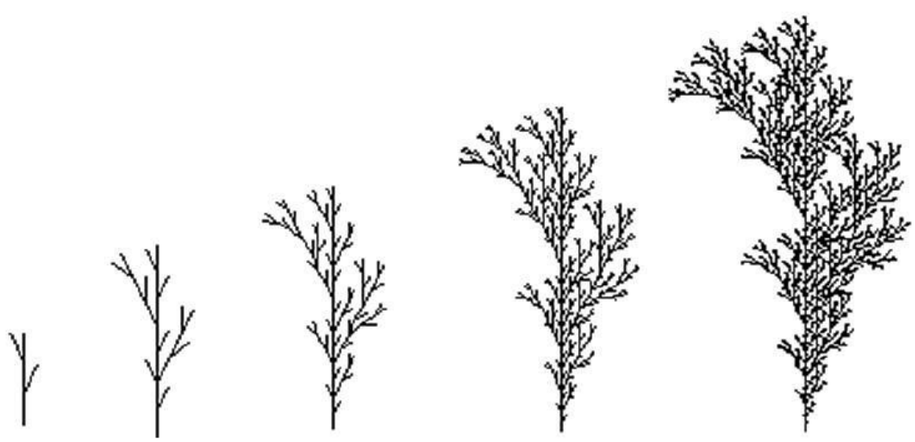


Figure 2: *Using an L-System to draw plants (Bornhofen et al., 2012)*

Although often associated with vegetation, and used by popular tools such as SpeedTree (Interactive Data Visualisation, 2002-2020), L-Systems can be used for many other types of systems. Tracery, for example, uses a text-replacement L-System to create stories (Compton, 2015). L-Systems can also be used for generating levels, or even missions within levels (Shaker et al., 2016, Chapter 5).

The strength of these systems is that with a simple starting point and a small rule set, complex, organic seeming patterns can arise. However these systems can also be quite repetitive. Furthermore it can be hard to tweak the system to achieve your desired output, as minor changes to the input or rules can have large changes on the result (Smith et al., 2015). Furthermore, grammars can be used to encode knowledge about an artifact and its structure in a piece of data, though it is difficult to handle constraints and high-level relationships in the generated content (Compton, 2016).

Constraint-based Constraint-based PCG is a top-down approach to generation, where the designers and programmers specify a number of high-level end goals for the generator's output. Unlike grammars and constructive methods, these systems can make promises about the validity of the content which is produced, as the generated content will be sure to meet the requirements specified by the designer. However these systems can be difficult to produce, as they need to express these design requirements in logic programming (Smith et al., 2015). This extends to even seemingly "common sense" goals (e.g. a room should have 4 walls). Since these systems can often grow into a complex web of logic, they can be difficult to debug.

Optimisation-based (Search-based) Optimisation systems generally use evolutionary algorithms to produce content. In simple terms, this procedure involves generating content, assessing it via a *fitness function*, and discarding content which performs poorly, while keeping and mutating content which performs well. The strength of these systems relies on the quality of the fitness function, which determines which content is passable. Yavalath (Browne, 2014) was created using this type of method. These systems can be quite general and powerful, and can produce unexpected results. However it can be difficult to find a suitable fitness function, which is the cornerstone of the generation. Furthermore, these systems tend to be slow, meaning they are more suited to offline than online generation (Smith et al., 2015; Togelius et al., 2010).

Parametric Methods Compton (2017) refers to "parametric" systems, which in many ways are similar to Smith et al.'s constraint-based systems in that they are generally top-down systems which allow the designer a large degree of control over the content. Parametric systems are particularly useful for interactive generative content (Compton, 2016). For example the creature creator in *Spore* (Maxis, 2008) uses a parametric generator which generates and applies textures to "metaballs", which are geometric objects whose shape is based on a skeleton, which is in turn generated by user-controlled parameters. Similarly, *Civilisation*

(Firaxis Games, 2016) allows players to adjust parameters such as size, number of islands, density of resources, etc. for generating the game maps. Even without user-facing interactivity, these parameters can be controlled by the generator. *No Man's Sky* (Hello Games, 2016), for example, uses a parametric generator for its creature creation system (Morin, 2016).

2.1.4 Delimitation of PCG

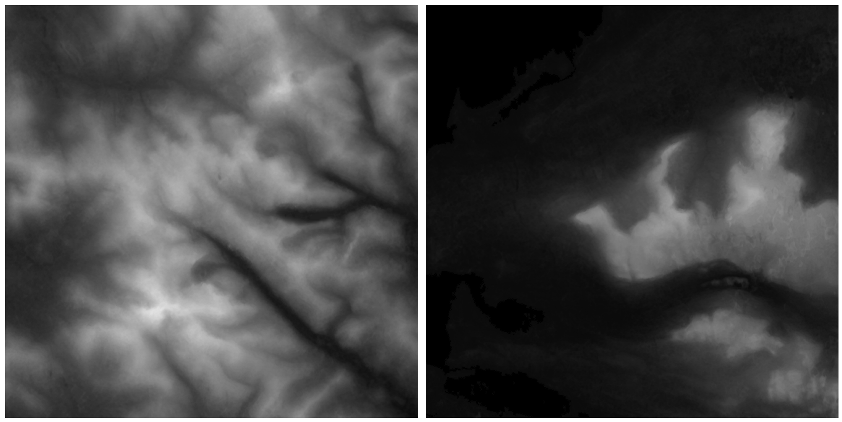
The previous sections have just scratched the surface of the vast topic of PCG, but have provided both the context and vocabulary to delimit the field to what is required for this thesis. As mentioned in Section 1, the primary purpose of PCG in this thesis is generating environments for narrative games. Considering the taxonomy presented in Section 2.1.1, some requirements and descriptors of a PCG system can already be laid out.

- The terrain that will be generated is **necessary** content, and thus it is important that it is valid for the design of the game.
- The system should allow a high level of designer control. As such the focus will be on **parametric** and **constraint-based** approaches.
- The generator should produce different terrain for each user. As such it needs to be an **online** system.
- The system could use either a **constructive** approach to reduce load times, or **generate-and-test** to ensure higher quality.
- The system should be **deterministic**, and there should at least be a **single dimension of control** in a random seed value in order to allow reproduction of environments in a testing scenario.
- The terrain system can be **generic** and use **automatic generation**, as the player's behaviour will not influence the terrain, at least before introducing narrative elements.

2.1.5 Generating Virtual Environments

In 3D games, terrain or landscapes are generally represented in one of two ways. It can be represented as a 2D greyscale texture known as a heightmap, or as voxels. Just as a pixel is a "picture-element", a voxel is a "volume-element". Voxels are most often represented as cubes, perhaps most famously in *Minecraft* (Mojang, 2011). As a 3D representation, voxels have some advantages over heightmaps, such as the ability to represent more complex 3D topography such as caves and overhangs. However voxels cost much more in terms of memory and storage, and are often more complex than required. For the purposes of terrain generation, this thesis will therefore focus on the use of heightmaps, as opposed to voxels.

As seen in Figure 3, real-world locations can be represented as greyscale heightmap textures, where black pixels represent low areas and white pixels



(a) *(b)*

Figure 3: *Real-world topography represented as a 2D heightmap texture. These texture represent (a) a dense, mountainous region with deep valleys, and (b) a region with smaller, isolated mountains with steep edges and flat tops*

represent higher altitudes. Similar textures can be created with noise algorithms for procedural generation of heightmap textures, and thereby generation of terrain. In order to make these maps look and feel natural, gradient noise such as Perlin or Simplex noise is often used. These gradient noises produce much more organic changes in values, as the intensity of each pixel is related to that of surrounding pixels. These noise based methods can be considered parametric, as the noise can be controlled with parameters such as amplitude, frequency, and scale.

While Perlin noise creates more organic patterns than uniform random noise, there is still a large gap between the patterns in Perlin noise (see Figure 4b) and real-world terrain (as seen in Figure 3). One way to improve upon this basic Perlin noise is by using Fractal (also called Fractional) Brownian Motion (fBm).

There are many ways to implement fBm, and using Perlin noise is a popular approach. Perlin noise based fBm involves summing noise values of decreasing amplitude and increasing frequency together. As illustrated in Figure 5, this results in a noise map with much more variety. Each layer of noise is known as an *octave*, while the rate of change of the amplitude between octaves is known as *persistence* and the change in frequency is known as *lacunarity.*

Diamond-Square Another way to generate noise maps using fBm is the *Diamond-Square* algorithm. This is a simple and fast algorithm which can quickly fill space and generate fractal noise patterns in a 2D grid (Shaker et al., 2016, Chatper 4). Firstly, the corner values of the grid are initialised. Then

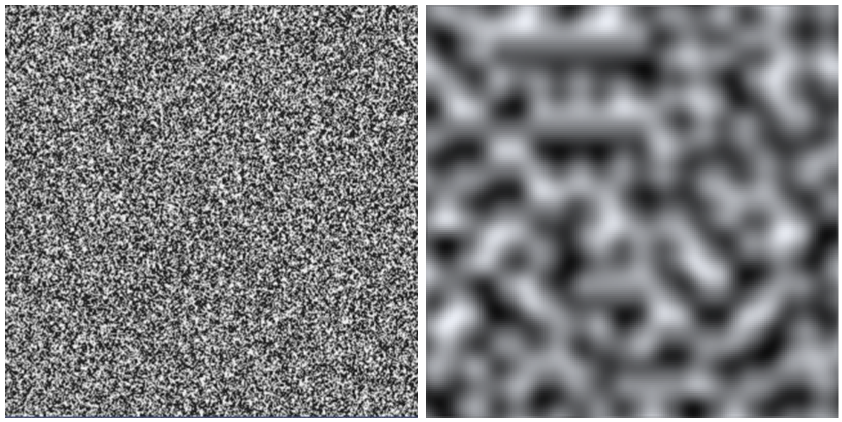


Figure 4: *White noise (a) produces uniformly random maps, while Perlin Noise (b) produces smooth transitions between high and low values*

items in the grid are assigned values by alternating the diamond and square steps. These two steps are repeated until all items have values assigned to them. This process is illustrated in Figure 6. The rules for the diamond and square steps are as follows:

The diamond step: For each square in the array, set the midpoint of that square to be the average of the four corner points plus a random value.

The square step: For each diamond in the array, set the midpoint of that diamond to be the average of the four corner points plus a random value.

The magnitude of the random value which is added is known as the *roughness*, since large values will produce rough surfaces and small values will produce smooth surfaces. This value is reduced slightly after each pair of diamond-square steps.

2.1.6 Evaluating PCG Systems

One issue that faces developers using PCG systems is that it can be difficult to evaluate the system in terms of the content it produces. Developers may be certain from their implementation for example that they can generate vast amounts of mathematically unique content, but it may not be perceptually unique to the end-user (Compton, 2017). Or perhaps not all the levels which are generated are able to be completed. Some of these steps can be overcome with certain generative approaches (e.g. constraint-based systems may ensure that a level can be played from start to finish), however it is still important to understand the abilities and performance of the generator. It can be tempting

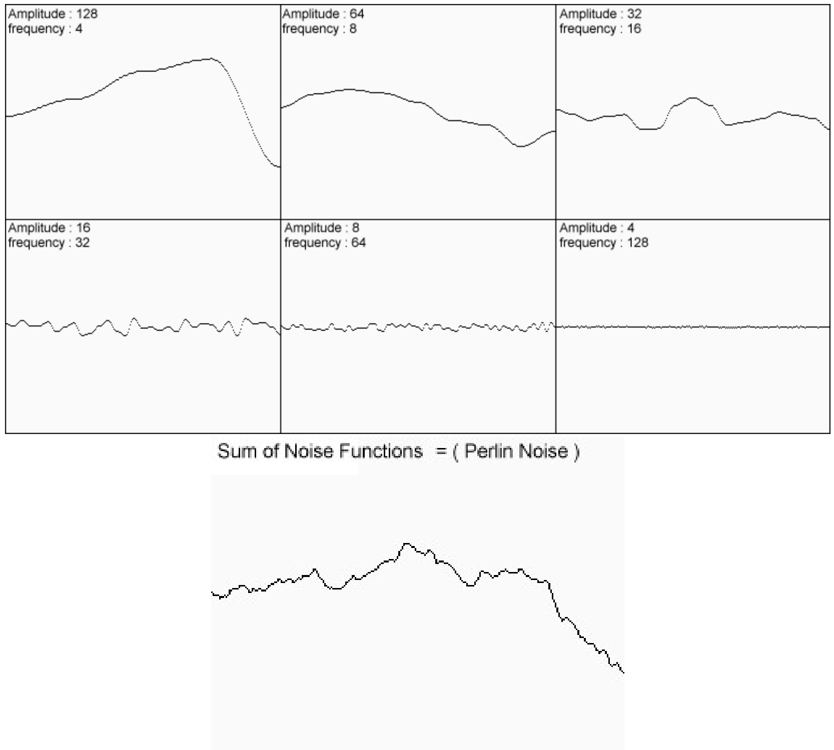


Figure 5: *Perlin Noise can be made even more natural seeming by summing octaves with different amplitudes and frequencies to create minor variations in the noise pattern (Biagioli, 2014)*

to take a few samples of content and use those to assess the generator, but in reality it is difficult to get a representative sample of output (Smith et al., 2015).

One method of evaluating content is to simply make a whitelist of random seeds or other control parameters (Compton, 2017). The whitelist could be selected with *generate-and-test*, as mentioned above in Section 2.1.1. In order to make sure an approach is feasible, the testing should be done algorithmically with a discrimination function to determine what content is passable. This has similar strengths and weaknesses to the optimisation-based generation described in Section 2.1.3, though is applicable to other generative approaches. The quality of this evaluation hinges on the algorithm. If the constraints for content to be approved by the algorithm is too tight, it could result in never accepting anything (Compton, 2017).

Another method of evaluating a generator is to visually map its “expressive range”. This gives a visual representation of the space in which the generator is working. It can help get an understanding of the type of content that is being

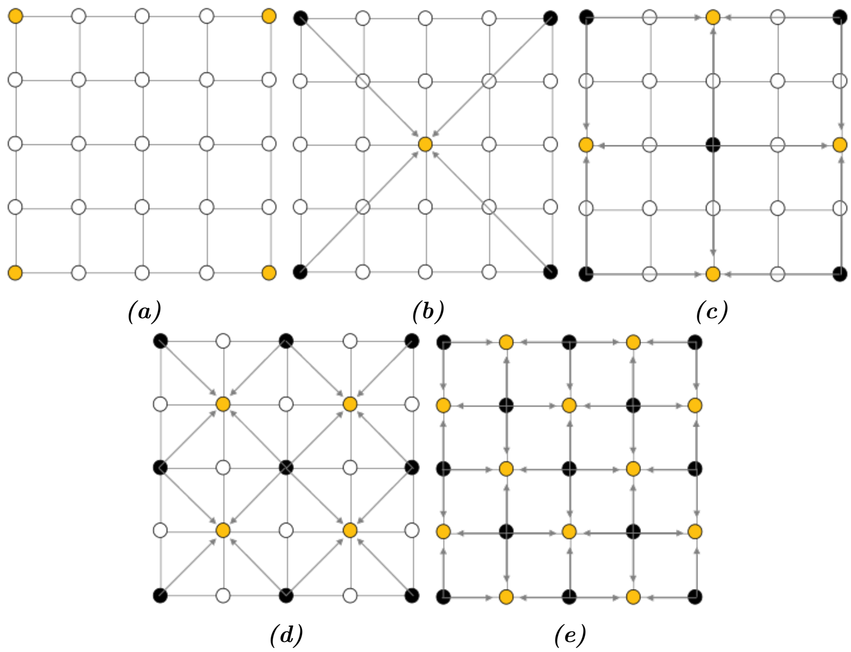


Figure 6: *Illustration of the Diamond-Square algorithm (Erwin, 2015). In each step, values which are being assigned are shown with a yellow dot, and values which have been assigned are marked with a black dot. White dots are unassigned values that are not involved in that step. Arrows indicate which values are influencing each other. The corner values are first initialised (a), before performing the diamond step (b) and square step (c). These steps are then repeated (d, (e) until all values have been assigned*

produced, and help identify any biases in the system.

In order to evaluate a generator in this manner, it is important to have some numerical metrics to describe the content. These metrics should represent something which is *emergent* from the generator, as opposed to a parameter which can be directly controlled (Shaker et al., 2016, Chapter 12). For example, Smith and Whitehead (2010) use this method for evaluating a level generator for *Launchpad*, a 2D platformer game. They use the metrics of "linearity" and "leniency", which are not directly controlled by designers.

Once suitable metrics have been chosen, the system should generate a huge amount of content, which should then be scored based on these metrics. The frequency of these scores is then mapped on a histogram or a heat map, which will then show the expressive range.

This representation gives developers and designers an easy way to understand their system's output, and understand how changing parameters affects the end result. For example a heat map for Smith and Whitehead's Launchpad level generator might identify a certain set of parameters is biased towards very

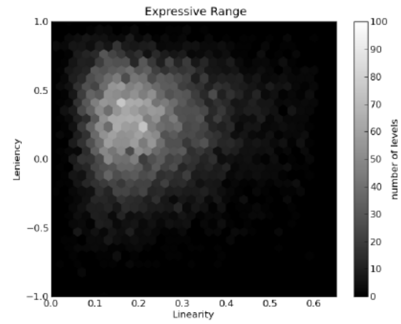


Figure 7: *A visualisation of the expressive range of a level generator for a 2D platformer (Smith and Whitehead, 2010)*

linear levels with high leniency. Designers can then quickly tweak parameters such as distance between platforms, and run the evaluation again to see how the change in parameters affects the space in which the system generates content.

Before continuing further with the design of a prototype PCG system, the next section of the analysis will first look into theories in the field of narratology, so that a prototype design may better incorporate aspects from both areas.

2.2 Narratology

Creating narratives is a complicated process even though storytelling is a big part of our lives. Knowledge has been passed on through stories for thousands of years e.g. when looking at cave paintings dating more than 40,000 years back (Aubert et al., 2019). Within the last 100 years both the Russian formalists (Propp, Tomashevsky etc.) and the French linguists (Benveniste, Barthes, and Todorov) started to theorise about narratives and begun to put elements of narratology into order to make for more structured analysis (Toolan, 2006). To create a narrative one needs to comprehend the many details and quirks that comprises a complete and compelling modern narrative. Those cave paintings were, for all we know, not meant to entertain and keep the viewer invested, but today the amount and supply of narratives are plentiful and therefore knowing what separates narratives from each other is paramount in creating one that has the ability to capture the attention of the receiver.

For this thesis, it is important to not only create a narrative but create one that has a certain standard, to really investigate the effect the environment has on that narrative. Therefore this section will delve into the theories created by the formalist, linguists, and others. This will then be used as a base to continue onward, and examine how narratives can be constructed in games.

Before continuing there are a few taxonomy details that need to be outlined. There are different ways the “receiver” of a narrative is defined. It can be a *reader* of a book, a *viewer* of a film or show, or a *player* of a game or other interactive experience. To keep the following discussion of narratives as simple as possible and to keep a consistency throughout this thesis, one term will be used for all of these terms. Therefore, *receiver* will be used as a catch-all term, as the same theory can be applied across mediums.

2.2.1 Defining Narratives

A narrative can be a difficult entity to define. There are many different definitions all dependent on who you ask or the context. Ryan (2007) offers one definition comprising of a set of rules that must be fulfilled in order to be considered a narrative. Others might argue that any definition should be less restricting and offer more space for interpretation (Genette, 1982). Within interactive narratives the receiver might be put in charge of creating the narrative themselves within the context of the provided world. In such cases, the narrative they create might not check many of the boxes set up by Ryan (2007) but the receiver still, nonetheless experience what for them is a narrative. This can also be true for some of the types of ES (see Section 2.3).

Instead, when discussing narratives two terms are quite important, that of *story* and *discourse*. The above mentioned French linguists call it *histoire* and *discours* and the Russian formalist call it *fabula* and *sjuzhet* but in general, they have the same meaning (Toolan, 2006). The term story refers to the basic objective description of the events and their chronology in the narrative. Discourse on the other hand refers to the rendering of the story, how the story is told (Abbott, 2007). In addition to these there are also the key terms of *events*, *narration*, and *plot*. Events being the individual parts that makes up the story, narration is who tells the story which has an impact on the discourse, and lastly the plot is a more flimsy term that relates to the structure of the events in the story and the discourse (Abbott, 2007; Toolan, 2006). These are all commonly agreed terms which are often used to analyse and describe a narrative.

Prince (2003) and Genette (1982) both define narratives closely in relation to these terms and mostly in regards to an event or a series of events that is merely what makes up a narrative. These definitions are of course older and many revisions has been made since (like Ryan (2007)) but for this thesis, their definitions are well suited; creating detailed narratives that would fulfill more rigid definitions would be more difficult and put a lot of strain on narrative designers within PCG environments. So we define a narrative as "*a series of events that together form a coherent meaning for the receiver*". The design process of the narrative is thus about designing the single events with regard to their meaning in terms of the overall narrative so they make sense to the receiver. Within this is then the considerations that needs to be put in to a narrative and these will be elaborated in the following sections.

2.2.2 Space and Time in Narratives

Two key components of narratives are the space and the time. For both terms there are multiple definitions but Zoran (1984) defines space as both the environment in which the narrative takes place and the objects that inhabits it and their movement in the narrative world. As a reader of a book, for example, it is important to know the relation of objects in the environment to visualise and make sense of them. As a viewer of a film it is important to show this relationship and it is not necessary to tell about the relationship as it can simply be shown (Bridgeman, 2007).

Space is an important part of narratives as it is the container in which the narrative takes place. Everything that shapes the receiver's experience of the narrative is within that space and everything that is contained in there has a potential meaning and could be analysed by the receiver. The space of the narrative can also vary in size and a narrative can be expressed from within a very confined space to the entire universe. The space within a narrative does not necessarily have to mirror the real world. Instead, elaborate worlds can be created by the author and communicated to the receiver by the words or images shown (Bridgeman, 2007; Zakowski, 2016). If the narrative takes place on a pirate ship, for example, placing a computer there would imply that this pirate ship might not be in the time in history one would assume it would and therefore space is directly connected to time and vice versa.

The term *time* that is being discussed here is the narrative time and not the time in which the narrative was created. Rather, it refers to the sequence of events and timing within the *story*, as well as the time period in which it takes place (Bridgeman, 2007).

Within narrative events there are multiple different ways of looking at them, including order, duration, and frequency (Bridgeman, 2007). A narrative event is defined as a scene or incident within the larger narrative. In the Little Red Riding Hood the fact that she meets the wolf in the wood, and the exchange of information that takes place, can be considered as an event within that narrative. The other scenes in the narrative can in similar fashion be considered as narrative events and these event together create the *story* of that narrative (Abbott, 2007). The order of these events has a great impact on the way the receiver makes sense of the narrative, because a big part of understanding the underlying *discourse* is the way in which events unfold, and the order in which different pieces of information are communicated (Abbott, 2007; Bridgeman, 2007). The order of the events can create different dynamics within the narrative. For example it is possible to present events out of time and non-linearly, also known as *flashback* (an event prior to current time) and *flashforward* (an event ahead of current time) (Bridgeman, 2007). In the same way, the duration of each of these events underlines the importance of the event for the greater narrative (Bridgeman, 2007). The frequency of which a given event is repeated also has an impact on how the importance of certain parts of the narrative. For example, if a certain character or object is encountered multiple times it underlines that character or object's importance.

2.2.3 Diegesis

The important thing to notice about the time and space within the narrative is that the narrative, and thus the time and space, is set within a confined world often referred to as a storyworld or the *diegesis* of the narrative (Abbott, 2007; Bridgeman, 2007; Schønau-Fog, 2015; Zakowski, 2016). The *diegesis* is the world that is evoked by the narrative and is inhabited by the characters of the narrative (Mittell, 2007). The *diegesis* can be completely different from the real world or it can be the exact same or close to reality. A narrative such as *Star Wars* (Lucas, 1977) takes place in a completely different *diegesis* whereas a narrative such as *Of Mice and Men* (Steinbeck, 1937) takes place in a *diegesis* that is very close to the real world.

2.2.4 Plot and Causality

A big part of comprehending a narrative is the connection between events or the *causality*. The causality between events is important because it helps to guide the receiver and thus affect the understanding of the narrative. More specifically, this inter-event analysis is known as the *plot* of the narrative and it arises not only from the story but also from the discourse of the narrative (Herman et al., 2005). Multiple definitions of plot have been made over the years and consensus has moved from plot being only related to the story to being mostly related to the discourse (Herman et al., 2005). Forster (1924) defined plot as the "creation (and also the suspenseful suppression) of causal connections between the individual events that constitute the chronology of the story". By this definition the chronological events are seen as a story, and only when the causality between these events is introduced does the plot become visible. In other words, the plot of the story lies within the causality between the events that makes up the story. This is also supported by other theorist within the field and they see these causal connections as a necessary condition for a narrative (Herman et al., 2005).

2.2.5 Linearity and Suspense in Narratives

In Section 2.2.2, the order of events was introduced as an important part of the time parameter in a given narrative. The order is indeed important also when understanding the causality between the events. A given event in a narrative follows the prevalent notion of time, meaning it moves forward in time. However, an interesting side to narratives is their ability to present events in any given order, also out of order or in reverse order (Bordwell et al., 2008). This of course has some implications as the causality between them still needs to be accounted for. In other words, a narrative does not need to uphold linearity. Instead an event can be explained, for example, by a flashback after it has taken place or a flashforward can give information that changes the dynamics of a current event.

The linearity and timing of events is important as it if used badly can almost stall a narrative and make the receivers disengage from the narrative. On the other hand, if these parameters are tweaked correctly another very important notion within narratives arises: suspense.

Suspense is a feeling that arises in the receiver while waiting for something uncertain to happen. The ability for the author to create narrative suspense is important for keeping the receiver engaged and invested in the narrative (Balint et al., 2017). Suspense arises when not just keeping details from the receiver, but letting them know you are keeping these secrets. It is about having the story progressing with the right, or variable, pace and revealing information in the right order and amount to the receiver. When designing for an engaging experience, building suspense by creating tension in the narrative is therefore paramount if the narrative is the main driver for engagement.

2.2.6 Narratvity as Simulation

Frasca (2003) argues that in order to properly understand the narrative abilities of games and other interactive experiences, they need to be separated not just from classic narratives, but from classic narrative theory. He therefore denotes games as simulation as opposed to the representational form of other narratives. He defines simulation within this matter as:

> to simulate is to model a (source) system through a different system which maintains to somebody some of the behaviors of the original system
>
> (Frasca, 2003)

This is in his own words meant to be seen as provisional as he expected a lot of progress in the understanding within this field since 2003. It is therefore important to take other elements such as the game's mechanics into account when evaluating the narrative experience as games should not just be seen as a representation but a simulation. As he states; "games are just a particular way of structuring simulation, just like narrative is a form of structuring representation" (Frasca, 2003).

If a game is a simulation rather than a representation of the authors intent, where does that place the author, and who is the author? Without digging into the many philosophical discussions on this topic, we merely look into the author as the narrator or the *creator* of the intended narrative. In our case and in many games' cases, this will be a combination of the game director and writers but also to some degree the designers as the space plays a significant role in especially games as stated above. There is many opinions on this matter and the discussions are ongoing. The main concern here is the blurred line between the game creator and the player due to the interactive nature of a game (Aylett and Louchart, 2007; Frasca, 2003).

Aylett and Louchart presented the term *narrative paradox*, which is the conflict between a pre-authored narrative and the freedom a virtual environment (VE) such as a game offers in terms of physical movement and interaction (Louchart and Aylett, 2003). In other words, how can a game be interactive and offer freedom to the user if at the same time, an author has pre-scripted a narrative that the user has to follow? The game offers much more agency to the receiver of the narrative than other mediums such as books or motion pictures. If this sense of agency is removed from the player to tell the narrative, it defeats the purpose of the medium (Aylett and Louchart, 2007; Schønau-Fog, 2015). The discourse of a game can simply not be covered by the what classical narratology offers. Instead it has to be supported by the ludological techniques, the game mechanics, and the interactivity (Larsen and Schoenau-Fog, 2016). Therefore, it is, to some extent, necessary to look away from the classical methods of storytelling and focus on a new set of theories that can incorporate the interactivity and other mechanics. So when designing a narrative for a game it is important to preserve the agency for the player. We will therefore now look towards some of the new possibilities for narratology that is offered within games. Before doing so, we will however reiterate on some of the key findings from this section.

2.2.7 Summary of Narratology

Classical narratology is a very interesting area that has been researched for many years. An important part of these narratives is the space. Often the diegesis and relation between different spatial elements are tightly controlled by the author as it often helps tell that particular narrative. When introducing PCG a lot of this control over the space is lost and therefore it can be more difficult for the authors to tell a coherent narrative. It is however important to still use the classical terms depicted in this section when evolving the narrative language beyond books and motion pictures.

In newer narrative mediums such as games, or similar interactive experiences, the space parameter actually plays an even bigger role as the receiver is not necessarily guided by the words or images from an author. Instead the receiver can often freely roam a VE and the space therefore becomes much larger and more open. As written above, the space has a huge role in the storytelling and this role therefore becomes more evident in these newer forms. As clearly seen in this section, the classical theories within narratology does not quite cover the wast opportunities that games and other interactive experiences offer. It does not mean that these older theories have to be dismissed completely, they simply need to be expanded. There has therefore been conducted a lot of research within the last 20 years in this field. And part of this has been concerned about the added layer that a VE offers.

2.3 Environmental Storytelling

When discussing narratives in games a large part comes down to the large elaborate environments in which the characters roam. Games also offer many other possibilities to the user including the mechanics and rules of the game and these parts must also not be forgotten when discerning the narrative possibilities a game offer (Larsen and Schoenau-Fog, 2016). In this thesis however, the main focus will be on the VE and how this affects the narrative as the PCG will affect the VE and in turn might impact the narrative. Looking into game mechanics in this case would therefore be less interesting.

Historically, storytelling has mostly been viewed in terms of time, i.e. the timing of events in the story (e.g. (Ricoeur, 1980)). However, in works like a game not only the time but also the space plays an important role in the storytelling. Due to the existence of a perceivable environment in which the player roams, the environment does not only serve as a container, but can be used to enhance the narrative or even serve as the main actor in the storytelling process, i.e. environmental storytelling (ES) (Baynham, 2003; Fina, 2003; Zakowski, 2016).

ES originates from Carson (Carson, 2000*a*,*b*) but is especially known by the definitions from Jenkins (2004). He defined ES as different ways of utilising the VE to convey the narrative. In particular he defined these four different types:

- Evoked narratives
- Enacted narratives
- Embedded narratives
- Emergent narratives

Evoked narratives means that an existing narrative or diegesis is enhanced by the details in the VE. This could be a VE where objects known from a certain storyworld are placed (e.g. TIE fighters could tell that this VE is part of the Star Wars universe) (Jenkins, 2004). An enacted narrative puts the player and their character centre stage. Here the micro narratives that are created by the player's actions contribute to the overall narrative. The encounter or props in the VE can all contribute to the narrative that is experienced by the player (Jenkins, 2004). Embedded narratives utilise player's ability to construct the narrative themselves based on the details presented to them. Here the VE is a "memory palace" where props and the mise-en-scène can reveal details to the players which in turn can help them reconstruct the plot (Jenkins, 2004). Lastly, emergent narratives are much more open to the player. Here the player can construct their own narrative based on encountered events and props in the diegesis. The game does not offer a specific story but instead makes it possible for the player to make stories based on their actions within the game (Jenkins, 2004).

Figure 8: *Portal uses embedded narratives to provide narrative information not given by the narrator (Valve, 2007)*

A possible goal for this thesis is to keep narratives coherent within the VE, even though the VE itself would change each time the game is played. Evoked narratives rely on an already existing diegesis which would require a lot of work on imitating such a diegesis (not to mention creative rights). In addition, the changes to the VE that the PCG introduces would be less interesting in such a design, as the narrative would still need to be curated. This also holds true for enacted narratives and thus these are similarly uninteresting for this thesis.

Embedded and emergent narratives are both interesting in terms of PCG. Emergent is interesting as it lends itself to the player constructing their own narratives from the VE and as the VE changes from play through to play through so could one expect the emergent narrative would for the player. However, the narrative already changes from time to time in an emergent narrative as this is the foundation itself for this type of narrative. Furthermore, emergent narratives often require many elaborate and detailed systems interacting to facilitate emergence. Embedded narratives are interesting in terms of PCG as the VE itself contributes to the telling of the narrative and thus, changing this could dramatically change the narrative the player experience.

2.3.1 Embedded Narratives

Embedded narratives can take centre stage or be a supplement tool to game designers. Some games rely almost solely on embedded narratives and they require a high level of comprehension from the player. More popular is using embedded narratives in conjunction with other modes of storytelling. Popular games such as the Uncharted series (Naughty Dog, 2007-2017) makes use of this a lot. The attentive player will notice details in the space design that reveals clues of past events and these might be confirmed by the characters after some time with dialogue or cut scenes. Other popular games such as *Portal* (Valve, 2007) and *Half-Life 2* (Valve, 2004) use embedded narratives to reveal narrative information, or hint at sub-plots, which are not provided by the game's narrator. In some cases these embedded narratives even help teach the player game mechanics and concepts.

Embedded narratives can be seen in many popular "walking simulator" style games, such as Firewatch (Campo Santo, 2016) or Gone Home (Fullbright, 2013). In such games, the player wanders around open spaces that are designed to contain valuable pieces of narrative information. Often, they also make use of different kinds of dialogue to express the thoughts of the characters and thus underline the meaning of the pieces. Embedded narratives can however stand on their own as touched upon above, but it does open up for a lot interpretation by the receivers. As an example, Bevensee et al. (2012*a*) developed the game *Aporia: Beyond the Valley*, with which they researched whether the "open-ended" format of these kinds of ES games affects the continuation desire of the player (Bevensee et al., 2012*a*).

2.3.2 Interpretation of Embedded Narratives

An embedded narrative is still very much curated by the game designers but the telling is much more reliant on the space design. Often the narrative is enhanced in this way and details can be embedded for the player to discover but often the main points that the designer wants to communicate are not only relying on the space design. There is however, nothing stopping designers from telling the story entirely using the space design. This of course puts a lot of strain on the player and might end up yielding totally different narrative experiences from player to player as ES is more of a bottom-up process for the player compared to classic storytelling.

This relationship is addressed by Bruni and Baceviciute (2013) as they propose the notion of the *Author-Audience Distance* (AAD) which is a measure of the interpretation gap that occurs between the author and the receiver when telling a narrative. On this continuum, larger levels of abstraction yields more ways the narrative can be interpreted versus smaller levels of abstraction where the narrative is more precise and thoroughly explained (didascalic) thus diminishing the gap between the author and receiver (illustrated in Figure 9). When designing a narrative it is good to have the level of abstraction in mind, and in terms of ES, and especially in the case of embedded narratives, it is preferred to create a narrative with low abstraction levels as otherwise some users might have the intended interpretation but many others might have different ones. This is of course dependent on whether such a gap in interpretation is wanted or not.

2.3.3 Staged Areas

An important part of embedded narratives is the use of *Staged Areas* (Jenkins, 2004). A staged area (SA) is a place in the VE that has been designed to specifically convey a part of the narrative. Instead of designing the entire space as an embedded narrative, the game designers instead design smaller specific areas that the player can discover and by that discovery gain more information about the narrative itself, such as the area from *Portal* shown in Figure 8. These areas then become equivalent to an event in a classic narrative (as described

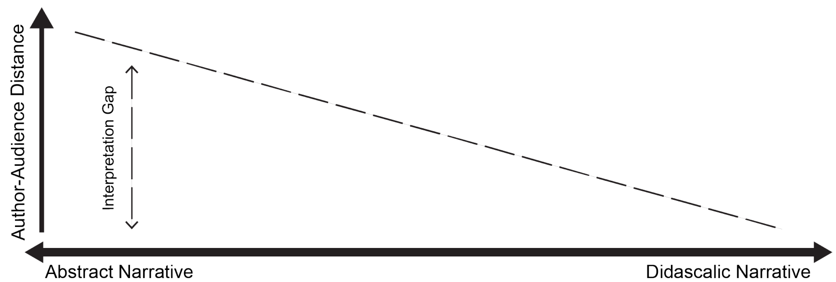


Figure 9: *Illustration of the Author-Audience Distance as created by Bruni and Baceviciute (2013)*

in Section 2.2.2). So where a writer would repeat specific events or pieces of information to underline the importance, the game designer can create multiple SAs or have the same information in multiple of the SAs of an embedded narrative. In regards to the PCG approach this thesis has, the usage of SAs are important as they can potentially serve as a great tool in keeping narrative information similar across different VEs. Designers could hand-craft a number of SAs, which could then be used to populate the generated environment. This maintains some level of designer control, but could still alleviate the workload by allowing designers to focus on key narrative areas, rather than the entire environment.

2.3.4 Delimitation on Environmental Storytelling

ES is in many ways very powerful in terms of conveying narratives in games both as a supplement and even as a standalone method. We will focus on embedded narratives, as they seem suitable for use with a PCG environment. The use of SAs in particular could prove to be valuable in the design of a PCG prototype. ES, in general, evidently clashes with the use of PCG due to the loss of autonomy from the developers on the environment and thus the storytelling aspects of this. As Jenkins (2004) argues, for all games, space plays a major role and therefore it is important that the effect the PCG has on the space part of the narrative is known. By only using embedded elements in the narrative space we will create a more extreme version of an embedded narrative without additional methods of conveying the narrative (similar to Bevensee et al. (2012*b*)). This enables us to measure what effect changing the space by PCG has on the narrative understanding for the players. From here on, when discussing embedded narratives, we refer to this version of an embedded narrative and not just any narrative with embedded elements. In Section 4 the details of such a narrative experience will be outlined. However, another important part to designing narratives in games is understanding how the narrative affects the game (mechanics, VE, etc.) and equally, the other way around.

2.4 Designing for Environmental Storytelling

When designing a game there are many considerations to take into account. When putting the narrative at the centre, the amount of considerations does not become less. All the above mentioned narrative design choices has to be implemented in a VE that might be very large or very small. The narrative might be dependent on specific places, timing, or a combination of the two. Maybe an event does not make sense if the player has not found an item and maybe the player are free to roam and thus maybe never finds that item. The list goes on and therefore there is many theories and tools that has been developed to aid game designers in combining a game and its narrative.

2.4.1 Space and Time of Narratives in Games

There is a difference from the general implementation of ES in our case, as this game will be utilising PCG. Normally, the game designers have full control over the implementation of everything connected to the narrative, as they have control over everything else in the process. In games that rely heavily on ES the need for careful consideration only becomes bigger, and so letting go of the control over these elements would seem impossible if the end result has to be on par. Normally the placement of props and general design of the space would be all meticulously crafted to strike the right balance and achieve the desired result.

The introduction of PCG removes some control over the space from the designers and in turn the control over the ES. The PCG algorithms are able to generate almost everything in relation to the space. The PCG system could be expanded to distribute props relating to the narrative in the same way that it does trees, rocks, and other environmental props. However, randomly distributing narrative props would defeat the purpose of ES as it is not just about the right props but also about the right placement, timing, etc. (Jenkins, 2004). A better approach, as suggested in Section 2.3.3, would be to hand craft SAs, and then have the PCG algorithm place these areas in the VE.

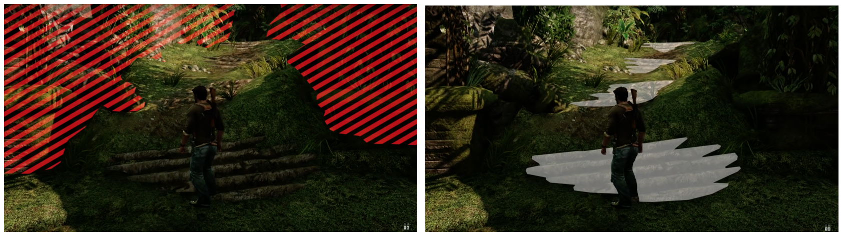
This might solve part of the space problem but the spacing can also impact the timing of the narrative. Within timing there is two main considerations: order and frequency. This is something that are key to any narrative (as explained in Section 2.2.5), but both are being impacted by the introduction of PCG. The order can be a concern in many games especially in those with more open environments with less control over the players movement like sandbox games. Even if measures are in place for creating the right order, the frequency and timing might still be difficult to strike. If the player does not encounter the right things at the right times they might loose valuable information.If the frequency is too low they might get bored and loose interest from lack of engagement. If it is too high they may not have time to process the presented narrative information before receiving more. Getting the right balance in the narrative is therefore difficult when moving from a completely hand-crafted game to one relying heavily on PCG and many of the normal game design methods becomes void.

To overcome this challenge a complete system taking both space and time into consideration when implementing the narrative in the PCG VE is needed. Schønau-Fog (2015) has proposed a method for handling a similar problem in games like the aforementioned sandbox games. The problem occurs when a player should have complete freedom in the environment and feel that their actions affect the game, but at the same time experience an engaging narrative. It is what has also been referred to in Section 2.2.6 as the narrative paradox. As Schønau-Fog argues that only locking narrative events to space in such free-roaming experiences increases the chance of players experiencing them in the wrong order, at wrong times, or even loosing valuable information all together, ultimately increasing the chances of experiencing a different narrative than intended (Schønau-Fog, 2015). He therefore proposes the Space-Time Drama Manager (STDM) which works from the assumption that events do not necessarily need to be locked in space or time. Instead, they can be placed in space when the player needs them to be. The events can then be much more dynamic and uphold a variety of temporal orders while integrating into the VE based on the player position, trajectory, or other parameters. The events can further be divided into different kinds of event capsules that have different requirements for how tight or loose they might be related to the player in space and time (Schønau-Fog, 2015).

- Events that are not locked in time and space
- Events that are locked in time but not in space
- Events that are locked in space but not in time
- Events that are locked in both time and space

A game designer utilising this method will be able to define an endless amount of different events in the STDM and then dynamically make it spawn these. The requirements for these will in most cases be based on the specific VE that is the game. However, these requirements do not need to be directly related to a specific VE but can also be more generalised, and this is where the STDM becomes interesting in terms of PCG. If a designer is creating events, then they can also create SAs for an embedded narrative. These SAs can be passed to the STDM, which will ensure that the SA will be encountered at an appropriate position in both time and space, thereby upholding a temporal order and frequency in the narrative which can help maintain suspense and engagement in the player.

There will still be a need to help guide the player. Not to control their specific movements, but to give them a sense of purpose and direction in the game. We can also use these same methods to not only aid the game design but guide the player and thus better predict their movement and behaviour which can in turn allow the STDM to anticipate where best to place SAs.



(a) *(b)*

Figure 10: *Two examples of how the Uncharted games has only one path for the player but make it seem like they have a free choice. They use both (a) untraversable areas that also frame the path and (b) provide subtle cues that hints a usable path (Brown, 2015; Naughty Dog, 2007-2017)*

2.4.2 Player Guidance in Games

Guiding players in games can be very straight forward. It can also be redundant if the design of the game calls for it. Often, designers will need some kind of control over where the player moves and when. Rotzetter (2018) identifies six distinct methods of player guidance: Informative, Interactive, Processual, Spatial, Emotional, and Narrative. These all have pros and cons and they range from very explicit systems like the informative "mini maps" to the more implicit systems like spatial (Rotzetter, 2018), which similar to ES, uses the environment to guide players.

For this thesis the easiest method would be to implement something adaptable like an informative system. In this way, aids like way points could be implemented and no matter the design of the VE caused by the PCG the player would easily be able to find their way through the environment. However, due to the unguided nature of an embedded narrative, implementing such a system would counteract the purpose of this. Instead, spatial guidance systems offer many clever ways to guide the players much more subtle. In games like the Uncharted series (Naughty Dog, 2007-2017) the designers utilise many different guidance systems. The maps in this series are often not open world. Instead they are predetermined and do not leave room for much exploration (see Figure 10). However, the game is designed in such a way that the players get a feeling that they can go where ever they want when they are really guided by clever environment design with multiple guidance systems (Brown, 2015; Gunson, 2013). One very interesting method that is utilised in the Uncharted games is that of *Wienies*.

A Wienie is a term used by Walt Disney about central landmarks in his theme parks (Korkihort, 2016). By placing big distinct landmarks in the VE the player is given an aim for their wandering (see Figure 11). This works in Uncharted because the player is told "that tower looks interesting" and they then move towards that tower in what they think is their own way. By implementing Wie-

(a) *(b)*

Figure 11: *Two examples of Wienies giving the players a direction in the Uncharted games (Naughty Dog, 2007-2017). Screenshots from (Brown, 2015)*

nies the players can be guided, and given a sense of purpose from the beginning of the game without restricting their agency. In terms of implementing a STDM in a game, the wienies would add a good parameter for predicting where in the VE a player might go.

2.4.3 Delimitation of Design for Environmental Storytelling

There are many considerations when designing games and the list does not become shorter when introducing ES and PCG. This section has purposefully only researched methods for narrative timing and player guidance, firstly to identify specific methods for managing the narrative aspects of a VE, and secondly do so in a way that also supports ES. Here the STDM from Schønau-Fog (2015) is a perfect fit as the STDM's event capsules and SAs from embedded narratives can arguably be interchanged. The STDM can be improved by having more parameters which it works with and therefore clever player guidance can aid in improving timing and placement of the SAs. In Section 4 the implementation of these methods in conjunction with PCG will be detailed. Having the necessary tools for designing and implementing a game utilising PCG and ES, it is key to be able to properly record the effects of this combination and therefore the next section will look into evaluation methodologies for narratives.

2.5 Narrative Experience Measurement

There are many ways to go about evaluating a narrative but researchers are often faced with a problem - the experience of a narrative is quite subjective. Therefore, many different methods have been developed to objectify the individual experience of a narrative. As discussed in Section 2.2.1, the plot that is perceived by the receiver depends very much on the discourse of the narrative, and changing the VE is changing part of the discourse which may also lead to changes of the story. It is therefore key that we are able to properly evaluate the narrative experience the receivers have.

As the experience of a narrative is subjective to the receiver and the process is internal, observations is not the ideal method to use. Emotion recognition can be used to gain information on the receivers emotional response to the events, and these can be compared to the desired response. However, this does not necessarily reveal how the receiver experienced the discourse and causality between the events, which in this case is key to telling a coherent narrative. Our aim is for the players to create a narrative themselves from the information presented to them via the embedded narrative. We will give the puzzle pieces in the SAs but they have to put them together themselves to make sense of it. For the purposes of evaluating how players experience a narrative, it would be desirable that the narrative will not only make sense to one person, but for all players to have an understanding close to the intended narrative of the author, i.e. the narrative should have a short AAD.

In addition to AAD, engagement in a narrative can reveal much about a receiver's experience. This can affect both their ability to understand and perceive the quality of a narrative. We have therefore developed our own method of measuring the narrative experience, named, the Narrative Experience Measurement tool or in short, the NEM tool. This section will present this in full detail.

2.5.1 Overview and Purpose of the NEM tool

As stated above, the goal here is to convey an engaging narrative that makes sense to the player, and is as close to the author's intend as possible. As such, the NEM framework is split into three parts, to evaluate the engagement, individual understanding, and AAD. The three parts of the framework are:

Narrative Engagement A measure of how much each individual player was engaged in the narrative.

Internal Coherency A measure of how much each individual player perceived the narrative as being coherent and making sense.

External Consistency A measure of the extent to which multiple players experience and understand the narrative in the same way.

By combining multiple existing methods to explore these parts we can get a solid overview of the narrative experience a game yields for the players, both as individuals and as a collective. We will utilise this framework ourselves in this thesis to investigate how the introduction of PCG affects the player's narrative experience from ES. The remainder of this section will explain in more detail how the NEM tool can be used.

2.5.2 Narrative Engagement

Busselle and Bilandzic (2009) provides us with a well documented method for evaluating participants' narrative engagement. It has been adapted by many others like Richardson et al. (2018) and though it was developed for other non-interactive media, multiple studies like Christy and Fox (2016) have already adapted it for games. It can be argued that the *measurement* of narrative engagement is similar across platforms, and games might even prove to be more engaging as is the case when comparing audio books and motion pictures (Richardson et al., 2018).

It comprises of 12 Likert items separated in 4 groups evaluating *Narrative Understanding, Attentional Focus, Narrative Presence,* and *Emotional Engagement.* Each of the items can be answered on a scale from 1 to 7 ranging from strongly disagree to strongly agree. Since the narrative will not be character-driven, emotional engagement is of less concern in this instance. For this thesis, these items will be omitted from the NEM, though they could be reintroduced if the narrative in question warrants.

There are many other methods for measuring engagement in games but we chose the narrative engagement specifically as we are not interested in knowing whether the overall game experience itself is engaging. Instead we just want to know whether the narrative aspect of that experience is engaging. By also measuring game engagement with methods such as continuation desire (Bevensee et al., 2012*a*; Schønau-Fog, 2014) or the game engagement questionnaire (Brockmeyer et al., 2009), we could potentially triangulate the results and exclude biases from elements that are not included in Busselle and Bilandzic's method. If the player is not in general engaged in the game, that might be the reason they are not engaged in the narrative. However, we argue that if they are engaged in the narrative they must consequently be engaged in the game experience. We therefore might get a false negative in any usages of this.

2.5.3 Narrative Coherency

To measure internal coherency we will partly use the narrative engagement questionnaire (Busselle and Bilandzic, 2009) as part of that is narrative understanding which points towards the coherency. However, in order to measure this better and to see whether we get external consistency we need to know specifically how the participants experience the narrative.

A key point in regards of the narrative experience, is to differentiate between the narrative that the authors try to convey and the narrative the receiver experience. As environmental storytelling in general is less directly curated by the game designers there is more leverage for the player to generate his own narrative understanding. The narrative process even in an embedded narrative can very much be treated as an emergent narrative by the definitions in Larsen et al. (2019). From the designers point of view, they merely create and present events to the player, but without an author there is no one but the player to curate those events into a coherent story. Instead, we must rely on the receivers ability to create the causality between these events themselves. The process thus becomes a bottom-up process instead of a top-down process like other narrative experiences. There is however no guarantee that the causality a specific receiver creates is the same as the one intended by the authors. This of course proposes a challenge for the game designers as they want to (in most cases) ensure the players get the right, coherent, narrative they had intended.

To test which narrative a receiver experienced, many different methods can be used. These are all subjective by nature and therefore need to be chosen carefully. One solution is to ask the receiver what narrative they experienced. This can be done in multiple ways like asking directly what they thought the plot was or more indirect ways like having them retell the narrative with a specific focalisation (Hoek et al., 2014). Larsen et al. (2019) defines the term "Afterstory" which is the narrative that a specific receiver generates and are left with after the narrative experience is over.

> The afterstory can vary greatly from person to person in a more emergent game or be more similar in a linear game, but there will always be subtle nuances, since the player's interpretation, reading, and feeling of narrativity will be different, even in a completely scripted sequence.
>
> (Larsen et al., 2019)

To get the afterstory from the receiver a method also proposed by Larsen et al. (2019) is *retelling*. The retelling is a new narrative that is created from the afterstory when the receiver retells the narrative they experienced, their afterstory. Bevensee et al. (2012*b*) used a similar retelling method to explore the participant's afterstory which was then coded using grounded theory to compare to other participants. We however propose a slightly different approach. To keep it simple, we are only interested in a certain amount of understanding from the participants, what Bordwell et al. (2008) describes as the referential meaning and maybe to some extent the explicit meaning. Instead of doing a complete retelling we are going to formulate at least three key points that the participant needs to get from the narrative in order to assume they have the right referential or explicit meaning from it (see Section 4.1). To know if they have this they will be asked multiple direct questions that relates to the specific pieces of information e.g. "what do you think this specific prop meant?". The answers to this can then be used to see if they got this, and other pieces of information, reasonably correctly and in turn understood the referential meaning of the narrative. It would be beneficial to have measures that are less reliant on the participant's memory and ability to generate the narrative themselves like quantitative measures but the subjective nature of narrative understanding means that qualitative measures like this is the only viable solution.

It will however be beneficial to quantify the measures in order to do statistical analysis on them and better compare across conditions. The answers will therefore be coded using content analysis (Bjørner, 2015) based on the understanding of the narrative and compared across participants.

To perform this comparison, and to see how the participant's afterstory compares to the intended narrative we will use AAD from Bruni and Baceviciute (2013) (see Section 2.3.2). For each of the key points, the participant will be asked to chose between some pre-defined answers, or they can add their own answer based on their afterstory, in a multiple-choice question. Each of the possible answers represent a greater distance between the author and the audience. Here it is also important to note that an embedded narrative (and ES in general), no matter how simple a narrative is trying to be conveyed, is not a completely didascalic narrative as the room for interpretation of the SAs requires more effort from the audience compared to other mediums that has more direct control over the presentation and thus interpretation. The embedded narrative of this thesis can therefore be considered to be somewhere in the middle between a very abstract and a very didascalic narrative. The interpretation gap is therefore quite big already but within this gap is multiple levels of understanding which will be represented by the different answer possibilities in the multiple-choice questions.

There is of course a bias in defining answers for the participant in advance, but this is accounted for by including an option to add their own answer if they do not feel any of the pre-defined aligns with their afterstory. Before each of these multiple choice questions, there should be an open-ended version of that question. In this way, we get the participant to think about their experience of this before giving them options. The multiple choice questions should be on a different page in the questionnaire as well so the participant cannot see what they answered and thus biasing their answers.

There will be multiple questions concerning the same key point to triangulate the answers. In addition, there will also be some questions that are designed to be misleading, for example by hinting at narrative elements that were not present, which should make the participant reflect more about the narrative. The participants will be informed that there is no singular correct answer, and they should not answer what they think is "the right answer", but what they experienced. The options for each question will be determined in advance based on the narrative and the background story of the narrative that is designed in Section 4.1.

2.5.4 Delimitation of Narrative Engagement Measurement

In order to collect all the needed information the NEM tool will be created as a questionnaire that comprises of all the above methods. This questionnaire can be seen in Appendix A and the specific methods can be seen in Section 3.4. The usage of this tool is to enable developers to compare the narrative engagement using the narrative understanding, attentional focus, and narrative presence parts of the narrative engagement questionnaire developed by Busselle and Bilandzic (2009). This questionnaire also provides an insight into how the narrative understanding is for the player. We have however developed our own method for testing the player's narrative understanding based off the works of Larsen et al. (2019) and Bruni and Baceviciute (2013) creating a method for comparing the AAD to the player's experience. This further enables us to not only compare the internal coherency for each individual player, but to compare the external consistency between players.

In Section 3.4.2, we will elaborate on the specific methods for quantifying these qualitative measures in order to measure the external consistency. Having defined this methodology adds the last needed background for defining the specific methods and designing a prototype. Before doing so a conclusion on this entire analysis is in place.

2.6 Analysis Conclusion

This entire section has explored and delimited the existing research in order to form a basis on which firstly the methods and later a specific design of a prototype can built upon.

By looking into different types of PCG systems and methods, we found that the planned terrain generation system can be described as online automatic generation of necessary PCG materials (Section 2.1.1). The system should be deterministic with at least a single dimension of control. The system will likely be generic, but could yet be adaptive, and it could either use constructive or a generate-and-test approach.

The terrain should be generated with a heightmap, using techniques such as fBm to create realistic, natural looking topography. This can be done using multiple octaves of Perlin noise, or an algorithm such as Diamond-Square.

When working with a PCG system, it can be evaluated and understood by using a heat map to illustrate the expressive range based on numerical metrics of emergent properties of the generated content. Such a method of evaluation can identify biases in the generator, and help quickly give a representative illustration of the generated content.

When substituting a large part of the curated space element of the narrative by a PCG algorithm, the developers and authors lose a lot of control over those narrative elements. This section has therefore researched multiple approaches to storytelling both classic and novel methods, all with different dynamics. Within games an increasingly popular form of storytelling is ES, which is also one of the most vulnerable to the changes in the VE that PCG introduces. Within ES there are four different types defined by Jenkins (2004): evoked, enacted, embedded, and emergent. Due to the nature of PCG it was found that embedded would fit well with the limitations and possibilities the technology offer. Evoked and enacted was deemed to be too reliant on accurate spatial parameters and emergent would be interesting in combination with PCG but would not be good at revealing how it affects the narrative. The specific method of using SAs in relation to an embedded narrative seems especially promising for designing narratives in PCG environments.

When designing a game that makes heavy use of PCG the creators have to rethink how the game is designed especially in terms of the narrative. Putting a PCG algorithm in charge of creating the VE takes away a lot of the possibilities the designers normally have to implement the narrative exactly like they want. This can in some ways be compared to implementing narratives in a sandbox like game where the player is given full control. We therefore decided to adapt the Space-Time Drama Manager from Schønau-Fog (2015) and treat the SAs as events that should be triggered independent of space and with the right timing to ensure narrative coherency and tension (suspense) which in terms can improve the engagement in the narrative. The STDM should to some extend be seen as an additional PCG system as it also, in its core, procedurally distributes the SAs. To further aid the design of such a STDM help can be found from player guidance theory known from other types of games. Here, especially the use of wienies are of interest.

With the novel nature of the PCG/ES combination in mind, it was clear that a specific method for evaluating such a system was needed. We therefore developed the Narrative Experience Measurement (NEM) tool. This framework will be able evaluate players' narrative engagement by implementing part of the Busselle and Bilandzic (2009) narrative engagement questionnaire. In addition to engagement, the NEM tool is also measuring the narrative coherency, the players have based on their narrative understanding. In other words - does the player create meaning from the presented information. This is achieved in a combination of part of the questionnaire from Busselle and Bilandzic (2009) and a novel method utilising a combination of open-ended questions and multiple-choice questions seeking to uncover the players understanding of key points in the narrative. The possible answers to the multiple-choice questions will be defined in accordance to the Author-Audience Distance developed by Bruni and Baceviciute (2013). Combining this provides a solid foundation for evaluating any prototype that combines PCG and ES and it can also be utilised in other situations but for this thesis this will be the main focus.

2.6.1 Research Question

This thesis will be an experimental research design that seeks to answer a central research question. It will be delimited to utilising PCG in conjunction with embedded narratives from ES to create a procedural VE where a STDM will dynamically place predefined SAs to make it possible for the player to get a coherent and engaging narrative experience solely from the ES in the game. In addition, we will utilise our own methodology in form of the NEM tool to evaluate the developed prototype. The research question that guides the reminder of this thesis will therefore be:

Will using an online PCG environment affect a player's engagement and narrative coherency of a dynamically embedded narrative?

To answer this, two conditions should be created - one where the PCG system's random seed will stay the same and one where the random seed will be different each time the prototype is played. This means that half of the players will experience the same VE, while the other half will each have unique VEs. We hypothesise that, if our system works as intended, the mean scores and variance between these two group's measures of narrative engagement and narrative coherency will not be significantly different. If this holds true it will show that it is possible to not only tell a narrative based on ES in a PCG environment but to do so in different VEs each time without affecting the narrative engagement and coherency for the player. This leads to the following experimental. and null hypothesis:

H0: There is not a significant difference in engagement levels and narrative understanding between the two groups

H1: There is a significant difference in engagement levels and narrative understanding between the two groups

3 Methods

The methodology used to answer the research question of this thesis will consist of multiple methods. This section can therefore be seen as a road map of the remainder of the project through the methods that will be utilised. In order to answer the research question, multiple evaluations of the prototype will be in order. This thesis uses exploratory sequential mixed methods, where the analysis in the previous section aimed to collect and analyse qualitative data in the form of existing research in order to generate the research question, and the following sections will gather more data, both qualitative and quantitative. These data will partly be used to inform the design of a prototype which will be used for the outcome evaluation, with the methods depicted in this section to ultimately provide an answer to the research question presented in Section 2.6.1.

The methods from here on will be more of a convergent parallel mixed method which will have multiple evaluations yielding both qualitative and quantitative data (Bjørner, 2015). Firstly, a formative evaluation of the development of a prototype which will be refined over multiple iterations. This will be followed by a summative evaluation of the developed prototype to see if it achieves the set goals (Bjørner, 2015). As the prototype is completed it will be used in correlation with the developed methodological framework, NEM (see Section 2.5), to carry out an outcome evaluation seeking to quantify the narrative experience of the player, to triangulate the data and provide an answer to the research question. The detailed methods for all steps in the process will be explained here in chronological order as they are carried out.

3.1 Sampling & Target Group

The target group for this thesis is broad, with very few restrictions. This is both due to the lack of specific skills, experiences, or knowledge required to participate in the evaluation, and in order to facilitate the recruitment of large numbers of participants.

As the thesis is about understanding the narrative, as opposed to any skill or experience, it is not important to restrict the target group based on experience with video games, or player type. Due to this lack of restrictions, however, it is worth gathering this information as demographics so that if needed it can be accounted for.

While the game could be played by people of any age, due to the content, the narrative is targeted towards adults of 18 years of age or above. No gender is specified for the target group, though this will still be gathered as demographics.

The participants will be recruited online using convenience sampling and snowball sampling (Preece et al., 2015). These sampling methods and a broad target group should help achieve a sufficient number of participants to conduct proper statistical analysis when needed.

3.2 Initial Evaluation of Topography

A large part of the methods used for this thesis will be about creating and improving the PCG system. To get the best possible starting point for generating terrains a small evaluation should be carried out. The goal of this evaluation is to figure out, firstly, whether participants can recognise if a terrain is real or computer generated. Secondly, it should provide a basis for discussing what realistic looking terrains looks like and which descriptions are related to these. When we have developed a PCG prototype that is able to generate terrains it will be used to generate both real terrains, meaning generating a terrain from a heightmap from a place in the real world, and procedurally generate random terrains. It will then be a within-subject evaluation where each participant will be asked to look at still images of both the PCG terrains and the real world terrains. Both types of terrains will be generated in a game engine so only the topography is different from terrain to terrain. For each terrain there should in addition be images from three different perspectives to see how the descriptions change based on the point of view. There will be three terrains from real-world data and three PCG generated. With three images for each it amounts to 18 individual images each participant has to look at.

For each image the participant will be asked to state whether the terrain is based of real-world data or computer generated (see Figure 17 in Section 4.5.2 for examples of images). After this, they will be asked to state why they think one or the other. The nominal data from asking them whether they think it is one or the other will be treated using descriptive statistics to determine if people are able to recognise a real world terrain from a computer generated one. The qualitative data from elaboration questions will yield some statements that are an indication of why the participants answered as they did. Using grounded theory and open coding the statements will be coded and the codes themselves originate from the source material (Bjørner, 2015). This will provide good insights into which features are regarded as realistic looking and from this we can tweak the PCG algorithms to yield more realistic results for the terrain generation. The specifics of this evaluation can be seen in Section 4.5.2.

3.3 Formative and Summative Evaluations

The test of the topography serves as the first step in the iterative process of evaluating and improving the PCG system. After this, multiple more iterations should follow with the aim of improving both the performance and results of the algorithms. Both a formative and a summative evaluation will be conducted of the prototype to ensure it is on par before it will be used in the final outcome evaluation described in the next section. A general overview of methods used for the formative evaluation can be seen in Appendix B. The outcome of this formative evaluation is described in Section 5, and some of the early development is described in Appendix B. The methods for the summative evaluation can be seen in Appendix D together with a description of the results.

3.4 Outcome Evaluation of Narrative Experience

To answer the research question, a between-groups evaluation will be used. In the control condition, participants will experience an embedded narrative in an environment that is generated with a parametric PCG system, which at runtime has a single dimension of control, i.e. a random seed. This random seed can be used to reproduce environments, and to ensure that participants in this condition experience the same environment as each other i.e. factors such as the topography of the landscape, the distribution of foliage, etc. will remain the same for all participants.

In the experimental condition, participants will be presented with a similar VE, also containing an embedded narrative. However in this case the environment will be generated from a different random seed for each participant. This means that no two participants should experience the same environment in this condition.

Data will be collected on four topics: narrative engagement, narrative coherency, tension in narrative composition (suspense), and the participants' experiences of the VE. All data is gathered via a post-game questionnaire. This data will be used to compare the two conditions using the NEM tool (see Section 2.5).

The evaluation and methods will be pilot tested to ensure the questionnaire yields the correct results and the prototype functions as expected. The detailed experimental setup for this evaluation and the proper data analysis procedure will be explained below. The results can be seen in Section 7 and their meaning in relation to the research question will be discussed in Section 8.

3.4.1 Experimental Setup

As mentioned in Section 3.1, the participants will be recruited for the test using convenience and snowball sampling methods. Participants will be provided with a link to a website which hosts the application. When launching the application, participants will be randomly assigned to either the control or experimental condition. The random distribution should result in approximately equal numbers of participants in each condition.

The application begins with a welcome screen which will introduce participants to the test scenario. Continuing through the application, participants are presented with another screen which introduces them to, and provides context for the narrative, as described in Section 4.1. While participants read this narrative introduction, the PCG system is at work in the background generating and populating the environment. When the PCG is ready, the participants can play the game.

Once the game is finished, they are redirected to an online questionnaire, which will keep a record of what random seed the participant had so that the answers in the questionnaire can be associated to what condition they experienced. The questionnaire consists of five sections, which are outlined below:

1. Narrative Engagement The first section of the questionnaire will consist of nine Likert items adapted from Busselle and Bilandzic's Narrative Engagement Questionnaire (2009). This will first and foremost point towards the narrative engagement of the participants but the part concerning narrative understanding will also be used together with the results from section 3.

2. Narrative Tension The second section is a Likert scale concerning the tension in the narrative. This consists of six Likert items that address the progression of the narrative and the participant's attention over time and whether we succeeded in creating suspense. This can also be used to point towards narrative coherency as players would not be able to experience tension in the narrative without also experiencing some level of coherency.

3. Narrative Coherency The third section consists of questions about the specific details of the narrative that we want the participant to get from the game experience (as described in Section 4.1). There are two questions for each key piece of information in this section, and two are "dummy questions", which ask about non-existent narrative elements.

These dummy questions should help indicate if participants have properly understood the narrative (i.e. can identify that these elements were not present), and that they are truthfully responding to the questionnaire. This method has been used in a previous study which was also concerned with participants noticing narrative elements in a VE (Kvisgaard et al., 2019).

These questions are first asked in an open-ended format, followed by multiple-choice. As explained in Section 2.5.3, each question is presented on a different page of the questionnaire. This way the participant will first give their own unbiased understanding of the narrative, before giving an answer that may be biased by the options presented.

4. Virtual Environment The fourth section is a user-focused evaluation of the output from the PCG system. This consists of a number of Likert items addressing the usability, suitability, and navigability of the generated environment, as well as if the environment was interesting for the participants. Each Likert item is counter-balanced and asked twice in different terms.

5. Demographics The final section collects demographics about the participant, such as age, gender, and experience with playing and making video games. They will also be asked to chose what type of player they think they are when playing games for us to compare what motivates them as players and therefore provide an insight into how player motivation might affect our results. These player types are based on Yee (2016) player motivation model, and the participants will be introduced to player types by looking at the table seen in Figure

Figure 12: *Table with descriptions of the different player types as defined by Yee (2016)*

12. The last part will also enable the participant to comment further on the game and narrative they just experienced. After completing the questionnaire the test is over and should have lasted about 25 minutes in total. All items from the questionnaire can be seen in Appendix A.

3.4.2 Data Analysis

The questionnaire yields a lot of data that needs to be analysed in different ways to answer these five overarching questions:

- Were the participants engaged in the narrative?
- Did each participant experience a coherent narrative?
- Did the participants feel an increasing tension in the narrative over time?
- Did the two conditions significantly differ in narrative understanding?
- Did the participants notice anything unusual about the VE?

The first part of the questionnaire is based on Busselle and Bilandzic (2009), and some items needs to be reverse coded before doing the analysis and any unusable data needs to be removed. The score for each participant should be calculated by adding the score for each item to yield a result from 9 to 27. The mean and standard deviation should then be calculated based on these collective scores for each condition (Richardson et al., 2018). Here a higher mean would point towards stronger engagement for the participants.

Individual Likert items yield ordinal data, which are non-parametric. However, when using a scale consisting of multiple items pointing to the same information, the sum of these scales can sometime be treated as parametric (Morgan, 2015; Weaver, 2019). As such, each of the scales used will be treated this way. An Anderson-Darling test will be used to determine if the rank-sum data is normally distributed, and Levene's test will be used to check the assumption of homogeneity of variance. If the data are not normally distributed, they may be log-transformed and checked again (Field and Hole, 2003). If after these tests the data are considered parametric, an independent t-test will be used. Otherwise, the Mann-Whitney U test will be used to compare the two samples to detect any significant difference. For a Mann-Whitney U test, the effect size r is calculated by using the formula $r = \frac{z}{\sqrt{N}}$. In turn, z is calculated by the formulae below (Kallner, 2018), where n_s is the number of samples from the group with the lower rank sum, n_l is the number of samples in the group with the larger rank sum, N is the total number of observations, and U_s is the test statistic from the Mann-Whitney U test:

$$\mu_s = \frac{n_s \times n_l}{2} \tag{1}$$

$$\sigma_s = \sqrt{\frac{n_s \times n_l \times (N+1)}{12}} \tag{2}$$

$$z = \frac{U_s - \mu_s}{\sigma_s} \tag{3}$$

This same procedure will also be used to compare the items in part 2 and part 4, however, for part 2 all the items will be added and the mean will then be calculated based on this value between 6 and 42 as all items point towards the same measure. This will then indicate whether the participants experienced a narrative with a rising tension curve (suspense) and also indicate whether they experienced the narrative as being coherent. In part 4, each individual set of items will be added and compared between conditions.

The data from part 3 will need more processing as there are a few more steps before the data can be analysed. The goal is to show how far the participants' understanding is from the intended understanding. All the free-form parts of each of the questions will be analysed using content analysis to code the answers into either "they got the intended narrative" or "they did not get the intended narrative". This in turn enables comparisons of the answers between the participants, both within each condition but also between conditions. These coded answers will also be used to determine how many participants had the same narrative experience and whether that experience is similar to the author's intended narrative.

The multiple-choice answers are already ranked according to understanding (see Section 6.3, Figure 4). For each question there are four possibilities; the understanding intended by the authors, two answers that are increasingly far from the intended understanding, and an option to chose none of the other options but add an option themselves. The possible answers reflect different steps within the AAD and each answer can therefore be coded from 3 for intended understanding to 0 for completely different understanding. The goal of all this, is to compare the narrative understanding between conditions and more specifically, to see whether or not significantly more participants got the right narrative understanding. The scores will therefore be added for each participant and used as a single score to tell something about the narrative understanding of each participant and these scores will then be used to compare both within conditions and between conditions using the non-parametric Mann-Whitney U test to see if any difference is significant.

3.5 Design Requirements

Before proceeding with the design of the prototype, both Sections 2 and 3 have resulted in a host of delimitations that will guide the design of the prototype. We can therefore now present an initial list of requirements for the design. Some of these are requirements based on the research, some are requirements in order to perform the evaluation, and others are requirements based on the resources available to the designers. These will then be used as the point of departure for the next section which will describe the specific design of the prototype.

Narrative Requirements

- The narrative prototype should be able to be completed within 10 minutes
- The narrative should be easy to comprehend, to keep a short Author-Audience Distance
- The narrative should be able to be expressed through a number of key points
- The narrative should be able to be told exclusively through staged areas (SAs)
- The SAs should be dynamically placed in the environment using a Space-Time Drama Manager (STDM)
- The STDM should manage the timing and spacing of SAs in order to generate suspense

Procedural System Requirements

- The PCG system should be able to generate environments suitable to the narrative
- The system should be online and not generated in advance
- Designers should be able to tweak the parameters of the PCG system to create different types of environments
- Beyond these parameters, at runtime the PCG system should use a random seed as its single dimension of control

- The PCG system should be deterministic, so that a single random seed will produce the same content each time

Technical Requirements

- The application should be able to run on a web browser
- The application should run with a smooth, consistent frame rate (approx. 30fps)
- The application should maintain acceptable levels of performance on lower-end hardware

4 Design

With some initial design requirements in place, this section will proceed to discuss the design which will be used for the narrative. Once a narrative is in place, this will inform the design of both the SAs which will be used to convey this narrative, and the PCG system which will generate the environment to house the SAs.

4.1 Narrative Design

As discussed in Section 2.2, designing a narrative requires a lot of consideration and careful planning to create a good story that is interesting and engaging, with specific considerations being required for creating an embedded narrative. Since a part of the methodology for this thesis is to to evaluate the player's engagement, we will also design the narrative with the goal of creating an engaging experience. Section 2.2.5 explained how the spacing and timing of events can create suspense which, in turn can engage the receiver. As mentioned in Sections 2.4.1 and 3.5, the STDM can be used to maintain tension, through dynamic timing and spacing, in addition to maintaining narrative understanding by ensuring that players encounter SAs in the correct sequence.

As we want to create a narrative that has a rising tension curve to create narrative engagement, we still need to define a beginning and end to the narrative. Within this span there needs to be a development for the character and the player. The player will between these two points come by at least five different SAs that act as the narrative driver for the player.

4.1.1 Designing for Embedded Narratives

The ES is key as it will be the only source of storytelling in this prototype and so it has to be carefully designed to give the right information to the receiver and in the right order. There should not be any non-diegetic explanation of narrative details within the game. With the exception of an introductory text, the SAs will be the sole narration. The design of the SAs has to consider not just presenting the narrative information, but also that the player should come across them in the environment. As such they should have increased narrative importance, and stand out from their surroundings.

By designing the game in this way we rely on the player's ability to create a narrative themselves from the presented information, what Jenkins (2004) would classify as an emergent narrative. We however argue that this is still just an embedded narrative as a narrative is defined in advanced and merely emerges with the player. Like discussed in Section 2.3.2 a narrative like this should be quite didascalic to keep a short AAD to ensure there is not too many possible explanations to the presented narrative information.

4.1.2 Backstory

To drive the design of the specific narrative it is important to think beyond what the user will experience. The narrative will take place in its own diegesis and to guide the design of the specific narrative, an important part is therefore to define the diegesis to guide the specific narrative that is being conveyed to the player. A comprehensive backstory or description of the diegesis is therefore needed. This backstory can be used as a guide to compare the players experience to the intended narrative. A detailed backstory about what diegesis the specific narrative is within and what happened prior to the start of game is therefore defined and can be found in Appendix E. The following section will present the narrative which has been synthesised from the backstory.

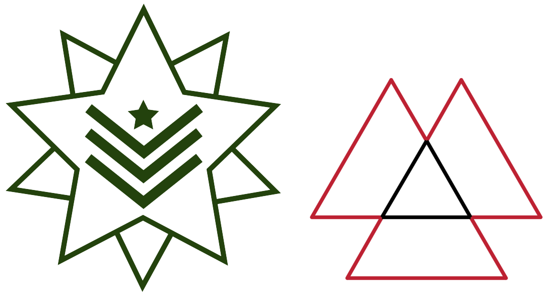
4.1.3 Specific Narrative

The main character that the player controls is a fisherman who arrives at an uncharted island that he has never seen before. This ensures some mystery and suspense from the beginning. He is driven by his desire to figure out if there is something on the island and he starts uncovering multiple structures of different kind around the place. The full synopsis reads as follows:

> **Synopsis** A fisherman comes across an undiscovered island. As he gets to shore and starts to explore the area he finds that it was once used by the government as a military area. As he walks deeper into the forest he gets a creeping feeling that someone or something is watching him. He gets to the base of a tall radio tower, and finds that it is totally abandoned. Just as he gets some relief, he sees some thick smoke rising in the distance. He quickly moves towards it, now with an even more creeping feeling. He sees occult carvings and tools at what appears to a ritual site in the forest and as he approaches the smoke, he sees that it comes from his own boat, and that it is shrouded in flames. He hears whispers and what sounds like chanting coming from all around him.

The continuous unveiling of different information pointing in different directions as for who has been on the island, and whether someone is still there, keeps the player on their toes. When he sees smoke coming from his own boat these suspicions starts to materialise. To guide the player in just the right amount, a narrative introduction is made to introduce them to the setting and the player's reason to be there before the game begins. It also works for giving the player a feel from the start that something is off with this island and they have to keep an eye out for things. By doing so we also indirectly tell them that whenever they encounter something out of place, like an SA, they should stay alert as to what that might tell them about the island.

> **Introduction** As a fisherman you know your way around the seas of your country, at least you thought so. One day, you veer away



(a) *(b)*

Figure 13: *Symbols for representing the military and occult presence on the island. Both are made specifically for this game*

> from your normal area and see a small island on the horizon. It does not show up on any of your sea maps and as you approach you get more and more curious about why you have never heard about this island. You board your dinghy and as you circle the island and approach the shore you see a tall tower that might reveal more about the mysteries of this island...

The narrative revolves around the interplay between a military presence and the fact that the island seems deserted but at the same time has clues indicating another presence that is more primitive and has an "occult" symbol that is painted or carved on many of their belongings. By utilising symbols to represent the military and the cult respectively we can better divide each SA as to what they represent and indirectly tell the player that there has been more than one group on this island. In Figure 13 these symbols can be seen where one has a very classic military look with stars and insignia in green and the occult one is centred around symmetry and red and black colours.

4.1.4 Key Narrative Information

From the narrative we specify three pieces of key information that the players should be able to get from encountering all the SAs and in the right order. The NEM tool (see Section 2.5) will then be used to find out if the players got these pieces of information correct. The Key information will be:

- There were military personnel at the island at some point
- Something bad happened to them
- There is now a cult on the island

Info \ Time	Staged Area #1	Staged Area #2	Wienie #1 Staged Area #3	Staged Area #4	Wienie #2 Staged Area #5
It was a military area		Supplies are found and there are a military logo on everything	There is a base with supplies and protection of the radio tower		
Something bad happened		There are ammo and much of it is used	Everything is empty and more used ammo is found		Your boat is found burning
A cult is now on the island	There is primitive tools and tents with symbols			A ritual ground with the same symbol and a dagger in the middle	Your boat is found burning

Figure 14: *Overview of the staged areas and how they relate to the information pieces the player are ought to grasp in the narrative*

4.2 Staged Areas

For each SA there will be some props that relate to the narrative and that will inform of the player of what might have happened on the island. There will be five staged areas that the player will come across in order. Each area is designed to reveal some information about the narrative and in the end, the player will hopefully be able to create a coherent narrative.

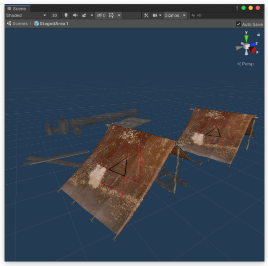
Figure 14 gives an overview of the SAs and what props will be where. For each SA it is possible to see which piece of information that particular prop is designed to convey. This is all designed to repeat important information multiple times to ensure the players are less likely to miss it. This means that all the props have to have a meaning for the overall narrative and bring new information to the table.

To guide the implementation of the narrative in the game a script is made which includes all the steps the player will go through, how they are introduced and start the game, and how each SA is presented to them. In Appendix F the complete script for the narrative can be seen.

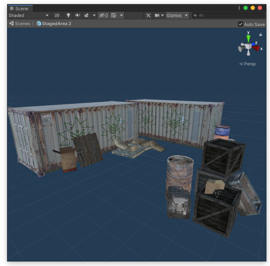
4.2.1 Design of Staged Areas

According to the overview in Figure 14 each staged area has a function in terms of the narrative. Below all of them are described in terms of their function and in Figure 15 the final design for all of them can be seen. All the assets used have been found on the Unity Asset Store except the radio tower which is created specifically for this game.

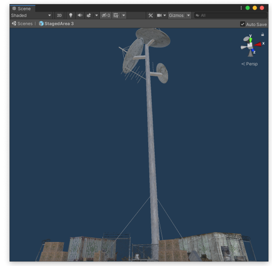
SA1 will be a primitive campground that should reflect the lifestyle of the cult and look like a small outpost for them. The player might think there is some natives on the island that lives in the primitive tents but the addition of



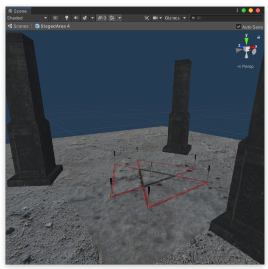
(a) SA #1



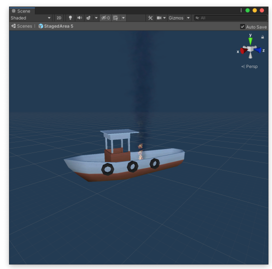
(b) SA #2



(c) SA #3



(d) SA #4



(e) SA #5

Figure 15: *All the staged areas made for the game*

the cult symbol at certain props should provide an early glimpse of something more than just natives.

SA2 should be a storage unit for the former military operations on the island and have containers and other stuff left behind including ammo to hint at this being a military operation. This will also be underlined by the inclusion of the military symbol on the props. This SA will be a contrast to the primitive look of the first SA and be the first real hint of military presence.

SA3 will be a tall radio tower at the highest point on the island complete with containers and bunkers where the military personnel and researchers would sleep and work. Here player really get to see how this has been a big military operation at some point but also that it seems abandoned in a hurry and that something might have happened.

SA4 is gonna be the first confirmation of something occult happening on the island. It should be a ritual site complete with carved pillars, carvings, and a dagger stuck in the ground. There will also be some torches that are smoking to indicate that a ritual has just taken place and someone might still be on the island.

SA5 should be the fisherman's boat that is found burning at the shore to confirm that someone else is on the island with you and that they do not have friendly intentions.

4.3 Player Guidance

As described in Section 2.4.2 it can serve as a great tool for the STDM to know within reason where the player is heading. Here the principle of wienies can be an especially powerful diegetic way point also in open world games. We therefore want to use wienies to both elevate the narrative by giving the player more of a purpose from the start and to improve the accuracy of the STDM when spawning the SAs. If we introduce the player to a wienie and indirectly tell them to move towards that, we can delimit the most likely points for spawning to an area between the player and the wienie. When deciding where to spawn an SA it is important that the point is in front or almost in front of the player and on their trajectory but if they roam freely, they might change direction often and not have a specific trajectory. By introducing a wienie we can assume they will be moving towards it and lets us assume their trajectory.

The radio tower from SA3 is an excellent candidate for a wienie and therefore the player is introduced to the radio tower from the beginning (see Section 4.1.3). This also means that SA3 will need to be spawned before the game starts but as it should be on the highest point on the map, it will be fixed in space, unlike the other SAs. Instead, SA3 should now be used to find the starting point for the player. This point should be along the shore as the player is supposed to start the game right as they get to shore in the island. This spawn should be decided based on being a certain distance from the radio tower but at the same time have a line of sight directly to the radio tower so the player can see it right as the game starts.

As the player gets to the radio tower they would only have reached halfway through the game and as they are now at the wienie we will no longer have a guide for the last two SAs. However, as the boat at SA5 is supposed to be on fire, we will add smoke coming from the fire to act as a new wienie for the rest of the game. This wienie should of course only be visible as the player gets to SA3 to ensure they will not wonder about the smoke and go for that before having visited the other SAs. Both the wienies can be seen in Figure 16

4.4 Audio

The last important aspect of the narrative part of the prototype is the audio used for the game. The audio design will be simple with diegetic sounds that would be expected outside on an island. Therefore nature sounds will be audible throughout the game (klankbeeld, 2013). In addition to the diegetic sounds, we will add some ambience to enhance the audio design and give the player a creeping feeling that all is not as it seems (Kinoton, 2018). When the player gets near the burning boat, another darker ambience should be heard increasing the aforementioned feeling (RoganMcDougald, 2015). As the player gets really close, they should also start hearing some chanting closing in on them to further

(a) First wienie: radio tower *(b) Second wienie: smoke*

Figure 16: *The two wienies used in the prototype*

add to the occult vibe and increase the suspense for the player (djgriffin, 2006; Surly, 2010).

With a design in place for the key elements of the narrative, the next step is to start designing constraints for the PCG system.

4.5 Design of Procedural System

Although it is often considered to be the domain of engineers, PCG is design. A programmer with no design sensibility could not write a good algorithm to generate levels, because they don't know how to evaluate the output of the system (Pittman, 2015). Therefore this section will outline multiple design considerations that are key when designing the PCG terrain system.

4.5.1 Terrain

As mentioned in Section 2.1.4, the PCG system which will be developed for generating the game environment will be a top-down, parametric system. This means that the system, with the basis of generation algorithms such as fBm and other constraints, should be able to generate environments appropriate to the goals of the designer. That is, the designer defines what the result should be, and the generator will produce content meeting those criteria. As such this section will consider what possible constraints could be used, and what the desired output of the PCG system is.

One of the first constraints to think about is the size of the playable area, and its boundaries. PCG environments offer an interesting possibility of an infinitely large playable space which can be generated at runtime. However, the specific narrative designed for our prototype takes place on an island, and as such this is the first constraint for the PCG system.

In addition to the size of the area, the topography of the terrain may have certain constraints. In the case of this prototype, there is a lot of freedom for the generator in terms of topography. However, the narrative calls for a Radio Tower atop the highest point of the island. As such, another constraint for the system is that there must be at least one hill, if not more.

The generator should be able to primarily produce islands. These should be large enough, with enough variation in topography that players will not have a clear view of the entire coast from their starting position, and that SAs may be placed along the player's expected trajectory, but not directly in their line of sight.

One of the goals of the terrain generation will also be to generate terrains that are similar to those known from the real world. The topography should in other words be realistic to the player to make them feel they are in a place in the real world and not in a game.

To inform the rather fluffy term of creating "realistic" terrains, we will be carrying out an evaluation of different kinds of terrains to deduct what are natural properties and to see if people are able to differentiate real and computer generated topography from each other.

4.5.2 Evaluation of Terrains

An evaluation was performed on an early prototype of the PCG terrain generator, in order to assess firstly if the generated landscapes were deemed "realistic", and secondly what kind of aspects of a terrain's topography do users look at when considering what is realistic. The complete methods used for this evaluation can be seen in Section 3.2.

This evaluation consisted of six different terrains, three of which were generated from real-world data, and three generated with different parameters to yield different results. The three generated ones were all generated using a combination of Perlin noise, Diamond-Square, and Voronoi tessellation algorithms, which are all explained in detail in Section 5.1. Each terrain was presented as images from three different points of view and so 18 images were shown the participants.

In Figure 17 some of the images can be seen and in the Electronic Appendices the questionnaire containing all 18 images can be seen. The order of the images was made random to avoid biases and in addition to the 18 images there was a demographics section concerning participants' experience with playing games, creating games, using image and video editing software, and using 3D modelling software. All this is due to a bias found in the pilot test showing that people being used to these programs might be more familiar with the use of some of the techniques like Perlin noise, and may be able to recognise these patterns.

Test Specifics The evaluation was carried out from March 25$^{\text{th}}$ to March 27$^{\text{th}}$ 2020 online via a questionnaire. Within this time 47 people participated ($N = 47$), 33 males (70 %) and 14 females (30 %) with an average age of 32 years old ($\mu = 32, \sigma = 9.8$). Each participant was asked whether a certain

		TD	ISO	FP
Computer Generated	1	0.71	0.51	0.22
	2	0.82	0.92	0.86
	3	0.49	0.43	0.41
Real World	1	0.59	0.63	0.27
	2	0.55	0.63	0.63
	3	0.71	0.82	0.49

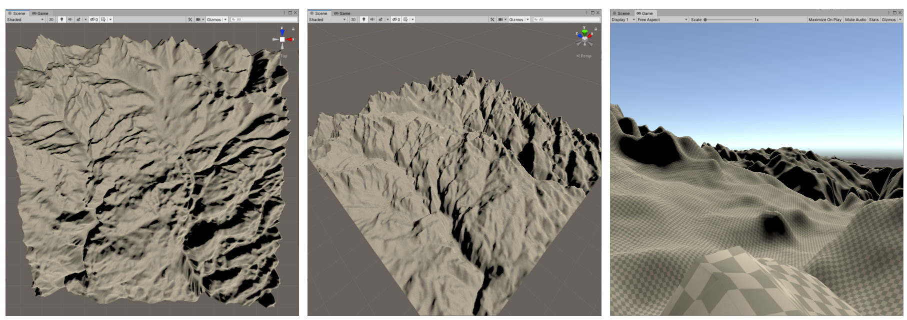
Table 1: *Results showing the average of participants classifying the terrain correctly for each of the 18 images. There were 3 computer generated terrains and 3 based of real-world heightmaps each of which had 3 perspectives: viewed from the top down (TD), viewed from an angle in isometric perspective (ISO), and viewed from a first person point of view at terrain level (FP)*

image of a generated terrain, to the best of their abilities was generated randomly or based directly of real-world data. This was a simple yes/no question and similarly, the data was stored as either a 0 for wrong answer or 1 for a right answer. In Table 1 the results for each of the 18 images can be seen. Noteworthy here is to compare all the 9 images from real-world data to the 9 others to see if a difference in correct classifications can be found. Here we get that on average, participants did not get more corrects for the real world data images ($\mu = 0.591, \sigma = 0.49$) compared to the computer generated data ($\mu = 0.596, \sigma = 0.49$). When comparing the three different views of the terrains there were more who were able to recognise right from wrong when it was viewed from a top-down ($\mu = 0.646, \sigma = 0.479$) or isometric perspective ($\mu = 0.656, \sigma = 0.476$) compared to viewing it from a first person perspective on the terrain ($\mu = 0.48, \sigma = 0.5$). However, the large standard deviation also shows that participants were not in line when it comes to this. For each image the participants were also asked to elaborate on what made them answer the way they did. These statements were all coded using open coding (Bjørner, 2015) and each statement was coded separately by two facilitators. The results of the coding can be seen in table 2 and both complete qualitative and quantitative data can be seen in the Electronic Appendices.

Results In general, participants seemed to have a difficult time accurately classifying terrains. However, some of the generated terrains were much more easily identified than others, suggesting that designers need to take great care when tweaking parameters, to maintain a realistic topography. From the qualitative results we can see that people used statements about organic shapes three times more often for the real-world terrains and used less organic wordings for the generated terrains which indicates that the generated terrains are less organic in the shape and that this is something people relate to being natural. Other results are more conflicting as they for example indicated that the computer generated ones were more random but at the same time saw more of a pattern in these.

	Repetitive	Randomness	Depth	Organic	Rough	Unique	Believable	Fake	Assumption
Computer Generated	16	17	6	4	33	4	27	36	12
Real World	5	8	11	15	9	7	58	23	7

Table 2: *The results of the open coding, yielding 9 unique codes, and how many statements/comments belonged to each code divided into whether the terrain was based off computer generated data or real-world data*



(a) *Top down view* ***(b)*** *Isometric view* ***(c)*** *First person view*

Figure 17: *Three different images from different angles of the same terrain. The terrain here is created directly from heightmaps of the Yosemite National Park in California*

Discussion of Results There are of course some biases to this method as, for example from the first person perpective, it can be subjectively easier or harder to recognise the distinct features based on the specific viewpoint that was chosen for the image. For all first person images, the view was selected in order to get as good a view of as much of the terrain as possible. The results for this could of course indicate that it is more difficult to discern specific features when being closer compared to having an overview from afar. However, as players will view the terrain from this perspective in-game, this is the most important one to consider. The results show that creating "realistic" terrains are not an exact science and people in general have a hard time seeing a difference between one directly generated from real-world data and one procedurally generated based on certain parameters. In Figure 17 three examples of the test images can be seen, all of the same terrain but from different angles.

4.5.3 Textures

Having a realistic topography is not enough to make a realistic experience. The topography makes up one part of the terrain. The other part is the textures.

In the images of the terrains in Figure 17 the terrain is not yet painted with textures that would make the otherwise boring and unrealistic terrain come to life. It is therefore key to apply the right textures at the right places to complete a realistic terrain.

A steep high part of the topography would probably be a mountain and would look like rock. A low area around a lake will probably be lush and have grass or dirt around it. It will catch the attention of the players if these things are not like what is expected from the real world. For this prototype it will be an island which means the lowest areas close to the water should be beach or have dirt, rocks and mud. There should be some hills or mountains which will have rocky cliff shown, but only on steep patches. If the mountain sides are flatter they will be more lush with grass and plants growing there.

All this should be incorporated in the PCG system so that when the terrain is generated it is automatically painted with the right textures at the right places.

4.5.4 Foliage

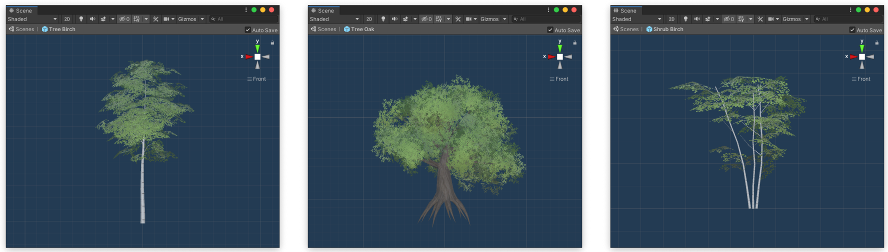
In addition to generating topography and painting textures, the PCG system should also populate the environment with trees and other foliage. As with the textures, placing this foliage is dependent on how it would grow naturally, at least if realism is the goal.

A small number of assets should be passed to the generator, which can then be placed in the environment based on some rules similar to the ones in the previous section. Trees and bushes in general do not grow on steep surfaces like a cliff or rock. It is therefore important that the slope of the terrain is a parameter and in the same way, certain trees might not grow above a certain height. Some foliage will also grow closer to the ocean, and different foliage will grow in clumps of different densities.

The specific trees and textures used should reflect the same graphical style. As this will be a browser game, and the PCG system will take some processing power already, we will not go for a photo-realistic style to the assets. Instead we will choose something that is more stylised with a lower polygon count for better performance (see Figure 18). The assets that will be used are imported from the Unity Asset store (Visual Design Cafe, 2019).

This particular package offers 15 different trees, 5 shrubs and bushes, 10 different weeds, and 4 ground textures. From this we should use the textures for the terrain and different trees, bushes and weeds for the foliage creating a coherent visual experience for the players.

With this in place we can now create a PCG system that makes realistic terrains that are painted with the right textures in the most natural places and has a variety of different foliage to inhabit it. The VEs that this system create now needs to be used as the backdrop for a narrative that uses the VE to tell an embedded story.



(a) ***(b)*** ***(c)***

Figure 18: *A few of the trees used in the prototype all from the same asset package keeping a cohesive style for the prototype (Visual Design Cafe, 2019)*

4.6 Design Conclusion

When using PCG to generate environments, particularly in this case when they are intended for use with ES, there are a number of elements that require careful consideration, to enable it to generate suitable content.It needs to be able to not just a create terrain, but paint textures in the right way to make as realistic a VE as possible. In addition, it also needs to, in our case, generate foliage to populate the terrain. Furthermore, the generator needs to be able to include specific features that support the designer's vision - in this case it must ensure that the output takes the form of an island, and that there are a sufficient number of hills.

Being able to embed a narrative in the VE at the same time does not make it any less challenging. By developing a full narrative and then afterwards splitting it into a design of specific SAs, we can better keep a coherent narrative which will be introduced to the player using the STDM dynamically but in the right order and with tension intact. Furthermore developing this narrative before the PCG system helps define the constraints needed to ensure the right environment can be produced.

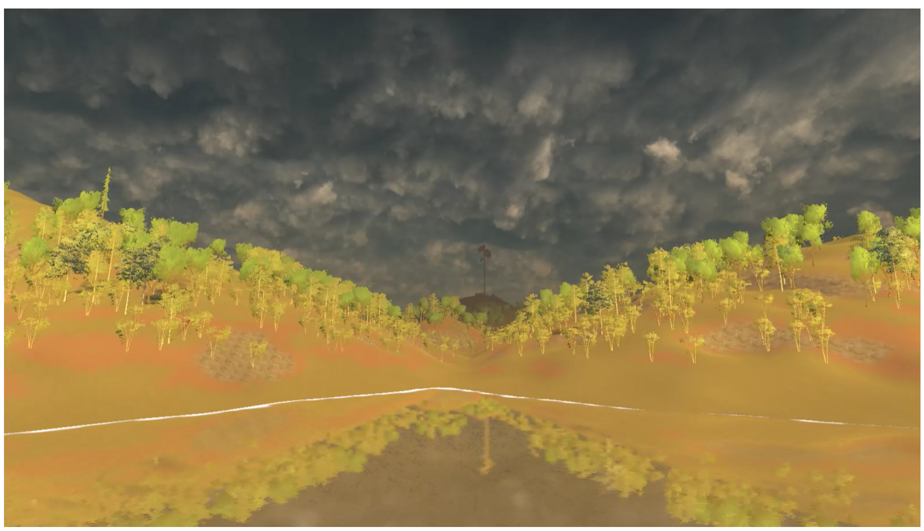


Figure 19: *Final design of the prototype with terrain, textures, foliage, and wienie*

5 Implementation

This section details the development of the PCG terrain system, the STDM, and the final prototype which is used for the evaluation as detailed in Sections 3 and 6. The implementation took a somewhat exploratory, iterative approach. Early prototypes tried to use Unreal Engine 4 due to it's powerful landscape tools. However these were revealed to powerful for designers who are hand-crafting terrains, and less accessible for PCG systems. This was considered for combination with a Generative Adversarial Network (GAN) to make realistic heightmaps, though this was not pursued as it would not have been suited to online generation. A custom-mesh terrain was first attempted in UE4, and then after initial difficulties ported over to Unity. Appendix B provides more details about these approaches, particularly the Unity prototype. In the end it was decided to keep working with Unity, but to utilise the terrain tools in favour of a custom solution. This section will detail some of the key elements of the final prototype which was developed.

5.1 Terrain Generation

The final version of this prototype was developed with Unity. Heightmaps were generated using fBm algorithms, and applied to the terrain object, using Unity's terrain tools. This allowed the developers to make use of much of the existing terrain framework with more advanced Level of Detail (LOD) system, support for trees, details, splatmap textures, and wind among others, rather than having to build an entire system from scratch.

5.1.1 Basic Topography

The basis of the system was to use fBm, as described in Section 2.1.5, in order to make a terrain with natural-looking topography and variations.

Perlin Noise Perlin noise is certainly more natural than random noise, and fBm certainly more than a single layer of Perlin. However, a single iteration of fBm is still somewhat lacking. As such, the Perlin noise heightmap generator was expanded to allow multiple sets of Perlin fBm to be layered on top of each other, each with different parameters such as scale, offset, octaves, persistence and lacunarity. This resulted in the designers being able to define multiple layers of features and produce more varied terrains with the generator than possible with a single pass of Perlin noise.

Midpoint Displacement In addition to the Perlin noise, a Diamond-Square algorithm was implemented, as detailed in Section 2.1.5 and Figure 6. While this method is also fBm and can produce similar noise maps to using octaves of Perlin noise, it can also produce vastly different landscapes, such as more "jagged" surfaces. Using this method in addition to Perlin noise fBm added another layer of control for the designers.

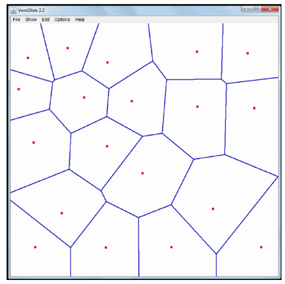


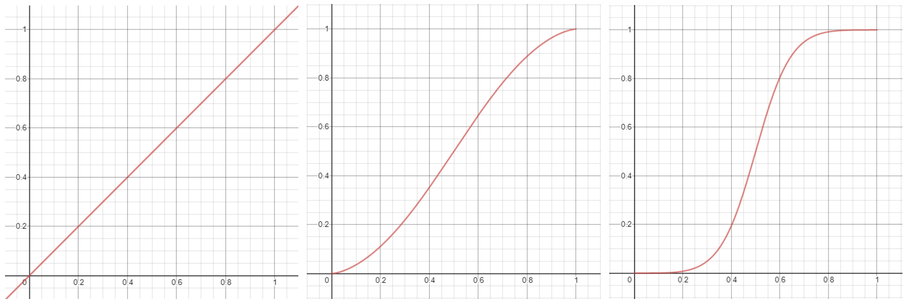
Figure 20: *An example of a Voronoi diagram (Shouraboura and Bleher, 2011)*

5.1.2 Fulfilling Designer Constraints

In order to increase the top-down control of the designers to better facilitate the design as presented in Section 4.5.1, two more methods were added to the heightmap generation. Firstly, Voronoi tessellation was used to generate mountains and hills. Secondly, a falloff map was added to reduce the edges of the heightmap, forcing an island-like terrain to be generated.

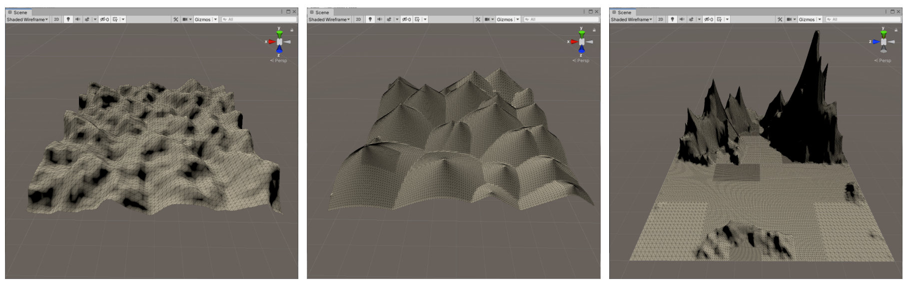
Voronoi Tessellation In order to ensure that there were a number of peaks for either hills or mountains, Voronoi tessellation was added. Voronoi tessellation is the division of a plane into segments, where the borders are equidistant between points. In practice this means that areas within these cells are closer to that cell's point than any other (illustrated in Figure 20). This technique is often used in shaders and textures, for effects such as cellular noise, for example in a leaf or an insect's wing, or cracked rocky surfaces and lava (Gonzalez Vivo and Lowe, n.d., Chapter 12). In terms of generating mountains, each point can be defined as a "peak", and given a new height value to reflect this. The heightmap values within each segment can be interpolated based on the distance from the peak. If the current value in the heightmap is less than the interpolated value, then this means that the point is closer to the current peak than any others. If the value is higher, that means there is another peak which is closer, and it should retain the old value. In the most simple model, this is linear interpolation, but a non-linear interpolation can cause this to generate sharper peaks, or smoother rolling hills. A falloff value is used to control the steepness of the slope of the hills, and for non-linear interpolations a "dropoff" value is used to control the curve of the slope. If a small number of peaks are defined, this will simply result in a number of separate hills or mountains. However if the peaks are somewhat densely packed, this will produce boundaries and valleys between them. This method will not generate natural terrains by itself however. Instead, this should be used to insert some low-level features into the heightmap, before making the terrains more natural with the fBm procedures described above.

Falloff Map In order to make the generator create an island, a falloff map is generated and subtracted from the heightmap. This is a simple gradient texture,



(a) $a = 1$ ***(b)*** $a = 1.5$ ***(c)*** $a = 3.5$

Figure 21: *Falloff curves produced with different values of a*



(a) ***(b)*** ***(c)***

Figure 22: *Sample terrains generated with (a) Perlin noise, (b) Voronoi tessellation, and (c) Midpoint Displacement with Diamond-Square*

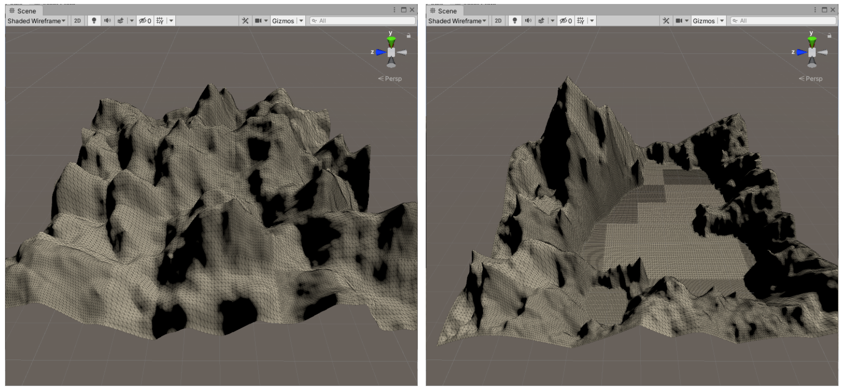
with high values at the edge and approaching 0 in the middle. The gradient was non-linear, to ensure that the size of the island could be controlled by the designer. The gradient was calculated with the following formula:

$$f(x) = \frac{x^a}{x^a + (1-x)^a}$$

This produces a non-linear dropoff, where values of a above 1 create an S-curve, with increasing values making the curve steeper, as shown in Figure 21.

Erosion Simulation After evaluating the perceived realism of the generated landscapes, (as described in Section 4.5.2), it was decided to enhance the landscapes with erosion simulation. Five different types of erosion were simulated: rain, river, landslide, tidal, and wind. Each of these were implemented as simple, low-cost abstractions to achieve a desired aesthetic, rather than realistic, detailed erosion simulations.

Rain Erosion by rain is the simplest of the erosion simulations. A number of random points are selected across the terrain based on the number of



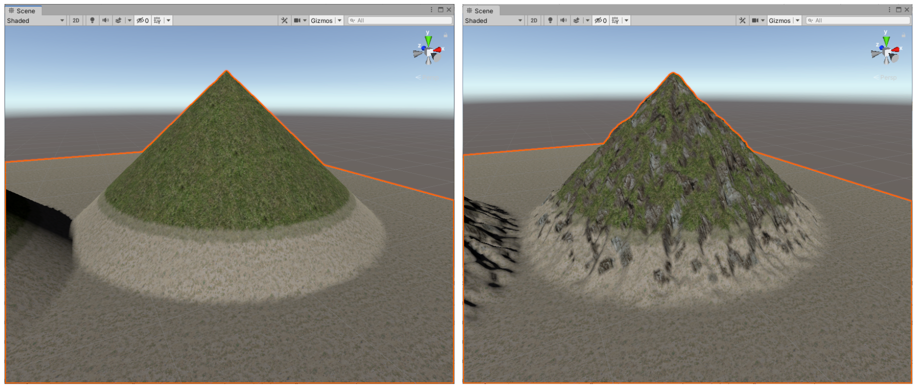
(a) *(b)*

Figure 23: *Combining methods produced some much more natural terrains*

droplets. Each of these points has its height reduced by a value determined by `erosionStrength`.

River Like the rain erosion, river erosion begins with randomly selecting a number of source points. Each of these "springs" has a solubility value, as a representation of how much earth the water can move before becoming saturated. From the spring, a path is traced downhill using the `RunRiver` function, which simulates a river eroding the earth while gathering and depositing sediment. This function can be seen in C#-style pseudo code below. The output of `RunRiver` is a 2D float array with the same dimensions as the heightmap. This is an erosion map, which is subtracted from the height map when the algorithm is completed.

```
while (erosionMap[currentPosition] > 0)
{
    List<Vector2> neighbors = GetNeighbors();
    neighbors.Shuffle();
    bool foundLower = false;
    foreach (Vector2 neighbor in neighbors)
    {
        if (heightMap[neighbor] < heightMap[currentPosition])
        {
            erosionMap[neighbor] =
                erosionMap[currentPosition]
                 - riverSolubility;
            currentPosition = neighbor;
            foundLower = true;
            break;
        }
    }
    if (!foundLower)
    {
```



(a) *(b)*

Figure 24: *River Erosion applied with many paths and a heavy erosion strength*

```
            erosionMap[currentPosition] -= riverSolubility;
    }
}

return erosionMap;
```

Listing 1: *Psuedocode of RunRiver() which performs a downhill random walk to populate an erosion map*

Despite the label, this type of erosion will not place rivers on the map. As such it is not concerned with other aspects of a river such as converging with other paths, or reaching the shore. Rather, this erosion simulates deep vertical ridges in slopes caused by water flowing down them.

Landslide Landslides are also simulated, taking steep and relatively linear sections of terrain and moving some of the mass from the higher area to the lower area. This algorithm simply loops through the heightmap and checks each position against its neighbours. If the current point is higher than the neighbour plus `erosionSensitivity`, then the current point is reduced by `erosionStrength` and the neighbour is increased by that much. This turns smooth, steep hills into rougher cliffs with deposits at the bottom, as illustrated by the graph in Figure 25.

Tidal Tidal erosion simulates the formation of beaches and cliffs by the ocean. This erosion is simply implemented by setting a `waterHeight` float variable. The shore line is found by searching for neighbouring pixels on the heightmap where one pixel is equal to or lower than `waterHeight` and the neighbour is above it. All pixels in a radius around this point are simply set to `waterHeight`, before smoothing the slope with a simple mean filter.

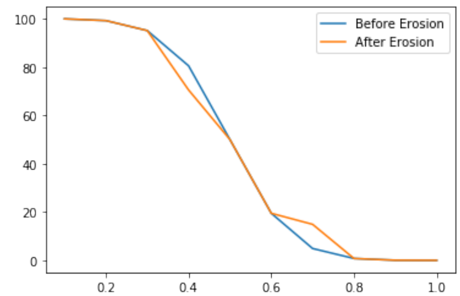
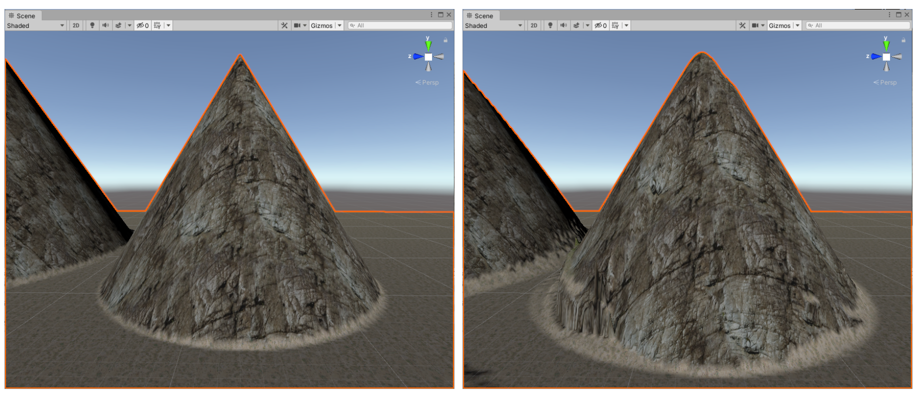
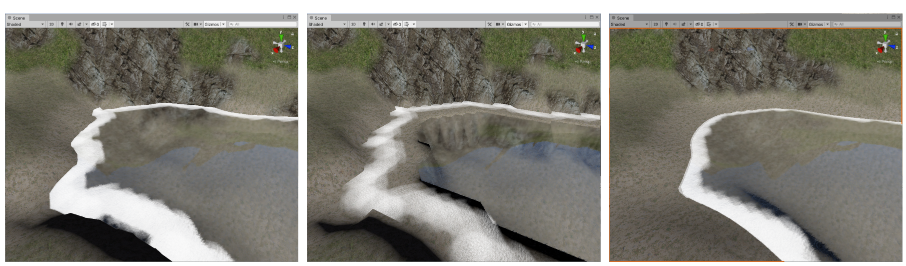


Figure 25: *Illustration of the effect of landslide erosion on a slope*



(a) ***(b)***

Figure 26: *The effects of landslide erosion can be subtle. Notice how the sharp peak on the left is smoothed out, and the slope becomes more uneven at the top (missing mass) and the bottom (added mass)*



(a) ***(b)*** ***(c)***

Figure 27: *The effects of tidal erosion. (a) before erosion. (b) after erosion. (c) after mean filter. Note that the white foam is a dynamically created mesh which is regenerated after the erosion*

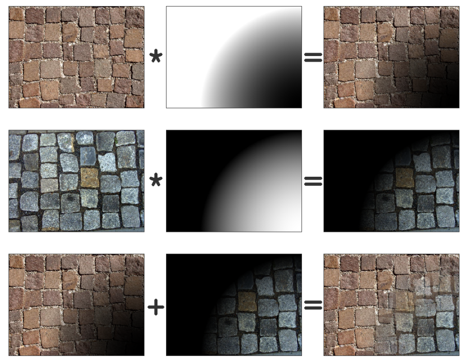


Figure 28: *An example of how texture splatting with alpha masks work (Wikimedia, 2008)*

Wind The final type of erosion is simulating sediment being moved and deposited across surfaces by wind. This was done by "digging" and "piling" using a simple sine wave. Futhermore, the dig and pile coordinates were modulated with Perlin noise, to create a more natural spread of these trenches, rather than being perfectly aligned in a grid. This produces a natural variation to the terrain resembling a wavy pattern seen on beaches, for example.

Texturing As visible in Figure 24, the terrain textures in this prototype are reactive to slope, in addition to height bands (before erosion in 24a, the smooth slope is covered in grass. After the erosion in 24b, steep surfaces are exposed, which are rocky and no grass grows there). While all of the previous techniques for terrain generation could have been used on the custom mesh from the previous prototype (see Appendix B for more details) with relative ease, it is at this stage of painting and decorating the terrain that the advantages of using Unity's terrain tools becomes apparent. The *terrainLayers* used in this tool allow for texture splatting, where textures can be blended by alpha maps (illustrated in Figure 28). These alpha maps, just like the heightmaps, can be defined through code. As such, custom terrain layers can be generated based not only on height as in the prototype described in Appendix B, but also by the slope in the corresponding area on the heightmap. This could be further expanded to take into account other factors such as proximity to the shore or vegetation, or other maps that could define biomes, mineral richness, wind exposure, or any other number of factors which could affect the terrain's texture. For the purpose of this prototype, it was decided that height and slope would be sufficient.

Trees & Details In addition to texture splatting, one of the large advantages of using the terrain tools was the trees and details system. This allows a number of trees and bushes to be defined, in addition to other details such as grasses, flowers, and rocks. These tools are designed for designers to paint them onto the terrain, but can also be applied through scripts. Importantly, this system allows automated LODs and billboards switching (as long as the assets provides them). This allowed the terrains to be populated with a large amount of trees visible from long distances without a performance hit. Similarly to the splatmaps for terrains, placement of trees and details can be defined by a detail map. Once more, height and slope are used to determine suitable locations for each tree prototype.

5.2 Space-Time Drama Manager

With the terrain generation PCG system in place, the next stage of implementation was the development of an STDM. As mentioned in Section 2.4.1, the "event capsules" from the STDM framework (Schønau-Fog, 2015) are equivalent to SAs, which are described in Section 2.3.3. In this thesis the STDM will always be about SAs and as such this term will also be used here even thought the correct term form the original framework is event capsules.

The STDM can be considered to be another PCG system, in addition to the terrain generator. For the purpose of this implementation, this would be an online system, which generates necessary materials, i.e. the space-time coordinates of the SAs. Considering the PCG methods described in Section 2.1.3 and the outline of the framework presented in Section 2.4.1, this could be made as a constraint-based PCG system. The system should be given constraints about what the necessary conditions are for both a place and time to be suitable candidates for the SA, and will generate a series of suitable points in space-time. There are three levels of constraints for the SAs: fixed-space constraints, relative-space constraints, and time constraints. Fixed-space constraints are specified on a per-SA basis. This generally refers to environmental descriptors, such as maximum and minimum height and slope for the SA. Relative-space constraints are more dynamic, and relates to where the SA should appear in relation to the player's position, the currently active wienie, or both (see Section 4.3). These are calculated and assessed at runtime, provided an SA has met the time constraints. Time constraints concern the order in which SAs are encountered, and the time elapsed between SAs. These constraints are defined in the STDM as they are relational between SAs.

Fixed-Space Constraints Before the beginning of the main loop, the STDM performs some initial steps, assessing the fixed-space constraints of each SA:

1. Find positions for static and key locations, such as wienies and player spawn
2. Find candidate positions for each SA. Each SA should store a reference to this list of candidate positions

The fixed-space constraints for each SA are summarise in Table 3

Time Constraints Once the main loop has begun, the STDM evaluates the time constraints of the SAs. At the most basic level, this involves holding a

	Min. Height	Max. Height	Min. Slope	Max. Slope	Area X	Area Y
SA #1	0.03	0.2	0	10	15	15
SA #2	0.03	0.18	0	15	20	20
SA #3	0.2	1	0	15	30	30
SA #4	0.03	0.226	0	20	20	20
SA #5	0.03	0.03	0	5	10	10

Table 3: *All the requirements for the five staged areas in the prototype*

sequential list of SAs, and a reference to one of these as the target SA. When the target has been encountered, the next SA is made the target. This creates a simple sequence between events. This is built upon with an elapsed time measure, where the next SA will not be assigned until a certain amount of time has passed.

Dynamic-Space Constraints In order to ensure the player encounters the SA, once the time constraints have been met, then the system evaluates the dynamic-space constraints. This involves finding positions from the list of candidates (i.e. that have already passed the fixed-space constraints), and trying to find suitable SA locations based on where the player is expected to go. In this prototype, wienies are used to guide the player, and it is assumed that the players will try to navigate towards them. As such, a the dynamic-space constraints limit possible areas to those which are between the player and the current wienie.

Re-assessing Spawned SAs If all of the above constraints have been met, the STDM spawns the SA at a suitable space-time coordinate. However it has not finished its work, and needs to continually re-assess the dynamic constraints. It is possible that an SA may be spawned, and the player may pass by without noticing, or change direction completely. The STDM will check parameters such as the difference between the player-wienie distance and the SA-wienie distance to see if the player has passed by it. If this is the case, it will restart the search for suitable coordinates based on the dynamic-space constraints with the current player position in mind.

5.3 Linking Systems with an Observer Pattern

An observer pattern was implemented in Unity in order to facilitate communication between systems without creating too many rigid dependencies. This pattern was implemented using *Scriptable Objects*, based on a talk about game architecture by Hipple (2017). This system consists of a `GameEvent` scriptable object, and a `GameEventListener` monobehaviour script.

In the observer pattern, a subject (the `GameEvent`) maintains a list of observers (the `GameEventListeners`). The subject has functions for registering and un-resgistering observers, which the observers call when they are enabled and disabled. The key behaviour of the subject is the `void Raise()` method, which calls the response function, `void OnEventRaised()` of each observer. The list of observers is iterated through in reverse order, in case the response involves an observer removing itself from the list.

The `GameEventListener` has a reference to a `GameEvent`, which it will subscribe to when enabled (and unsubscribe from when disabled). The core functionality of this system is the integration with Unity's Event system. Each observer has a `UnityEvent` variable, which can contain any number of references to scene objects, or calls to other methods on other scripts. This response is invoked in the `OnEventRaised()` method.

```
using System.Collections.Generic;
using UnityEngine;

[CreateAssetMenu]
public class GameEvent : ScriptableObject
{
    private List<GameEventListener> listeners = new List<
    GameEventListener>();

    public void Raise()
    {
        for (int i = listeners.Count - 1; i >= 0; i--)
        {
            listeners[i].OnEventRaised();
        }
    }
    public void RegisterListener(GameEventListener listener)
    {
        listeners.Add(listener);
    }
    public void UnregisterListener(GameEventListener listener)
    {
        listeners.Remove(listener);
    }
}
```

Listing 2: *Game Event Scriptable Object*

```
using UnityEngine;
using UnityEngine.Events;

public class GameEventListener : MonoBehaviour
{
    public GameEvent Event;
    public UnityEvent Response;

```

Figure 29: *Screenshot from the final prototype, where the wienie is visible atop a hill*

```
    private void OnEnable()
    {
        Event.RegisterListener(this);
    }
    private void OnDisable()
    {
        Event.UnregisterListener(this);
    }
    public void OnEventRaised()
    {
        Response.Invoke();
    }
}
```

Listing 3: *Game Event Listener*

This event system is used in conjunction with coroutines and callbacks in order to schedule operations in both the runtime terrain generation and the STDM. For example, the `TerrainPainter`, which applies trees and splatmaps to the terrain, has a `GameEventListener`, which listens for the *TerrainGenerationComplete* event. As such, once the heightmap has been fully generated and eroded, this event raises and the terrain painter begins working. Likewise, the `TerrainPainter` has another `GameEventListener` for the `StagedAreaSpawned` event, which responds by calling the `ApplySplatmaps` function. Since the STDM may flatten an area of the terrain, this observer pattern lets the terrain system respond to changes made by the STDM without needing a direct reference.

With both the terrain generator and the STDM in place the prototype is practically done. The SAs are all created as prefabs in Unity which enables instantiating them as one object at the right points as explained above. Audio was also added in accordance with the requirements from Section 4.4 and splash screens was created to facilitate the narrative introduction (see Section 4.1.3) and to inform the players about the game and the evaluation of it (see Section 3.4.1). When the game is over the player is redirected to an online questionnaire where the terrain generation seed is passed to the questionnaire automatically to link responses to the terrain they experienced. With this in place, the prototype is ready to be evaluated in conjunction with the methods from Section 3.4.

6 Evaluation

With a prototype ready, the methodology outlined in Section 2.5 can be used to answer the final research question. An evaluation was set up according to the description in Section 3.4. This section will outline the details of the test, and the results of the evaluation will be presented subsequently in Section 7.

6.1 Test Description

The prototype used for the evaluation was a browser-based game hosted on Itch.io. When ending the game, an online questionnaire would open and prompt participants to answer the questions presented to them. The game was distributed through multiple groups on Facebook to gather as many participants as possible in accordance with the target group sampling method outlined in Section 3.1. The distribution began on May 13^{th} and ended on May 19^{th}. Over this span of time, 69 participants completed the game and answered the questionnaire. As this is a between-groups test design, there were two different conditions. All participants were randomly assigned either the control or the experimental condition which resulted in 37 (29 male, 7 female, 1 other) in the control condition and 32 (24 male, 8 female) in the experimental condition. Both groups has a similar average age of 28.9 for the control and 28.6 for the experimental condition.

6.2 Independent Variable

For the control condition, all participants would experience an environment generated from the same random seed, which meant that the procedurally generated VE would be the same for all of them. Participants in the experimental condition would all be assigned a different random seed meaning that each participant would experience a different VE from one another. The independent variable was therefore whether a fixed random seed or different random seeds was used to generate the VE. The random seed for the control condition was selected from a whitelist of random seeds which was collected using a generate-and-test approach, in which the designers looked for well balanced VEs that seemed to be representative of the generator. The fixed random seed of 48 was selected from this list of suitable seeds, and used for all participants in the control condition.

The random seed in the experimental condition was chosen from a pool of 1000 seeds. The random seed used when the VE was generated was based on the current time in milliseconds when the game was initialised. By using this method it is very difficult to replicate a given seed as players have little to no control over at what millisecond the game would initiate. Other methods for assigning the different random seeds would have yielded more different seeds but as we did not expect many participants 1000 was deemed to be enough to minimise the chance of two participants getting the same random seed.

Question	Rank 3	Rank 2	Rank 1
1 (1)	It is government property	It is private property	It is undiscovered
2 (3)	Yes	Could be	No
3 (m)	I did not see any skeletons	The former inhabitants	Animals
4 (1)	The military	A private company	Natives to the island
5 (2)	Still on the island	Dead	They left the island
6 (2)	No one left	Everyone left in a hurry	It was useless
7 (m)	They are not dead	A cult	The government
8 (3)	A cult	The military	It was an accident

Table 4: *The specific questions asked and the possible answers to those questions, ranked according to how close to the intended meaning they are, i.e. rank 3 answer is the authors intend and rank 2 and 1 are further and further away. The number in the parenthesis indicates what key piece of information that question refers to (1 = The military was on the island; 2 = Something bad happened to them; 3 = There is a cult on the island now; m = misleading)*

6.3 Dependent Variables

From Sections 2.5 and 3.4.1 it can be seen that there are a few dependent variables that will be compared across the two conditions.

The narrative engagement scale has 9 items ranging from strongly disagree to strongly agree on a 7-point scale. The same is true for the narrative tension except here there is only 6 items.

For the narrative coherency part there are 8 questions, two for each piece of key information (see Section 4.1.4) and two dummy questions not concerning any of it (see Section 3.4.1). Each of the 8 questions will consist of a free form answer and the same question but as a multiple-choice question with 3 answer possibilities or the option to write their own answer. The possible answers and their rank can be seen in Table 4. These ranks will then be summed for each of the pieces of the key information and dummy questions, resulting in a score for each piece of information as well as a collective one for all 8 questions.

The fourth section of the questionnaire consisted of 9 Likert items, 2 concerning the usability, two about the suitability, two about navigability, and three about the interest in the VE, all counter-balanced. These are for evaluating the PCG system together with the narrative from a user point of view. See Section 3.4.2 for complete description of the data analysis.

6.4 Test Procedure

As stated above, the game was hosted and played online in a browser. This in turn means that we as test conductors had limited control over how the participant would go about doing the test, and if any problems were to arise we were not able to step in. This also means that there is no chance for the participant to ask questions or have something elaborated. It is therefore very important that the procedure is designed in a way that leaves little room for interpretation.

When a participant pressed the link and opened the website they would see the game as well as a description of what the game is, and what the game's controls are. When they start the game they would be met with a description explaining what they were about to experience so that they knew it would have a narrative and that they should keep an eye out for information about this. In addition they were asked to use headphones when playing the game before they were introduced to the narrative (see Section 4.1.3).

In the following section, the results from the evaluation will be presented followed by a short section on what the key findings are. The full data set and the questionnaire can be found in the Electronic Appendices.

7 Results

For all applicable scales, Anderson-Darling and Levene's tests were used to determine if the data could be treated as parametric. If the data were non-normal, the data would be log-transformed and tested again for normality. If the data were parametric, they were evaluated with an independent t-test, otherwise with Mann-Whitney U.

7.1 Narrative Engagement

Condition	N	Mean Rank	Std Dev	Median	Sum of Ranks
Control	37	9.513	3.576	9	352
Experimental	32	10.687	3.876	12	342
Total	69				

Table 5: *Descriptive statistics of narrative understanding*

Narrative Understanding The log transformed data from the control condition were normally distributed ($\alpha = 0.05, A^2 = 0.519, A^2 < 0.722$), though the data from the experimental condition were not ($\alpha = 0.05, A^2 = 1.629, A^2 > 0.715$). Therefore the data could not be treated as parametric. Mann-Whitney U indicated that there was no significant difference in understanding between conditions ($U = 467, p = 0.06, r = -0.18$).

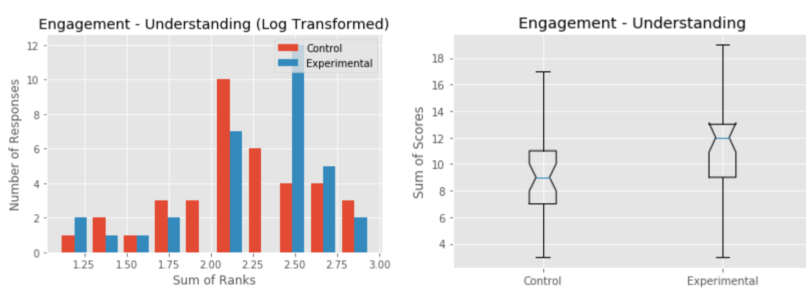


Figure 30: *Distribution (Log-transformed) and box plot of narrative understanding. Despite not having significant results, the disribution and box plot both indicate a higher mean in the experimental condition.*

Attentional Focus The data from the control condition were normally distributed ($\alpha = 0.05, A^2 = 0.287, A^2 < 0.722$), as were the data from the experimental condition ($\alpha = 0.05, A^2 = 0.677, A^2 < 0.715$). Levene's test indicated that the difference in variance between conditions was approximately zero, $F(36, 31) = 2.048, p = 0.16$. Therefore the data could be treated as parametric. An independent t-test indicated no significant difference in attentional focus between conditions ($t = -0.23, p = 0.81, r = 0.02$).

Condition	N	Mean Rank	Std Dev	Median	Sum of Ranks
Control	37	14.594	3.694	15	540
Experimental	32	14.781	2.814	15	473
Total	69				

Table 6: *Descriptive statistics of Attentional Focus*

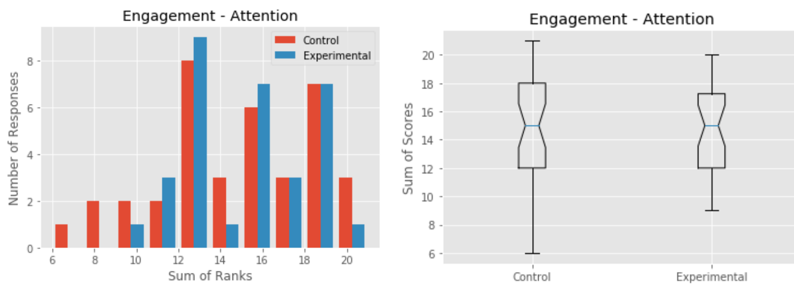


Figure 31: *Distribution and box plot of Attentional Focus*

Condition	N	Mean Rank	Std Dev	Median	Sum of Ranks
Control	37	12.459	4.469	14	461
Experimental	32	13.312	2.983	13	426
Total	69				

Table 7: *Descriptive statistics of Presence*

Presence The data from the control condition were normally distributed ($\alpha = 0.05, A^2 = 0.554, A^2 < 0.722$), as were the data from the experimental condition ($\alpha = 0.05, A^2 = 0.506, A^2 < 0.715$). Levene's test indicated that the difference in variance between conditions was approximately zero, $F(36, 31) = 3.591, p = 0.06$. Therefore the data could be treated as parametric. An independent t-test indicated no significant difference in presence between conditions ($t = -0.93, p = 0.36, r = 0.11$).

Overall Narrative Engagement The summed results from the engagement scale for the control condition were normally distributed ($\alpha = 0.05, A^2 = 0.516, A^2 < 0.722$), as were the data from the experimental condition ($\alpha = 0.05, A^2 = 0.287, A^2 < 0.715$). Levene's test indicated that the difference in variance between conditions was approximately zero, $F(36, 31) = 0.014, p = 0.91$. Therefore the data could be treated as parametric. An independent t-test indicated no significant difference in narrative engagement between conditions ($t = -0.98, p = 0.33, r = 0.12$).

Play Again As it was not part of the engagement scale, the Likert item regarding participants desire to play again was evaluated separately. Since it

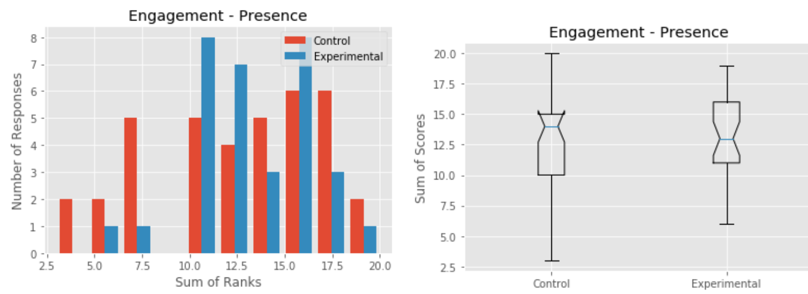


Figure 32: *Distribution and box plot of Presence*

Condition	N	Mean Rank	Std Dev	Median	Sum of Ranks
Control	37	24.108	5.774	25	892
Experimental	32	29.469	5.562	25	815
Total	69				

Table 8: *Descriptive statistics of Narrative Engagement*

was an individual Likert item, it could not be treated as parametric. This data is summarised in Table 9. A notched box plot of the data can be seen in Figure 34. A Mann-Whitney U test indicated that there was no significant difference in the desire to play the experience again between participants in the control and experimental conditions ($U = 488.5, p = 0.10, r = -0.15$).

Condition	N	Mean Rank	Sum of Ranks
Control	37	3.649	135
Experimental	32	4.219	135
Total	69		

Table 9: *Summary of "Play Again" data*

7.2 Narrative Tension

The log-transformed data were normally distributed in both the control ($\alpha = 0.05, A^2 = 0.282, A^2 < 0.722$) and experimental ($\alpha = 0.05, A^2 = 0.591, A^2 < 0.715$) conditions. However Levene's test showed that the assumption of homogeneity of variance was broken in both the original ($F(36, 31) = 4.616, p = 0.035$) and log-transformed data ($F(36, 31) = 5.740, p = 0.019$). As such the data are treated as non-parametric. Mann-Whitney U indicates no significant difference in Narrative Tension difference between conditions ($U = 498.5, p = 0.13, r = -0.135$).

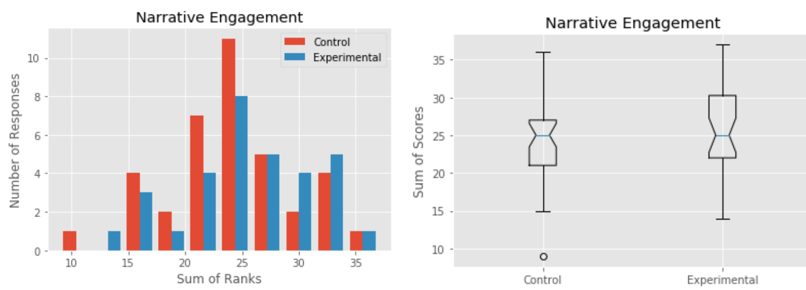


Figure 33: *Distribution and box plot of Narrative Engagement*

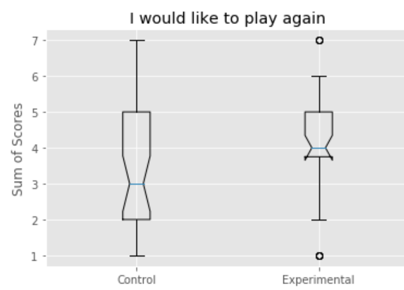


Figure 34: *Box plot of players"desire to play the game again. Once more, no significant result was found, but the box plot indicates that the true mean is likely higher in the experimental condition*

7.3 Narrative Coherency

An Anderson-Darling test indicated that the log-transformed data of narrative coherency were neither normally distributed in the control condition ($\alpha = 0.05$, $A^2 = 2.057, A^2 > 0.722$) nor the experimental condition ($\alpha = 0.05$, $A^2 = 1.323, A^2 > 0.715$). However Levene's test did show that the difference in variance between the conditions was approximately zero, $F(36, 31) = 0.292, p = 0.59$. Since the data were not parametric, Mann-Whitney U was used to test the narrative coherency section. This test indicated that between the control group ($Mdn = 15$) and the experimental group ($Mdn = 15$) there was no significant difference in narrative coherency ($U = 533.5, p = 0.24, r = -0.08$)

7.4 PCG

Usability An Anderson-Darling test indicated that the log-transformed data of usability were neither normally distributed in the control condition ($\alpha = 0.05$, $A^2 = 0.836, A^2 > 0.722$) nor the experimental condition ($\alpha = 0.05$, $A^2 = 0.842, A^2 > 0.715$). Mann-Whitney U indicated no significant difference in usability between conditions ($U = 498.5, p = 0.128, r = -0.135$).

Condition	N	Mean Rank	Std Dev	Median	Sum of Ranks
Control	37	27.892	4.032	28	1032
Experimental	32	28.813	2.639	29	922
Total	69				

Table 10: *Descriptive statistics of Narrative Tension*

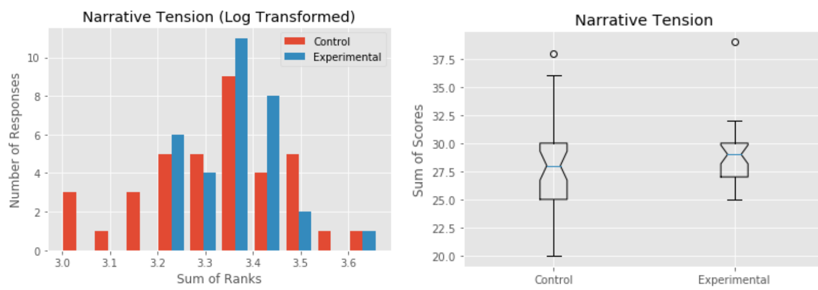


Figure 35: *Distribution (log-transformed) and box plot of Narrative Tension. The mean values in both conditions are similar, though the experimental conditions seems to have much lower variance*

Suitability An Anderson-Darling test indicated that the log-transformed data of suitability were neither normally distributed in the control condition ($\alpha = 0.05$, $A^2 = 0.895, A^2 > 0.722$) nor the experimental condition ($\alpha = 0.05$, $A^2 = 2.014, A^2 > 0.715$). Mann-Whitney U indicated no significant difference between conditions ($U = 479.5, p = 0.086, r = -0.163$).

Navigability An Anderson-Darling test indicated that the log-transformed data of navigability were neither normally distributed in the control condition ($\alpha = 0.05$, $A^2 = 2.077, A^2 > 0.722$) nor the experimental condition ($\alpha = 0.05$, $A^2 = 1.578, A^2 > 0.715$). Mann-Whitney U indicated no significant difference between conditions ($U = 470, p = 0.07, r = -0.177$).

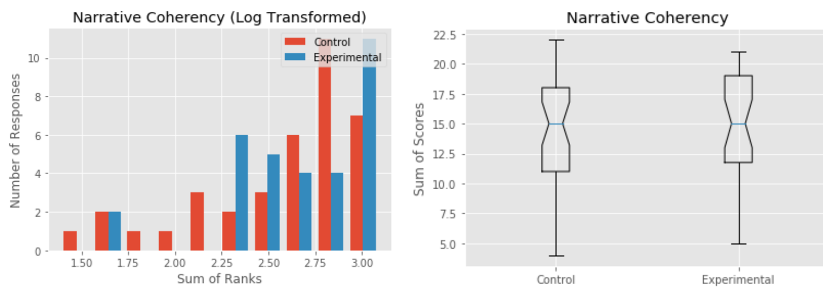
Interest An Anderson-Darling test indicated that the log-transformed data of interest were neither normally distributed in the control condition ($\alpha = 0.05$, $A^2 = 2.616, A^2 > 0.722$) nor the experimental condition ($\alpha = 0.05$, $A^2 = 2.559, A^2 > 0.715$). Mann-Whitney U indicated no significant difference between conditions ($U = 584.5, p = 0.46, r = -0.01$).

7.5 Player Types

Part of the demographics was collecting information about the player types of participants. In Table 16 the mean summed scores for the narrative coherency and the narrative engagement based on player types can be seen.

Condition	N	Mean Rank	Std Dev	Median	Sum of Ranks
Control	37	14.135	4.816	15	523
Experimental	32	15.031	4.398	15	481
Total	69				

Table 11: *Descriptive statistics of Narrative Coherency*



(a) ***(b)***

Figure 36: *Distribution and box plot of Narrative Coherency*

Furthermore, the narrative engagement of players was analysed once more, this time considering player type to be the independent variable. The mode response for player type was "Immersion', so the data were split based on whether participants identified as "Immersion"players, or "Other'. An Andersen-Darling test revealed both the data for "Immersion"player types ($\alpha = 0.05$, $A^2 = 0.632, A^2 < 0.699$) and "Other"player types ($\alpha = 0.05$, $A^2 = 0.405, A^2 < 0.732$) were normally distributed. Levene's test showed that both conditions had approximately the same variance, $F(22, 45) = 0.089, p = 0.766$. Therefore the data could be considered parametric. An independent t-test showed that there was no significant difference between these conditions ($t = 1.94, p = 0.056, r = 0.12$).

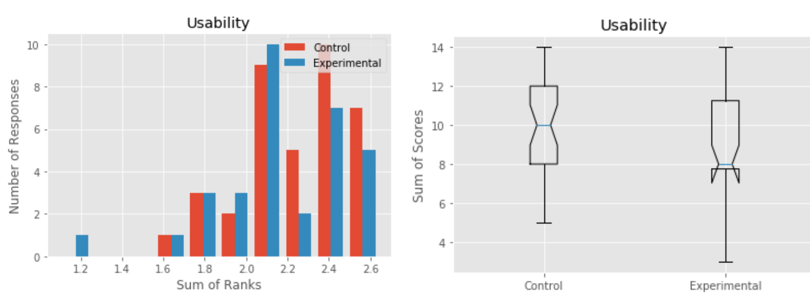


Figure 37: *Distribution (log-transformed) and box plot of Usability. Most responses were in a similar range, though the box plot indicates that the mean usability is potentially much lower in the experimental condition*

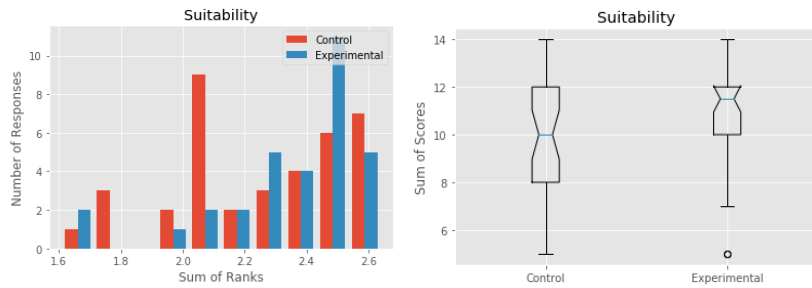


Figure 38: *Distribution (log-transformed) and box plot of Suitability. These plots indicate that despite no significant result being found, the mean suitability was higher in the experimental condition*

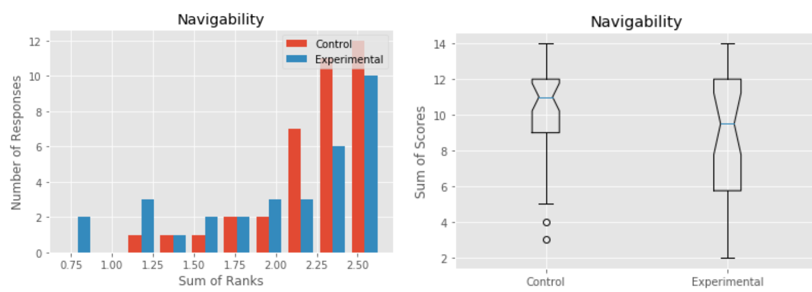


Figure 39: *Distribution (log-transformed) and box plot of Navigability. These plots indicate that there was much less variance in the navigability in the control condition than the experimental.*

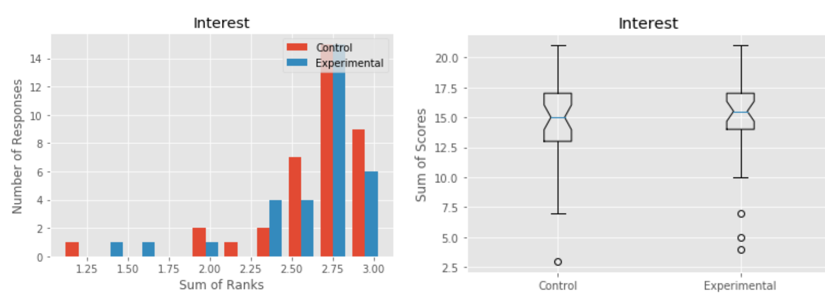


Figure 40: *Distribution (log-transformed) and box plot of data on Interest in the environment*

Condition	N	Mean Rank	Std Dev	Median	Sum of Ranks
Control	37	9.946	2.588	10	368
Experimental	32	9.187	2.755	8	294
Total	69				

Table 12: *Descriptive statistics of Environment Usability*

Condition	N	Mean Rank	Std Dev	Median	Sum of Ranks
Control	37	9.946	2.588	10	368
Experimental	32	10.781	2.260	11.5	345
Total	69				

Table 13: *Descriptive statistics of Environment Suitability*

7.6 Findings

From these results a lot of interesting findings can be extracted. Most noticeable is how none of the inferential statistics returned significant results. When comparing the narrative engagement the two samples have equal variances and the independent t-test also shows that there is no significant difference in narrative engagement between them. No significant difference was found between conditions when asking participants if they want to play again ($p = 0.1$), but the box plot of this data (seen in Figure 34) suggests that participants in the experimental condition wanted to play again more.

The narrative tension scale also shows no significant different between conditions. As for the narrative coherency, the variances are once again equal between the groups and the Mann-Whitney U also showed no significant difference. This indicates that the two groups had similar levels of narrative coherency when analysing the ranks in the multiple-choice questions. Comparing the summed narrative coherency score to player types shows that no specific player type scores higher than others. Additionally, no significant difference was found in

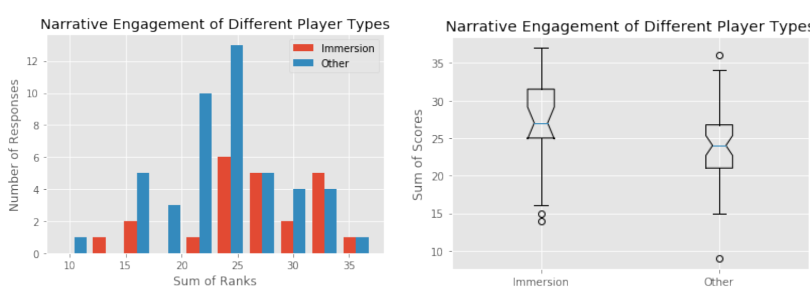


Figure 41: *Distribution and box plot of Narrative Engagement based on Player Type. This box plot shows that Immersion players generally reported higher levels of engagement that other player types*

Condition	N	Mean Rank	Std Dev	Median	Sum of Ranks
Control	37	10.0	2.721	11	370
Experimental	32	8.562	3.699	9.5	274
Total	69				

Table 14: *Descriptive statistics of Environment Navigability*

Condition	N	Mean Rank	Std Dev	Median	Sum of Ranks
Control	37	14.918	3.858	15	552
Experimental	32	14.781	4.059	15.5	473
Total	69				

Table 15: *Descriptive statistics of Interest in the environment. The responses appear almost identical in distribution across conditions.*

narrative engagement of different player types, although the "Immersion" players ($\mu = 26.6, \sigma = 5.8$) almost had a significantly higher engagement than others ($\mu = 23.8, \sigma = 5.4$) ($p = 0.056$).

The items concerning how the environment fitted the narrative were also all non-significant, indicating that even though the VE changes it still feels as suited as if it is the same over and over. While there was not significant difference between conditions, there was an overall high level of suitability reported, suggesting that the PCG terrain's design is appropriate for the narrative.

These findings are all interesting for us, and even though they might not seem very good, for this thesis it is positive that they are all non significant. Why this is so, will be discussed in more detail in the next section.

Player Type	N	Mean Coherency	Mean Engagement
Action	5	15.2	23.2
Social	13	13.15	24.08
Mastery	9	13.22	22.0
Achievement	5	13.6	25.8
Immersion	23	15.78	26.61
Creativity	8	14.25	26.12
No type	6	15.5	21.66

Table 16: *Narrative coherency and engagement based on player types*

8 Discussion

The results from the evaluation described in the previous section are plentiful but also quite similar to each other. The relevance, meaning, and possible causes of these results in relation to the research question will be discussed in further detail in this section. And in addition, biases and the validity and reliability of the evaluation itself will be discussed.

8.1 Interpreting Results

All the data analysis was performed between the two groups to examine how experiencing a different VE influences narrative engagement, tension (suspense), and understanding in players. None of the comparisons between groups yielded any significant results and we can therefore not reject the null hypothesis stating that *there is not a significant difference in engagement levels and narrative coherency between the two groups.* This suggests that changing the VE for every participant does not significantly change the narrative experience on average compared to experiencing the same one.

For the narrative engagement the Levene's test shows that the variances are equal. Beyond checking if the data were parametric, it was important to look for a difference in variance, as it would indicate that the difference in the VEs would mean a difference in the experience for the participants. That this is not the case is interesting, and using different procedurally generated environments cannot be said to influence the narrative engagement of the player. The item concerning whether participants would play the game again does also not show a significant difference though it can be seen from the box plot in Figure 34 that the variance is smaller for the experimental condition but the Levene's test points towards it not being significantly smaller.

These data were further examined through the lens of player types. This yielded no significant results, but with a low p-value of 0.056, and a visible difference in the box plot in Figure 41, this could be a type-II error. This would suggest that players who identify as the 'Immersion' player type were more engaged in the narrative than others. This would make sense as a result, as the desire for engaging narratives is an aspect of this player type by definition.

For the narrative tension Likert scale no significant difference was found but the Levene's test did show that the variances were not homogeneous between the two groups. Interestingly, from the box plots it can be seen that the variance is smaller for the experimental group which is unexpected. It was anticipated that if the variances were unequal, it would be lower in the control group, as the variations in responses from individual participants' differences would be outweighed by those caused from experiencing different VEs. This could perhaps indicate that despite the random assignment of conditions, that these two sample groups are not in fact representative of the same population.

The results from the narrative tension was in part about measuring the narrative coherency, as if the participant would experience a rising tension curve we can assume this means that they had a somewhat coherent narrative experience individually. This together with the narrative understanding can both act as indicators of the narrative coherency.

For the narrative coherency also no significant difference was found showing an equal level between the two groups when looking at the multiple-choice questions. The box plots of the different key pieces of information and the summed responses also shows that the variance is the same for the two groups. This is interesting because for the experimental condition the participants did not experience the same VE, but that did not affect the narrative to such a large degree that it meant a lower narrative understanding. One sub-scale from the narrative engagement measure was similarly about the narrative understanding; looking at this sub-scale also reveals no significant difference. It does however return a low p-value of 0.06 which could indicate a type-II error. By looking at the box plot it can be seen that had the results been significant it would have been in favour of the experimental condition. The reason for a better narrative understanding in this condition might be because the chosen random seed for the control condition actually is not as representative as we thought, and therefore it is more a matter of this yielding lower results, than the experimental yielding higher. This however, can not be proven from these results. It is also worth mentioning that the overall narrative coherency scale is in the range of 0 to 24. The box plots show that in both conditions, 75% of participants scored above 12, meaning that despite the broad distribution, there was a generally high level of narrative understanding.

The section about how the VE, created by the PCG system, suited the narrative once again showed no significant differences but did return quite high scores. The measures of usability, suitability, and navigability all have medians for both conditions above 8, which puts most respondents on the higher end of these measures (on a scale of 0 to 14). This indicates that players found the environment well suited to the narrative which was not exactly expected as this is not hand-crafted, though perhaps reflects the strength of a constraint-based PCG system such as this.

All the non-significant results found in the evaluation is in many ways a positive sign for combining PCG and ES. The aim here was to detect any significant impacts changing the VE for each player would have. Not finding any difference indicates that it is possible to utilise ES even in an ever changing environment without impacting the engagement and narrative coherency for the player. There is a few considerations and biases to have in mind however.

8.2 Setup, Procedure, and Test Specifics

Within this specific evaluation and its procedures are some biases that needs to be taken into account when discussing these results. The prototype was browser-based and so, restraints to both the graphical fidelity and the amount of assets in the game were introduced. In other words, the game could have looked better, and had more detailed SAs, if the setup for the test was different. Having participants enter, play the game, and answer the questionnaire themselves meant that a lot of control was lost from our side. We were not able to help with bugs or ensure everything would run smoothly for the participant. The prototype was extensively pilot tested to ensure all participants would have the same procedure during the test, but it was impossible to be 100% sure of this. Being physically present to facilitate the test would have helped avoid a lot of these unknowns. On the other hand, the online distribution helped in recruiting a higher number of participants, thereby increasing the reliability of the results.

Having the game be browser-based also introduced significant loading times before the participants were able to play the game. This would not have been solved entirely by moving to a non browser-based version, but control over the hardware which was used to the test could have helped alleviate these issues. From the hosting website we were able to get basic analytics about how many viewed the page, and played the game. Comparing these numbers to the final number of participants shows that only approximately one third of people who played the game ended up submitting a response to the questionnaire. This could be attributed to many things, with known issues on both the long loading screens and issues in opening the questionnaire from the game, many participants may have stopped there. Since we were not present to facilitate the testing, there was no guarantee that we would get the data as needed. These numbers could also be inflated by returning participants trying the game again, but not submitting data on replaying it.

Measuring narrative engagement when the setup is this uncontrolled also introduces a few biases. The measure of the engagement can be affected by many factors which in a controlled environment would have been easy to rule out. We have no way of knowing if participants played the game in a quiet place or even if they were alone or affected by the presence of others. This was why we encouraged everyone to wear headphones when playing the game, but this is again not something we were able to control. However, these circumstances are equally likely to affect participants in the control and the experimental condition so the comparisons will still be okay, but looking at the results themselves are much more uncertain.

The method for measuring the narrative coherency seemed to work as intended. Participants would write their own short answer to each question and then pick one afterwards in the multiple-choice which would reflect the one they wrote themselves. No analysis of how many changed their answer when faced with the choice was made as this would not have added any valuable information in answering the research question. Here we were only interested in the specifics of whether they got the right understanding or not, which an answer to the multiple choice question would reveal, but having them give their own answer first enabled them to consider their own interpretation before being faced with the choice which would be less likely to bias them when choosing the answers.

Only relying on the multiple-choice answers in our analysis was a purposeful choice that ensured good quantifiable data. It is however worth noting that efforts was made to code the open-ended answers to see how these would compare to the answers in the multiple-choice one. Coding these did prove to be difficult and also more or less redundant as most of the answers pointed towards somewhat of the same answer as the multiple-choice ones. It was therefore decided it would not bring valuable data to the efforts of answering the research question.

These multiple-choice questions introduce some biases. They are formulated by the narrative designers, who of course have the "right" interpretation of the narrative in mind when making the questions. Perhaps a better solution would have been to run a preliminary test of narrative coherence with qualitative responses, which would have been coded to generate the multiple-choice items. In general the NEM was supposed to evaluate the narrative coherency but lacks some more robust measures to test the individual person's feeling of coherency.

It could in that regard also have been useful to change the methods a bit and have some interviews in addition to the existing measures. By following up with such a qualitative measure for at least some participants, we could potentially find out and correlate their complete detailed experienced narrative with their answers to the multiple choice to better figure out how these might relate to peoples actual afterstory.

Another part that would enhance this methodology is better telemetry from the game. We worked on setting up a server for saving telemetry in the form of player behaviour, completion time, which SAs was visited, and information about the system the game was played on, among others. Due to time constraints this was deemed expendable as these data would be less important than the ones already collected in the questionnaire. It would have served as good insights into how different VEs affected the player behaviour as well as provided knowledge on whether participants would come across less SAs in one of the conditions and how this would affect the narrative coherency.

8.2.1 Technical Issues

In addition to the biases mentioned above, there was one technical issue which we were not able to fix before the final evaluation. Then participants would get to the end of the game they would be presented with a splash screen prompting them to continue to the questionnaire which would happen when a button was pressed. Unfortunately, pressing this button was difficult and would in most cases require a few presses of the escape key before pressing the button would be registered and the participant were send to the questionnaire. This is known to have caused at least some participants to play the game but not answer the questionnaire as they would give up. However pilot tests indicated that most participants were able to work past this bug, so it was not considered a test-breaking issue.

8.2.2 Validity

A few things have impacted the validity of the evaluation. Measuring narrative engagement and coherency are difficult as these are both very subjective measures. In a more controlled setting, there would be some psycho-physiological measures that could have pointed towards general engagement in a more objective manner. However, by using the pre-defined and well documented Likert scale from Busselle and Bilandzic (2009) we ensured a good measure for the narrative engagement. As seen in Section 2.5.3, measuring narrative coherency is no easy task and very difficult to quantify. However, by combining methods of afterstory (Larsen et al., 2019) and the AAD (Bruni and Baceviciute, 2013), we developed our own method which quantified the answer to a multiple choice question to enable comparisons between groups. The validity of this method would need to be evaluated for itself as this was not feasible within the scope of this project unfortunately. Having chosen a more qualitative approach could have provided us with more in-depth results but they might also have been much more scattered and harder to conclude anything from.

Comparing our two groups gave us some great results but it would have been interesting to also compare these groups to a much more controlled condition like a VE that was not created by PCG but hand-crafted by game designers. In this way we could not only see if the changing VEs had an impact over the same, but see if it would be significantly worse at engaging and tell the narrative compared to a hand-crafted one. This would have increased the validity of our results but as is, they are still useful in relation to creating VEs using PCG in narrative focused games.

If we had this extra condition we would also be able to see if our prototype was too specialised to our specific narrative. In other words, a threat to the validity might be that the non-significant results is due to the fact that the prototype was so specialised in placing the SAs in the right order, at points where the player would not miss them, and that they were so distinct compared to the rest of the VE, that whatever the VE looked like they would always experience a coherent narrative, effectively removing the VE as a factor for the narrative. This evaluation has used just a single independent variable, in the form of the environment. However, even though the STDM is used in both conditions, it is an unpredictable element that could negatively impact the validity of these results. The stochastic nature of the input to the STDM could have at least been accounted for in the analysis, if there was sufficient telemetry collected about both the system's performance (e.g., where and when did it try to place SAs), and players' movement through the environment. The non-deterministic nature of the STDM could call the reliability of these results into question, too.

8.2.3 Reliability

The setup and procedure behind this thesis is elaborate and replications might result in a different outcome. This is partly due to how we are specifically evaluating our specific prototype and so repeating the same experiment with another prototype might not yield the same results meaning the reliability could be better. While the PCG system for terrain generation is somewhat generalisable and allows a large variety of different outcomes and designer control, the STDM is very much an ad-hoc PCG system. It is a constraint-based system, which is designed to facilitate the specific constraints required for this narrative. Future implementations of this system should try to improve this for a more off-the-shelf solution. The core functionality of the STDM is generalisable and allows for generic prefab GameObjects to be passed in as SAs, with their own defined fixed-space constraints, and time-constraints (see Section 5.2 for more on types of constraints). However the dynamic space constraints are less open to designers, and the rules for spawning wienies are very specific to this narrative's requirements.

This evaluation used the NEM, a tool we developed, in order to gather and analyse the data. However, this tool itself has not been extensively tested, and could be described in finer detail. As such, it is unclear how well other researchers may be able to recreate this tool, although the questionnaire used is available in the Electronic Appendices accompanying this thesis. Furthermore, while the NEM was deemed suitable for our requirements, without rigorous testing it cannot be said with 100% conviction that it is a reliable measure.

The narrative itself was not evaluated separately, and there was no baseline measure of how well participants would understand the story based on the SAs alone. While an exact reproduction of this experiment may yield similar results, it is unclear if a recreation with a different narrative would yield similar results.

9 Conclusion

This thesis developed a prototype game with a PCG environment and an embedded narrative in order to answer the research question which was initially presented in Section 2.6.1.

> Will using an online PCG environment affect a player's engagement and narrative coherency of a dynamically embedded narrative?

Using the developed Narrative Experience Measurement (NEM) tool we were able to conduct a full evaluation of our developed prototype. The NEM tool consists of different sections all concerned with the overall narrative experience of a game. These sections were "Narrative Engagement", "Internal Coherency", and "External Consistency". These in turn are measured by a series of Likert scales: "narrative engagement" (with sub-scales of "understanding", "attentional focus", and "presence"), "narrative tension", and "narrative coherency".

The results in this between-subject test showed that there was no significant difference in the narrative experience between players all experiencing the same virtual environment (VE) and players all experiencing different VEs. Furthermore, similar levels of variance of narrative coherency, narrative tension, and narrative understanding across conditions indicate that there was external consistency between participants. This shows that it is possible to dynamically embed a narrative in the space of a game and still uphold similar levels of narrative engagement and coherency in all participants regardless of which environment they experience.

Though it is difficult to infer a lot in regards to specific levels of these measures compared to more classical designed game spaces it does show promise for the combination of techniques of Procedural Content Generation (PCG) and Environmental Storytelling (ES).

There are many different methods which can be used for PCG, but a constraint-based PCG system allows designers to specify high-level requirements for the environment, which is well suited to being combined with ES and embedded narratives. This allows designers to still specify requirements for the environment, while reducing the amount of time needed to craft expansive worlds, allowing them to focus on the the finer detail of important narrative elements and SAs. Furthermore, this opens up possibilities for enhanced replayability for narrative games in PCG environments.

This thesis creates a foundation for discussing the combination of PCG and narratives in games. It offers an initial approach to combining these elements and opens for much more research in this field.

10 Future Works

This thesis creates a solid foundation for research within this field and the methodology can be used for future studies with similar objectives. This was however only a vertical slice of the area and so research should be conducted in multiple other tracks from here on.

As discussed in Section 8 it will be useful to expand the methodology proposed in the NEM tool to include non-PCG environments to see what effect the use of PCG in itself has on the narrative when directly compared to hand-crafted environments. In addition, the aim of this thesis was to create a simple narrative and even though deeper layers exist, we only wanted the players to get the referential meaning of the narrative (Bordwell et al., 2008). It would therefore be interesting to repeat the evaluation but design a different narrative and aim to embed also the deeper, explicit or implicit, meaning in the space similar to Bevensee et al. (2012*b*) to see if a similar non-significant result would show.

In that regard, it should be explored how longer, and in turn, more detailed embedded narratives would hold up in a similar evaluation. We only aimed for a 5-10 minutes game experience which limited the narrative information that could be chronicled to the player. It would be interesting to examine if expanding the narrative to contain more details and a deeper implicit meaning would yield similar results.

SAs are a really useful design tool when embedding narratives in any VE, whether it is PCG or not. The SAs are always bringing important narrative information which in many ways really underline when narrative information is passed on. So it would be interesting to use more types of events from Schønau-Fog (2015), as the only one used right now is those locked in time, and to a lesser degree space.Introducing other types of events, like some which are not locked at all but could appear at any time or place, would create more dynamics within the narrative experience. These events would need to be designed carefully if the narrative should keep a certain tension curve and they should be much more universal to the narrative in comparison to the SAs.

In relation to the above, the STDM which was adapted from Schønau-Fog (2015) could be expanded upon and have more parameters to more precisely direct where SAs would spawn and also enable other types of events. More parameters could also ensure it would become more general and expandable to other narratives and types of environments.

A power of PCG that has been left untouched in this thesis, as it deserves one for itself, is that of procedural narratives. The STDM could be expanded upon and given more freedom as a PCG narrative system. Perhaps such a system could be used to create a dynamic or reactive narrative, which could provide a unique moment-to-moment experience for players, while maintaining the same broader narrative strokes.

Moving past the STDM, another approach of merging ES and PCG could be to create a PCG designer, which would actually generate the SAs in order to tell a narrative, similar to how games such as Yavalth (Browne, 2014) have been created by PCG systems, and how some text-replacement L-Systems such as Tracery (Compton, 2015, 2016) can already generate procedural stories.

This thesis provides some interesting results both in terms of PCG and ES and can be used as a foundation for multiple future studies within the empty space between these two fields.

References

Abbott, H. P. (2007), *Story, plot, and narration*, Cambridge Companions to Literature, Cambridge University Press, p. 39–51.

Aubert, M., Lebe, R., Oktaviana, A. A., Tang, M., Burhan, B., Hamrullah, Jusdi, A., Abdullah, Hakim, B., xin Zhao, J., Geria, I. M., Sulistyarto, P. H., Sardi, R. and Brumm, A. (2019), 'Earliest hunting scene in prehistoric art', *Nature* **576**, 442–445.

Aylett, R. and Louchart, S. (2007), Being there: Participants and spectators in interactive narrative, *in* M. Cavazza and S. Donikian, eds, 'Virtual Storytelling. Using Virtual Reality Technologies for Storytelling', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 117–128.

Balint, K., Kuijpers, M. and Doicaru, M. (2017), *The effect of suspense structure on felt suspense and narrative absorption in literature and film*, pp. 177–198.

Bay 12 Games (2006), '*Dwarf Fortress*', Bay 12 Games and Kitfox Games.

Baynham, M. (2003), 'Narratives in space and time: Beyond "backdrop" accounts of narrative orientation', *Narrative Inquiry* **13**(2), 347–366.

Beckham, C. and Pal, C. (2017), 'A step towards procedural terrain generation with gans'.

Bevensee, S. H., Boisen, K. A. D., Olsen, M. P., Schoenau-Fog, H. and Bruni, L. E. (2012*a*), Aporia – exploring continuation desire in a game focused on environmental storytelling, *in* D. Oyarzun, F. Peinado, R. M. Young, A. Elizalde and G. Méndez, eds, 'Interactive Storytelling', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 42–47.

Bevensee, S. H., Boisen, K. A. D., Olsen, M. P., Schoenau-Fog, H. and Bruni, L. E. (2012*b*), Project aporia – an exploration of narrative understanding of environmental storytelling in an open world scenario, *in* D. Oyarzun, F. Peinado, R. M. Young, A. Elizalde and G. Méndez, eds, 'Interactive Storytelling', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 96–101.

Biagioli, A. (2014), 'Understanding perlin noise', `https://flafla2.github.io/2014/08/09/perlinnoise.html`. [Accessed: February 12th 2020].

Bjørner, T. (2015), *Qualitative methods for Consumer Research*, 1 edn, Hans Reitzels Forlag.

Blizzard Entertainment (2012), '*Diablo III*', Blizzard Entertainment.

Bordwell, D., Thompson, K. and Smith, J. (2008), *Film art : an introduction*, McGraw-Hill higher education, 8. ed. edn, Mcgraw Hill, Boston.

Bornhofen, S., Gardeux, V. and Machizaud, A. (2012), 'From swarm art toward ecosystem art', *International Journal of Swarm Intelligence Research (IJSIR)* **3**, 1–18.

Bridgeman, T. (2007), *Time and space*, Cambridge Companions to Literature, Cambridge University Press, p. 52–65.

Bridson, R. (2007), Fast poisson disk sampling in arbitrary dimensions, *in* 'ACM SIGGRAPH 2007 Sketches', SIGGRAPH '07, Association for Computing Machinery, New York, NY, USA, p. 22–es.
URL: *https://doi.org/10.1145/1278780.1278807*

Brockmeyer, J., Fox, C., Curtiss, K., McBroom, E., Burkhart, K. and Pidruzny, J. (2009), *The deveopment of the Game Engagement Questionnaire: A measure of engagement in video game-playing*, University of Toledo.

Brown, M. (2015), '*Why Nathan Drake Doesn't Need a Compass — Game Maker's Toolkit*', Youtube `https://www.youtube.com/watch?v=k70_jvVOcG0`. [Accessed: March 13th 2020].

Browne, C. (2014), '*Yavalath*', `http://cambolbro.com/games/yavalath/`.

Bruni, L. E. and Baceviciute, S. (2013), Narrative intelligibility and closure in interactive systems, *in* H. Koenitz, T. I. Sezen, G. Ferri, M. Haahr, D. Sezen and G. Çatak, eds, 'Interactive Storytelling', Springer International Publishing, Cham, pp. 13–24.

Busselle, R. and Bilandzic, H. (2009), 'Measuring narrative engagement', *Media Psychology* **12**(4), 321–347.
URL: *https://doi.org/10.1080/15213260903287259*

Campo Santo (2016), '*Firewatch*', Campo Santo.

Carson, D. (2000*a*), 'Environmental storytelling: Creating immersive 3d worlds using lessons learned from the theme park industry', `https://www.gamasutra.com/view/feature/131594/environmental_storytelling_.php`. [Accessed: March 8th 2020].

Carson, D. (2000*b*), 'Environmental storytelling, part ii: Bringing theme park environment design techniques to the virtual world', `https://www.gamasutra.com/view/feature/131593/environmental_storytelling_part_.php`. [Accessed: March 8th 2020].

Christy, K. R. and Fox, J. (2016), 'Transportability and presence as predictors of avatar identification within narrative video games', *Cyberpsychology, Behavior, and Social Networking* **19**(4), 283–287.
URL: *https://doi.org/10.1089/cyber.2015.0474*

Compton, K. (2015), 'Tracery', Computer Software. `https://www.tracery.io/`.

Compton, K. (2016), '*So you want to build a generator*', Blog Post. https://galaxykate0.tumblr.com/post/139774965871/so-you-want-to-build-a-generator. Accessed: Febrary 27th 2020.

Compton, K. (2017), '*Practical Procedural Generation for Everyone*', GDC. YouTube https://www.youtube.com/watch?v=WumyfLEa6bU. [Accessed: February 27th 2020].

djgriffin (2006), 'tibetan chant 4 colargol 2.aif', https://freesound.org/people/djgriffin/sounds/15488/. [Accessed: April 24th 2020].

Erwin, C. (2015), 'Diamond square', https://en.wikipedia.org/wiki/File:Diamond_Square.svg. [Accessed: February 12th 2020].

Field, A. and Hole, G. (2003), *How to Design and Report Experiments*, SAGA Publications.

Fina, A. D. (2003), 'Crossing borders: Time, space, and disorientation in narrative', *Narrative Inquiry* **13**(2), 367–391.

Firaxis Games (2016), '*Sid Meier's Civilization VI*', 2K Games.

Forster, E. M. (1924), *Aspects of the Novel*.

Frasca, G. (2003), *Simulation versus Narrative: Introduction to Ludology*, Routledge.

Fullbright (2013), '*Gone Home*', The Fullbright Company.

Gearbox Software (2009-2019), '*Borderlands*', 2K Games.

Genette, G. (1982), *Figures of Literary Discourse*, Columbia University Press.

Gonzalez Vivo, P. and Lowe, J. (n.d.), 'The book of shaders', https://thebookofshaders.com/.

Gunson, L. (2013), '*Composition techniques and player direction*', Shape of Play https://shapeofplay.wordpress.com/2013/06/25/composition-level-design/. [Accessed: March 13th 2020].

Halperin, X. (2019), '*"Marvel's Spider-Man": Procedural Lighting Tools*', GDC. https://www.gdcvault.com/browse/gdc-19/play/1026315. [Accessed: February 11th 2020].

Hello Games (2016), '*No Man's Sky*', Hello Games.

Hellquist, P. (N.D.), '*Inside The Box: The Borderlands 2 Loot System*', https://www.gearboxsoftware.com/2013/09/inside-the-box-the-borderlands-2-loot-system/.

Herman, D., Jahn, M. and Ryan, M.-L., eds (2005), *Routledge Encyclopedia of Narrative Theory*, Routledge, London.

Hipple, R. (2017), '*Game Architecture with Scriptable Objects*', Unite Austin 2017 - https://www.youtube.com/watch?v=raQ3iHhE_Kk. [Accessed: March 25th 2020].

Hoek, M., Theune, M. and Linssen, J. (2014), Generating game narratives with focalization and flashbacks.

Hørdum Jæger, P. (2018), Procedural generated trees: How distribution and growth factors affect percieved realism, Master's thesis, AAlborg University.

Insomniac Games (2018), '*Marvel's Spider-Man*', Sony Interactive Entertainment.

Interactive Data Visualisation (2002-2020), '*SpeedTree*', Computer Software.

Jenkins, H. (2004), Game design as narrative architecture, *in* N. Ardrip-Fruin and P. Harrigan, eds, 'First Person: New Media as Story, Performance, and Game', MIT Press, Cambridge, pp. 118–130.

Kallner, A. (2018), Formulas, *in* A. Kallner, ed., 'Laboratory Statistics (Second Edition)', second edition edn, Elsevier, pp. 1 – 140.
URL: *http://www.sciencedirect.com/science/article/pii/B9780128143483000010*

Khaled, R., Nelson, M. J. and Barr, P. (2013), Design metaphors for procedural content generation in games, *in* 'Proceedings of the SIGCHI Conference on Human Factors in Computing Systems', CHI '13, Association for Computing Machinery, New York, NY, USA, p. 1509–1518.
URL: *https://doi.org/10.1145/2470654.2466201*

Kinoton (2018), 'Dark cave drone', https://freesound.org/people/Kinoton/sounds/421826/. [Accessed: April 24th 2020].

klankbeeld (2013), 'estate february nl haanwijk 130217_00.flac', https://freesound.org/people/klankbeeld/sounds/178374/. [Accessed: April 24th 2020].

Korkihort, J. (2016), '*The Origin of the Disneyland Wienie*', MousePlanet https://www.mouseplanet.com/11371/The_Origin_of_the_Disneyland_Wienie. [Accessed: March 13th 2020].

Kvisgaard, A., Klem, S., Nielsen, T., Rafferty, E., Nilsson, N., Høeg, E. and Nordahl, R. (2019), Frames to zones: Applying mise-en-scène techniques in cinematic virtual reality, *in* '2019 IEEE 5th Workshop on Everyday Virtual Reality, WEVR 2019', IEEE, United States. 2019 IEEE 5th Workshop on Everyday Virtual Reality, WEVR ; Conference date: 23-03-2019 Through 23-03-2019.

Larsen, B. A., Bruni, L. E. and Schønau-Fog, H. (2019), The story we cannot see: On how a retelling relates to its afterstory, *in* R. E. Cardona-Rivera, A. Sullivan and R. M. Young, eds, 'Interactive Storytelling', Springer International Publishing, Cham, pp. 190–203.

Larsen, B. A. and Schoenau-Fog, H. (2016), The narrative quality of game mechanics, *in* F. Nack and A. S. Gordon, eds, 'Interactive Storytelling', Springer International Publishing, Cham, pp. 61–72.

Louchart, S. and Aylett, R. (2003), Solving the narrative paradox in ves – lessons from rpgs, *in* T. Rist, R. Aylett, D. Ballin and J. Rickel, eds, 'Intelligent Virtual Agents', Vol. 2792, Springer, Heidelberg, p. 244–248.

Lucas, G. (1977), 'Star wars episode iv: A new hope'. [film] Directed by G. Lucas. United States: Lucasfilm.

MacDonald, K. (2018), '*No Man's Sky developer Sean Murray: 'It was as bad as things can get'*', The Guardian `https://www.theguardian.com/games/2018/jul/20/no-mans-sky-next-hello-games-sean-murray-harassment-interview`. [Accessed: February 12th 2020].

Maxis (2008), '*Spore*', Electronic Arts.

Maxis (2014), '*The Sims 4*', Electronic Arts.

Mittell, J. (2007), *Film and television narrative*, Cambridge Companions to Literature, Cambridge University Press, p. 156–171.

Mojang (2011), '*Minecraft*', Mojang and Microsoft Studios.

Morgan, D. (2015), 'Is it appropriate to use mann whitney u test to compare between two unrelated groups (unequal no. of participants) where response is in likert scale?'. `https://www.researchgate.net/post/Is_it_appropriate_to_use_Mann_Whitney_U_test_to_compare_between_two_unrelated_groups_unequal_no_of_participants_where_response_is_in_likert_scale/55edd4205f7f7129158b45fb/citation/download?fbclid=IwAR0sIFqU4-2zfMbGHsLTrjvjlm5L8O99umtEsELnqQDkaSteELruztRtPiE`.

Morin, R. (2016), '*Inside the Artificial Universe That Creates Itself*', The Atlantic. `https://www.theatlantic.com/technology/archive/2016/02/artificial-universe-no-mans-sky/463308/`. Accessed: Febrary 27th 2020.

Mossmouth (2008), '*Spelunky*', Mossmouth and Microsoft Studios.

Naughty Dog (2007-2017), '*Uncharted*', Sony Interactive Entertainment.

Pittman, D. (2015), '*Procedural Level Design in Eldritch*', GDC. YouTube `https://www.youtube.com/watch?v=BYNOPJOdvzs`. [Accessed: February 24th 2020].

Portnow, J. and Floyd, D. (2015), '*Procedural Generation - How Games Create Infinite Worlds - Extra Credits*', YouTube https://www.youtube.com/watch?v=TgbuWfGeG2o. [Accessed: February 11th 2020].

Preece, J., Rogers, Y. and Sharp, H. (2015), *Interaction Design: Beyond Human Computer Interaction*, John Wiley & Son Ltd.

Prince, G. (2003), *A Dictionary of Narratology*, University of Nebraska Press.

Richardson, D. C., Griffin, N. K., Zaki, L., Stephenson, A., Yan, J., Curry, T., Noble, R., Hogan, J., Skipper, J. I. and Devlin, J. T. (2018), 'Measuring narrative engagement: The heart tells the story', *bioRxiv* .
URL: *https://www.biorxiv.org/content/early/2018/09/13/351148*

Ricoeur, P. (1980), 'Narrative time', *Critical Inquiry* **7**(1), 169–190.

RoganMcDougald (2015), 'Sci-fi horror ambience', https://freesound.org/people/RoganMcDougald/sounds/260440/. [Accessed: April 24th 2020].

Rotzetter, F. (2018), *Nonverbal Guidance Systems Seamless Player-leading in Open-world Games*, pp. 169–190.

Ryan, M.-L. (2007), *Toward a definition of narrative*, Cambridge Companions to Literature, Cambridge University Press, p. 22–36.

Santiago, D. (2019), '*Procedurally Crafting Manhattan for "Marvel's Spider-Man"*', GDC. https://www.gdcvault.com/play/1025765/Procedurally-Crafting-Manhattan-for-Marvel. [Accessed: February 11th 2020].

Saunders, M., Lewis, P. and Thornhill, A. (2007), *Research Methods for Business Students*, Pearson Education.

Schønau-Fog, H. (2014), 'At the core of player experience: Continuation desire in digital games', pp. 388–410.
URL: *https://sfx.aub.aau.dk/sfxaub?sid=pureportal\&doi=10.1002/9781118796443.ch14*

Schønau-Fog, H. (2015), Adaptive storyworlds: Utilizing the space-time continuum in interactive digital storytelling, *in* H. Schønau-Fog, L. E. Bruni, S. Louchart and S. Baceviciute, eds, 'Interactive Storytelling', Springer International Publishing, Cham, pp. 58–65.

Shaker, N., Togelius, J. and Nelson, M. J. (2016), *Procedural Content Generation in Games*, Springer International Publishing. [Accessed: February 11th 2020].

Shouraboura, C. and Bleher, P. (2011), 'Placement of applications in computing clouds using voronoi diagrams', *J. Internet Services and Applications* **2**, 229–241.

Smith, G., Togelius, J. and Sturtevant, N. (2015), '*Making Things Up: The Power and Peril of PCG*', GDC. YouTube https://www.youtube.com/watch?v=B11RlHZsmGE. [Accessed: February 24th 2020].

Smith, G. and Whitehead, J. (2010), 'Analyzing the expressive range of a level generator'.

Steinbeck, J. (1937), *Of Mice And Men*, Covici Friede, United States.

Surly (2010), 'Buddhist chanting.wav', https://freesound.org/people/surly/sounds/91195/. [Accessed: April 24th 2020].

Suzuki, M. (2007), 'A three dimensional box counting method for measuring fractal dimensions of 3d models', *Proceedings of the 11th IASTED International Conference on Internet and Multimedia Systems and Applications, IMSA 2007* .

Togelius, J., Kastbjerg, E., Schedl, D. and Yannakakis, G. N. (2011), What is procedural content generation? mario on the borderline, *in* 'Proceedings of the 2nd International Workshop on Procedural Content Generation in Games', PCGames '11, Association for Computing Machinery, New York, NY, USA.
URL: *https://doi.org/10.1145/2000919.2000922*

Togelius, J., Yannakakis, G. N., Stanley, K. O. and Browne, C. (2010), Search-based procedural content generation, *in* C. Di Chio, S. Cagnoni, C. Cotta, M. Ebner, A. Ekárt, A. I. Esparcia-Alcazar, C.-K. Goh, J. J. Merelo, F. Neri, M. Preuß, J. Togelius and G. N. Yannakakis, eds, 'Applications of Evolutionary Computation', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 141–150.

Toolan, M. (2006), *Narrative: Linguistics and Structural Theories*, Vol. 13 of *Encyclopedia of Language & Linguistics*, Elsevier, p. 459–473.

Valve (2004), '*Half-Life 2*', Valve Corporation.

Valve (2007), '*Portal*', Valve Corporation.

Valve (2008), '*Left 4 Dead*', Valve Corporation.

Visual Design Cafe (2019), '*Open World Nature Kit*', https://assetstore.unity.com/packages/3d/environments/landscapes/open-world-nature-kit-150748#content. [Accessed: May 17th 2020].

Weaver, B. (2019), 'How can one help people create simple scales from likert-scored items?'. https://www.researchgate.net/post/How_can_one_help_people_create_simple_scales_from_Likert-scored_items/5cb4fe5da7cbaf213b03108b/citation/download?fbclid=IwAR19cBfiasa5rJMNO_al1CDlv5ugvoR802y9mYuj_ZH7-cnXriaVGyuIOwo.

Wikimedia (2008), 'Texture splatting', `https://en.wikipedia.org/wiki/File:Texture_splatting.png`. [Accessed: April 18th 2020].

Yannakakis, G. N. and Togelius, J. (2011), 'Experience-driven procedural content generation', *IEEE Transactions on Affective Computing* **2**(3), 147,161.

Yee, N. (2016), The gamer motivation profile: What we learned from 250,000 gamers, *in* 'Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play', CHI PLAY '16, Association for Computing Machinery, New York, NY, USA.
URL: *https://doi.org/10.1145/2967934.2967937*

Zakowski, S. (2016), Environmental storytelling, ideologies and quantum physics: Narrative space and the bioshock games, *in* 'DiGRA/FDG &#3916 - Proceedings of the First International Joint Conference of DiGRA and FDG', Digital Games Research Association and Society for the Advancement of the Science of Digital Games, Dundee, Scotland.

Zoran, G. (1984), 'Towards a theory of space in narrative', *Poetics Today* **5**(2), 309–335.

Appendices

A Questionnaire Items for Narrative Experience Evaluation

Questions on Engagement:

- Narrative Understanding
  - At points, I had a hard time making sense of what was going on in the game (-)
  - My understanding of the characters is unclear (-)
  - I had a hard time recognising the thread of the story (-)
- Attentional Focus
  - I found my mind wandering while the game was on (-)
  - While the game was on I found myself thinking about other things (-)
  - I had a hard time keeping my mind on the game (-)
- Narrative Presence
  - During the game, my body was in the room, but my mind was inside the world created by the story
  - The game created a new world, and then that world suddenly disappeared when the game ended
  - At times during the game, the story world was closer to me than the real world
- I would like to play this game again

(-) = reverse coded

Questions on Narrative Tension

- I did not experience any narrative progression (-)
- At the end, I knew more about the narrative than when I began
- I was more interested in the narrative at the beginning than the end
- I became less and less interested in the narrative as it progressed (-)
- My attention towards the narrative increased over time
- At no point did I find the narrative interesting (-)

(-) = reverse coded

Questions on Narrative Coherency

- (1) Why is the island hidden on maps?
- (4) Who built the radio tower?
- (5) Where are the ones who built the radio tower now?
- (6) Why is so much stuff left on the island?
- (2) Is there anyone else on the island?
- (8) Who set fire to the boat?
- (3) Where did the skeletons come from?
- (7) Who killed the former inhabitants on the island?

(-) = reverse coded

Number in parenthesis points to the order the questions is asked in the questionnaire to avoid biases.

Questions on PCG

- Usability
  - I was not able to navigate between two points in the environment (-)
  - It was easy to get to the places I wanted to go
- Suitability
  - The atmosphere of the environment suited the narrative
  - The narrative did not feel well suited to this environment (-)
- Navigability
  - When I moved around in the environment I felt lost (-)
  - I always knew where to go in the environment
- Interest
  - I found the environment boring (-)
  - I was curious to explore the environment
  - I want to continue exploring this environment

(-) = reverse coded

Demographics

- What is your age?
- What is your gender?
- How often do you play video games?
- How would you classify yourself as a player based on these definitions of player types?
- How much do you know about creating video games?
- Do you have any comments about the performance of the game (e.g. frame rate issues, loading times, graphical bugs etc)?
- Do you have any additional comments or feedback on the game or this questionnaire?

B Formative Evaluation of System Performance

The evaluation of the topography serves as the first step in the iterative process of evaluating and improving the PCG prototype. After this, multiple additional iterations should follow with the aim of improving both the performance and results of the algorithms. Every time a new feature is implemented in the prototype a small technical evaluation will be performed to ensure the system is still responding as it is supposed to. Each of these evaluations will be carried out by the developers based on these parameters:

- Initial loading time
- Frame rate
- Frame drops
- Aesthetics of generated content compared to intent

For individual iterations there might be additional parameters depending on the stage of prototyping. This is an important process as each iteration provides valuable feedback which in terms generates new design requirements that will be implemented before carrying out the evaluation again. The individual steps in this process are not very interesting and is only sparsely reported if big design changes has been established. The resulting product from this process will be described in detail during the implementation section (Section 5).

C Initial Prototyping Overview

The PCG system underwent multiple iterations. Initial attempts were made to use Unreal Engine 4 (UE4), due to its powerful terrain tools, materials and lighting. The terrain tools are unfortunately not readily extensible through scripts, and no viable solution using Unreal Engine was found.

Early attempts were made at creating a Generative Adversarial Network (GAN) to generate realistic height maps (Beckham and Pal, 2017) to import into UE4. With a small amount of training this produced height maps with many artefacts, but indications that they could be suitable with more training and tuning of hyper-parameters (see Figure 42). However while UE4 terrains can be generated from heightmaps, this is an editor-only feature, meaning that it cannot be dynamically used at runtime. As such this approach was decided against as it was more suited to an offline developer tool than to online generation of materials.

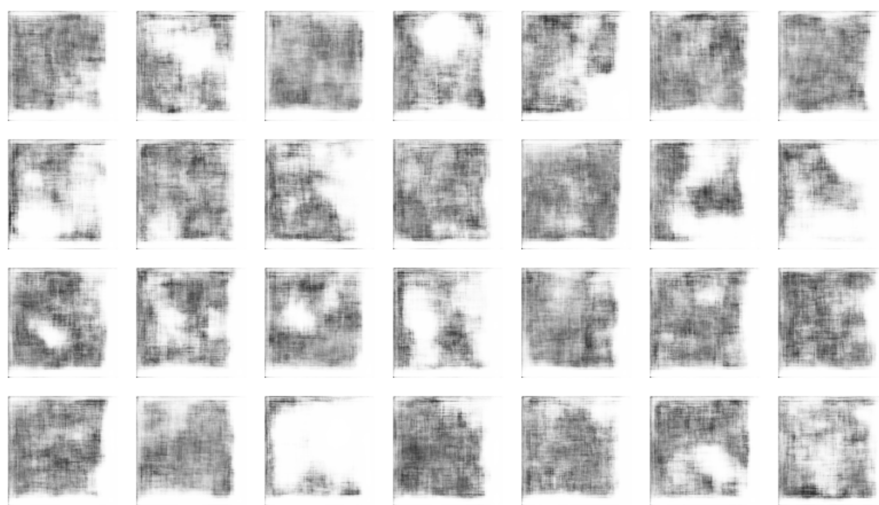


Figure 42: *Output from the height map GAN had many artefacts, but could have been improved upon*

Further attempts were made to use custom and procedural mesh tools and add-ons for UE4, but none of these solutions were deemed to be viable by the developers. As such, Unity was finally chosen as the game engine of choice for development, as it provides more tools for customisation and extension of existing systems.

Custom Mesh Terrain

The first prototype created in Unity was a custom terrain mesh. This solution involved using fBm with Perlin noise to deform the vertices of a plane. At this early stage of development it had not been decided that the narrative would take place on an island (see Section 4.5.1). As such, an early assumption was that the player could continue indefinitely in any direction. In this case, the terrain would have to continuously generate as the player traversed the environment, so they would not encounter the edge of the map.

Endless Perlin Noise Terrain As described in Section 2.1.5, using Perlin noise in this manner involves layering multiple octaves of noise, with decreasing amplitudes (controlled by a *persistence* value, and increasing frequencies (controlled by *lacunarity*).

An important part of the generation process is normalising the values generated by the noise based on the minimum and maximum heights. This helps keep the entire heightmap in the range of 0 and 1, even after applying multiple octaves.

```
noiseMap[x, y] = Mathf.InverseLerp(minHeight,
                                   maxHeight,
                                   noiseMap[x, y]);
```

Listing 4: *Normalizing the values of the heightmap*

This works well if generating a single terrain chunk, but creates seams between chunks if using continuous terrain generation. These seams are not only a visual artefact, but create gaps in the collision mesh which players could fall through.

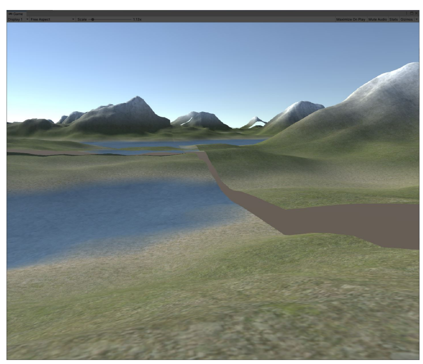


Figure 43: *Seams Between Terrain Chunks*

In order to remove the seams, it is important that each terrain chunk is normalised using the same values. Instead of using a local minimum and maximum for normalising chunks, the values should be normalised between 0 and the Maximum possible height. This can be calculated by summing the amplitude of the octaves as they are being created.

```
float maxPossibleHeight = 0;
float frequency = 1f;
float amplitude = 1f;
```

```

for (int i = 0; i < octaves; i++)
{
    // After calculating octaves & offsets ...

    maxPossibleHeight += amplitude;
    amplitude *= persistance;
}
```

Listing 5: *Finding MaxPossibleHeight*

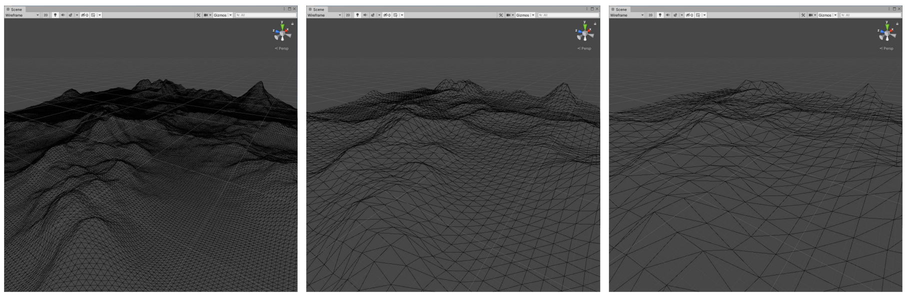
While this fixes the seams issue, it creates a new issue. The upper bound of the normalisation is so high, the normalised values tend to be very low, as very few of the noise values approach anywhere near the maximum possible height. As such, a constant value, `MaxHeightReduction` is created to reduce the maximum value to something more reasonable. Unfortunately this acts as somewhat of a "magic number", and the value varies based on the desired terrain, the height multiplier and height curve used. From trial and error, a value between 1.75 and 2.0 were deemed the best fit. A `NormalizeMode` enum was created to allow the designer to chose between normalising each chunk based on local minimum and maximum values, or global ones.

```
if (normalizeMode == NormalizeMode.Local)
{
    noiseMap[x, y] = Mathf.InverseLerp(
        MinLocalNoiseHeight,
        maxLocalNoiseHeight,
        noiseMap[x, y]
    );
}
else // Global Normalize Mode
{
    float normalizedHeight =
        (noiseMap[x, y] + 1)
        / (2f * maxPossibleHeight / MaxHeightReduction);
    noiseMap[x, y] = Mathf.Clamp(
        normalizedHeight,
        0,
        int.MaxValue);
}
```

Listing 6: *Local and Global Normalize Modes*

Level Of Detail In order to maintain decent a frame rate, while still rendering all visible terrain to the horizon, a level of detail (LOD) system is needed. This system renders the terrain chunks nearest the player at high resolution, while the resolution decreases as the chunks get further from the player. This can be seen in figure 44.

The LOD system re-introduced the seams bug however, as chunks of different resolutions would not align properly. In order to fix this, each chunk was rendered with a high-resolution border - thereby ensure the borders would always align properly, regardless of the LOD of the rest of the mesh.



(a) *(b)* *(c)*

Figure 44: *The same mesh chunk at 3 different LODs*

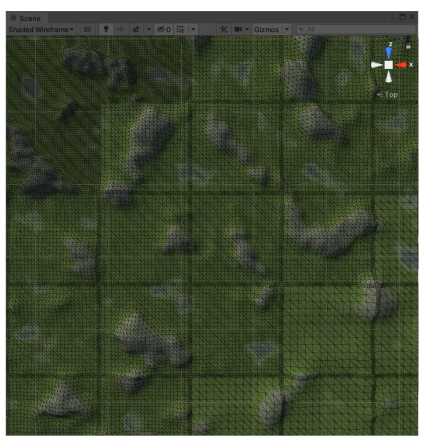
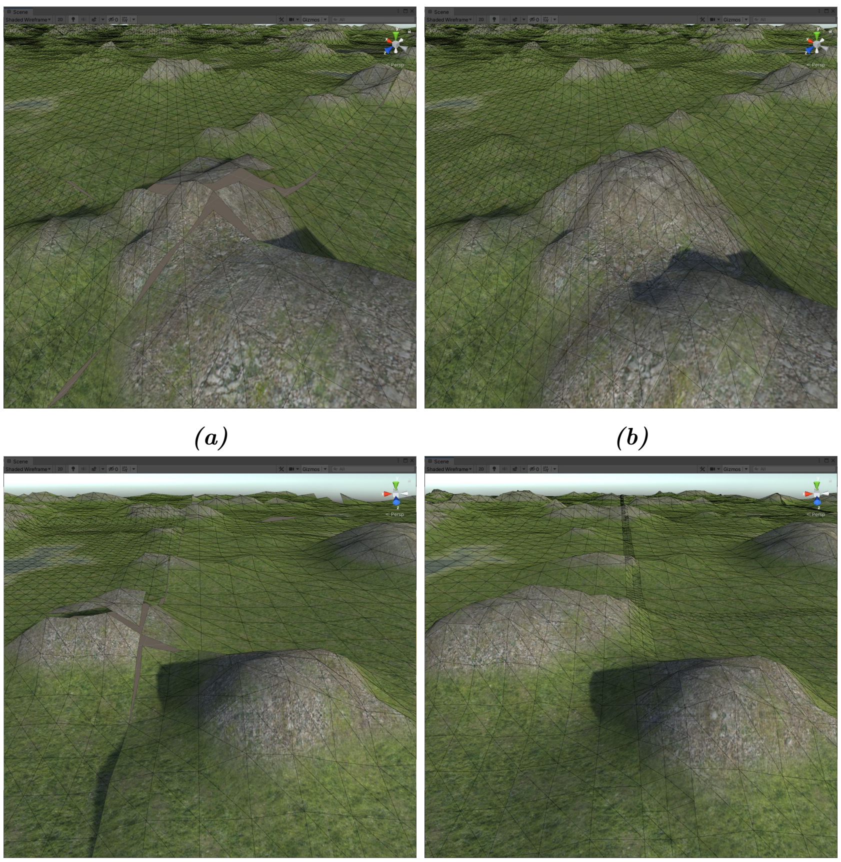


Figure 45: *Top view of multiple different LOD meshes with high-LOD borders*



(a) *(b)* *(c)* *(d)*

Figure 46: *Adding a high detailed border to each chunk fixed gaps caused by different LODs*

This still leaves a visible seam, but the subtle changes in lighting are less noticeable at a distance than a gap in the mesh. It would be possible to fix the lighting seam too, however since these should not be viewed up close due to the nature of the LOD system, this issue is of no real concern.

Collision Generation At this point the terrain generation is working well, but there is still a large performance issue. While the LOD system helps significantly reduce the rendering overhead, there is still a large amount of processing being dedicated to physics calculations, as each generated mesh bakes collisions when being generated.

This was changed so that collision would only be baked when the player approaches a new chunk. This resulted in a maximum of four active collision meshes at any time, while there would generally only be one active. Furthermore, although the meshes needing collision would be rendered with high LOD, the equivalent low-LOD mesh is used to approximate the surface for collision, further reducing the overhead.

Terrain Shader A basic surface shader which accepts an array of textures was created for the terrain mesh. Each material can be given a height range in which it will be applied (this can be seen in figure 43, where water, sand, grass, rock and snow textures are all applied at different heights). However since these meshes are generated at runtime and do not have uv maps, simply applying these textures results in stretching along the sloped areas. A simple triplanar mapping was added to remedy this. The stretching occurs on slopes as the textures are simply projected onto the Y-axis on the XZ plane. If this projection was to occur on the XY or YZ planes, the flat areas of the terrain would be stretched, while some of the slopes would be better mapped. Triplanar mapping fixes this by blending these three projections, rather than just using one. The blending is based on the absolute value of the world space surface normal at each point.

```
float3 triplanar ( float3 worldPos , float scale , float3 blendAxes )
{
  float3 scaledWorldPos = worldPos / scale;

  float3 xProjection = tex2D (baseTexture , scaledWorldPos.yz) * blendAxes.x;
  float3 yProjecection = tex2D ( baseTexture ,  scaledWorldPos.xz) * blendAxes.y;
  float3 zProjecection = tex2D ( baseTexture , scaledWorldPos.xy) * blendAxes.z;

  return xProjecection + yProjecection + zProjecection;
}

void surf (Input IN, inout SurfaceOutputStandard o)
{
    /* ... other surface shader functionality ...*/

```

```
  float3 blendAxes = abs ( IN.worldNormal );
  blendAxes /= blendAxes.x + blendAxes.y + blendAxes.z;

  o.Albedo = triplanar ( IN.worldPos , baseTextureScale , blendAxes);
}
```

Listing 7: *Simple triplanar texture mapping*

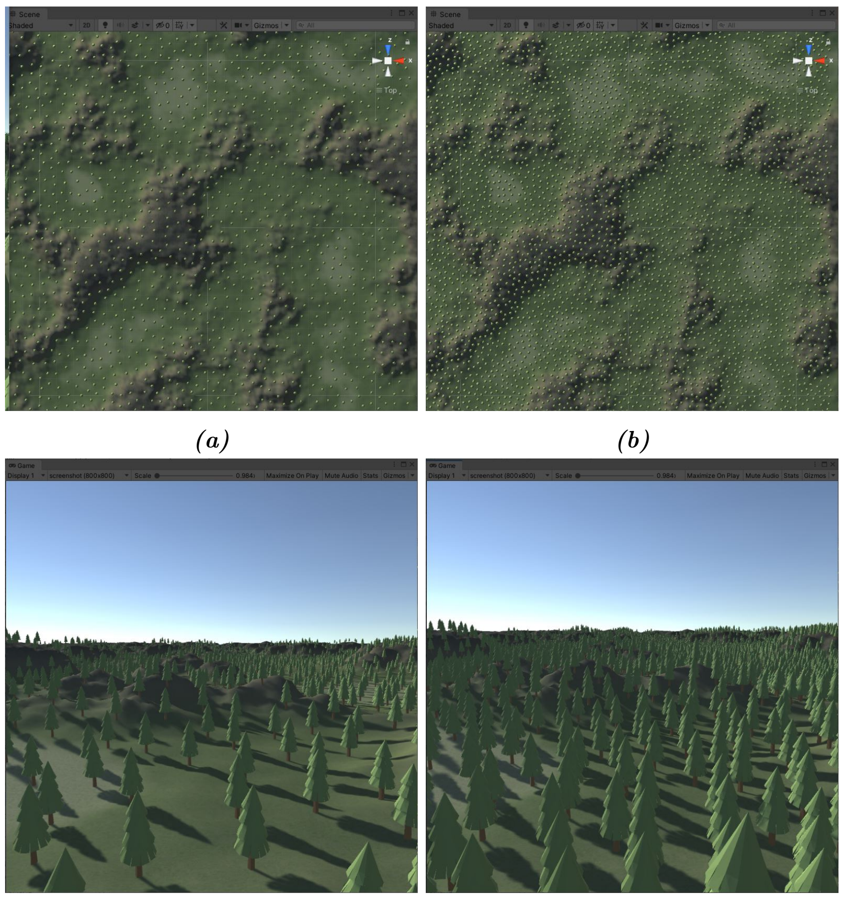
Trees & Vegetation After generating and texturing the terrain, the next step is to populate the terrain with foliage. A fast-Poisson disk sampling algorithm was implemented based on the work of Bridson (2007). This algorithm will find N sample Poisson disks separated by a minimum distance of r, mediated by a constant k which is the number of iterations to spend on each search. The algorithm functions for an arbitrary n dimension, though in this case will be considered for 2 dimensions. It is performed in three steps:

1. Initialise an 2-dimensional grid, where the cell size is bounded by $r/\sqrt{2}$. In two dimensions, this can be understood as making the diagonal length of each cell r. This therefore ensures that each cell can contain at most one sample. In this case, the list of samples can be stored as a simple 2-dimensional array of integers where -1 indicates no sample, otherwise the value indicates the index of the sample contained in that cell.
2. Select the initial sample, x0, randomly chosen uniformly from the domain. Insert it into the background grid, and initialise the "active list" (an array of sample indices) with this index (zero).
3. While the active list is not empty, chose a random index from it, and search up to k points within r and $2r$ from the sample. For each of these points, check if they are within r of another sample. If a point is far enough away, set it as the next sample, and add it to the active list. If no suitable point it found, remove the current one from the active list.

(Bridson, 2007)

This is an O(n) algorithm which produces a quite natural-seeming distribution. Points are found on the XZ plane of the the terrain mesh, and tree objects are instantiated at the Poisson sample positions.

Informal Evaluation At this point of the development, an informal internal evaluation was performed. The online generation system worked smoothly, and a player could traverse the terrain continuously in any direction without interruption as long as they wanted. The terrain was visually *acceptable*, although they looked distinctly computer generated, as opposed to based on real-world data or being hand-crafted. The triplanar mapping allowed textures to be applied without stretching, though it was somewhat jarring visually that the textures were applied in very linear height bands. Finally, while the calculation of



(a) *(b)*

(c) *(d)*

Figure 47: *Top-down and first person perspectives of trees distributed with fast-Poisson disk sampling at two different densities*

sample positions for the trees was performant, instantiating the number of trees required for the terrain, caused a significant drop in performance. Furthermore, this system did not function with the LOD for terrains, and the stochastic nature of the sampling combined with the deterministic sampling for the terrain meant that leaving an area and returning would result in the same landscape but different forests.

At this point, it was decided that while this implementation was on a good track, the amount of work required to get this system ready for even a minimal viable product would be equivalent to re-inventing the wheel, and as such continuing on this path would be beyond the scope and resources of this project. It was thus decided to re-assess the design goals of the system, and move from this custom-mesh terrain, to use Unity's terrain tools.

D Visualising the Expressive Range of the Terrain Generator

As mentioned in Section 2.1.6, one way to evaluate and understand PCG systems is to visualise the "expressive range" of the system, based on some numerical metrics of emergent properties of the system. For this generator, *fractal dimension* is used as a metric to evaluate the "natural" qualities of the generated landscapes. Fractal dimension for this procedure is calculated with a box-counting method. In brief, this method involves the following steps:

1. Get an outline or silhouette of the object to be measured
2. Place the object on a grid of boxes, and count how many boxes touch the object, n
3. Change the scale of the boxes by a factor, s
4. Perform this step k times.
5. Plot n against s for every step. If the object is a fractal, this should result in an exponential curve.
6. Plot $log(n), log(s)$. This should approximate a straight line.
7. The slope of this line that best fits this data is the fractal dimension.

Although this method can be applied to 3D objects (Suzuki, 2007), it was deemed impractical for this evaluation to voxelize and perform a 3D analysis on the hundreds of terrains that would be needed to run each time the parameters would be changed.

As such it was decided to split the height maps into multiple height bands using K-Means clustering, and find the fractal dimension of each band. Then using a dimension reduction process such as PCA or t-SNE, the fractal dimension of multiple height bands can be visualised in two dimensions.

While this is somewhat more abstract than the other measures as presented in Section 2.1.6, it can still provide at least an overview of how broad the expressive range is, or if the terrains being produced are all overly similar.

While Fractal Dimension is an interesting measure of the natural qualities of a terrain, it was probably not the best tool for this type of evaluation. This evaluation is very unperformant: it firstly requires the generation of a large data set, followed by numerous expensive operations including K-Means clustering, followed by k iterations of a thresholding operation, for each n item in the data set. This had such a high demand that even with 100 samples in 6 clusters, the application would often run out of memory.

Even aside from issues of performance, the results did not give the visualisation of the generator that was hoped for. This evaluation was run on a data set comprised of heightmaps generated with the individual techniques used, in addition to the final, combined generation. It was hoped that this could give some

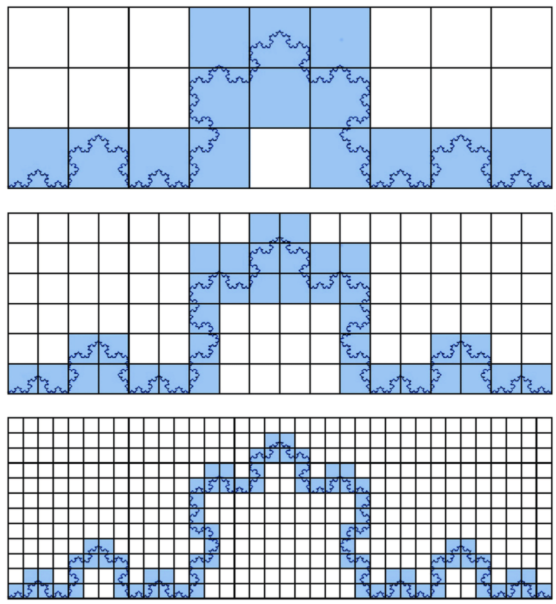
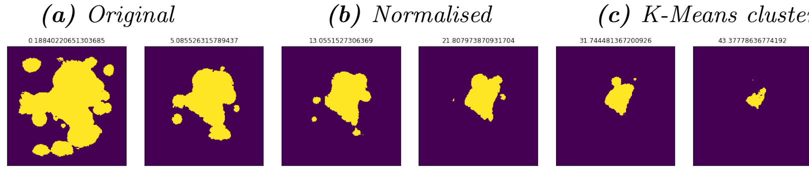


Figure 48: *An Illustration of how the Box-counting method is used when calculating Fractal Dimension of a Koch curve*

(a) *Original* ***(b)*** *Normalised* ***(c)*** *K-Means clusters*



(d) *Separated layers from k-means clustering*

Figure 49: *K-Means clustering and segmentation of heightmaps for analysis*

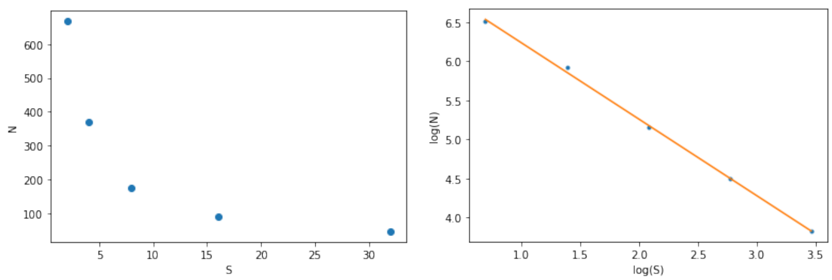


Figure 50: *When plotting the number of boxes hit (N) at different sizes factors (S), the data follows an exponential curve. Log(N) plotted against log(S) therefore produces a straight line, the slope of which is the Fractal Dimension*

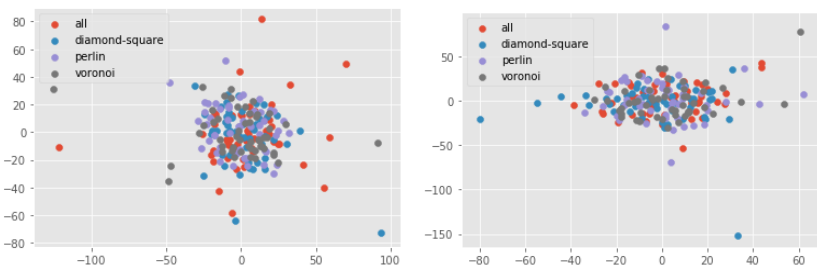


Figure 51: *t-SNE plots do not give an intuitive understanding of the generators output. It could be interpreted that most of the generated content shares similar features, though there are some outliers that produce quite different content.*

idea of how the different algorithms influenced the final output. A t-SNE visualisation of this data shows one large cluster in the centre, both when applied to the raw data and the after reducing the data with PCA.

This is somewhat interesting, but ultimately not nearly as useful as the expressive range visualisation proposed in Section 2.1.6. Rather than specifically trying to represent natural qualities, future works in this area could be better served by finding emergent measures that quantify the suitability of an environment for the embedded narrative, such as the number of possible areas for SAs. However, it is difficult to find such measures that can be taken just from the terrain generators output, particularly when using the STDM for dynamic SA placement. It could be performed based on results from the evaluation and players' experience. This too has its own issues, firstly that it would be very slow to generate sufficient data when relying on player-testing, and secondly that it would probably come about at the end of the processes, rather that between iterations, so would only provide data to explain the system, rather than help designers change the system.

E Backstory for Narrative

- In a nation the government started to pick up some electromagnetic signals from on of their minor remote islands.
- It has never been inhabited so they decided to send some of their top researchers along with a small collection of military personnel out to the island to set up camp and find out where these signals comes from.
- The military set up a base camp in the middle of the island along with a tall radio tower to enable communication with the main land and to help the researchers tools.
- The whole operation was blacked out and no one knew about this. The island was therefore also deleted from sea maps to cover up the operation.
- After a few months, everyone on the island started to feel a bit weird. There was this creepy feeling on the island as if there was a force affecting everyone there.
- The highest ranking officer on the island was a colonel from the military.
- After some time he started to isolate himself and act different around the other people.
- After half a year on the island, the researchers still haven't solved the mystery of the signals.
- The colonel had started having a lot of small private gatherings with the other military personnel and the researchers started to worry about their behaviour.
- The head of the research therefore went into the colonel's quarters late one night to talk with him but instead of seeing the colonel alone he saw most of the military personnel gathered in there.
- They were all dressed in simple clothes and had painted their faces and bodies completely red except a black sign on their chest that seemingly looked like three triangles on top of each other.
- The scientist was chocked and didn't know what to do so he sneaked back out of the quarters and rushed over to his lab.
- He just made it the door and as he opens he is faced with the colonel dressed in the same outfit as the other military personnel.
- The colonel grabs the scientist in the arm and drags him over to the radio tower where he is tied up and stripped of his clothes.
- All the other personnel now start coming forward. They have grabbed the 8 other scientist and they are now sat on the ground in front of the lead scientist in what looks like a triangle.

- The colonel comes forward with a long dagger and starts chanting in an unfamiliar language. The other starts echoing what he chants and it becomes louder and louder until the colonel steps forward and puts the dagger in the lead scientist's chest.
- The military has turned to an occult cult, worshipping the invisible force in the island.
- They quickly leaves the base camp and disappear into the forest and they are never seen again.
- They now live in primitive structures built mostly from the trees on the island and they have completely abandoned the stuff they brought in the first place.
- A fisherman is out alone one day to catch some fish.
- He has seen on his equipment that a large amount of cod was starting to gather and he therefore followed them in a direction he had never sailed before.
- After a few hours he can see an island in the distance that does not seem to show up on his sea maps.
- The cods seem to be heading towards the island so he continues to follow them closer and closer to the island.
- As he get close he is intrigued by the fact that he has never heard about an island out here and that it does not show up on the maps.
- He therefore decides to sail up close and lay anchor before boarding his dingy to go ashore.
- As he gets to the beach he can see what looks to be a man-made radio tower on the island and he decides to go and check it out.

F Script for Narrative

- Splash screen with text to introduce the narrative/game
- Fade in
- Camera looks around and stops at a tower in the distance
- Leave controls to the player
- *Player starts walking*
- **Staged Area #1:**
  - After a little while you start seeing small man-made structures and you find a small area where there is what looks like a small outpost built of wood and tarps
  - On some of the structures you see some carvings of triangles
- *Player continues walking towards the tower*
- **Staged Area #2:**
  - You come across a collection of wooden crates and ammo that seems old and left. It looks like military property due to markings on the crates
- *Player continues walking towards the tower*
- **Staged Area #3:**
  - You finally arrived at the tower. You can see that it is in the middle of a large small overgrown camp. It looks like a radio tower of some sort but it does not look functional anymore. Whoever built it and the camp probably left a long time ago. As you come to peace with the island being abandoned, you see thick black smoke rising in the distance
  - The camera pans and zooms on the smoke
  - You think you better check it out
- *Player starts walking towards the smoke*
- **Staged Area #4:**
  - You suddenly find yourself in the middle of a small clearing in the forest. You see some primitive tools laying around but you get terrified when you see on the ground there is a carving looking like three intersecting triangles with a dagger in the middle You can hear some whispers in the distance almost like chanting

- *Player starts walking towards the smoke*
- **Staged Area #5:**
  - You get to the beach and just as you do so, you see your boat is burning out on the water. You start hearing chanting all around you
  - The camera starts to fade out as the chants becomes louder. It pans around but just as it do so the camera goes completely black and the sounds stop immediately
- Shortly after, the text fades in prompting the player to answer a questionnaire