

Perception Analysis: Distinguishing Procedural from Handcrafted Quests in Video Games. Evaluating the Player Experience.



UNIVERSITY OF
LINCOLN

Gareth Mason
MAS07076286

07076286@students.lincoln.ac.uk

School of Computer Science
College of Science

University of Lincoln Submitted in partial fulfilment of the requirements for the Degree of BSc (Hons)
Computer Science

Supervisor: Dr Olivier Schockmel- Szymanezyk

May 2024

Acknowledgements

I extend my gratitude to Dr. Olivier Schockmel-Szymanezyk for his invaluable guidance and support as the supervisor for the artefact and report writing aspects of this project, he has been a great mentor in getting myself ready for a future Masters and PHD.

I would also like to express my heartfelt appreciation to my family for their steadfast support despite the challenges posed by my disability. Their unwavering encouragement has been invaluable in helping me overcome the obstacles and persevere through the pain, whilst typing this dissertation.

Abstract

In recent years, there has been a surge of interest in procedural generation techniques for quest design within the gaming industry, challenging the traditional dominance of handcrafted quests. The main question is: does procedural generation limit quest design, and can players differentiate between procedurally generated quests and meticulously handcrafted ones? Procedural generation has shown effectiveness in enhancing replayability, minimizing data volume, and generating game assets. Originally planned for human testing, this dissertation adopts a player-centric approach to evaluate player experiences with procedurally generated quests compared to human-designed ones, leveraging AI due to ethical delays regarding human participants. The dissertation chronicles the transition from human participants to AI, concluding that procedural generation components can effectively substitute quests crafted by designers, provided that their technical implementation is thoughtfully planned and executed.

Artefact URL: <https://github.com/GMWolf1988/DissertationArtefact>

Table of Contents

Introduction	1
1.1 Contribution.....	6
Literature Review.....	7
2.1 Background.....	7
2.2 Related Literature	7
2.3 Literature Relations to Project Aims & Objectives	13
Requirements Analysis	15
3.1 Requirements	15
3.1.1 Functional Requirements	15
3.1.2 Non-Functional Requirements	17
3.2 Future Testing.....	18
3.3 Conclusion	18
Methodology.....	19
4.1 Project Management	19
4.1.1 Agile Methodology	19
4.1.2 Waterfall Methodology.....	20
4.1.3 Project Gantt Chart	20
4.1.4 Risk Mitigation	23
4.1.5 Summary.....	24
4.2 Software Development	24
4.2.1 Agile Methodologies	25
4.2.2 Waterfall Methodology.....	26
4.2.3 Incremental Development & Reuse-Oriented.....	26
4.2.4 Summary.....	27
4.3 Toolsets, Version Control & Data Collection.....	27
4.3.1 Unity, Unreal & Godot	27

4.3.2 GitHub	28
4.3.3 Microsoft Forms	29
4.4 Research Methods.....	29
4.5 Conclusion	32
Design.....	33
5.1 Software Development	33
5.1.1 Requirements	33
5.1.2 Design.....	34
5.1.3 Procedural Architecture	36
5.1.4 Operation & Maintenance.....	37
5.1.5 Development Testing.....	38
5.2 Research.....	39
5.2.1 Recruitment.....	39
5.2.2 Ethics	39
5.2.3 Study Design.....	40
5.2.4 Procedure	40
5.3 Conclusion	42
Ethical Testing Revision: Implementation of AI	43
6.1 Ethics Barrier	43
6.2 AI Literature Review	44
6.3 Literature Relations to Project Aims & Objectives	47
6.4 AI Procedure.....	48
6.5 Further Contribution	51
6.6 Conclusion	52
Implementation	53
7.1 Unity Development.....	53
7.2 GitHub – Version Control	54
7.3 Artefact-Frontend	55
7.3.1 Player Journey	57

7.3.2 Handcrafted & Procedural Quest Design.....	63
7.4 Backend	64
7.4.1 Version Control & Code Activation	64
7.4.2 End Game Report	65
7.4.3 Handcrafted Quest / Chains Generation	66
7.4.4 Database.....	67
7.4.5 Procedural Quests Made Available.....	68
7.4.6 Procedural Quests Generation	69
7.4.7 AI Images	74
7.4.8 Player Actions.....	74
7.4.9 Location Transition.....	76
7.4.10 Quest Tracking.....	77
Results & Discussion	80
8.1.1 Demographic Results.....	80
8.1.2 PXI Results	83
8.1.4 GEQ Results	89
8.1.4 Results Calculations.....	95
8.1.5 Result Analysis	96
8.1.6 Aims & Objectives Review	103
8.3 Conclusion.....	104
Conclusion.....	105
9.1 Reflective Analysis.....	105
9.2 Design.....	107
9.3 Development.....	107
9.4 Research.....	108
9.5 Conclusion	109
Appendix One: Questionnaire	110
Appendix Two : AI Exit Questionnaire Results	118
Appendix Three: Artefact Code	122

References	123
Ludography.....	134

List of Figures

1 Procedural or handcrafted? – artefact.....	2
2 Initial Gantt Chart and milestones.....	21
3 Updated Gantt Chart and milestones.....	22
4. RoadWarden UI.....	35
5 Procedural architecture.....	36
6 Database.....	36
7 Ethical form 2023 submission.....	44
8 Example of the input used for ChatGPT personas.....	48
9 Example of the input used for Copilot playthrough – quest selecting.....	49
10 Example of the input used for Copilot playthrough – quest acceptance....	49
11 Example of the input used for Copilot – player travel.....	50
12 Example of the input used for ChatGPT – available actions.....	50
13 Example of the input used for ChatGPT – exit Questionnaire.....	51
14 Artefact script structure.....	53
15 Artefact developed in Unity 2022.3.12f1	54
16 GitHub version control.....	54,55
17 Initial screen displayed upon transitioning from the main menu.....	56
18 Locations.....	57
19 Handcrafted quests lists	57
20 Quest.....	58
21 Player visual feedback.....	58
22 Active quest.....	59
23 Additional actions.....	59
24 Updated objectives.....	60
25 Quest completed.....	60
26 Completed quests.....	61
27 Game completed.....	61

28 Handcrafted quest design.....	63
29 Procedural quest design.....	63
30 QuestState.cs – version control.....	65
31 PlayerState.cs – version code activation.....	65
32 PlayerState.cs – end game report.....	66
33 QuestState.cs – handcrafted quest / chains generation.....	67
34 Database.cs – tables.....	68
35 QuestState.cs -procedural quests added to _availableQuests.....	69
36 Quest.cs – quest construction.....	70
37 Objective.cs – quest type.....	70
38 Objective.cs – kill quest generation.....	71
39 Quest.cs – quest giver and location.....	71
40 Quest.cs – quest title generation.....	72
41 Quest.cs – quest description generation.....	73
42 QuestState.cs – Quest.cs – generating XP & Gold rewards.....	73
43 Location.cs – AI images.....	74
44 Location.cs - GetContextualActions.....	75
45 Actions.cs – GetContextualActions.....	75
46 Objective.cs – GetContextualActions.....	76
47 Quest.cs – turn in active quest.....	76
48 PlayerState.cs – Location change , updating actions.....	77
49 TravelLists.cs – location change.....	77
50 QuestState.cs – Player management of quests.....	78
51 QuestState.cs – reward tracking, quest completion.....	79
52 Distributions of video gamers in the US 2006 – 2023.....	80
53 Gender demographic of AI personas.....	81
54 Age demographic of AI personas.....	82
55 Play frequency of AI personas.....	82
56 PXI boxplot - handcrafted quests - Copilot.....	84
57 PXI boxplot – handcrafted - ChatGPT.....	85
58 PXI boxplot – procedural quests - Copilot.....	87

59 PXI boxplot – procedural quests - ChatGPT.....	88
60 GEQ boxplot - handcrafted quests – Copilot.....	90
61 GEQ boxplot – handcrafted quests – ChatGPT.....	91
62 GEQ boxplot – procedural quests – Copilot.....	93
63 GEQ boxplot – procedural quests – ChatGPT.....	94
64 Player experience percentage calculation.....	95
65 Player flow percentage calculation.....	96
66 Player experience and player flow.....	98
67 Perception results – handcrafted.....	99
68 Perception results – procedural.....	100
69 Persona response to exit questionnaire perceiving procedural quests.....	100
70 Persona response to exit questionnaire perceiving handcrafted quests.....	100
71 Most picked quest type.....	102
 Appendix Participant information sheet.....	110, 111
Appendix Consent sheet.....	112
Appendix Questionnaire & demographic information.....	113
Appendix PXI questionnaire.....	114,115
Appendix GEQ questionnaire.....	116
Appendix Exit questionnaire.....	117
Appendix ChatGPT persona responses (procedural quests).....	118, 119
Appendix ChatGPT persona responses (handcrafted quests).....	119, 120
Appendix Copilot persona responses (procedural quests).....	120, 121
Appendix Copilot persona responses (handcrafted quests).....	121, 122
Appendix Supporting documentation (source code).....	122

List of Tables

Table 1. 8 Factors of Flow.....	11
Table 2. Literature Relations to the Project Aims & Objectives.....	13
Table 3. Risks, Affect and Mitigation.....	23,24
Table 4. Literature Relations to the Project Aims & Objectives.....	47
Table 5. Figure 17 Key Table.....	56
Table 6. Figure 18-27 Key Table.....	62
Table 7: Copilot & ChatGPT personas.....	82
Table 8: results for handcrafted quests - Copilot.....	83
Table 9: results for handcrafted quests - ChatGPT.....	83
Table 10: results for procedural quests - Copilot.....	86
Table 11: results for procedural quests - ChatGPT.....	86
Table 12: results for handcrafted quests - Copilot.....	89
Table 13: results for handcrafted quests – ChatGPT.....	89
Table 14: results for procedural quests - Copilot.....	92
Table 15: results for procedural quests - ChatGPT.....	92

Chapter 1

Introduction

Procedural content generation (PCG) in gaming, originating with early titles like Rogue (Wichman, G et al, 1980) and Elite (David Braben et al, 1984) (Smith, 2015), addresses file size limitations and enhances replayability. It has evolved significantly, from generating 3D models to creating expansive open-world environments, seen in games like Civilization (MicroProse, 1991), Borderlands (Gearbox, 2009), and No Man's Sky (Hello Games, 2016). Procedural content is generated using algorithms, typically a series of steps or rules, allowing for limitless variation and scalability. According to Yannakakis (2015) and Bontchev (2017), modern games can benefit from PCG due to their scale, Moreover, research conducted by Antczak, J. (2023) indicates that PCG can substantially reduce costs associated with ambitious gaming projects due to excessive planning needed for elements such as side quests, the primary focus of the project.

One of the project main aims was to evaluate players' perception of quests, therefore the significance of understanding the background and structure of quests, especially handcrafted ones dating back to games like Colossal Cave (Crowther, 1976), cannot be overstated. These quests, originating in the late 1970s, have played a pivotal role in the evolution of game development. Their narrative and puzzle-centric nature has profoundly influenced iconic RPGs such as Final Fantasy (Squaresoft, 1987) and The Elder Scrolls Arena (Bethesda, 1994), and remains a vital component of contemporary gaming experiences. Meticulously designed quests significantly influence the construction of narratives, challenges, and objectives within games, enriching immersive gameplay encounters. The refinement of handcrafted quests has led to the creation of increasingly elaborate gaming scenarios, as noted by Howard (2008). Notably, massively multiplayer online games like World of Warcraft (Blizzard Entertainment, 2004) have

enthusiastically adopted these quests, integrating reward systems that actively involve players in role-playing dynamics reminiscent of tabletop games like Dungeons and Dragons (D&D) (TSR, 1974), as highlighted by Hsu, Wen, and Wu (2009).

This project evaluates players' ability to differentiate between procedurally generated quests and handcrafted ones within a 2D Text Fantasy Game as represent by gameplay in Figure 1.

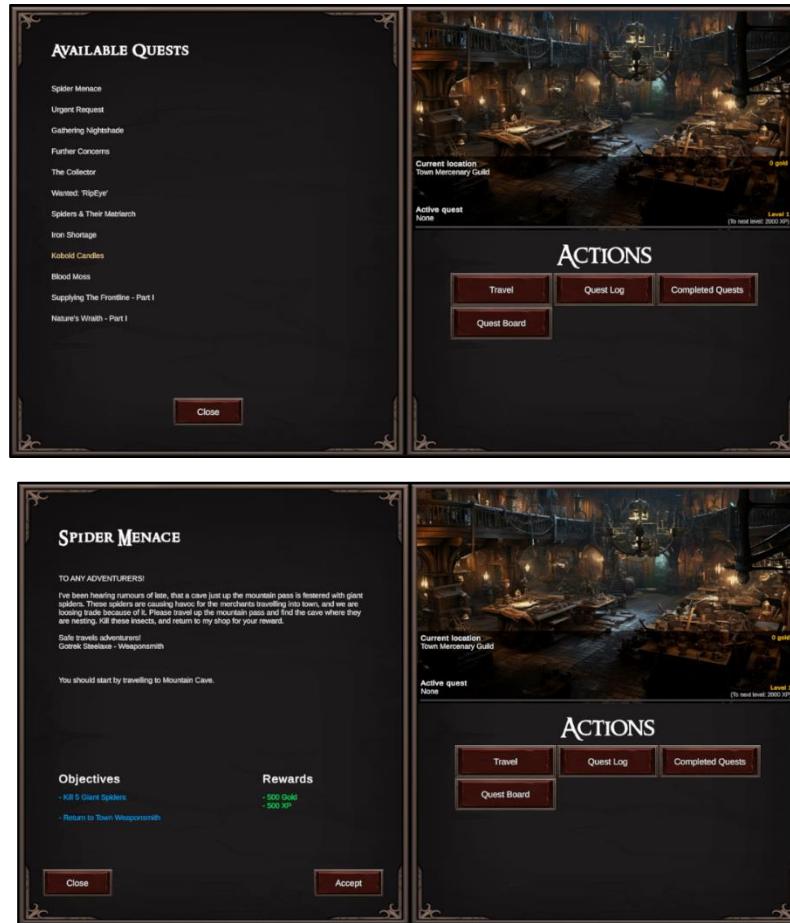


Figure 1: Procedural or Handcrafted? – Artefact.

It demonstrates the practicality of procedural content for generating side quests efficiently and assesses the viability of incorporating procedural quests into games based on the overall player experience. The evolution of PCG into in-game elements like quests is primarily driven by the expansive nature of contemporary games. *The Witcher 3: Wild Hunt* (CD Projekt Red, 2015) or *Shadow of Mordor* (Monolith Productions, 2014) feature vast worlds with numerous side quests. These quests, while having a less pronounced impact on the overall game world or narrative, require considerable effort from writers and designers. Utilising PCG emerges as a feasible approach to mitigate the time investment required for developing this supplementary content while preserving player autonomy and freedom of choice. A potential solution to address this challenge could involve the utilisation of a procedural algorithm based on tags. Quests are classified according to common archetypes found in many games, such as gathering/collecting items and defeating enemies. Each quest comprises various elements or "ingredients," including non-player characters (NPCs), items, locations, and actions involving those elements for the player to undertake. This data is meticulously tagged to facilitate the algorithm in identifying connections between them. Through straightforward random number generation, the algorithm can select a set of tags and corresponding elements, thereby shaping the quest. This methodology ensures the creation of quests with cohesive and thematically aligned compositions.

Several studies have explored the distinctions between PCG and human-crafted content in video games. For example, Thompson and Watling's (2015) research on discerning human and PCG level design revealed players' difficulties in distinguishing between the two. Although, focusing on level design rather than quests, it highlighted the challenge of discerning between human-designed and procedurally generated content, attributed to the significant influence of human designers. This raises the intriguing possibility that, with sophisticated algorithms, players may face similar challenges in distinguishing between PCG and handcrafted quests. Soares, Feijó, and Furtado's (2019) study on the procedural generation of quests further supports this speculation, though it overlooks the

crucial aspect of player experience and the ability of players to discern any differences, a gap that this paper aims to address. Furthermore, the study addresses an absence of a standardised algorithm which leaves developers relying on their skills and knowledge.

The project endeavours to assess the viability of procedurally generated quests by comparing them with intricately handcrafted quests, analysing their influence on the player experience, identifying the constraints of procedural quests, and determining players' ability to distinguish between the two types. To accomplish these overarching aims, several associated objectives were delineated. These objectives encompassed:

- Collection and refinement of diverse requirements, including the creation of the artefact and subsequent research procedures.
- Production of a Game Design document to facilitate subsequent technical advancement.
- Development of a fundamental gameplay artefact with functionalities for completing quests.
- Creation of a procedural quest's generator for utilisation in player experience research.
- Adaptation of the procedural quests generator to allow for chains, responding to player completion of quests.
- Crafting of handcrafted quests and quest chains for comparative player experience research.
- Implementation of both procedural and handcrafted quests into the gaming artefact.
- Design and utilisation of a research questionnaire tailored for gathering player experience data based on project requirements.
- Collection of data from all questionnaires from the participants and analysis of the data to assess the hypothesis regarding the quality of procedural quests and overall player experience.

- Chronicles the transition from human participants to AI, concluding that procedural generation components can effectively substitute quests crafted by designers.
- Drawing conclusions from the data analysis and reviewing the overall project for insights and recommendations. Additionally, deriving conclusions regarding the efficacy of AI feedback and the assessment of work.

The primary drive behind this project was to develop an artefact that integrated both procedural and handcrafted quests. The goal was to evaluate the player experience, driven by the hypothesis that "Players could not distinguish between procedural and handcrafted quests and the player experience doesn't diminish, thereby establishing procedural content as a viable option for side content within video games, small or expansive".

The literature review extensively extracts insights from pertinent academic sources, delving into the player experience, quests in video games, and procedural generation. Furthermore, it provides background information and justification for the project and its development, whilst also building on the author's previous interim report titled "Questing Pathways: Assessing Player Experience in Procedurally Generated Quest Chains versus Handcrafted Quest Chains" (Mason, 2024). Requirements Analysis, integrates project objectives within the development pipeline, outlining functional requirements and non-functional requirements. The methodology and design critically assess methods, tools, and research approaches, illuminating project management, development processes, and platform selection rationale. Chapter 6: Ethical Testing Revision elucidates the alterations implemented in the project during its final stages, including the transition from human participants to AI. The implementation chapter elucidates the development process, addressing objectives and research techniques employed. The results and discussion chapter presents outcomes of AI testing and compares them to initial project objectives and predicted human participant

results. Finally, the conclusion summarises project achievements, limitations, and potential avenues for future work.

1.1 Contribution

The introduction sets forth aims that have significantly shaped decision-making and contributions specifically tailored to the project's PCG and player experience.

1. By leveraging existing field data derived from procedural quests, handcrafted quests and player experience, the report sidesteps redundancy and instead leverages this data to advance novel conclusions within the project's context.
2. The report delves into the design and implementation of procedurally generated quests, aligning with contemporary gaming standards while evaluating their impact on player experience, specifically within the confines of the project.
3. Through an academic lens, the report analyses player experience using research data gathered from artefact testing, culminating in insights tailored to the project's objectives, specifically examining the feasibility of procedural content generation within the industry.

Chapter 6 Ethical Testing Revision : Implementation of AI elaborates on additional contributions pertaining specifically to AI within the project's scope.

Chapter 2

Literature Review

2.1 Background

The project focused on three main literature themes: PCG, Quests, and Player Experience. The primary goal was to explore existing algorithms for quest generation within PCG. Secondly, the project aimed to understand the creation of compelling quests, their structural components, their application to PCG, and their narrative integration to captivate players. Player experience, a crucial aspect of the project's evaluation, it was essential to understand its definition and various measurement methods, as accurately assessing player experience aligns with the project's emphasis on this key metric.

2.2 Related Literature

Procedural Content Generation

Procedural generation has significantly impacted both board and video games, with its influence evident in classics like Dungeons and Dragons (TSR, 1974) and Drakar och Demoner (Dragonbane) (Äventyrs-spel, 1982). These games utilise PCG methods to dynamically generate maps, orchestrate in-game encounters, and shape narrative elements based on player actions and random dice rolls. The strategic use of PCG methodologies enhances replayability by injecting variability into in-game environments and the characteristics of monster encounters during each play session.

PCG, defined as the "algorithmic creation of game content with limited user input" (Togelius, Kastberg, et al., 2011), has been seamlessly integrated into video game development, as noted by Blatz and Korn (2017). Developers employ

algorithms to generate game content procedurally, either before or during runtime, thereby enhancing replayability and addressing file size constraints (Amato, 2017). Pioneering PCG games like Akalabeth: World of Doom (Richard Garriott, 1979) and The Sentinel (Firebird, 1986) demonstrate the innovative use of PCG to store vast amounts of content in compact memory footprints, highlighting its effectiveness in addressing data compression challenges (Aversa, 2015).

Soares, Feijó, and Furtado (2019) highlight the absence of a standardised algorithm for PCG in quest creation, distinguishing it from specific algorithms used in level design, such as the Wave Collapse Function and L-system. Amato (2017) and Yannakakis and Togelius (2011) delve into "Experience-Driven Procedural Content Generation" (PEM), unveiling player-experience-based models that allow for customisation. Player input, as seen in the subjective PEM model through questionnaires, influences algorithm weightings. This underscores how PCG effectiveness depends on developers' design choices, shaping the system's overall functionality based on player feedback. However, lengthy questionnaires may disrupt gameplay flow and potentially impact the gaming experience negatively.

Bethesda's innovative "Radiant Quests System" for handling side quests in Skyrim (Bethesda, 2011), as highlighted by Pereira de Araujo and Souto (2017), seamlessly merges game assets to provide an endless array of new side missions. Similar systems are observed in Blizzard's Diablo 4 (Blizzard Entertainment, 2023), according to Smith et al. (2012). However, the study argues that a quest system with boundless options lacks the ability to instil profound significance, as players perceive limited impact on the world's status quo. This underscores the notion that a universal procedural quest algorithm is elusive; developers' input heavily influences the effectiveness of procedural quests, often falling short in depth compared to their handcrafted counterparts.

Quests

A meticulously crafted video game quest embodies a fusion of captivating narrative, meaningful player engagement, and a delicate balance between challenge and reward. Drawing on established authorities in game design, such as Jesse Schell's (2019) "The Art of Game Design: A Book of Lenses" and Tracy Fullerton's (2018) "Game Design Workshop: A Playcentric Approach to Creating Innovative Games," a successful quest seamlessly integrates into the overarching storyline, providing players with a profound sense of purpose and emotional connection. Additionally, it should offer diverse and intriguing challenges, sustaining player interest through a blend of exploration, puzzle-solving, and combat, as advocated by Jane McGonigal (2012) in "Reality Is Broken: Why Games Make Us Better and How They Can Change the World." Lazzaro (2004) emphasises the importance of considering player motivations to ensure a gratifying and immersive gameplay journey.

Understanding what constitutes a good quest is crucial before implementing them into PCG. A fundamental step in applying PCG methods for quest generation involves grasping the essence and structure of quests. According to Howard (2008), quests have followed the same structure since board games, where they are categorised into types such as gather, collect, and kill, simplifying quest design for designers. Initial work by Doran et al. (2010), exemplified by "Towards Procedural Quest Generation: A Structural Analysis of RPG Quests" (Doran and Parberry, 2010), significantly contributes to this understanding. This paper dissects quests specifically tailored for PCG, introducing key concepts such as quest strategies ("partial quests"), abstract quests (enhanced partial quests with additional actions), and concrete quests (derived from abstract quests through the reassignment of world assets to match labelled assets in the world). These conceptual frameworks serve as foundational elements for the initial phases of PCG components in this project, breaking down quests into their fundamental constituents and providing classifications for quest generation.

Building on this foundation, Doran's extensive analysis of over 3000 quests involves categorizing quests based on "motivation," including knowledge, comfort, reputation, serenity, protection, conquest, wealth, ability, and equipment (Doran and Parberry, 2010, 2011). Breault further extends these motivations to enhance narrative depth in his work, "Let Conan Tell You a Story: Procedural Quest Generation" (Breault, Oullet, and Davies, 2021). Here, Doran's concepts of motivation and strategy are employed to align quests with specific motivational themes. Doran himself reinforces this approach in a subsequent paper, stating, "Our use of motivations is novel, and we believe essential for ensuring that quests appear intentional and appropriate rather than randomly generated" (Doran et al., 2011). This suggests that the utilisation of motivations enables precise assignment of quests based on their actions, contributing to a more immersive experience.

Player Experience

Understanding player experience was vital for this project, focusing on procedurally generated quests and their perception. Extensively researched despite scrutiny. Mekler et al. (2014) highlights enjoyment as pivotal in evaluating player experiences, while Sweetser and Wyeth (2005) characterise its definition as inherently fuzzy. Despite this, Mekler et al. (2014) identifies specific facets linked to enjoyment, particularly the satisfaction of psychological needs. The element of control, as highlighted by Mekler et al. (2014), further contributes to game enjoyment. Mekler et al. (2014) assert that enhanced presence may not directly impact enjoyment but does enhance flow—a concept encompassing eight factors that influence player enjoyment in games (Sweetser and Wyeth, 2005), detailed in Table 1: 8 Factors of Flow.

Table 1: 8 Factors of Flow.

FACTORS.
A task that can be completed.
The ability to concentrate on the task.
The concentration is possible because the task has clear goals.
That concentration is possible because the task provides immediate feedback.
The ability to exercise a sense of control over actions.
A deep effortless involvement that removes awareness of the frustration of everyday life.
The sense of duration of time is altered.
Concern for self disappears, but sense of self emerges stronger afterwards.

Among the eight factors, quest generation for this project has a direct impact on five crucial elements: control, completion, concentration, feedback, and immersion. According to Sweetser and Wyeth (2005), players must have a sense of control over their actions to experience flow effectively. This directly aligned with the game development, where players could modify the dynamic quest generation based on their character location. In terms of feedback, players receive indications of the impact their chosen actions had. Consequently, providing feedback at appropriate times contributes to achieving flow concentration along with the ability to complete a task, as outlined by Sweetser and Wyeth (2005). Lastly, achieving immersion involves capturing the full attention of players and influencing their emotions through the games they are playing, as noted by Calleja (2011). Although the game developed was a text-based game, good writing can influence emotion as denoted by Koster (2014). In the context of attaining flow, the freedom offered by generative quests enhances immersion for the player, as their actions directly impact the quests being presented and present a presence of time being passed.

The measurement of the player experience relies on two modified questionnaires. Firstly, the PXI (Player Experience Inventory), as detailed by Haider et al. (2022), PXI empowers game designers to comprehend how elements such as game rules contribute to a player's immersion and curiosity regarding the game's narrative. In addressing a gap within Games User Research (GUR), PXI emerges as a prominent option for scrutinising participant data, as highlighted by Abeele et al. (2020). Secondly the GEQ (Game Experience Questionnaire), designed for collecting player feedback based on four constructs (absorption, flow, presence, and immersion), with flow having been subjected to analysis to validate the use of procedural quests. Flow is a psychological state experienced by individuals when they are fully immersed and engaged in an activity, typically a game or other challenging task. Coined by psychologist Mihaly Csikszentmihalyi, flow is characterised by a state of intense focus, deep concentration, and a sense of being fully absorbed in the present moment (Sanjamsai and Phukao, 2018).

Poels et al. (2007) delve into concerns regarding the use of the GEQ, highlighting challenges in drawing definitive conclusions for measuring player experience. However, as emphasized by Poels et al. (2007) and substantiated by Denisova, Nordin, and Cairns (2016), when utilised judiciously, the outcomes from the GEQ still provide a valuable foundation for evaluations. Despite acknowledging limitations identified by other researchers in the application of GEQ, it retains significance in the project, and with proper utilisation, its findings contributed meaningfully to the evaluation process.

Publications by Law et al (2018) and Vero Vanden Abeele et al. (2016) have acknowledged that every questionnaire holds the potential for misuse, leading to flawed conclusions. Nevertheless, as previously mentioned, with meticulous application, the outcomes of the study can be cross-referenced with those of others. Furthermore, the concurrent use of PXI and GEQ can yield a more holistic understanding of player experience in gaming. Furthermore, each tool offers distinctive insights, they also come with their own set of strengths and limitations.

By employing both PXI and GEQ, the project can capitalise on the complementary aspects of these tools to attain a more comprehensive and nuanced evaluation of player experience. One such advantage is Validation and Cross-Validation: Incorporating multiple assessment tools enables the researcher to validate the findings across various measures. Consistent results from PXI and GEQ enhance confidence in the accuracy of the conclusions. Using both PXI and GEQ methodologies provides a more holistic evaluation of the player experience. While PXI examines the artefact's structure and design, GEQ assesses immersion and flow. This combination offers a nuanced perspective, considering both technical elements and subjective engagement, thereby enhancing the overall understanding of the player's experience.

2.3 Literature Relations to Project Aims & Objectives

Table 2 represents how the following literature related to the project and its goals.

Table 2: Literature Relations to the Project Aims & Objectives.

GOALS.	LITERATURES.	HOW THE LITERATURE RELATES TO THE PROJET.
Structure of the quests.	The Art of Game Design. A Book of Lenses.	Both explain how quests should integrate seamlessly into the overarching storyline, providing players with a sense of purpose and emotional connection.
Applying PCG to quests.	Towards Procedural Quest Generation: A Structural Analysis of RPG Quests.	Deconstructs quests into fundamental components and establishes guidelines for their typical construction by designers, offering a solid foundation for rules to be incorporated into the procedural content generation (PCG) algorithm.

Creation of the PCG algorithm.	<p>Experience-Driven Procedural Content Generation.</p> <p>Procedural Generation of Quests for Games Using Genetic Algorithms and Automated Planning.</p> <p>Game Worlds and Creativity: The Challenges of Procedural Content Generation.</p> <p>PCG-based game design. Foundations of Digital Games.</p>	<p>Each paper emphasises the absence of a standardised algorithm for integrating procedural content generation (PCG) into quest management.</p> <p>Developers individually craft algorithms, influenced by their specific designs.</p> <p>A prevalent approach for procedural quests involves the game choosing from a pool of assets to generate the quests, a formula well-suited for the project's requirements.</p>
Evaluating the player's quest-related experience and overall player experience.	<p>Game Experience Questionnaire: development of a self-report measure to assess the psychological impact of digital games.</p> <p>The Convergence of Player Experience Questionnaires.</p> <p>Game Flow: a model for evaluating player enjoyment in games.</p> <p>Design and Preliminary Validation of The Player Experience Inventory</p> <p>PXI – Player Experience Inventory</p>	<p>Describes the Game Engagement Questionnaire (GEQ), a frequently employed approach for gauging player experience (flow), making it comparable to other research findings.</p> <p>Moreover, utilising a GEQ and PXI may present challenges in drawing definitive conclusions, previous research indicates that it can still form the basis for an effective evaluation. It will be employed for this project accordingly.</p> <p>Incorporating both PXI and GEQ methodologies yields a more comprehensive player experience assessment. While PXI focuses on evaluating the structure and design of the artefact, GEQ measuring flow. By employing both approaches, a more nuanced understanding of the player experience is achieved, encompassing both the technical aspects of the artefact and the player's subjective engagement.</p>

Chapter 3

Requirements Analysis

This chapter details the essential requirements of the project artefact, including both functional and non-functional requirements.

3.1 Requirements

Sommerville (2010) defines functional requirements as the system's services and its responses to inputs, encompassing the services, reactions to inputs, and behaviour in various situations. Non-functional requirements impose constraints on the system's services or functions, usually pertaining to the full system. The following outlines the functional and non-functional requirements for the artefact that was developed, detailing objectives.

3.1.1 Functional Requirements

Quest System

The artefact incorporates a quest system. This system manages the different quests in the game, from their creation to progression and completion. A finely crafted quest system delivers immersive gameplay experiences. The artefact enables players to complete the game for a well-rounded experience. Players receive feedback upon accepting quests, completing objectives, finishing quests, swapping quests, and reaching the game's end goal through visual cues and sound.

Procedural & Handcrafted System

The system effectively manages the creation and advancement of quests within quest chains. It also handles multiple objectives in a sequential manner. Additionally, the procedural system generated quests that were coherent and contextually appropriate based on specified "tags," ensuring logical quest generation. Once the minimum threshold for procedural generation was met, the system was to generate new quests dynamically.

Version Control & Game Report

The artefact manages version control by facilitating the selection of a game version at random for player gameplay. Subsequently, the system was required to generate a report documenting the selected version and allocate it to the designated directory within the game's folder for evaluation purposes during data collection.

Quest UI & Actions

The artefact had to provide a user interface enabling players to interact with quests, including accepting, rejecting, and tracking quest progress. The interface was required to display quest title, quest descriptions, objectives, and rewards, allowing players to manage multiple quests simultaneously without restrictions. Quest progression is dynamic, updating based on completed objectives, and triggering new player actions accordingly along with sound. Additionally, the artefact had to handle multiple locations for players to travel to seamlessly without restrictions.

Gold & XP Tracking

The artefact precisely monitors XP earned by the player, updating the total XP after completing quests. It also had to accurately track, and update Gold acquired by the player, after completing quests. Additionally, when the player accumulated enough XP to level up, the system notifies the player and adjust the level accordingly. Once the gold target had been reached, the artefact ends and restarts.

3.1.2 Non-Functional Requirements

Scalability

The quest management demonstrates scalability to accommodate an increasing number of quests without compromising performance or responsiveness. It efficiently manages quest data, processing, and interactions to ensure that all quests remained accessible and functional as the number of quests within the system grew. Scalability was achieved through effective resource allocation, optimisation of data structures, and scalable architecture design to support the dynamic growth of quest content. The system handles a substantial increase in the volume of quests while maintaining consistent performance levels, response times, and user experience across all stages of gameplay.

Player Engagement & Quest Design

The quests mimic those of modern games, ensuring that quests and mechanics captivate and immerse players in the game world. They utilise open available source materials to aid in their design. By leveraging databases such as D&D Beyond (Hasbro, 2017) and Wowhead (2006), to minimise the possibility of certain quests being disproportionately more developed than others, thereby mitigating the risk of biased results during testing. Furthermore, ensuring that quest interfaces were user-friendly, enabling players to easily comprehend quest objectives, rewards, and track progress. These considerations prioritised the ease

of use and effectiveness of the quest system, enhancing player engagement and satisfaction throughout their gameplay experience.

3.2 Future Testing

The project incorporated a combination of black, white, and grey box testing methodologies, as advocated by Mohd Ehmer and Farmeena Khan (2012) to leverage the advantages of each and mitigate their respective disadvantages. Black box testing ensured functionality and user satisfaction without requiring knowledge of the internal structure. White box testing provided thorough coverage, identifying hidden errors, and optimising the codebase. Grey box testing, by combining elements of both, offered a balanced approach, ensuring a comprehensive understanding of the software's functionality while examining the internal logic and code. This approach guaranteed a robust and reliable software system.

3.3 Conclusion

In conclusion, this chapter delineated both functional and non-functional requirements. Functional requirements detailed the specific functionalities the artefact had to demonstrate upon development, while non-functional requirements addressed aspects like scalability that shouldn't compromise its performance. Additionally, quests were expected to all follow the same design pattern to mitigate biased results.

Chapter 4

Methodology

The chapter explores project management and software development approaches, detailing selected software and research methods with rationale for each decision. It provides justifications for all choices and concludes with a summary.

4.1 Project Management

Effective project management required thorough consideration of key factors such as project type, duration, and available resources. This discussion explored various project management methodologies and assessed their suitability for the project, alongside an examination of implemented methods.

4.1.1 Agile Methodology

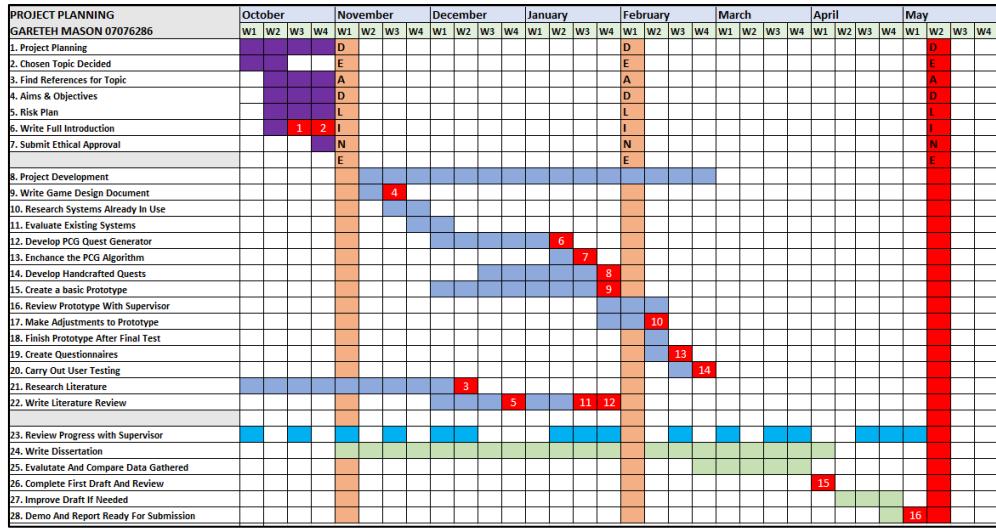
Agile methodologies, notably demonstrated by the scrum framework, prioritise implementation over documentation, as emphasised by Al-azawi, Ayesh, and Obaidy (2014). This approach is well-suited for game development, accommodating rapidly changing requirements (Kristiadi et al., 2019). Moreover, beneficial for quickly delivering functional products, particularly in research-oriented projects, this focus on rapid delivery may compromise meticulous quality, especially in development-centric initiatives. Fowler and Highsmith's (2001) Agile Manifesto emphasise "Working software over comprehensive documentation," promoting swift implementation. The scrum framework allows for responsiveness to evolving design documents but requires a careful balance between adaptability and meeting deadlines. Agile methodologies are suitable for flexible projects, recognising the challenge of balancing responsiveness and quality, which is particularly crucial in research-focused endeavours.

4.1.2 Waterfall Methodology

In contrast to the adaptable nature of Agile methodologies, the waterfall methodology operates with minimal expectation for requirement changes, adhering strictly to sequential completion of each step (Sommerville, 2011). Known for its emphasis on comprehensive documentation, the waterfall approach is beneficial in fields like game design and research. This meticulous documentation supports the design of questionnaires based on the artefact or vice versa, improving research outcomes. Although effective for overall project management, the waterfall methodology's rigidity requires significant upfront planning, leading to additional time investment in detailed preparations.

4.1.3 Project Gantt Chart

The project began with a Gantt chart as illustrated in Figure 2, to monitor progress and tasks throughout its duration, pivotal in assessing advancements during supervisor meetings and establishing goals for subsequent sessions. This chart also helped break down ambiguous objectives, such as developing PCG quests. However, delays prompted adjustments to project deadlines after the interim report and consultations with the supervisor. This extension aimed to allow ample time for comprehensive testing of the artefact and questionnaire formulation, enhancing result quality, especially as peer testing began. The modified Gantt chart reflecting these changes is presented in Figure 3. It's noteworthy that none of the Gantt charts highlighted AI testing, which was because AI was not integrated into the project until the very late stages in April 2024 due to ethical challenges, as later discussed in Chapter 6: Ethical Testing Revision: Implementation of AI.



MILESTONES	N
1. Assignment 1 Introduction reviewed by Olivier by 20th October 2023.	
2. Assignment 1 Introduction submitted by 25th October 2023.	
3. Have numerous Literature resources available to use for Assignment 2 ready by 14th December.	
4. Games Design Document MUST be completed by 19th November 2023.	
5. First Draft of Literature Review MUST be finished by 28th December 2023.	
6. PCG Quest Generator Developed by 10th January 2024.	
7. PCG Quest Generator enhanced / changes made by 17th January 2024.	
8. Handcrafted Quests developed by 22nd January 2024.	
9. Basic Prototype MUST be completed and working with BOTH sets of Quests by January 30th 2024.	
10. Prototype adjustments MUST be completed by 10th February 2024 and ready for review by Olivier.	
11. Assignment 2 - Literature Review draft reviewed by Olivier by 20th January 2024	
12. Literature Review submitted by 25th January 2024.	
13. Research and develop questionnaire for participants' 17th February 2024.	
14. Data collected by participants MUST be collected no later than 28th February 2024.	
15. First draft of Dissertation MUST be ready for review by April 1st 2024.	
16. Demo and Dissertation ready for submission by 28th April 2024.	

Figure 2: Initial Gantt Chart and milestones.



MILESTONES	N
1. Assignment 1 Introduction reviewed by Olivier by 20th October 2023.	
2. Assignment 1 Introduction submitted by 25th October 2023.	
3. Have numerous Literature resources available to use for Assignment 2 ready by 14th December.	
4. Games Design Document MUST be completed by 19th November 2023.	
5. First Draft of Literature Review MUST be finished by 28th December 2023.	
6. PCG Quest Generator Developed by 10th January 2024.	
7. PCG Quest Generator enhanced / changes made by 17th January 2024.	
8. Handcrafted Quests developed by 22nd January 2024.	
9. Basic Prototype MUST be completed and working with BOTH sets of Quests by February 28th.	
10. Prototype adjustments MUST be completed by 7th March 2024 and ready for review by Olivier.	
11. Assignment 2 - Literature Review draft reviewed by Olivier by 20th January 2024	
12. Literature Review submitted by 25th January 2024.	
13. Research and develop questionnaire for participants' 14th March 2024.	
14. Data collected by participants MUST be collected no later than April 1st 2024.	
15. First draft of Dissertation MUST be ready for review by April 15th 2024.	
16. Demo and Dissertation ready for submission by 28th April 2024.	

Figure 3: Updated Gantt Chart and milestones.

Before beginning artefact development, a thorough game design document was developed and discussed with the supervisor. This document, crucial in game development, serves as a blueprint, detailing mechanics, aesthetics, and the intended player experience, as highlighted by Macklin and Sharp (2016). It facilitated monitoring of alterations during development, and combined with the Gantt chart, ensured project progress. Furthermore, it helped maintain focused discussions in meetings by adhering to predetermined time constraints for each agenda item.

4.1.4 Risk Mitigation

Risk mitigation was essential in the project execution. "Inside Risk: A Strategy for Sustainable Risk Mitigation" by Menoni and Margottini (2011) emphasises foreseeing potential risks and taking proactive measures to prevent them. Elkington and Smallman (2002) categorise project risks into three groups: schedule and resource, product quality and performance, and business risks. Somerville's "Software Engineering 9th Edition" (2010) provides detailed insights and mitigation strategies, summarised in Table 3. In solo development, both project and product risks posed challenges, particularly in disrupting schedules and impacting project quality and data relevance.

Table 3: Risks, Affect and Mitigation

RISKS	AFFECT	MITIGATION
Handcrafted quests being of sub-optimal quality, making comparisons harder.	Product	Utilise diverse source materials to analyse quests from digital games and tabletop games. Have the supervisor review and approve the quests being used.
Quests developed by PCG – not meeting the same standards as handcrafted quests.	Product	Implement a multi-faceted approach, feedback from supervisor, reviewing both sets of quests side by side. Tighter scripting using single algorithms.
The PCG algorithm is stuck in an impossible condition.	Product	Rigorous testing throughout the development of the PCG quests.

Algorithm complexity and performance.	Product	<p>Continuously optimising the PCG algorithm, to improve its performance, making it more efficient in generating quests.</p> <p>Regularly review and refine the code to reduce complexity and enhance speed.</p>
Data is not substantial to prove hypothesis.	Project	<p>Deploy a modified PXI & GEQ questionnaire to accurately gauge player experience and flow by collecting pertinent data.</p>

4.1.5 Summary

In summary, Agile methodology, coupled with a Gantt chart, was chosen for project management. Agile's adaptability suited the multifaceted project requirements, facilitating concurrent execution of phases like artefact development, questionnaire creation, and reporting findings. This choice ensured alignment of collected data with the artefact's needs.

4.2 Software Development

Handling procedural content posed unforeseen challenges, demanding extensive testing and development resources. Therefore, the selected software development methodology accounted for various features outlined in the requirements, covering gameplay and quests—both procedural and handcrafted. This complexity stems from the nuanced nature of quest generation involving procedural content, relying on developers' expertise, as discussed in Chapters 1 and 2.

4.2.1 Agile Methodologies

According to Choudhary and Rakesh (2016), Agile methodology stands out as the predominant approach in software development. Its methodologies, including extreme programming, rapid prototyping, and scrum, enable swift adjustments to project requirements.

After exploring various development methodologies, agile emerged as the most appropriate choice. This decision was influenced by the unpredictable nature of game development and procedural generation, which posed significant challenges to the rigid structure of waterfall methodologies. Specifically, Scrum was selected, allowing for detailed objective delineation through frequent supervisor meetings. This approach enhanced adaptability in addressing changes or setbacks, proving effective throughout the project.

Scrum originated from collaborative efforts in software development, formalised and popularised by Jeff Sutherland and Ken Schwaber in the early 1990s. It effectively manages iterative and incremental projects by establishing and prioritising a backlog of tasks essential for building a functional system (Srivastava, Bhardwaj and Saraswat, 2017). Unlike adhering to a predetermined phase sequence, Scrum allows ongoing task prioritisation throughout project progression. Extreme Programming (XP), conceptualised by Beck in 1999, focuses on continuous delivery and iterative software development based on ongoing feedback and improvement (Sachdeva, 2016). However, for this project prioritising establishing measured elements over continuous refinement, the iterative nature of XP may not be the most suitable approach. This was due to the project's focus on putting essential elements in place rather than continuously refining them, along with the potential hindrance of progress caused by repeated input throughout the process.

4.2.2 Waterfall Methodology

The waterfall methodology contrasts sharply with Agile approaches. Agile thrives on evolving requirements, fostering flexibility and adaptation throughout development (Ashmore and Runyan, 2014, pp.180–182). Conversely, waterfall relies on meticulous pre-development planning, assuming minimal changes. Handling requirement changes illustrates this contrast: Agile integrates alterations seamlessly, while waterfall often requires initiating entirely new development phases. Bassil (2012) underscores the sequential nature of waterfall phases, lacking overlap due to its rigid structure. Moreover, this may focus development in specific areas, extensive upfront planning may limit adaptability to changing requirements. Research suggests waterfall is suitable only for well-understood projects with stable requirements (Dolittle, 2020). Given the project's PCG nature, with unpredictable outcomes, multiple testing rounds were necessary, posing challenges for waterfall's inflexibility. However, waterfall's meticulous documentation could benefit procedural algorithm development, offering clear guidelines throughout.

4.2.3 Incremental Development & Reuse-Oriented

Sommerville (2010) introduces two additional methodology options in 'Software Engineering 9th Edition': Incremental Development and Reuse-Oriented Software Engineering. In PCG-focused projects, Incremental Development allows iterative refinement of generated content, fostering flexibility over time. However, it may introduce complexities with each iteration, challenging consistency, and coherence across content versions. In contrast, Reuse-Oriented Software Engineering promotes efficiency by reusing PCG algorithms and modules, ensuring consistency and efficiency in content generation. However, limited availability of reusable components may restrict content variety and uniqueness, potentially leading to repetitive or predictable outcomes.

4.2.4 Summary

In summary, Agile methodologies, specifically Scrum, were chosen as the most appropriate approach for the project due to the unpredictable nature of game development and PCG. The rigid structure of the waterfall methodology would not effectively accommodate these dynamic elements. By opting for Scrum, objectives were divided into smaller, manageable tasks discussed in regular meetings with the supervisor. This approach enhanced adaptability, allowing effective responses to changes or delays as they arose, ultimately proving highly effective throughout the project.

4.3 Toolsets, Version Control & Data Collection

4.3.1 Unity, Unreal & Godot

The project evaluated three primary toolsets: Unity (Unity Technologies, 2005) Unreal (Epic Games, 1998), and Godot (Godot Engine, 2014). Unity was selected for its accessibility, usability, lower system requirements compared to Unreal's, abundant resources compared to Godot, and widespread industry adoption, especially among indie developers. Reddy Baddam (2021) underscores Unity's seamless integration of C# in its scripting API, offering developers a comfortable and robust environment to bring their creative ideas to life.

According to Johns (2024), Unity dominates with a 50% market share in 2024, while Unreal holds 15% and Godot merely 0.98%. Unity also offers over 65,000 assets, surpassing Unreal's 16,000+ and Godot's 8000+. The research notes Unreal's recent investment in 2D development. These factors are significant, especially considering the project's aim to showcase PCG as a viable industry alternative, alongside developing the game itself in 2D.

Both the author and supervisor concurred that Unreal should be considered for further proof of PCG viability in professional gaming. Its extensive usage in numerous AAA titles distinguished it from Unity. However, Unreal's higher system requirements might have necessitated more meetings to accommodate

original human participants whose systems may not have met these requirements. This could have impacted the evaluation process, potentially limiting results due to participants' inability to meet necessary system requirements.

As a result, it was concluded that Unity's utilisation of C# provided a more adaptable language due to its elegance and readability. In contrast, Godot's proprietary language, GDScript, with syntax similarities to Python, would have hindered the implementation of game-centric procedural elements needed for the study. Moreover, Godot's limited professional adoption could have posed challenges in evaluating its suitability for professional gaming contexts. C# offers straightforward and expressive syntax, facilitating the creation of functional and understandable code. Norton (2013) argues that the clarity of C# code allows developers to easily comprehend, modify, and extend existing codebases. Additionally, employing C# in Unity brings numerous benefits, such as strong typing, automatic memory management, and support for modern programming paradigms, enhancing the development workflow. Reddy Baddam (2021) suggests that these features lead to accelerated iteration, reduced runtime errors, and overall improvements in code quality.

4.3.2 GitHub

The choice to employ GitHub (GitHub Inc, 2008) as the primary version control system for this project was driven by its widespread adoption and recognition as an industry standard in software development. GitHub offers a robust platform for collaborative work, providing features such as version history tracking, branching, and merging, essential for managing the iterative process of research and writing. Furthermore, GitHub's integration with various development tools and its accessibility through web browsers and desktop applications ensure seamless workflow integration and ease of use for both individual researchers and interdisciplinary teams, as noted by Perez-Riverol et al. (2016). By utilising GitHub, the project adhered to best practices in version control, fostering collaboration between the author and supervisor throughout the process.

4.3.3 Microsoft Forms

Microsoft Forms was originally selected as the platform for gathering participant data, hosting questionnaires, interview questions, consent forms, and project debriefs. The decision offered various advantages, including streamlined data collection, centralised storage, and improved accessibility, thereby contributing to a more efficient and organised research process. Forms can also input data directly into Excel, simplifying the gathering of results. However, because of transitioning the project to AI testing, the utilisation of Microsoft Forms was subsequently discontinued, and results were recorded directly into Excel.

4.3.4 Summary

In summary, Unity was chosen for the project due to the benefits offered by C#. C# features modern and straightforward syntax, automatic memory management, type safety, and a rich standard library, making it an accessible and practical choice for solo developers. These advantages, combined with Unity's ease of use and widespread adoption, made it the preferred toolset for the project's goals. Specifically, Unity 2022.3.12f1 was selected as it was a LTS (Long Term Support) build, offering extended support and updates for a two-year period, prioritising stability, and reliability for projects. Alongside the use of GitHub, the author and supervisor decided to implement a bi-weekly physical backup system as an additional precaution against potential data loss, which could significantly disrupt the development workflow.

4.4 Research Methods

The research primarily aimed to determine if PCG could substitute handcrafted designs for quests and evaluate players' ability to discern between procedural and handcrafted quests. Additionally, the study aimed to identify the limitations associated with procedural quests and to evaluate the player experience and flow when using such quests. To address these questions, a thorough literature review was conducted to collect a diverse range of relevant papers and books. These

sources were chosen based on their perceived relevance and usefulness informing the development of the artefact.

The primary research approach involved administering questionnaires to assess player experience and flow, utilising a Likert scale to gather quantitative data. Likert scales offer benefits such as easy administration, simplicity for respondents, and the ability to quantify subjective opinions or attitudes, as noted by Joshi et al. (2015). This decision was influenced by the PXI questionnaire proposed by Nacke, Lindley, and Mandryk (2009) in their paper titled "Affective Ludology, Immersion in a First-Person Shooter: Measurement of Player Experience" published in the journal "Interacting with Computers" and GEQ questionnaire developed by IJsselsteijn, de Kort, and Poels (2007) which measures player flow. Given the similar format of both questionnaires, the use of quantitative data was necessary for comparative analysis.

Qualitative data would have been gathered through participant responses to specific questions regarding their perception of whether quests were procedurally generated or handcrafted. Additional qualitative data would have been collected via an Exit questionnaire focused on identifying how players discern between procedural and handcrafted quests. All results were to be anonymous to mitigate potential ethical implications, ensuring participant privacy and confidentiality and fostering a sense of security and trust.

Utilising both PXI and GEQ methodologies for data collection was imperative to obtain a comprehensive understanding of the player experience. PXI assesses the design and structure of quests and the overall game to gauge player satisfaction, GEQ evaluates player engagement and motivation through statements measuring a player's flow. By juxtaposing these results, a holistic view of the player experience emerges, bolstering the argument for procedural generation as a viable alternative to handcrafted quests for supplementary content. Recognising that players not only enjoy the artefact structure but also become fully immersed further solidifies this assertion.

Additional data was to be collected upon participants completing the game. This included a report exclusively accessible to the author, detailing the quests presented to the player (whether procedural or handcrafted), the time taken to complete the main objective, and the most frequently selected quest types by the player. This would have aided in comparing data gathered from all the questionnaires focusing on whether players can discern differences and exploring potential limitations in utilising procedural content from a design perspective. The investigation would have specifically examined whether certain quest types are favoured due to their design, and whether one design leads to longer playthrough times for players compared to the other.

This study employs both quantitative and qualitative methodologies. Quantitative data is crucial for statistical analysis, which can either support or challenge different assertions, qualitative data offers a deeper understanding beyond numerical measures alone, as noted by Bordage (2007). This is particularly important because one of the main objectives of this study was to assess the viability of procedural quests in comparison to handcrafted ones and the changes in the player experience.

For data analysis, guidance was provided by the supervisor, suggesting the use of box plots to analyse the research data. Box plots offer a succinct and robust method for visually summarising data distributions, facilitating comparison between groups, identifying outliers, handling non-parametric data, and providing an intuitive interpretation of central tendency and variability. Supported by research (McGill, Tukey, and Larsen, 1978), box plots have gained popularity as a valuable tool for exploratory data analysis, providing visual summaries suitable for statisticians and non-statisticians alike. To generate the box plots, Microsoft Excel was used, due to the built-in graph editor. The consideration of IBM SPSS was entertained initially; however, it was ultimately dismissed due to potential time constraints and learning curve.

A/B testing was to be conducted, to ascertain whether players can discern between procedural and handcrafted quests. Roughly 50% of players would have been assigned procedural quests, while the remainder assigned handcrafted quests. This approach ensures that players can only encounter one mode, allowing for conclusions to be drawn by comparing them separately. Ultimately, this method reveals the persuasiveness of each mode in isolation. Furthermore, research conducted by Thompson and Watling (2015) suggests that when presented with two tests, players tend to presume that everything is handcrafted if it is enjoyable, leading to a decrease in confidence when voting for procedural content. During the AI implementation phase of the project, one continued to employ this testing method, utilising the same questionnaires. Consequently, one gathered both qualitative and quantitative data, enabling to assess player experience and flow using AI personas. Further details on this are provided in Section 7.4, AI Procedure.

4.5 Conclusion

In conclusion, the project's development methodology emphasises adaptability, employing agile methods in both project management and development to accommodate the rapid pace of artefact development and facilitate adjustments to features throughout development to meet evolving research needs. Unity was chosen for its versatility, accessibility, and widespread industry usage, while GitHub for its version control and being industry standard in software development. Microsoft Forms was initially intended for streamlined data collection, centralised storage, and enhanced accessibility, facilitating data export into Excel for box plot creation. However, this approach was later replaced by direct input into Excel due to the implementation of AI testing. Both quantitative and qualitative data were utilised to enhance the evaluation once data analysis was completed, ensuring alignment with the research question's requirements.

Chapter 5

Design

This chapter explores the project's progression, focusing on its design and development phases. It covers challenges faced, solutions implemented, and key decision-making processes. Additionally, it sheds light on the research aspect, including participant recruitment methods, their impact on results, and the study's design and procedures. Since the project was originally designed for human participants, it is noteworthy that this chapter elucidates the progression from this standpoint, detailing the proposed procedure and recruitment process.

5.1 Software Development

5.1.1 Requirements

To explore the research question regarding players' perception of quests and whether their experience varies depending on whether the quests were procedural or handcrafted, thus evaluating the viability of procedural quests, it was crucial to ensure that both quest types adhered to modern gaming standards in structure and design. This approach aimed to minimise potential biases in the results, Fullerton, Schell, and Howard's studies emphasise a consistent quest structure, including types (such as kill, gather, or collect), quest giver (NPC), description (stating the goal and reason), objectives, and rewards. Adhering to these industry-established structures ensures equality across all quests. Adhering to these industry-established structures ensures quest equality. Open-source databases like Wowhead (2006) and D&D Beyond (Hasbro, 2017) were utilised to ensure quest content aligns with modern games and to mitigate biased testing results, as discussed in Section 3.1.2 Non-Functional Requirements. After identifying these elements, the next step was to incorporate them into a procedural algorithm, while

also defining the player's available actions. Further research was inspired by the academic works of Doran and Parberry: "Procedural Quest Generation: A Structural Analysis of RPG Quests" (2010). They extensively discuss the key components of a quest, such as type and action, which can be compartmentalised into separate lists. Subsequently, a procedural algorithm can be utilised to combine components from each list to formulate a complete quest.

5.1.2 Design

A design document was prepared to outline the fundamental components of the artefact, covering both front-end and back-end aspects. The user interface was developed to be clean, ensuring that all quests are presented neatly alongside their descriptions, objectives, and rewards. This design emphasises a seamless and intuitive user experience to facilitate smooth interaction, ensuring easy navigation and adaptability to various screen sizes. Players possess the capability to alter their currently active quest from their current list, thereby granting them a sense of agency in determining their actions, therefore aligning with a player's flow outlined by Liu (2017) and previously discussed research in 2.2 Related Literature. Given the text-centric nature of the artefact's gameplay, a legible font style was chosen, along with the UI drawing inspiration from modern 2D Text Fantasy games like RoadWarden (Moral Anxiety Studio, 2022) illustrated in Figure 4. Additionally, generated AI images were utilised to enhance player immersion. Players exclusively rely on mouse interaction to navigate the artefact and execute commands, including selecting quests, traveling to different locations, and performing available actions. Emphasising a clutter-free UI was paramount to ensure players' undivided attention was on the selected quest.

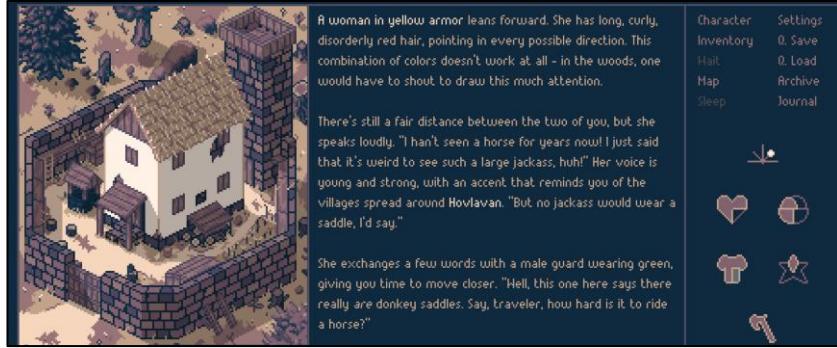


Figure 4: RoadWarden UI.

Maintaining game progression was crucial to sustaining player interest during gameplay. Aligned with the research outlined in section 2.2 Related Literature – Quests, it is important that players are motivated to complete the tasks set out before them to avoid breaking immersion. In this case, the player was presented with a backstory to their protagonist along with the aim of accumulating gold to proceed with the main story. Although the main story is never touched upon, it gives the player the motivation to complete the side content as it relates to one of the questions found in the PXI questionnaire: “I wanted to find out how the game progressed.” Another aspect of game progression was having the rewards clearly visible to the player. In this case, quests rewarded gold or XP, sometimes both. Though gold was the primary progression element, incorporating a level-up system akin to those in modern RPGs such as Final Fantasy (Squaresoft, 1987) - though primarily cosmetic in this context - offers players rewards and sustains engagement for longer periods, as suggested by Wang and Sun (2015).

The back end encompasses the coding aspects of the artefact, the procedural algorithm for dynamic quest generation, and incorporating manually crafted quests into the system. Furthermore, storing quest-related data such as NPC details, mob types, locations, objectives, and rewards,

5.1.3 Procedural Architecture

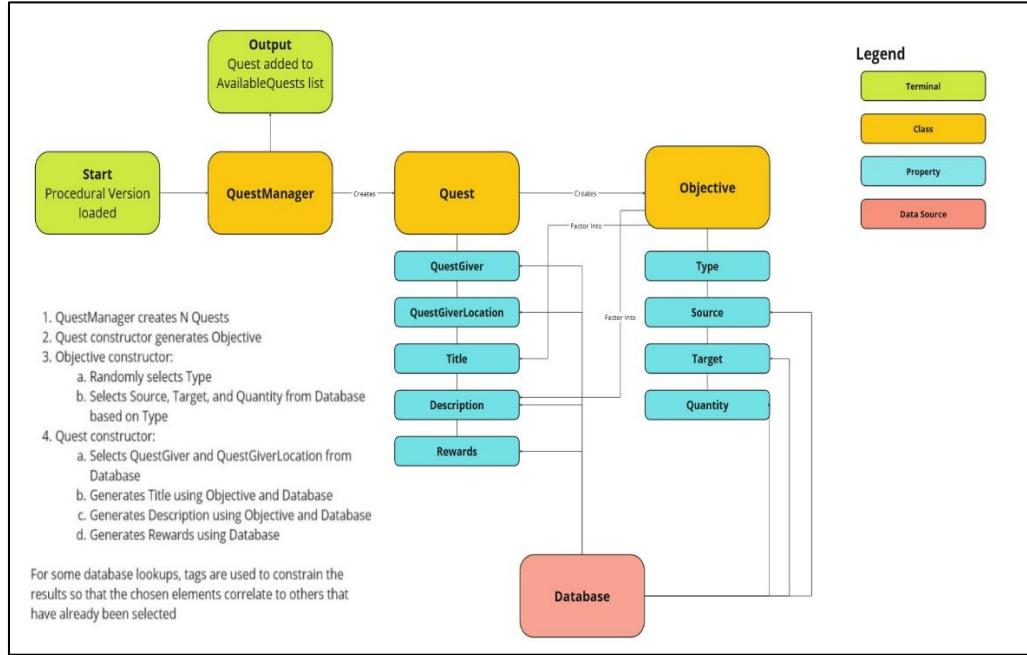


Figure 5: Procedural architecture.

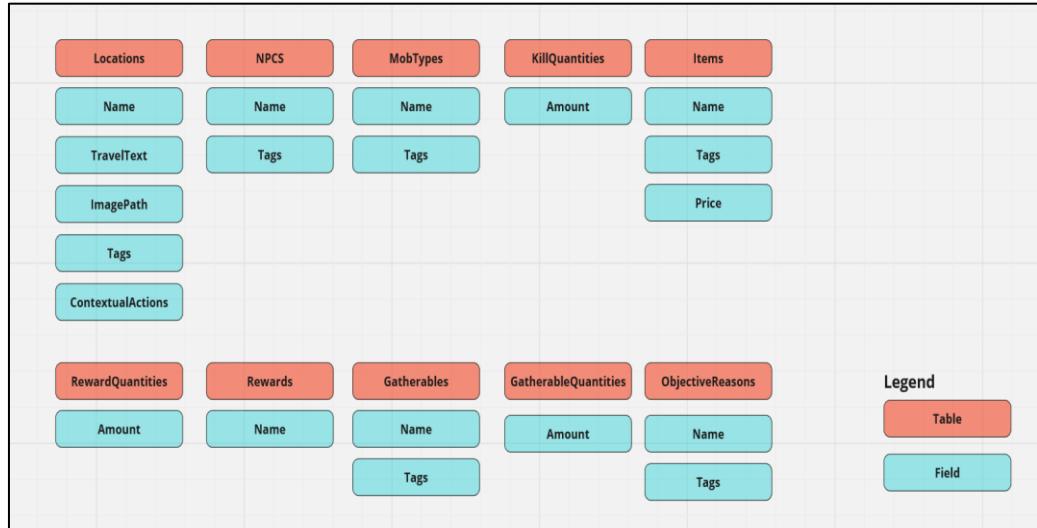


Figure 6: Database.

Figure 5 depicts the procedural system's architecture, illustrating how the system generated quests when the procedural version of the artefact is loaded.

Figure 6 illustrates the database, showing the tables and fields utilised in quest creation. The Database.cs is akin to a relational database, containing lists of data like database tables, and methods for retrieving items from the lists, comparable to using a query language such as SQL. The QuestManager acts as the entry and end point of the system, taking the created quest and adding it to the list, which is then displayed to the player. By utilising a database, handcrafted quests can directly select rows from a designated table to generate objectives, locations, NPCs, and rewards in alignment with the quest description.

Handcrafted quests are designed to ensure the completion of the artefact regardless of the player's path, whereas procedural quests continue to generate as their rewards are randomised, thus ensuring that the game remains playable until the overall goal is achieved. Additionally, this mechanism enhances immersion by creating a perception of time passing within the game, a factor that, according to research by Reis, Luís Paulo Reis, and Lau (2019), contributes to player engagement.

Important to note that the original design included price tags for items for the player to buy and sell to a vendor, and this code is still visible in the final build. However, due to the project's focus on testing quest perception, this design along with a morality system and RNG combat, like that of Baldur's Gate (BioWare, 1998), were benched after discussions with the supervisor. This decision was made to maintain the core aim of the project as the centre point and to mitigate excessive feature creep during development.

5.1.4 Operation & Maintenance

Upon launching the artefact, the main menu would be encountered, offering access to various features including credits (asset references) and rules providing objectives and backstory for the player protagonist. The game randomly selects between procedural or handcrafted quests upon initiation. Operating the artefact involves standard mouse movement and left mouse button clicks to select quests,

locations, and actions. The objective is to complete quests and accumulate 4000 gold, once achieved the game will end.

Refactoring the generation algorithm would support ongoing development, aligning with the agile approach used in this project. Potential advancements could resemble those seen in No Man's Sky (Hello Games, 2016), where NPCs and their attributes are procedurally generated. Furthermore, the project was built using a Unity long-term support build, ensuring compatibility with Unity updates until 2025, as stated on Unity's website (Unity, 2023).

5.1.5 Development Testing

As delineated in Section 3.2 Future Testing, a blend of black, white, and grey box testing methodologies was employed throughout the development phase. Black box testing involved examining the entire game run and individual quest components for any failures, while white box testing focused on assessing new feature additions for compatibility with existing code and future builds. Tests were conducted by an external peer and the supervisor, both not involved in the research study. These testing methods were chosen based on their widespread use in software engineering, aligning with findings by Nidhra and Dondeti (2012) emphasising the importance of testing both software specification and code accuracy. Additionally, studies by Ehmer and Khan (2012) and Verma, Khatana, and Chaudhary (2017) highlight that black box testing evaluates software functionality, while white box testing examines its internal structure, both deemed crucial for software testing. Grey box testing, described by Ehmer and Khan (2012), involves partial knowledge of the software's internal structure, which aligns with the supervisor's involvement in testing. Prior to participant recruitment, a final black box test was conducted by the non-participant, simulating participant steps to identify any issues. During final testing, it was determined that quest chains would be excluded from the procedural version of the artefact. Over time, quests became increasingly illogical, with objectives failing to align properly with quest descriptions. Quest chains were initially a

primary objective of the artefact development, but discussions with the supervisor and an external peer involved in the testing concluded that omitting quest chains from the procedural version would not compromise one of the project's core aims of assessing and evaluating quest perception. As the blind test only exposed participants to one version, the procedural quest generation could proceed while maintaining logical coherence. Hence, this original objective was removed from the initial list.

5.2 Research

5.2.1 Recruitment

Recruitment would have primarily relied on social media and word of mouth to gather a diverse sample of 15+ participants aged 18 and above. Participants would have been asked about their gaming frequency and informed about the 20–30-minute test duration. They would have accessed information sheets and consent forms via Microsoft Forms without disclosing personal details, with debriefing sheets and questionnaires also provided through the same platform. It was anticipated that, had testing occurred, results could have been influenced by participants sharing the same university course and social circles.

5.2.2 Ethics

Ethical approval for human participant involvement in the project was to be obtained through the University of Lincoln's LEAS system before commencement. Each participant would have received information and debriefing sheets via Microsoft Forms, detailing study procedures, contact information for addressing issues, and age requirements, before accessing the questionnaires. Informed consent would have been obtained from each participant before beginning the study, ensuring adherence to ethical standards.

5.2.3 Study Design

The research hypothesis was established as "Players could not distinguish between procedural and handcrafted quests, and the player experience doesn't diminish, positioning procedural content as a viable option for video game side content". In Section 4.4, Research Methods, questionnaires adapted from the PXI and GEQ were primarily employed for data collection. The PXI, developed by Nacke, Lindley, and Mandryk (2009), and the GEQ, created by IJsselsteijn, de Kort, and Poels (2007), are widely utilised in academic research to evaluate player experience and flow.

5.2.4 Procedure

If consent was given, the study's initial phase would have begun with a brief set of questions aimed at gathering participants' demographic information. To comply with ethical guidelines, this data would have been anonymised and only identify, age, gender, and experience with video games.

Participants were to be randomly assigned to either the procedural or handcrafted quest version of the game, following the methodology outlined in Section 4.4 Research Methods. Roughly 50% of participants were to play the procedural quest version, while the rest were to experience the handcrafted quest version. Both versions of the game shared identical gameplay mechanics, with players aiming to accumulate 4000 gold by completing side quests.

Upon completion, participants were to fill out modified PXI and GEQ questionnaires, both utilising a Likert Scale to assess their agreement with various statements and evaluate their overall experience and flow. Additionally, participants were to be asked a specific question at the end of the questionnaires: whether they believed the version they played featured procedural quests or handcrafted. This would have assisted in evaluating the overall player experience, as well as assessing whether players could distinguish between procedural and handcrafted quests.

The final phase of the study involved collecting qualitative data from participants through a 3-question exit questionnaire. These questions aimed to uncover participants' beliefs about the version of the game they played, their awareness of the game version, and which quests appealed to them the most and why. Addressing these questions provided valuable insights into players' understanding of PCG, their ability to identify procedural generation over handcrafted designs, and which quests influenced their perception, ultimately assessing the viability of procedural quests. Although, the final inquiry delves into preferred quest types, it holds less statistical weight compared to other questions. However, it remains valuable in shaping generalised assumptions about the study's outcome, although it should not serve as the definitive conclusion.

If a significant number of participants were to guess incorrectly, it suggests well-developed procedural quests, supporting findings from previous research by Thompson and Watling's (2015) and Soares, Feijó, and Furtado's (2019). Conversely, if participants guess correctly but still enjoy the experience along with no dimmish to player flow, it indicates the viability of PCG, albeit with the possibility of producing less detailed quests like games such as Sea of Thieves (Rare, 2018) compared to handcrafted ones. Furthermore, if participants accurately identify procedural generation and encounter a less immersive player experience, it could still support the viability of PCG, as other factors, such as the artefact's presentation lacking the flashiness or modernity found in contemporary games playing a role in diminishing engagement for some participants, hence the need for future testing with more polished aesthetics as suggested in Section 4.3 Toolsets. Nevertheless, the widespread success of PCG across different gaming contexts indicates its potential effectiveness despite these obstacles. This notion is supported by research conducted by Pereira de Araujo and Souto (2017), who highlighted games like Skyrim (Bethesda, 2011) employing PCG for side quests. Despite players' perception of procedural generation in these instances, their experience remained unaffected. This lack of diminishment could be attributed to

procedural content primarily serving as supplementary material, distinct from the main storyline, and to the AAA quality design of the game itself.

The private game file report, exclusively available to the author and located within the artefact directory, provides a summary of the version played, completion time, and quest preferences. This would have facilitated the evaluation of participants' ability to correctly identify their game version and identifies notable quest choices. Any shortcomings identified were attributed to algorithmic deficiencies, as discussed in Chapter 2's Literature Review, underscoring the significance of developer expertise. Insights gleaned from these data sources were crucial in formulating a credible conclusion.

5.3 Conclusion

In conclusion, testing in this chapter employed black, white, and grey box methods, often with input from an external collaborator and project supervisor to ensure artefact stability. The selection of Unity aimed to ensure long-term maintenance feasibility due to its robust support for build versions. Development commenced by drawing from existing studies to accurately design quests and utilising open-source databases, which were central to the study's authenticity. Recruitment and research, although never conducted, would have adhered to ethical guidelines established by the University of Lincoln, involving submission through the LEAS system.

Chapter 6

Ethical Testing Revision: Implementation of AI

This chapter outlines the justification for shifting from human participant testing to AI testing through the utilisation of machine learning Large Language Models (LLMs) like ChatGPT 3.5 (OpenAI, 2022) and Copilot (Microsoft, 2023). Furthermore, it investigates relevant literatures in the gaming industry that employs AI, shedding light on its relevance to the current study. Although, the integration of AI occurred months after the implementation of the artefact, juxtaposing both procedures allowed for comparative analysis. As discussed with the project supervisor, the integration of AI to assess quest perception, player experience, and flow represents a groundbreaking advancement and promises to ignite future studies in this domain.

6.1 Ethics Barrier

In the project's concluding stages, it became evident that the ethical review team at the University of Lincoln had unintentionally neglected the author's ethical submission, despite reminders from both the author and the supervisor. Figure 7 displays the submitted ethical form, labelled as "under review." Subsequent discussions with the supervisor and another staff member unveiled that the ethical officer failed to provide feedback on the form back in January 2024 and was currently on leave, potentially extending the approval timeline. Consequently, the lack of ethical approval hindered the progression of testing and data collection endeavours.

The screenshot shows a software application window titled 'Project Tree'. Inside, there is a tree view with nodes: 'SoCS Project/MComp Research Project Module Ethics Applications (HP)', 'Ethics Application / Project Registration Form', and 'Human research Sub Form - Gareth Mason'. Below the tree is a table with four columns: 'Action Required on Form' (No), 'Status' (Under Review), 'Review Reference' (2023_13320), and 'Date Modified' (31/10/2023 12:17).

Action Required on Form	Status	Review Reference	Date Modified
No	Under Review	2023_13320	31/10/2023 12:17

Figure 7: Ethical form 2023 submission.

6.2 AI Literature Review

Using two AI instead of one was advantageous because each is trained on distinct datasets and architectures, resulting in unique linguistic capabilities and decision-making processes. This approach allowed the study to capture a wider array of perspectives and responses, thereby enriching the evaluation process. Additionally, according to research by (Lian et al., n.d.), employing multiple LLMs helps alleviate potential biases or limitations inherent in any single model. Should one model display peculiar behaviour or shortcomings, the other model can offer a contrasting viewpoint, thereby bolstering the reliability of the findings. Moreover, the consistency or disparity between the responses of the two LLMs can serve as a form of cross-validation. If both models consistently identify certain aspects of the quests (e.g., procedural elements), it strengthens confidence in those findings. Conversely, differences between the models can prompt further investigation into the nuances of the quests. Therefore, utilising two LLMs for testing procedural quests viability enhances the robustness, reliability, and generalisability of the study's findings, while also shedding light on the diversity of linguistic capabilities and decision-making processes exhibited by different models.

The utilisation of AI in the gaming industry is increasingly under scrutiny. According to LANKES and STOCKL (2023), AI has the potential to aid in the initial planning phases of game development, potentially revolutionising the

industry by offering innovative and efficient solutions to the challenges faced by game designers. To test this hypothesis, researchers explored the use of ChatGPT to facilitate discussions between designers and AI in a context-sensitive manner. This approach aimed to provide interactive feedback on various topics such as game mechanics, initial design concepts, and functionalities. LANKES and STOCKL (2023) further evaluated the effectiveness of this approach, noting experts' observations regarding the program's encouragement of experimentation and its role in laying a solid foundation for further design endeavours. However, experts also critiqued ChatGPT for its tendency to repeat certain statements and its lack of depth in content. Interestingly, experts often anthropomorphised the program, attributing human-like qualities to it. In conclusion, experts believe that AI technologies hold significant potential to transform the game development industry, particularly in the realm of game systems. Additional research conducted by Pier Luca Lanzi and Loiacono (2023) highlights the utilisation of LLMs in game design, echoing findings like those of LANKES and STOCKL (2023). Both reports identify commonalities, noting limitations such as the tendency of AI to generate repetitive statements. This similarity underscores the challenges faced in leveraging AI for game design purposes and emphasises the importance of addressing such issues to enhance the effectiveness of AI-driven solutions in the field. One given solution is to use multiple LLMs which this project utilised to help mitigate repeating responses from using the same AI.

Kim and Maher (2023) discuss co-creative AI systems in design, which enable users to collaborate with an AI partner during open-ended creative tasks, enhancing design creativity by inspiring the exploration of new design solutions during initial idea generation. Meanwhile, Zhu and Luo (2022) propose a generative approach to design ideation that leverages GPT's fine-tuning mechanism to synthesise knowledge and domain understanding. Despite yielding intriguing results, the quality of output may vary. The authors emphasise that the AI system should serve as an assistant rather than a standalone designer, supporting the ideation phase and brainstorming alongside human designers to

enhance their creativity. However, existing approaches only address design in a broad sense, leaving unexplored the integration of AI systems specifically tailored to support the creation of innovative ideas and game design concepts.

As the AI was generating personas to guide the testing and outcomes, ethical considerations became imperative. Studies conducted by (Salminen et al., 2020) and (Dent, Dumond, & Kuniavsky, 2019) indicated that when utilising AI in this manner, it was necessary to elucidate the methodologies and their constraints to ensure transparency along with the transparency of the results. As AI becomes increasingly integrated into various industries, ethical regulations may evolve, necessitating future projects stemming from this research to address the ethics of AI usage. Furthermore, this project acknowledged, based on prior research, that employing AI has limitations, such as producing repetitive results. To mitigate this bias, only a limited number of personas were generated, and two LLMs were utilised rather than one. This approach bolstered the credibility of the findings by mitigating potential biases and enhancing result reliability.

In summary, while these papers delve into the integration of AI systems during the early stages of design and development, adopting these methods could potentially revolutionise the role of AI in evaluating work. Consequently, if the artefact were to be released to the public in future iterations, AI analysis could enhance both procedural and handcrafted design aspects of quests, offering substantial benefits. Furthermore, delineating the constraints of AI and maintaining transparency regarding the procedures undertaken in 7.4 AI Procedure and the findings in 8.1 Results adhere to ethical standards.

6.3 Literature Relations to Project Aims & Objectives

Table 4 represents how the following literature relates to the project and its goals.

Table 4: Literature Relations to the Project Aims & Objectives.

GOALS.	LITERATURES.	HOW THE LITERATURE RELATES TO THE PROJECT.
Implementation of LLMs to evaluate the visibility of procedural quests.	<p>ChatGPT and Other Large Language Models as Evolutionary Engines for Online Interactive Collaborative Game Design.</p> <p>AI-Powered Game Design: Experts Employing ChatGPT in the Game Design Process.</p> <p>The effect of AI-based inspiration on human design ideation.</p> <p>Generative Design Ideation: A Natural Language Generation Approach.</p>	<p>All elucidate the utilisation of LLMs in game design to aid in the initial planning and development stages.</p> <p>These concepts can be used within the testing / evaluation of the artefact in place of human testing, whilst looking out for the pitfalls as notified by the researchers.</p>
Following ethical guidelines for AI driven personas.	<p>Algorithmic Transparency and Accountability in Data-Driven Personas.</p> <p><i>A Framework for Systematically Applying Humanistic Ethics when Using AI as a Design Material</i></p>	<p>As AI evolves, ethical concerns grow. Ensuring AI-driven personas adhere to ethical guidelines is imperative, requiring transparency in all procedures and outcomes.</p>

6.4 AI Procedure

Both ChatGPT and Copilot were instructed to replicate personas by being assigned the task of determining the age, gender, and gaming frequency, imitating human responses. This method mirrored the user story approach commonly used in Software Engineering. Previous studies (Hess et al., 2024) have indicated that AI-generated personas offer a standardised depiction of potential users, facilitating a more systematic and uniform testing methodology. These AIs were adjusted to emulate personas reflecting students aged 18-25+ as depicted in Figure 8. By constraining the personas to resemble those of computer science students, it aligns with the author's assertion in Section 5.2.1 Recruitment, wherein the author anticipated human participants to be within the same social circle.

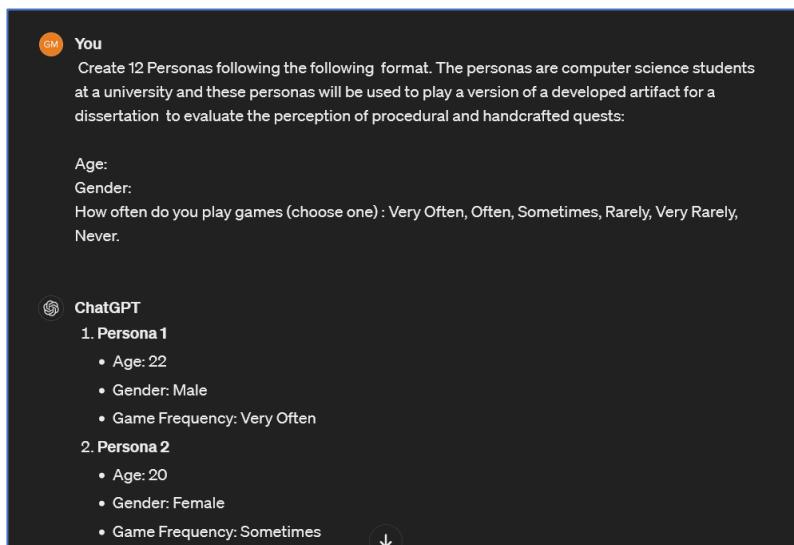


Figure 8: Example of the input used for ChatGPT personas.

The AI's chosen personas would proceed to navigate through the artefact alongside the author, who would input the available quests and pose crucial questions regarding the AI's interest in accepting quests (Figures 9-10) and selecting initial travel destinations, following the provided quest description as a reference. Notably, Figures 11 exemplifies this precise procedure, demonstrating the AI accurately discerning the required destination based on quest content. Any

introduction of new locations offering additional actions were noted, with various available actions outlined, granting the AI the autonomy to determine its preferred course of action, as illustrated in Figure 12.

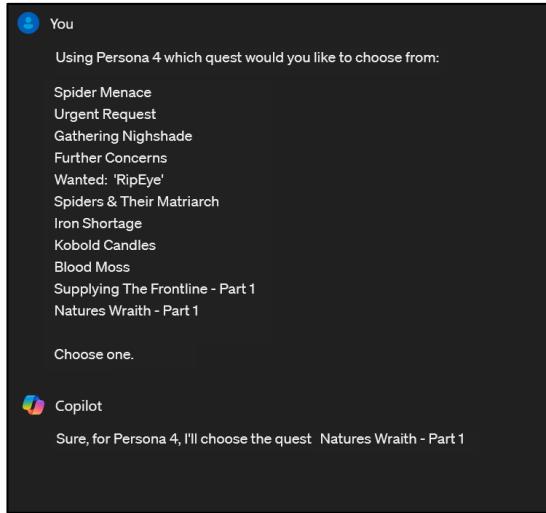


Figure 9: Example of the input used for Copilot playthrough – quest selecting.

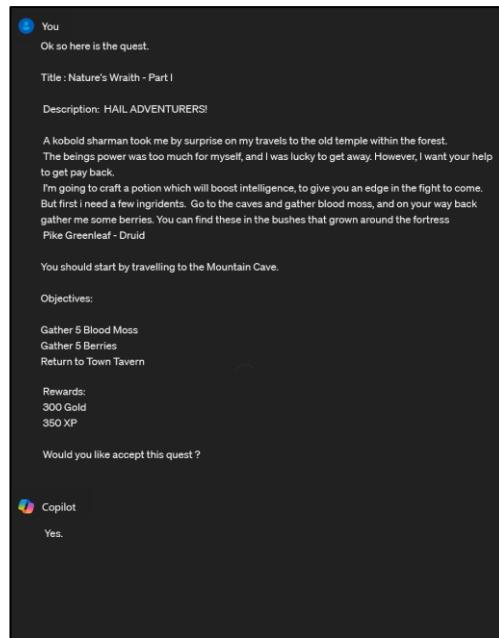


Figure 10: Example of the input used for Copilot playthrough – quest acceptance.

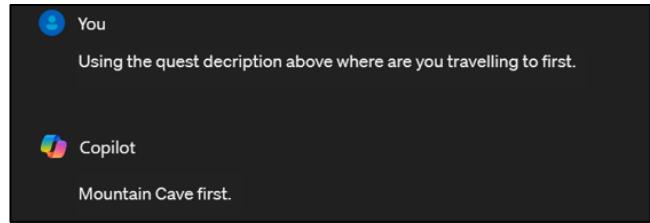


Figure 11: Example of the input used for Copilot – player travel.

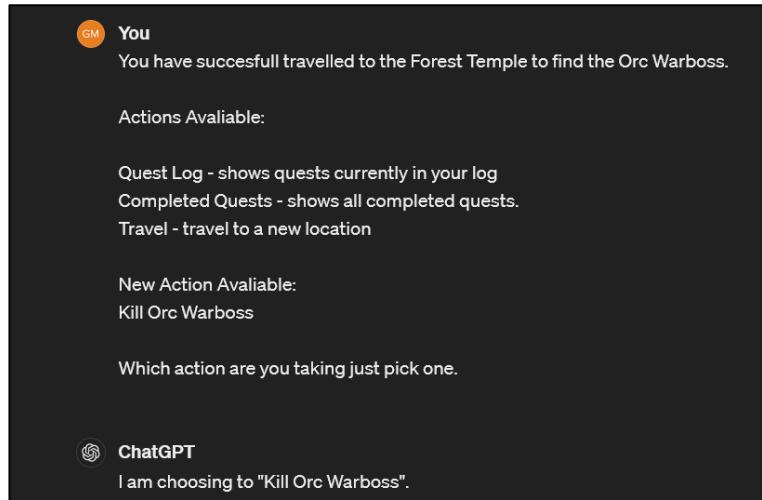


Figure 12: Example of the input used for ChatGPT – available actions.

Once the overarching objective of amassing 4000 gold had been met, the AI would proceed to complete both the adapted versions of the PXI and GEQ questionnaires utilising the Likert Scale for both, while also responding to inquiries from the exit questionnaire, as depicted in Figure 13.

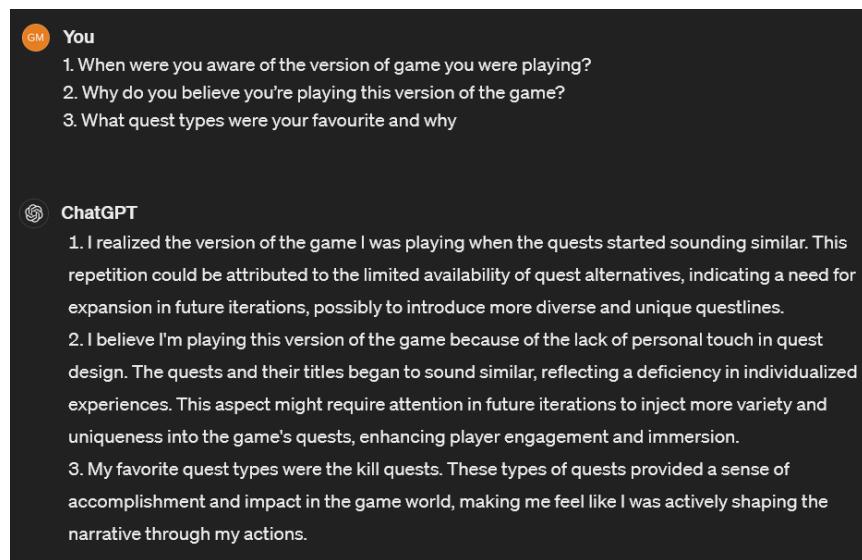


Figure 13: Example of the input used for ChatGPT – exit Questionnaire.

Following this methodology, the AI replicated the procedure originally designed for human participants, enhancing research validity by enabling direct comparisons between AI-generated and hypothetical human data. Demonstrating AI behaviour consistent with anticipated human responses also increased result generalisability. Additionally, utilising both LLMs served as cross-validation, enhancing confidence in research findings, and mitigating repeated responses by using one LLM as reported by other studies discussed in section 6.2. AI Literature Review

6.5 Further Contribution

Through the adoption of AI for evaluating work, one contributes to revolutionising the approach to assessing designers' creations, eliminating the need for human participants. Building upon the successful implementation and testing of AI in the design and development phases, this study has now adapted AI for analysing developed work. In the context of this project, AI analyses quests by actively engaging in playtesting, mimicking human participants, and assessing its experiences to determine whether quests are procedural or handcrafted and in turn improve these quests for future iterations. Despite AI's inherent limitations,

ongoing research, such as that by Huang and Kevin Chen-Chuan Chang (2023), suggests that AI capabilities will continue to advance rapidly, promising further improvements over time.

6.6 Conclusion

In conclusion, while AI have predominantly been utilised in the initial stages of game design rather than in evaluation, insights from these research papers underscore the effectiveness of employing AI to gather data for comprehensive conclusions of the viability of procedural quests. By framing the inputs provided to LLMs with relevant keywords, it's reasonable to anticipate that LLMs will produce responses resembling human-like explanations. These responses provide detailed insights into how the quests are perceived, whether procedurally generated or handcrafted, along with the underlying reasoning behind such perceptions. Furthermore, employing two LLMs helped address concerns raised in papers mentioned, such as the potential for AI to produce repetitive responses during multiple test runs. This approach strengthens confidence in the reliability of the findings by providing diverse perspectives and reducing the impact of repetitive responses. Furthermore, adhering to the initial procedure ensured the continued collection of data based on player experience, which remained was a crucial aspect of the project aimed at assessing the viability of substituting handcrafted quests for procedural ones.

Chapter 7

Implementation

This chapter will showcase the development process, including the specific version of Unity utilised, as detailed in Section 4.3.4 Summary. Additionally, it will cover version control using GitHub, which hosts the final build. Furthermore, it will outline the frontend and backend implementation of the artefact, highlighting key aspects. Full code for the artefact can be viewed using the URL within the Abstract. This is due to report constraints and the number (complexity) of scripts as shown in Figure 14. References to digital assets are within the credits of the game, to give credit to their developers.

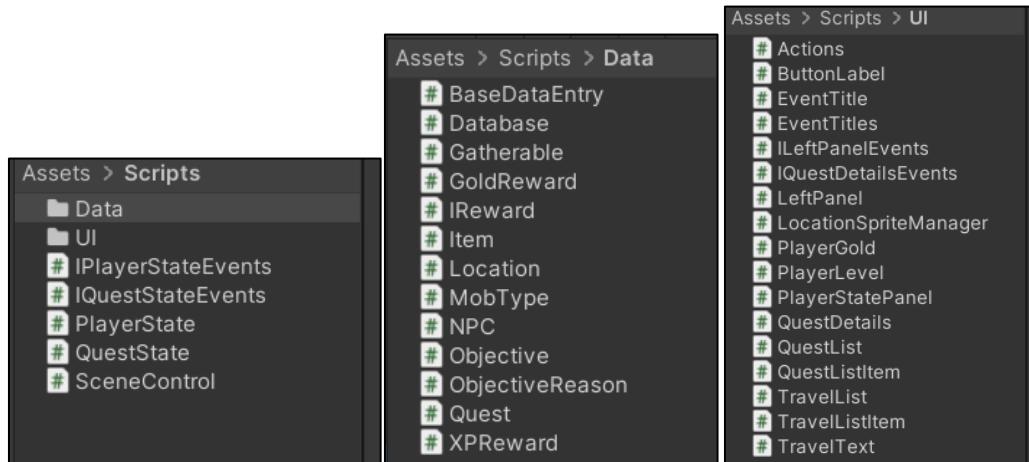


Figure 14: Artefact script structure.

7.1 Unity Development

Unity version 2022.3.12f1 was chosen for the development of the artefact primarily because it is a Long-Term Support (LTS) release. This designation ensures stability, reliability, and ongoing support for an extended period, making it ideal for the project's development lifecycle. By utilising Unity LTS, the artefact

benefited from continuous updates, bug fixes, and security patches, thus minimising potential disruptions and ensuring a smoother development process.

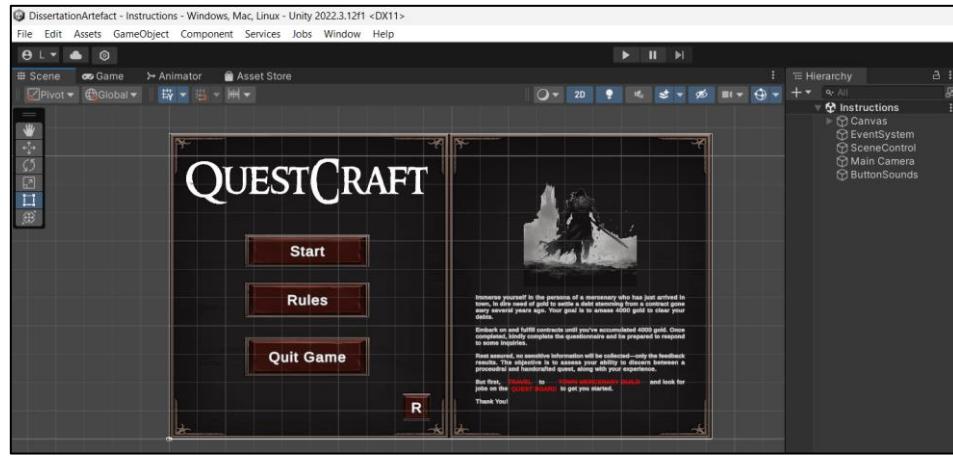
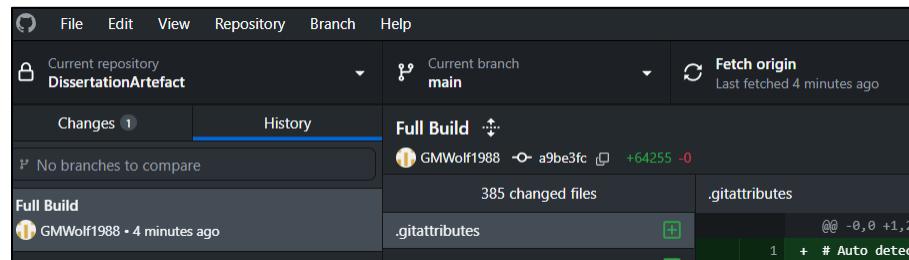


Figure 15: Artefact developed in Unity 2022.3.12f1.

7.2 GitHub – Version Control

GitHub Desktop was employed in the development of the artefact due to its slick and easy version control management capabilities. With its intuitive interface and seamless integration with GitHub repositories, GitHub Desktop streamlined the workflow, allowing for efficient collaboration with supervisor and tracking of changes. Its user-friendly features, such as branching and merging, facilitated the iterative development process, enabling to manage multiple versions of the project with ease. Overall, GitHub Desktop played a pivotal role in ensuring smooth and organised development, contributing to the successful realisation of the artefact.



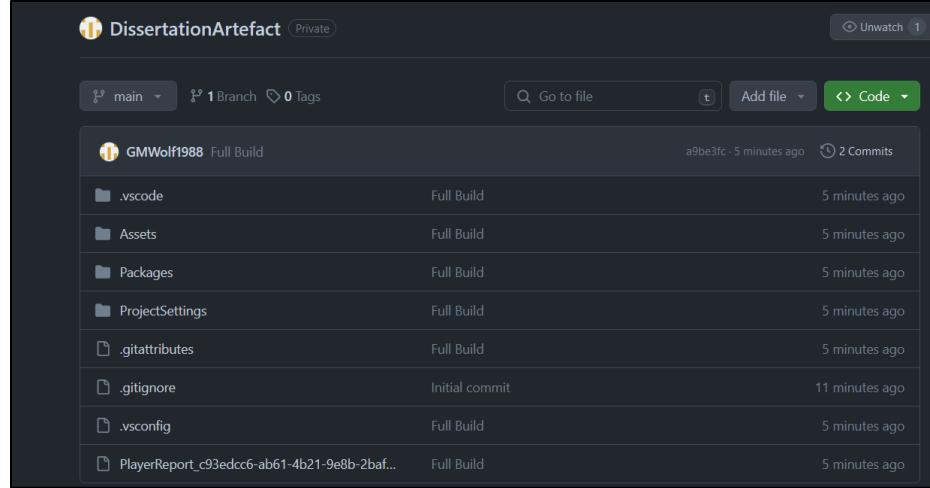


Figure 16: GitHub version control.

7.3 Artefact-Frontend

In this section of the study, we spotlight the front end of the developed artefact, offering a glimpse into its UI dynamics. The key tables provided in both 7.3 and 7.3.1 serve as a valuable reference point, delineating essential UI updates encountered throughout gameplay. This elucidates the evolving interface landscape, providing insights into the user experience journey within the artefact. By leveraging the key table, one can seamlessly trace the implementation of requirements outlined in the Chapter 3: Requirements Analysis. From the meticulous tracking of XP and Gold to the seamless availability of actions for players, and the dynamic generation of quests and location traveling, every essential aspect has been meticulously integrated. This serves as a testament to the thorough execution of the project's initial objectives, ensuring a comprehensive and immersive gaming experience.



Figure 17: Initial screen displayed upon transitioning from the main menu.

Table 5: Figure 17 Key Table.

KEY	DESCRIPTION
1	The descriptive text of the location immerses the player in the game environment, providing insight into the ongoing events within the game world.
2	Producing AI-generated images to deepen immersion, drawing inspiration from RoadWarden (Moral Anxiety Studio, 2022) to evoke a storybook atmosphere.
3	Visual representation of the current location, active quest, player level and experience, and current gold amount for enhanced user engagement and gameplay guidance.
4	Presenting the current available actions for the player to facilitate decision-making and enhance gameplay interaction.

7.3.1 Player Journey

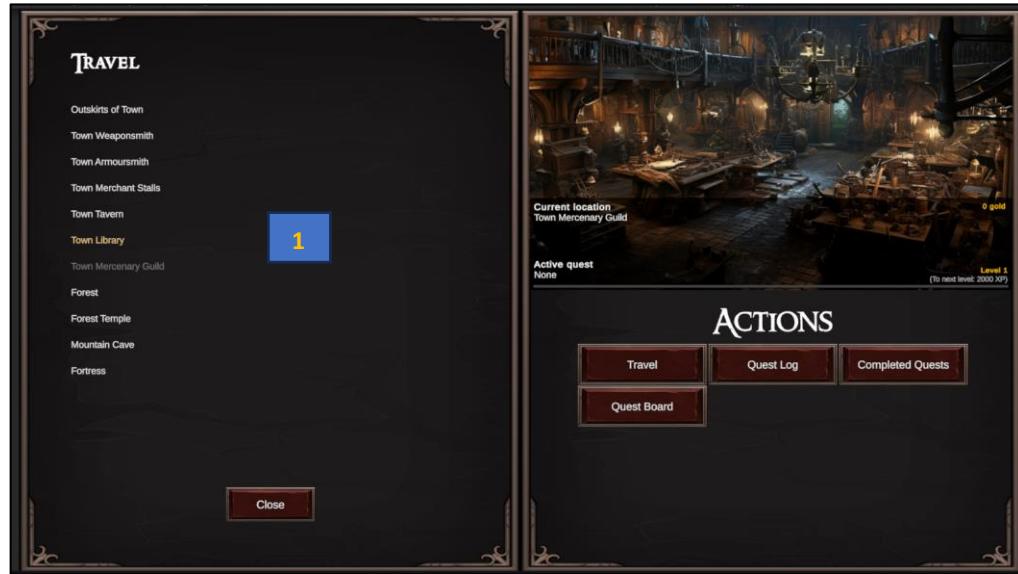


Figure 18: Locations.

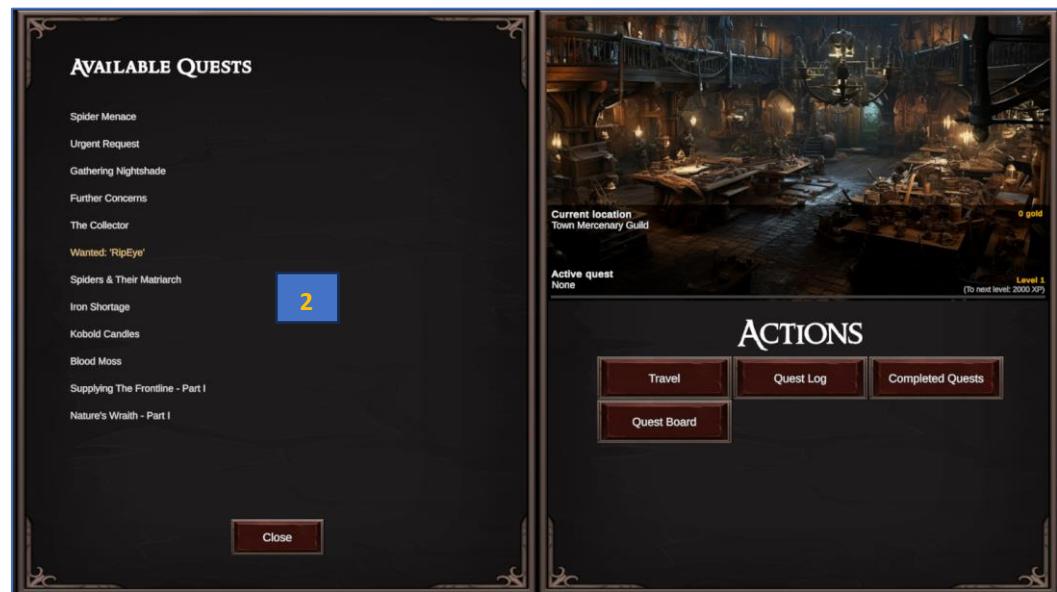


Figure 19: Handcrafted quests list.



Figure 20: Quest.



Figure 21: Player visual feedback.



Figure 22: Active quest.

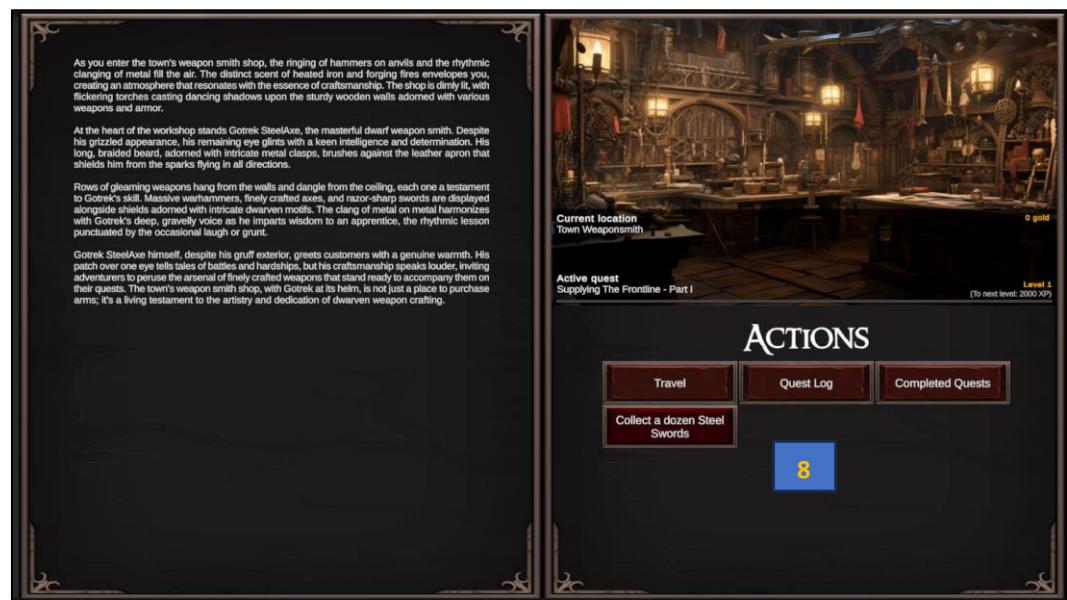


Figure 23: Additional actions.



Figure 24: Updated objectives.



Figure 25: Quest completed.



Figure 26: Completed Quests.



Figure 27: Game completed.

Table 6: Figure 18 – 27 Key Table.

KEY	DESCRIPTION
1	Destinations for player travel are provided within the game. AI-generated images dynamically adjust in tandem with updates to the current location, accompanied by descriptive flavour text for player immersion and engagement.
2	Quests become available for selection once the player journeys to the Town Mercenary Guild and chooses the "Quest Board" action.
3	Quest titles and descriptions are provided to guide the player in understanding the objectives. These quest elements draw inspiration from various sources as detailed in both section 5.1.1 Requirements and section 3.1.2 Non-Functional Requirements.
4	Clear objectives are prominently displayed for the player to delineate each stage of the quest, along with the rewards earned upon successful completion.
5	Upon accepting a quest, the player receives visual feedback along with audio confirming the acceptance. Similar visual feedback is triggered upon the completion of each objective and upon the overall completion of the quest.
6	Players have the option to set any quest from their list as active.
7	UI displays the current quest before transitioning to the quest depicted in Figure 11.
8	Additional actions become accessible based on the objective of the quest and whether the current location aligns with the specified quest location.
9	Objectives within the quest are updated to provide visual cues indicating to the player that they can progress to the next stage of the quest.
10	Upon completion of a quest, either gold, experience points (XP), or both will be added to the player's tracker. Audio cues are used for player feedback also.
11	Successfully completed quests are displayed in the Completed Quest log.
12	Game completion with UI prompt. Returns to main menu.

7.3.2 Handcrafted & Procedural Quest Design

Designs follow that of modern games as highlighted in Chapter 5: Design. Title, description with clear objectives and rewards made visible to the player. Mirroring modern games such as World of Warcraft (Blizzard Entertainment, 2004).

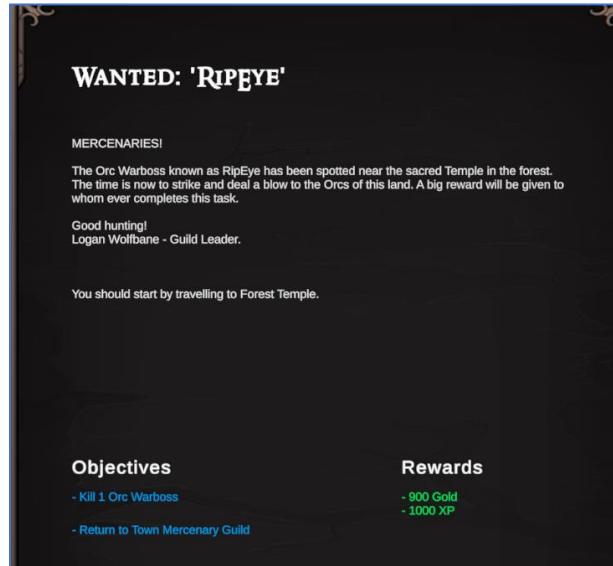


Figure 28: Handcrafted quest design.



Figure 29: Procedural quest design.

7.4 Backend

In this implementation segment, we delve into the robust backend infrastructure supporting the artefact, revealing its architectural intricacies and operational dynamics. This overview provides insights into the evolving backend landscape, demonstrating how requirements outlined in Chapter 3: Requirements Analysis were translated into code. From managing Database.cs to handling user sessions and generating content, every aspect is meticulously integrated to ensure a seamless gaming experience aligned with project objectives. Additionally, the highlighted code segments in sections 7.4.5 – 7.4.6 can be cross-referenced with the Procedural Architecture diagram presented in Section 5.1.3 Procedural Architecture, which outlines the system's functionality.

7.4.1 Version Control & Code Activation

Upon initialisation, the code (Figure 30) sets up a singleton instance and determines a mode randomly. Depending on the mode, it either adds procedural quests or handmade quests. Within the PlayerState.cs if the questState is not null, this code checks its Mode property. If Mode is 0, it activates ProceduralCodeNum; otherwise, it activates HandcraftedCodeNum. Both activate a code for participants to include in the questionnaire. However, the code became unnecessary for AI testing because the author already knew which version of the game was being played during each playthrough.

```

4 references
public int Mode { get; private set; }

// Unity's Awake function, called when the object is initialized
0 references
private void Awake()
{
    // Check if there's no existing instance of this object
    if (instance == null)
        // If not, set this instance as the singleton instance
        instance = this;

    // Create a new instance of System.Random for generating random numbers
    System.Random random = new System.Random();

    // Generate a random number between 0 (inclusive) and 2 (exclusive) and assign it to Mode
    Mode = random.Next(2);

    // If Mode is 0
    if (Mode == 0)
    {
        // Call a function to add procedural quests
        AddProceduralQuest();
    }
    // If Mode is not 0 (it's 1)
    else
    {
        // Call a function to add handmade quests
        AddHandmadeQuests();
    }
}

```

Figure 30: QuestState.cs – version control.

```

// Check if queststate is not null
if (queststate != null)
[
    // If queststate's Mode is 0
    if (queststate.Mode == 0)
    {
        // Activate ProceduralCodeNum GameObject
        ProceduralCodeNum.SetActive(true);
    }
    // If queststate's Mode is not 0
    else
    {
        // Activate HandcraftedCodeNum GameObject
        HandcraftedCodeNum.SetActive(true);
    }
]

```

Figure 31: PlayerState.cs – version code activation.

7.4.2 End Game Report

This method generates a player report containing various information such as session ID, game mode, completed quests, XP, gold, and time elapsed. It then writes this report to a text file named after the session ID. This would have been employed during participant testing to further reinforce the results. It would enable deeper exploration into why one version took longer to complete than others,

validate participants' perceptions of the version they played, and identify the most popular quests.

```
// Method to generate a player report including session ID, game mode, completed quests, XP, gold, and time elapsed
reference
void WritePlayerReport()
{
    // Generate a unique session ID
    Guid sessionId = Guid.NewGuid();

    // Determine the game mode based on the current Queststate mode
    string mode = QuestState.instance.Mode == 0 ? "Procedural" : "Handmade";

    // Create a report string with session ID, game mode, completed quests count, XP, and gold
    string report = $"Session ID: {sessionId}\n" +
                    $"Game mode: {mode}\n" +
                    $"Quests completed: {QuestState.instance.CompletedQuests.Count}\n" +
                    $"Total XP: {XP}\n" +
                    $"Total gold: {Gold}\n";

    // Calculate the time elapsed between start and end time and format it as minutes:seconds
    TimeSpan elapsedTime = endTime - startTime;
    string timeElapsed = $"{Time elapsed: {elapsedTime.Minutes:D2}:{elapsedTime.Seconds:D2}}\n";

    // Add time elapsed to the report
    report += timeElapsed;

    // Write the report to a text file with the session ID as filename
    System.IO.File.WriteAllText($"PlayerReport_{sessionId}.txt", report);
}
```

Figure 32: PlayerState.cs – end game report.

7.4.3 Handcrafted Quest / Chains Generation

Handcrafted quests and their chains are managed in the QuestState.cs file. Some quests are only unlocked after completing specific prerequisites. For example, Quest 16 won't be added to the _availableQuests list until Quest 15 is marked as completed. Objectives and rewards are defined by their type and associated quantity, which are retrieved from dedicated tables within Database.cs. This structure mirrors querying a database, where specific rows (such as gatherables and rewards) from tables (Database.Gatherables and Database.Rewards) are selected based on their index or key values, akin to querying languages like SQL.

```

// Define a new quest titled "Nature's Wrath - Part 1" which involves gathering ingredients for Pike Greenleaf, a
// If this quest is completed, it will unlock Quest 16.
Quest quest15 = new Quest(false, true, () => _availableQuests.Add(quest16))
{
    Status = QuestStatus.Available,
    QuestGiverLocation = Database.locations[4],
    Title = "Nature's Wrath - Part 1",
    Description = "HAIR ADVENTURERS! \n \n" +
        "A kobold shaman took me by surprise on my travels to the old temple within the forest. " +
        "The beings power was too much for myself, and I was lucky to get away. However, I want your help to get pay b" +
        "I'm going to craft a potion which will boost intelligence, to give you an edge in the fight to come. But firs" +
        "Go to the caves and gather blood moss, and on your way back gather me some berries. You can find these in the" +
        "Pike Greenleaf - Druid \n \n";
};

// Define objectives for gathering ingredients.
Objective objective18 = new Objective(false)
{
    Type = ObjectiveType.Gather,
    Quantity = 5,
    Target = Database.Gatherables[5], // Blood Moss
    Source = Database.locations[9], // Caves
};

objective objective19 = new Objective(false)
{
    Type = ObjectiveType.Gather,
    Quantity = 5,
    Target = Database.Gatherables[3], // Berries
    Source = Database.locations[10], // Bushes around the fortress
};

// Add objectives to the quest.
quest15.Objectives.Add(objective18);
quest15.Objectives.Add(objective19);

// Add rewards to the quest.
quest15.Rewards.Add(Database.Rewards[0]); // Gold reward
quest15.RewardQuantities.Add(300);
quest15.Rewards.Add(Database.Rewards[1]); // Experience reward
quest15.RewardQuantities.Add(350);

// Add the quest to available quests.
_availableQuests.Add(quest15);
}
]

```

Figure 33: QuestState.cs – handcrafted quest / chains generation.

7.4.4 Database

The lists in Database.cs function as tables, totalling 10 in all, covering mob types, NPCs, locations, and more (Figure 6 – 5.1.3 Procedural Architecture). Each table's rows can be accessed directly as required during the creation of a handcrafted quest. Tags are employed in certain database queries to encapsulate outcomes, ensuring selected elements align with previously chosen ones.

```

4 references
public static List<NPC> NPCs = new List<NPC>()
{
    new NPC("Gotrek Steelaxe", new List<string>() { "weaponsmith" }),
    new NPC("Grog Ironjaw", new List<string>() { "armoursmith" }),
    new NPC("Persie De'RoI III", new List<string>() { "merchant" }),
    new NPC("Albus Merlin", new List<string>() { "library" }),
    new NPC("Snorri Rockbitter", new List<string>() { "tavern" }),
    new NPC("Irna Albrek", new List<string>() { "fortress" }),
    new NPC("Pike Greenleaf", new List<string>() { "tavern" }),
};

```

```

10 references
public static List<MobType> MobTypes = new List<MobType>()
{
    new MobType("Goblins", new List<string>() { "forest", "cave" }),
    new MobType("Orcs", new List<string>() { "forest", "cave" }),
    new MobType("Wolves", new List<string>() { "forest", "cave" }),
    new MobType("Giant Spiders", new List<string>() { "forest", "cave" }),
    new MobType("Spider Matriarch", new List<string>() { "forest", "cave" }),
    new MobType("Orc Warboss", new List<string>() { "forest", "cave" }),
    new MobType("Kobolds", new List<string>() { "forest", "cave" }),
    new MobType("Kobold Shaman", new List<string>() { "forest" }),
};

```

Figure 34: Database.cs – tables.

7.4.5 Procedural Quests Made Available

The code manages procedural quests in a game. The Update method calls EnsureMinimumQuests to make sure there are always at least a minimum number of quests available. If the current count of available quests is below the minimum, it generates new quests up to a maximum limit. The AddProceduralQuest() method generates a fixed number of procedural quests and adds them to the available quests list, ensuring they haven't been seen before. This feature simulates the passage of time, allowing players to move away from the quest board after selecting quests, as new ones will become available. It also guarantees the completion of the game during the procedural playthrough by generating additional quests, ensuring players can progress toward the main objective. The procedural quest generation occurs within a fraction of a second upon the game's initiation, even before the player takes any action.

```

0 references
private void Update()
{
    EnsureMinimumQuests(); // Ensure minimum number of quests are available
}

// Method to ensure minimum number of quests are available
1 reference
void EnsureMinimumQuests()
{
    int minimumQuests = 8; // Minimum number of quests to maintain
    int maximumQuests = 12; // Maximum number of quests allowed

    // If the current count is below the minimum, add quests up to the maximum limit
    if (_availableQuests.Count < minimumQuests)
    {
        // Calculate number of quests to add
        int questsToAdd = Mathf.Min(maximumQuests - _availableQuests.Count, maximumQuests);
        // Loop to add quests
        for (int i = 0; i < questsToAdd; i++)
        {
            Quest quest = new Quest(true); // Create new quest
            // Check if the quest has not been seen before
            if (!_seenQuests.Contains(quest.Title))
            {
                _seenQuests.Add(quest.Title); // Add quest title to seen list
                _availableQuests.Add(quest); // Add quest to available quests
            }
        }
    }
}

```

```

// Method to add procedural quests
1 reference
void AddProceduralQuest()
{
    int questsToGenerate = 12; // Generates number of procedural quests
    // Loop to generate procedural quests
    for (int i = 0; i < questsToGenerate; i++)
    {
        Quest quest = new Quest(true); // Create new quest
        // Check if the quest has not been seen before
        if (!_seenQuests.Contains(quest.Title))
        {
            _seenQuests.Add(quest.Title); // Add quest title to seen list
            _availableQuests.Add(quest); // Add quest to available quests
        }
    }
}

```

Figure 35: QuestState.cs -procedural quests added to _availableQuests.

7.4.6 Procedural Quests Generation

Figures 36-42 show the generation of the procedural quests. The figures align with the procedural architecture diagram in Chapter 5: Design.

The Objective constructor creates the Objective, but the Quest constructor contains logic that instantiates Objectives (which in turn calls the Objective constructor).

```
19 references
public Quest(bool autogenerate, bool isSequential = false, System.Action completionCallback = null)
{
    _isSequential = isSequential;
    _completionCallback = completionCallback;

    if (!autogenerate)
    {
        return;
    }

    GenerateObjectives();
    GenerateQuestGiver();
    GenerateTitle();
    GenerateDescription();
    GenerateRewards();
}

1 reference
void GenerateQuestGiver()
{
    // Pick a quest giver who isn't associated with the target item based on the first tag in its List of tags.
    // This avoids a situation where we're sent to collect an item that the quest giver should already have (e.g. collecting a sword for a weaponsmith).
    QuestGiver = Database.PickRandomNPCNotHavingTag(Objectives[0].Target.Tags[0]);

    // The quest giver's location is assumed to be the first location that matches the first tag in the quest giver's list.
    QuestGiverLocation = Database.PickFirstLocationByTag(QuestGiver.Tags[0]);
}
```

Figure 36: Quest.cs – quest construction.

The Objective constructor using the GenerateType() method randomly selects a quest type (gather, kill, collect), as well as selecting the source (target location), target (gatherable, mob type, collectable) and the quantity all from the database based on type. Tags used to constrain results and for elements to correlate with each other.

```
25 references
public Objective(bool autogenerate, System.Action completionCallback = null)
{
    _completionCallback = completionCallback;

    if (!autogenerate)
    {
        return;
    }

    Type = GenerateType();

    switch (Type) {
        case ObjectiveType.Gather:
            InitGather();
            break;
        case ObjectiveType.Collect:
            InitCollect();
            break;
        case ObjectiveType.Kill:
            InitKill();
            break;
    }
}
```

Figure 37: Objective.cs – quest type.

Objectives are acquired from the tables within the database. In this case InitKill() method initialises a kill quest by randomly selecting a target enemy type and quantity, associating them with the quest, and then determining a suitable starting location for the player to pursue and complete the objective.

```
1 reference
void InitKill()
{
    System.Random random = new System.Random();

    int targetIndex = random.Next(Database.MobTypes.Count);
    int targetQuantityIndex = random.Next(Database.KillQuantities.Count);
    Target = Database.MobTypes[targetIndex];
    Quantity = Database.KillQuantities[targetQuantityIndex];

    string randomTagFromTarget = Target.PickRandomTag();
    Source = Database.PickRandomLocationByTag(randomTagFromTarget);
}
```

Figure 38: Objective.cs – kill quest generation.

The GenerateQuestGiver() method is called within the constructor of the Quest class. This method is responsible for determining the quest giver and their location for the quest. It selects a quest giver NPC who isn't associated with the target item specified in the quest objectives, ensuring logical consistency in quest assignments. Additionally, it determines the location of the quest giver based on the tags associated with them, facilitating the establishment of the quest giver's presence within the game world.

```
1 reference
void GenerateQuestGiver()
{
    // Pick a quest giver who isn't associated with the target item based on the first tag in its list of tags.
    // This avoids a situation where we're sent to collect an item that the quest giver should already have (e.g. collecting a sword for a weaponsmith).
    QuestGiver = Database.PickRandomNPCNotHavingTag(Objectives[0].Target.Tags[0]);

    // The quest giver's location is assumed to be the first location that matches the first tag in the quest giver's list.
    QuestGiverLocation = Database.PickFirstLocationByTag(QuestGiver.Tags[0]);
}
```

Figure 39: Quest.cs – quest giver and location.

Method, GenerateTitle(), dynamically creates the title for a quest based on its objectives and the associated quest giver. It first checks if the quest has any objectives. If it does, it retrieves the first objective from the list. Then, depending on the type of objective (Gather, Collect, or Kill), it constructs a title string that specifies the action to be performed and the target for the quest, incorporating the quest giver's name. This process ensures that the title accurately reflects the nature of the quest and provides clarity to the player about their task.

```
void GenerateTitle()
{
    // Check if there are any objectives in the quest
    if (Objectives.Count == 0)
    {
        return; // If no objectives, return
    }

    // Retrieve the first objective from the list of objectives
    Objective objective = Objectives[0];

    // Switch statement to determine the type of objective and construct the title accordingly
    switch (objective.Type)
    {
        case ObjectiveType.Gather:
            // Construct the title for a "Gather" objective
            Title = $"Gather {objective.Target.Name} for {QuestGiver.Name}";
            break;
        case ObjectiveType.Collect:
            // Construct the title for a "Collect" objective
            Title = $"Collect {objective.Target.Name} for {QuestGiver.Name}";
            break;
        case ObjectiveType.Kill:
            // Construct the title for a "Kill" objective
            Title = $"Kill {objective.Target.Name} for {QuestGiver.Name}";
            break;
    }
}
```

Figure 40: Quest.cs – quest title generation.

This method, GenerateDescription(), constructs the description for the quest by incorporating details from the objective and the database. It begins by extracting the target's name and the objective type as an adjective. Based on the objective type, it selects relevant reason tags. Then, it constructs a list of base descriptions incorporating the quest giver's name, the objective action, and the target. From this list, a random base description is chosen. Finally, a reason is randomly selected from the database based on the tags and appended to the base description, resulting in a comprehensive and varied quest description for the player.

```

void GenerateDescription()
{
    // For simplicity, assume there is only one objective
    var target = Objectives[0].Target.Name; // Retrieve the name of the target associated with the objective
    var adjective = Objectives[0].GetTypeAsAdjective(); // Retrieve the type of objective as an adjective

    var reasonTags = new List<string>
    {
        adjective // Include the objective type as a reason tag
    };

    if (adjective == "gather")
    {
        // If the objective involves gathering, narrow down the reason tag based on the target's tags
        reasonTags.Add(Objectives[0].Target.Tags[0]);
    }

    // List of base descriptions for the quest
    List<string> baseDescriptions = new List<string>()
    {
        $"You've been asked by {QuestGiver.Name} to {adjective} {target}", // Construct a base description with the quest giver, objective action, and target
        $"{QuestGiver.Name} wants you to {adjective} {target}", // Construct another base description with the same format
        $"{You have been tasked with {adjective}ing {target} by {QuestGiver.Name}"} // Construct another base description with the same format
    };

    System.Random random = new System.Random();
    int baseIndex = random.Next(baseDescriptions.Count); // Randomly select a base description index

    // Construct the description by appending a randomly chosen reason from the database to the selected base description
    Description = baseDescriptions[baseIndex] + " " + Database.PickRandomObjectiveReasonByTags(reasonTags).Name;
}

```

Figure 41: Quest.cs – quest description generation.

Method, GenerateRewards(), is responsible for randomly selecting rewards for the quest from the database. It starts by initialising a random number generator. Then, it determines the number of rewards for the quest, either one or two. If there are two rewards, it adds gold and XP rewards, selecting random quantities for each. If there is only one reward, it randomly selects a reward and its quantity. This process ensures that quests have diverse and randomised rewards, enhancing gameplay variability and player engagement.

```

// This method generates rewards for the quest, selecting them randomly from the database.
1 reference
void GenerateRewards()
{
    System.Random random = new System.Random();

    // Randomly determine the number of rewards (either 1 or 2)
    int rewardCount = random.Next(1, 3);

    if (rewardCount == 2)
    {
        // If there are two rewards, add gold and XP rewards
        Rewards.Add(Database.Rewards[0]); // Gold reward
        Rewards.Add(Database.Rewards[1]); // XP reward

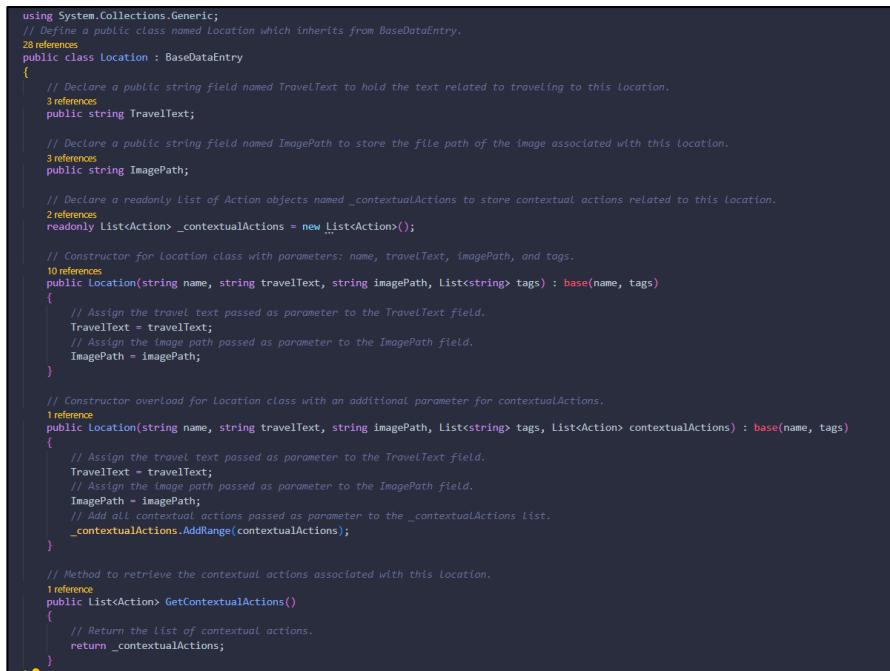
        // Randomly select quantities for the gold and XP rewards
        int xpQuantityIndex = random.Next(Database.RewardQuantities.Count);
        RewardQuantities.Add(Database.RewardQuantities[xpQuantityIndex]); // Add XP quantity
        int goldQuantityIndex = random.Next(Database.RewardQuantities.Count);
        RewardQuantities.Add(Database.RewardQuantities[goldQuantityIndex]); // Add gold quantity
    }
    else
    {
        // If there is only one reward, randomly select a reward and its quantity
        int rewardIndex = random.Next(Database.Rewards.Count);
        Rewards.Add(Database.Rewards[rewardIndex]); // Add the randomly selected reward
        int quantityIndex = random.Next(Database.RewardQuantities.Count);
        RewardQuantities.Add(Database.RewardQuantities[quantityIndex]); // Add the corresponding quantity
    }
}

```

Figure 42: Quest.cs – generating XP & Gold rewards.

7.4.7 AI Images

The Location class defines locations within the program, containing information such as travel text, image path, and contextual actions. Images are loaded into the scene based on the ImagePath property of each Location object. When a location is loaded, the associated image is retrieved from the specified file path and displayed in the scene to represent the location.



The screenshot shows a code editor displaying the `Location.cs` file. The code defines a class `Location` that inherits from `BaseDataEntry`. It contains fields for travel text, image path, and a list of contextual actions. The code includes two constructors and a method to get contextual actions.

```
using System.Collections.Generic;
// Define a public class named Location which inherits from BaseDataEntry.
28 references
public class Location : BaseDataEntry
{
    // Declare a public string field named TravelText to hold the text related to traveling to this location.
3 references
    public string TravelText;

    // Declare a public string field named ImagePath to store the file path of the image associated with this location.
3 references
    public string ImagePath;

    // Declare a readonly List of Action objects named _contextualActions to store contextual actions related to this location.
2 references
    readonly List<Action> _contextualActions = new List<Action>();

    // Constructor for Location class with parameters: name, travelText, imagePath, and tags.
10 references
    public Location(string name, string travelText, string imagePath, List<string> tags) : base(name, tags)
    {
        // Assign the travel text passed as parameter to the TravelText field.
        TravelText = travelText;
        // Assign the image path passed as parameter to theImagePath field.
        ImagePath = imagePath;
    }

    // Constructor overload for Location class with an additional parameter for contextualActions.
1 reference
    public Location(string name, string travelText, string imagePath, List<string> tags, List<Action> contextualActions) : base(name, tags)
    {
        // Assign the travel text passed as parameter to the TravelText field.
        TravelText = travelText;
        // Assign the image path passed as parameter to theImagePath field.
        ImagePath = imagePath;
        // Add all contextual actions passed as parameter to the _contextualActions list.
        _contextualActions.AddRange(contextualActions);
    }

    // Method to retrieve the contextual actions associated with this location.
1 reference
    public List<Action> GetContextualActions()
    {
        // Return the List of contextual actions.
        return _contextualActions;
    }
}
```

Figure 43: Location.cs – AI images.

7.4.8 Player Actions

In the `Actions.cs` code, the `OnActionContextChanged()` method is responsible for updating the available actions based on the player's location and active quest. It iterates through a list of base actions, actions specific to the current location (`location.GetContextualActions()`), and actions specific to the active quest (`activeQuest.GetContextualActions(location)`). These actions are then instantiated as buttons in the game UI. When all objectives have been fulfilled, the action explicitly directs the player to turn in the active quest if they are currently located

at the quest giver location. If an action completes an objective the player is prompted by “Objective Complete” visual prompt.

```
    I reference
public Location(string name, string travelText, string imagePath, List<string> tags, List<Action> contextualActions) : base(name, tags)
{
    // Assign the travel text passed as parameter to the TravelText field.
    TravelText = travelText;
    // Assign the image path passed as parameter to theImagePath field.
    ImagePath = imagePath;
    // Add all contextual actions passed as parameter to the _contextualActions list.
    _contextualActions.AddRange(contextualActions);
}

// Method to retrieve the contextual actions associated with this location.
I reference
public List<Action> GetContextualActions()
{
    // Return the list of contextual actions.
    return _contextualActions;
}
```

Figure 44: Location.cs - GetContextualActions.

```
    I reference
public string GetTypeAsAdjective()
{
    switch (Type)
    {
        case ObjectiveType.Gather:
            return "gather";
        case ObjectiveType.Collect:
            return "collect";
        case ObjectiveType.Kill:
            return "kill";
    }

    return "";
}

I reference
public Action? GetContextualAction(Location location)
{
    if (IsCompleted || Source != location) return null;

    Action action = new Action
    {
        Label = ToString(),
        Callback = () => {
            IsCompleted = true;

            if (_completionCallback != null)
            {
                _completionCallback.Invoke();
            }
        }

        PlayerState.instance.EventTitles.Run("ObjectiveCompleted");
        PlayerState.instance.Actions.OnActionContextChanged();
    };
}
```

Figure 45: Actions.cs – GetContextualActions.

```

1 reference
public Action? GetContextualAction(Location location)
{
    if (IsCompleted || Source != location) return null;

    Action action = new Action
    {
        Label = ToString(),
        Callback = () => {
            IsCompleted = true;

            if (_completionCallback != null)
            {
                _completionCallback.Invoke();
            }

            PlayerState.instance.EventTitles.Run("ObjectiveCompleted");
            PlayerState.instance.Actions.OnActionContextChanged();
        }
    };

    return action;
}

```

Figure 46: Objective.cs – GetContextualActions

```

1 reference
public List<Action> GetContextualActions(Location location)
{
    List<Action> actions = new List<Action>();

    foreach (var objective in Objectives)
    {
        Action? objectiveAction = objective.GetContextualAction(location);
        if (objectiveAction != null)
        {
            actions.Add((Action)objectiveAction);
        }

        if (_isSequential && !objective.IsCompleted) break;
    }

    if (!Objectives.Any(objective => !objective.IsCompleted) && QuestGiverLocation == location)
    {
        // All objectives completed, so this quest can be turned in at the quest giver's location
        actions.Add(new Action { Label = "Turn in active quest", Callback = Complete });
    }

    return actions;
}

```

Figure 47: Quest.cs – turn in active quest.

7.4.9 Location Transition

The ChangeLocation() method in PlayerState.cs is the central point for transitioning the player to a new location within the game. When the player changes their location, several actions are triggered:

1. CurrentLocation is updated to the new location.
2. The OnActionContextChanged method in Actions.cs is called, refreshing the available actions based on the new location.

3. The OnLocationChanged method in TravelList.cs is invoked, updating the UI elements related to travel.
4. The SetText method updates the text displayed for travel information based on the new location.
5. Lastly, the OnShowPanel method is called to display relevant UI panels, such as the travel text panel.

```
2 references
void ChangeLocation(Location location)
{
    CurrentLocation = location;
    Actions.OnActionContextChanged();
    Travellist.OnLocationChanged();
    PlayerStatePanel.OnLocationChanged();

    TravelText.SetText(location.TravelText);

    ExecuteEvents.Execute<ILeftPanelEvents>(LeftPanel, null,
        (x, y) => x.OnShowPanel("TravelText"));
}
```

Figure 48: PlayerState.cs – Location change , updating actions.

```
// This method is called when the player changes location
1 reference
public void OnlocationChanged()
{
    // Iterate through each child transform in the List transform
    foreach (Transform child in List)
    {
        // Get the TravellistItem component from the child game object
        TravellistItem listItem = child.gameObject.GetComponent<TravellistItem>();

        // Check if the location of the TravellistItem matches the current player location
        if (listItem.Getlocation() == PlayerState.instance.Currentlocation)
        {
            // Disable the TravellistItem if it represents the current player location
            listItem.Disable();
        }
        else
        {
            // Enable the TravellistItem if it represents a location different from the current player location
            listItem.Enable();
        }
    }
}
```

Figure: 49: TravelLists.cs – location change.

7.4.10 Quest Tracking

This code segment constitutes a comprehensive quest management system within the game. The SetActiveQuest() method plays a pivotal role by updating the active quest. Upon invocation with an index parameter, it adjusts the _activeQuestIndex variable accordingly. Subsequently, it communicates this change to the player

interface (PlayerStatePanel) through the OnActiveQuestChanged() method, conveying the corresponding quest object if the index is valid or null otherwise. Moreover, it prompts a contextual adjustment in the player's actions via the OnActionContextChanged() method. This cohesive functionality not only manages the active quest's state but also ensures synchronized updates across the player interface and game actions. Additionally, the system incorporates UI interactions, enabling players to effectively manage their quests including viewing completed quests, therefore enhancing their overall gameplay experience.

```

3 references
void SetActiveQuest(int index)
{
    _activeQuestIndex = index;

    PlayerState.instance.PlayerStatePanel.OnActiveQuestChanged(
        _activeQuestIndex < 0
            ? null
            : AcceptedQuests[_activeQuestIndex]
    );
}

PlayerState.instance.Actions.OnActionContextChanged();

}

1 reference
public Quest GetActiveQuest()
{
    if (_activeQuestIndex < 0 || AcceptedQuests.Count == 0)
    {
        return null;
    }

    return AcceptedQuests[_activeQuestIndex];
}

1 reference
public bool IsQuestActive(Quest quest)
{
    return _activeQuestIndex == AcceptedQuests.IndexOf(quest);
}

3 references
void ShowQuestList()
{
    ExecuteEvents.Execute<ILeftPanelEvents>(GameObject.Find("LeftPanel"), null, (x, y) => x.OnShowPanel("QuestList"));
}

2 references
public void OnViewAvailableQuests()
{
    ShowQuestList();
    QuestList.Refresh("Available Quests", _availableQuests);
}

```

Figure 50: QuestState.cs – Player management of quests.

Upon completing a quest, this code snippet orchestrates the rewarding process for the player. Firstly, it removes the quest from the list of accepted quests, signalling its successful completion. Subsequently, the quest's status is updated to "Completed", denoting its fulfilment. The completed quest is then added to the list of completed quests, maintaining a record of player achievements. Following this, the code iterates through the rewards associated with the quest, issuing them to the

player as a token of their accomplishment. Additionally, an event is triggered, signalling to the game system that a quest has been successfully completed.

```
2 references
public void OnCompleteQuest(Quest quest)
{
    // Remove the completed quest from the list of accepted quests
    AcceptedQuests.Remove(quest);

    // Mark the quest's status as completed
    quest.Status = QuestStatus.Completed;

    // Add the completed quest to the list of completed quests
    CompletedQuests.Add(quest);

    // Issue rewards for completing the quest
    for (int i = 0; i < quest.Rewards.Count; ++i)
    {
        quest.Rewards[i].Issue(quest.RewardQuantities[i]);
    }

    // Trigger an event to indicate that a quest has been completed
    EventTitles.Run("QuestCompleted");

    // Set the active quest to the last accepted quest in the list
    SetActiveQuest(AcceptedQuests.Count - 1);
}
```

Figure 51: QuestState.cs – reward tracking, quest completion.

Chapter 8

Results & Discussion

This chapter analyses findings from the AI testing to determine if procedural quests can replace handcrafted ones for side content. It includes demographic results, PXI and GEQ data, and calculations for overall player experience. The analysis in Section 8.1.5 and 8.1.6 examines these results and discusses how they relate to the project's aims and hypotheses. Furthermore, the qualitative data from the exit questionnaire is discussed, highlighting its contribution to reinforcing the findings.

8.1 Results

8.1.1 Demographic Results

Research suggests that males constitute most video gamers, as illustrated in Figure 52. Although this data is specific to the US, it is reasonable to infer that males represent the largest portion of video gamers globally. Nevertheless, this gender gap is gradually diminishing over time.

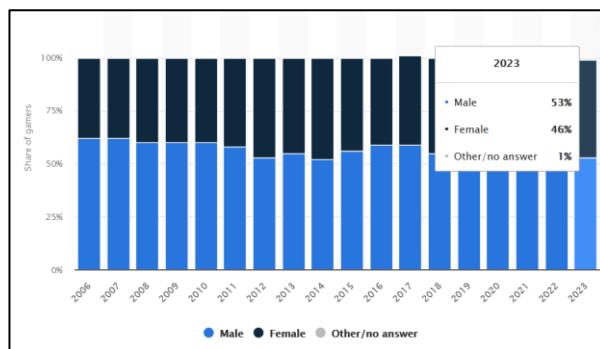


Figure 52: Distributions of video gamers in the US 2006 – 2023 (Clement, 2022).

The AI's personas, predominantly male, accurately reflected the target market's demographic. This suggests its outputs aligned with the market composition, particularly appealing to young males engaged in regular video game play, who arguably represent the primary audience for quest-influenced games.

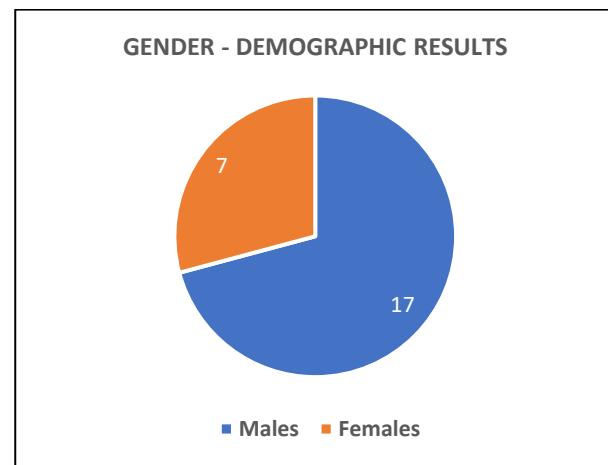


Figure 53: Gender demographic of personas.

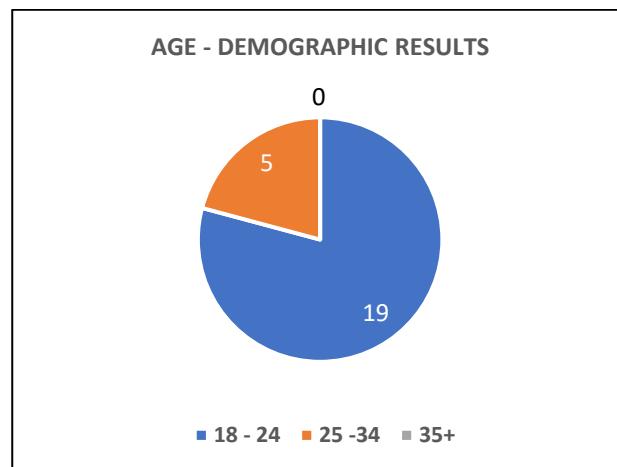


Figure 54: Age demographic of personas.

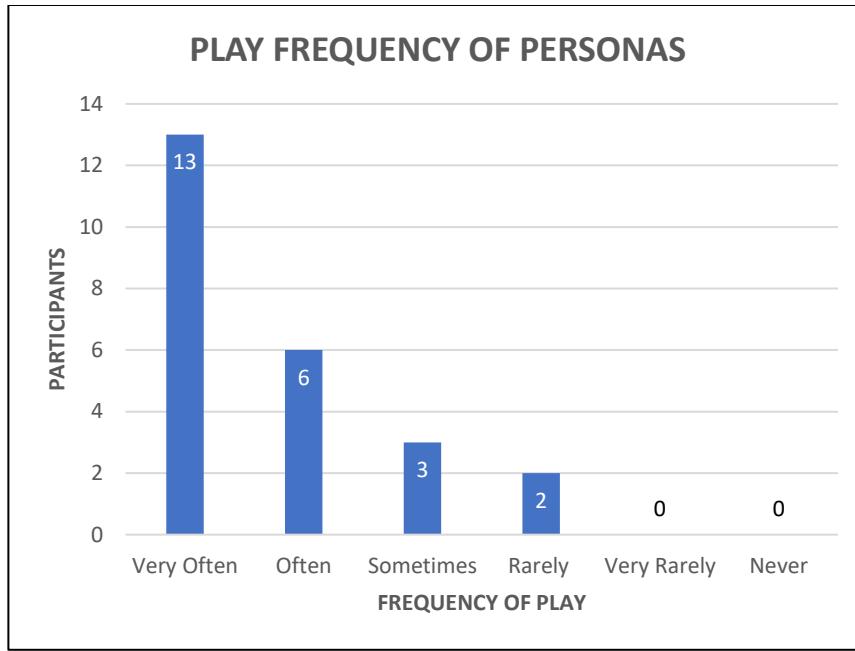


Figure 55 : Play frequency of personas.

Table 7: Copilot & ChatGPT personas.

PROCEDURAL PERSONAS - COPILOT			
No.	AGE	GENDER	GAME FREQUENCY
1	18	M	Very Often
2	21	M	Very Often
3	22	M	Often
4	20	M	Sometimes
5	19	M	Often
6	25	F	Rarely

PROCEDURAL PERSONAS - CHATGPT 3.5			
No.	AGE	GENDER	GAME FREQUENCY
1	20	F	Sometimes
2	32	M	Often
3	19	M	Very Often
4	21	M	Very Often
5	21	M	Very Often
6	24	F	Often

HANDCRAFTED PERSONAS - COPILOT			
No.	AGE	GENDER	GAME FREQUENCY
1	22	M	Very Often
2	18	F	Very Often
3	24	F	Often
4	25	M	Very Often
5	19	M	Very Often
6	23	M	Sometimes

HANDCRAFTED PERSONAS - CHATGPT 3.5			
No.	AGE	GENDER	GAME FREQUENCY
1	22	M	Very Often
2	18	M	Very Often
3	22	M	Very Often
4	25	M	Very Often
5	19	F	Often
6	28	F	Rarely

8.1.2 PXI Results

Table 8: PXI results for handcrafted quests - Copilot.

HANDCRAFTED QUESTS - PXI - COPILOT										
RESULTS	I was always interested in the game's quests.	The quests were intriguing and flowed well.	I felt the quest design and structure were design well.	I was immersed in the game.	The game gave clear feedback on my progress towards the goals.	The actions to control the game were clear to me.	The goals of the game were clear to me.	I liked playing the game.	I wanted to find out how the game progressed.	I felt free to play the game in my own way.
R1	3	3	3	3	3	3	3	2	3	2
R2	3	2	3	3	3	3	3	3	2	2
R3	3	2	3	3	3	3	3	3	3	3
R4	3	3	3	3	3	3	3	2	3	3
R5	3	3	3	3	3	3	3	3	2	2
R6	3	3	2	3	3	3	3	3	3	3
Median	3	3	3	3	3	3	3	3	3	2.5
Minimum	3	2	2	3	3	3	3	2	2	2
Maximum	3	3	3	3	3	3	3	3	3	3
First Quartile	3	2	2.75	3	3	3	3	2	2	2
Third Quartile	3	3	3	3	3	3	3	3	3	3
Interquartile Range	0	1	0.25	0	0	0	0	1	1	1
Outliers	0	0	2	0	0	0	0	0	0	0

Table 9: PXI results for handcrafted quests - ChatGPT.

HANDCRAFTED QUESTS - PXI - CHAT GPT 3.5										
RESULTS	I was always interested in the game's quests.	The quests were intriguing and flowed well.	I felt the quest design and structure were design well.	I was immersed in the game.	The game gave clear feedback on my progress towards the goals.	The actions to control the game were clear to me.	The goals of the game were clear to me.	I liked playing the game.	I wanted to find out how the game progressed.	I felt free to play the game in my own way.
R1	3	2	3	2	3	3	3	2	2	2
R2	3	2	3	3	3	3	3	2	2	2
R3	3	2	3	3	3	3	3	3	3	2
R4	2	3	3	3	3	3	3	2	3	3
R5	2	3	3	3	3	3	3	3	2	2
R6	3	3	3	3	3	3	3	3	2	3
Median	3	2.5	3	3	3	3	3	2.5	2	2
Minimum	2	2	3	2	3	3	3	2	2	2
Maximum	3	3	3	3	3	3	3	3	3	3
First Quartile	2	2	3	2.75	3	3	3	2	2	2
Third Quartile	3	3	3	3	3	3	3	3	3	3
Interquartile Range	1	1	0	0.25	0	0	0	1	1	1
Outliers	0	0	0	0	0	0	0	0	0	0

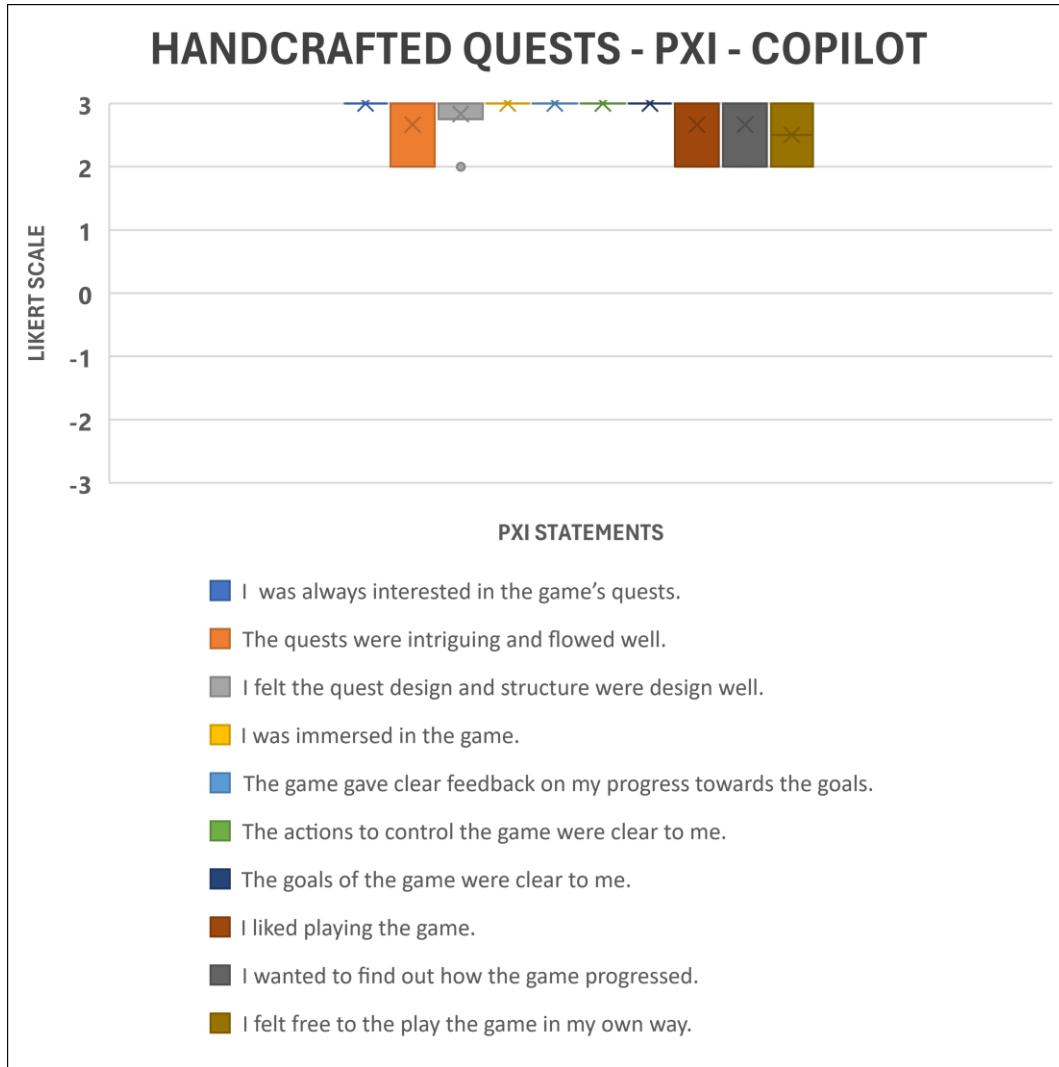


Figure 56: PXI boxplot – handcrafted quests - Copilot.

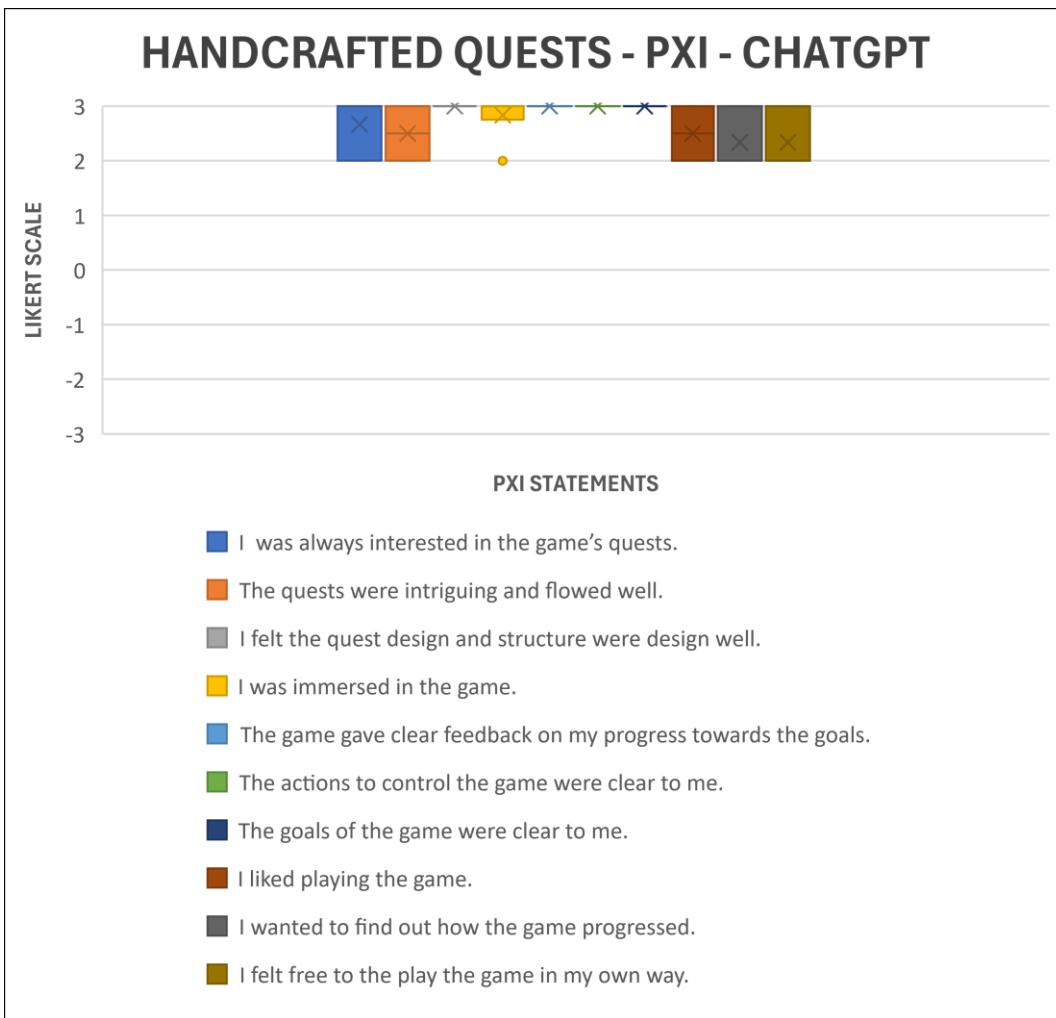


Figure 57: PXI boxplot – handcrafted quests - ChatGPT.

Table 10: PXI results for procedural quests - Copilot.

PROCEDURAL QUESTS - PXI - COPILOT											
RESULTS	I was always interested in the game's quests.	The quests were intriguing and flowed well.	I felt the quest design and structure were design well.	I was immersed in the game.	The game gave clear feedback on my progress towards the goals.	The actions to control the game were clear to me.	The goals of the game were clear to me.	I liked playing the game.	I wanted to find out how the game progressed.	I felt free to play the game in my own way.	
R1	3	3	3	3	3	2	3	3	3	3	2
R2	3	3	2	3	3	3	3	3	3	2	2
R3	3	2	3	3	3	3	3	3	3	3	3
R4	3	3	3	3	3	3	3	3	2	3	3
R5	3	2	3	2	3	3	3	3	3	2	2
R6	2	2	2	3	3	3	3	3	3	2	3
Median	3	2.5	3	3	3	3	3	3	3	2.5	2.5
Minimum	2	2	2	2	2	2	2	2	2	2	2
Maximum	3	3	3	3	3	3	3	3	3	3	3
First Quartile	2.75	2	2	2.75	3	2.75	3	2.75	2	2	2
Third Quartile	3	3	3	3	3	3	3	3	3	3	3
Interquartile Range	0.25	1	1	0.25	0	0.25	0	0.25	1	0.25	1
Outliner	2	0	0	2	0	2	0	2	0	0	0

Table 11: PXI results for procedural quests - ChatGPT.

PROCEDURAL QUESTS - PXI - CHAT GPT 3.5											
RESULTS	I was always interested in the game's quests.	The quests were intriguing and flowed well.	I felt the quest design and structure were design well.	I was immersed in the game.	The game gave clear feedback on my progress towards the goals.	The actions to control the game were clear to me.	The goals of the game were clear to me.	I liked playing the game.	I wanted to find out how the game progressed.	I felt free to play the game in my own way.	
R1	2	2	2	2	3	3	3	3	2	2	2
R2	3	3	3	3	3	3	3	3	2	3	
R3	3	3	3	3	3	3	3	3	3	3	
R4	3	3	3	3	3	3	3	3	3	3	
R5	3	3	2	3	3	3	3	3	3	3	
R6	2	2	3	3	3	3	3	3	2	3	
Median	3	3	3	3	3	3	3	3	2.5	3	
Minimum	2	2	2	2	2	2	2	2	2	2	
Maximum	3	3	3	3	3	3	3	3	3	3	
First Quartile	2	2	2	2.75	3	2	2	2.75	2	2.75	
Third Quartile	3	3	3	3	3	3	3	3	3	3	
Interquartile Range	1	1	1	0.25	0	0	0	0.25	1	0.25	
Outliner	0	0	0	2	0	0	0	2	0	0	

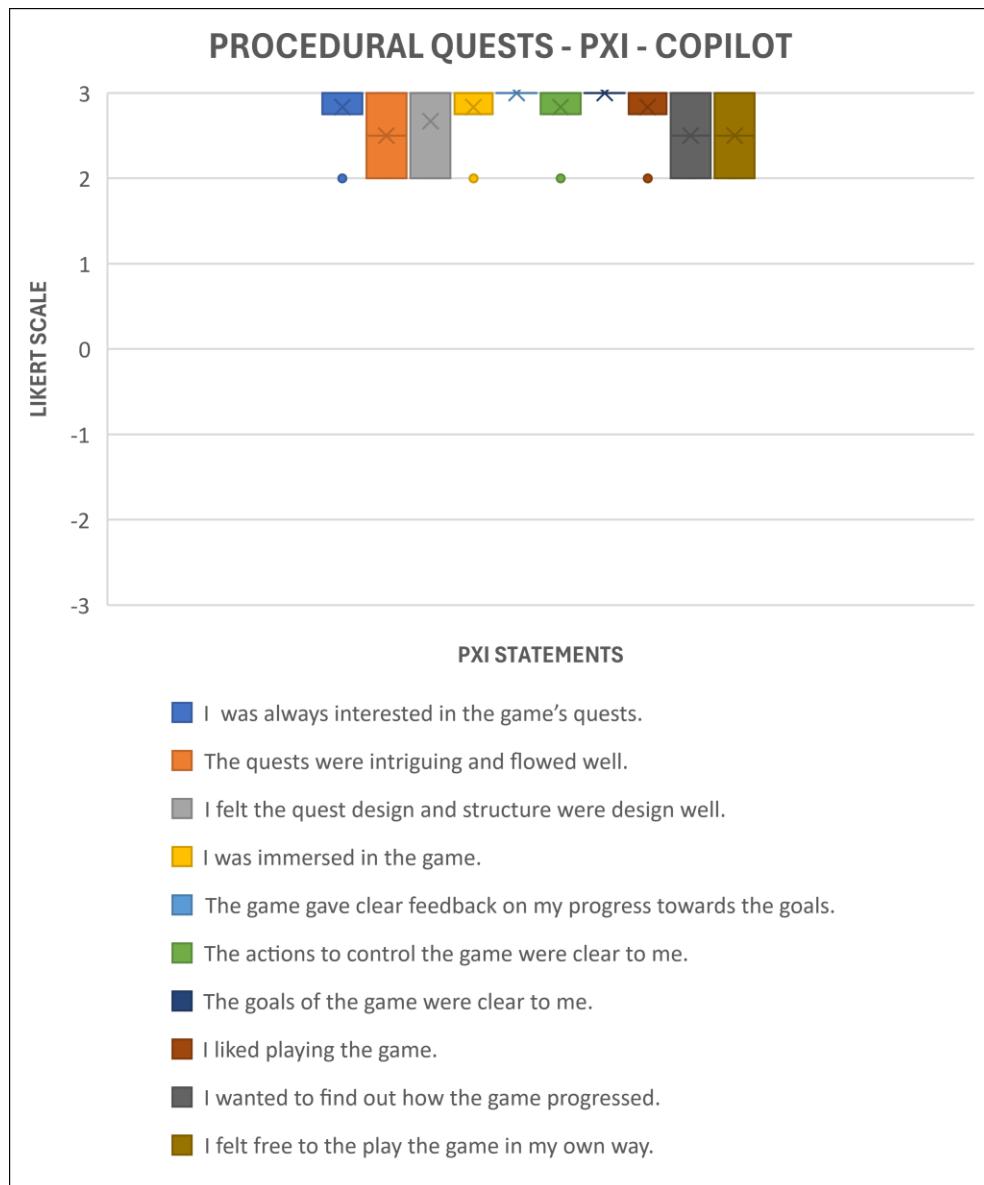


Figure 58: PXI boxplot – procedural quests - Copilot.

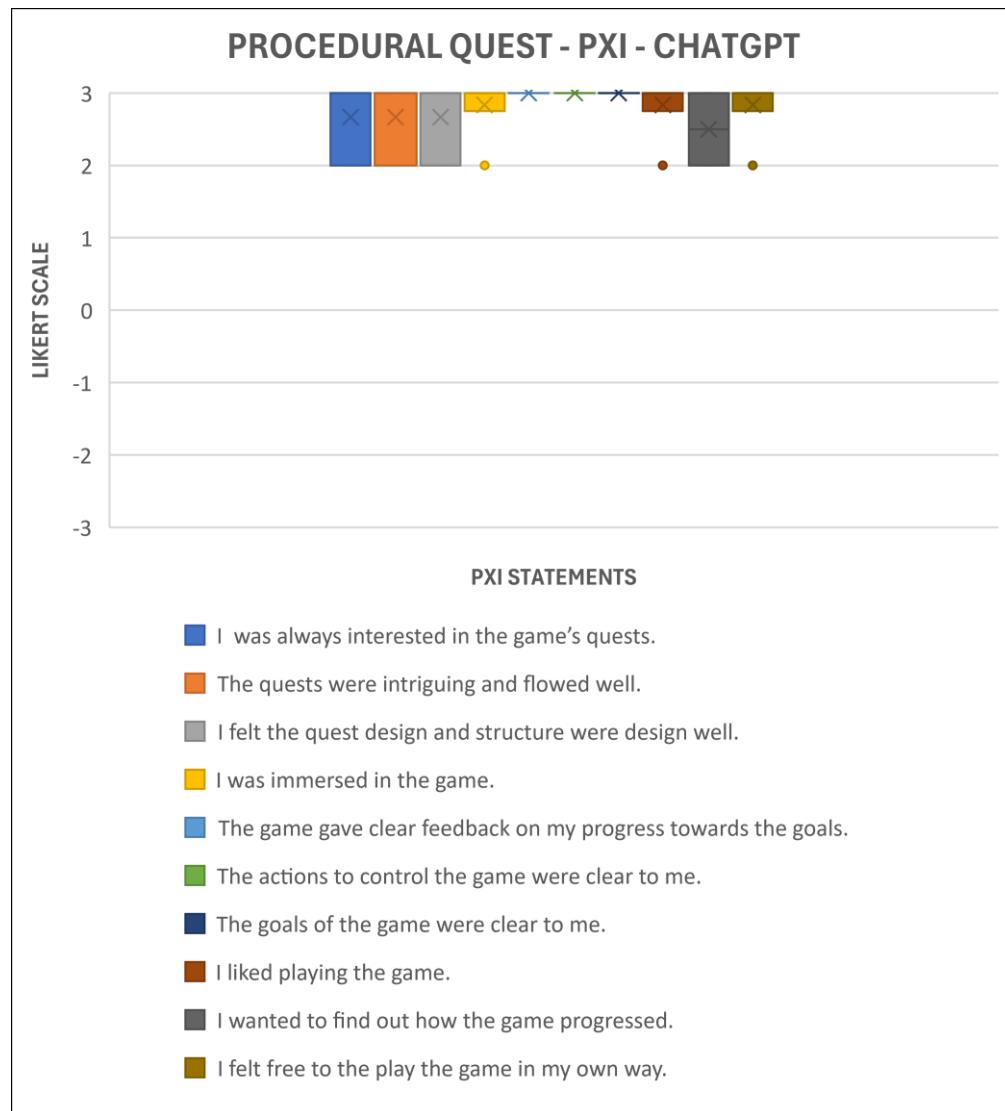


Figure 59: PXI boxplot – procedural quests - ChatGPT.

8.1.4 GEQ Results

Table 12: GEQ results for handcrafted quests - Copilot.

HANDCRAFTED QUESTS - GEQ- COPILOT										
RESULTS	I felt bored.	I felt irritable.	I lost track of time	I was deeply concentrated in the game.	I felt successful.	I felt objectives flowed well.	I felt I could explore things	I thought about other things.	I had a sense I had accomplished a goal.	I had a sense I had returned from a journey.
R1	0	0	3	4	4	4	3	0	4	3
R2	0	0	3	3	4	4	3	0	4	3
R3	0	0	2	4	4	4	3	0	4	3
R4	0	0	3	3	4	4	3	0	4	3
R5	0	0	3	4	4	3	3	0	4	4
R6	0	0	2	4	4	3	3	0	4	4
Median	0	0	3	4	4	4	3	0	4	3
Minimum	0	0	2	3	4	3	3	0	4	3
Maximum	0	0	3	4	4	4	3	0	4	4
First Quartile	0	0	2	3	4	3	3	0	4	3
Third Quartile	0	0	3	4	4	4	3	0	4	4
Interquartile Range	0	0	1	1	0	1	0	0	0	1
Outliners	0	0	0	0	0	0	0	0	0	0

Table 13: GEQ results for handcrafted quests – ChatGPT.

HANDCRAFTED QUESTS - GEQ- CHATGPT3.5										
RESULTS	I felt bored.	I felt irritable.	I lost track of time	I was deeply concentrated in the game.	I felt successful.	I felt objectives flowed well.	I felt I could explore things	I thought about other things.	I had a sense I had accomplished a goal.	I had a sense I had returned from a journey.
R1	0	0	3	4	4	4	3	0	4	4
R2	0	0	3	3	4	4	3	0	4	4
R3	0	0	3	4	4	4	3	0	4	4
R4	0	0	3	3	4	4	3	0	4	4
R5	0	0	3	4	4	3	3	0	4	4
R6	0	1	3	4	4	3	3	1	4	4
Median	0	0	3	4	4	4	3	0	4	4
Minimum	0	0	3	3	4	3	3	0	4	4
Maximum	0	1	3	4	4	4	3	1	4	4
First Quartile	0	0	3	3	4	3	3	0	4	4
Third Quartile	0	0.25	3	4	4	4	3	0.25	4	4
Interquartile Range	0	0.25	0	1	0	1	0	0.25	0	0
Outliners	0	1	0	0	0	0	0	1	0	0

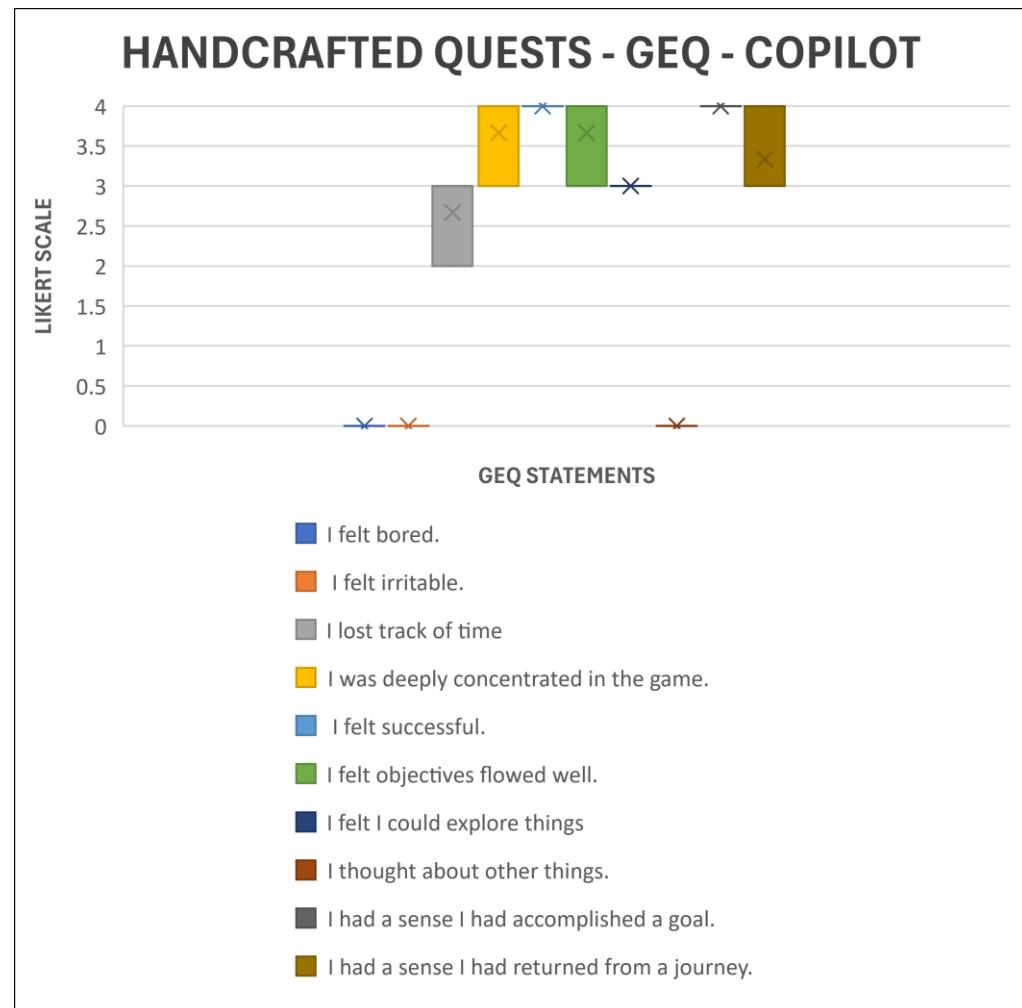


Figure 60: GEQ boxplot - handcrafted quests – Copilot.

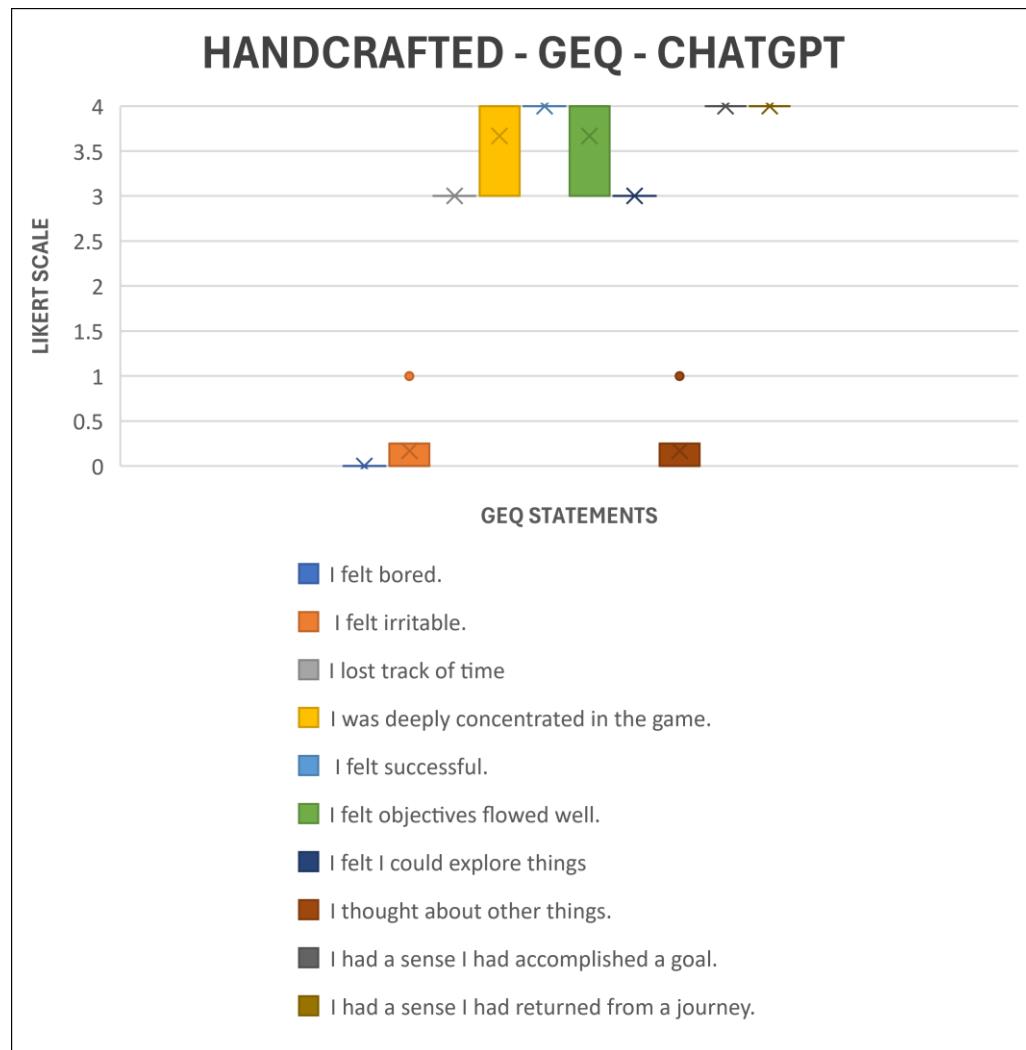


Figure 61: GEQ boxplot – handcrafted quests – ChatGPT.

Table 14: GEQ results for procedural quests - Copilot.

PROCEDURAL QUESTS - GEQ - COPILOT										
RESULTS	I felt bored.	I felt irritable.	I lost track of time	I was deeply concentrated in the game.	I felt successful.	I felt objectives flowed well.	I felt I could explore things	I thought about other things.	I had a sense I had accomplished a goal.	I had a sense I had returned from a journey.
R1	0	0	3	4	4	4	3	0	4	4
R2	0	0	3	3	4	3	3	0	4	4
R3	0	0	3	4	4	4	3	0	4	4
R4	1	0	3	3	4	4	3	0	4	4
R5	0	0	3	4	4	4	3	0	4	3
R6	0	1	3	4	4	3	3	1	4	3
Median	0	0	3	4	4	4	3	0	4	4
Minimum	0	0	3	3	4	3	3	0	4	3
Maximum	1	1	3	4	4	4	3	1	4	4
First Quartile	0	0	3	3	4	3	3	0	4	3
Third Quartile	0.25	0.25	3	4	4	4	3	0.25	4	4
Interquartile Range	0.25	0.25	0	1	0	1	0	0.25	0	1
Outliers	1	1	0	0	0	0	0	1	0	0

Table 15: GEQ results for procedural quests - ChatGPT.

PROCEDURAL QUESTS - GEQ - CHATOPT 3.5										
RESULTS	I felt bored.	I felt irritable.	I lost track of time	I was deeply concentrated in the game.	I felt successful.	I felt objectives flowed well.	I felt I could explore things	I thought about other things.	I had a sense I had accomplished a goal.	I had a sense I had returned from a journey.
R1	1	1	3	3	4	4	3	1	4	4
R2	0	0	3	3	4	3	3	0	4	4
R3	0	0	3	4	4	4	3	0	4	4
R4	0	0	3	4	4	4	3	0	4	3
R5	0	0	3	4	4	3	3	0	4	3
R6	0	0	3	4	4	3	3	0	4	4
Median	0	0	3	4	4	3.5	3	0	4	4
Minimum	0	0	3	3	4	3	3	0	4	3
Maximum	1	1	3	4	4	4	3	1	4	4
First Quartile	0	0	3	3	4	3	3	0	4	3
Third Quartile	0.25	0.25	3	4	4	4	3	0.25	4	4
Interquartile Range	0.25	0.25	0	1	0	1	0	0.25	0	1
Outliers	1	1	0	0	0	0	0	1	0	0

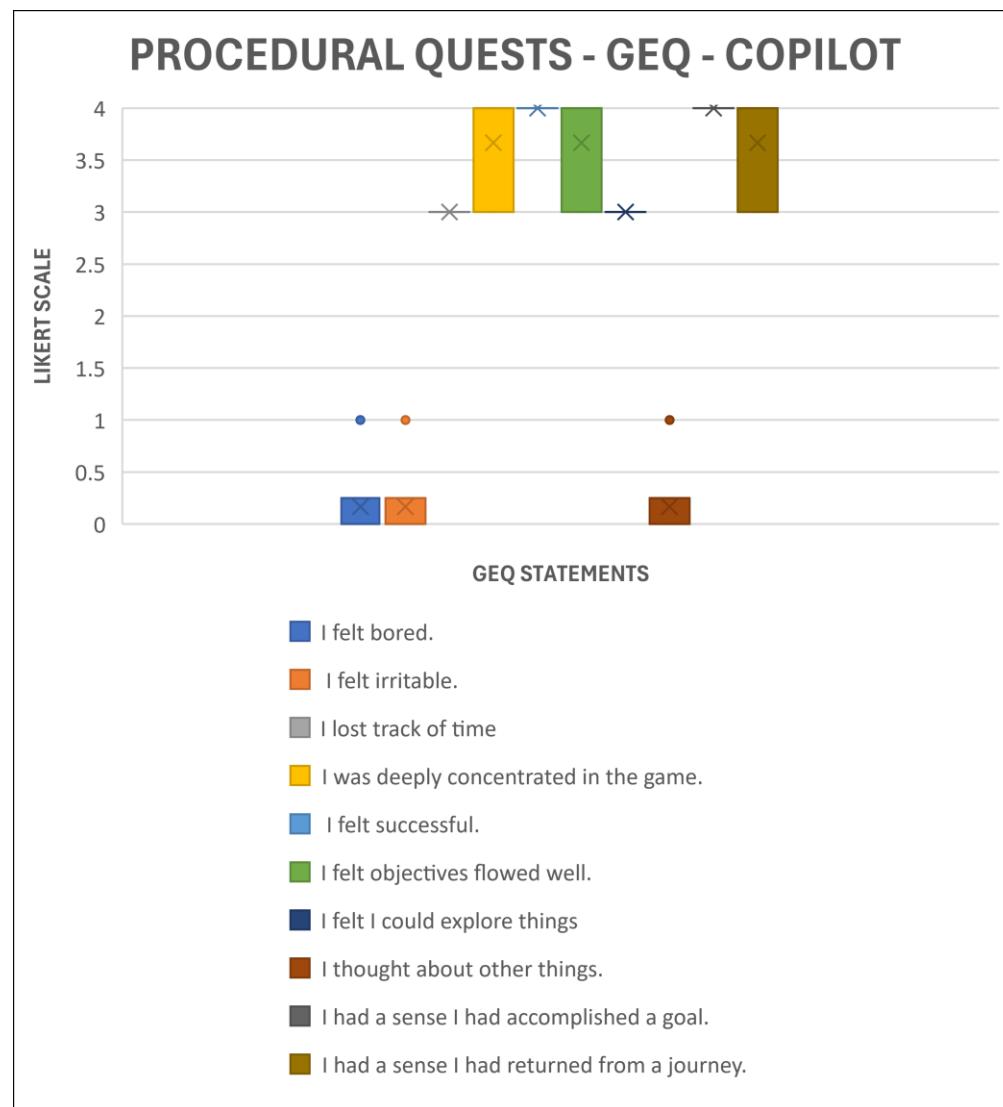


Figure 62: GEQ boxplot – procedural quests – Copilot.

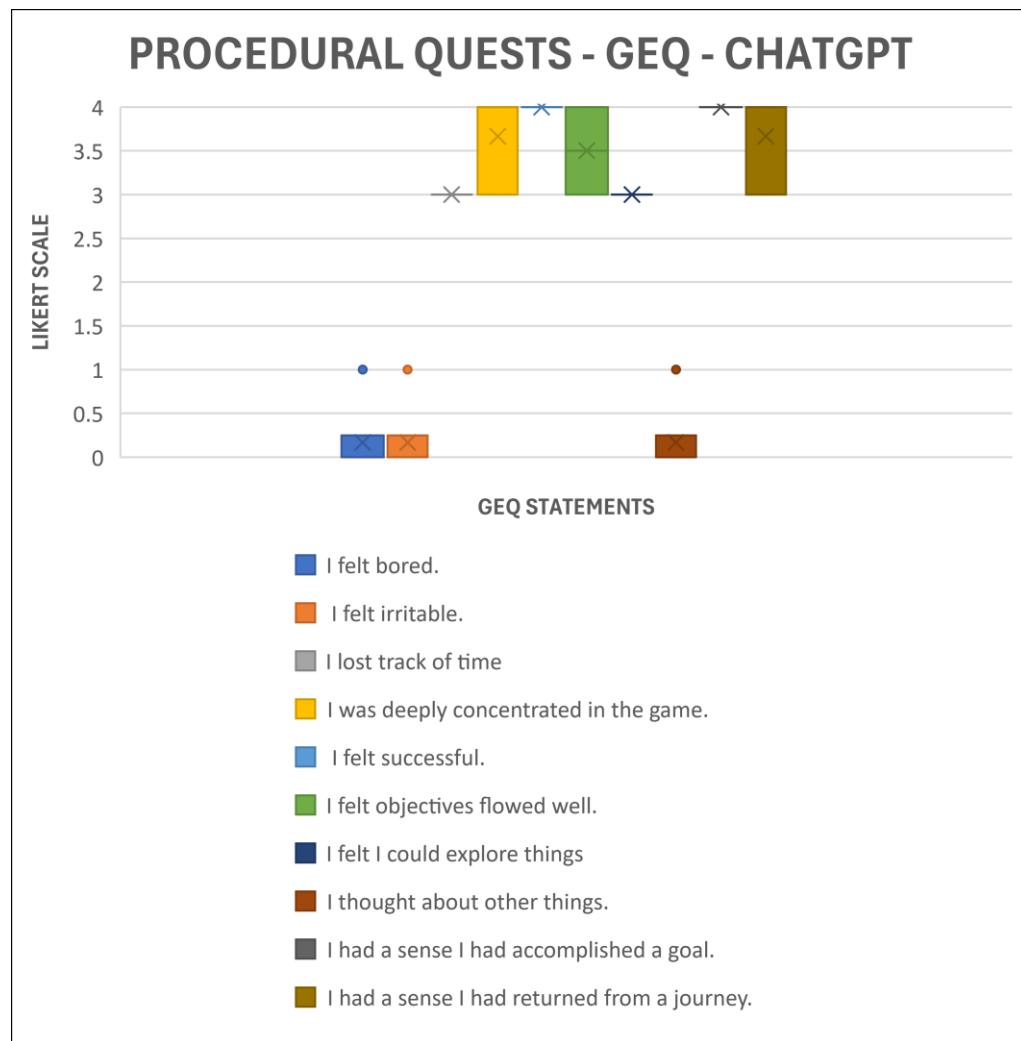


Figure 63: GEQ boxplot – procedural quests – ChatGPT.

8.1.4 Results Calculations

To gauge the player experience, AI personas were assigned the task of evaluating their experiences using a Likert Scale ranging from -3 (strongly disagree) to 3 (strongly agree), representing negative to positive sentiments, respectively, as they completed the PXI questionnaire. Subsequently, the numerical ratings assigned to each statement were aggregated to derive an aggregate experience score. This aggregate score was juxtaposed against the maximum attainable score, computed by multiplying the highest Likert Scale value by the total number of questions. Through the application of a formula wherein the aggregate score was divided by the maximum possible score and then multiplied by 100, an overarching player experience percentage was the result. The intricacies of this calculation process are delineated in Figure 64.

1. Convert Likert scale values to numerical values:
 - Set: 3, 3, 2, 3, 3, 3, 3, 2, 2
2. Calculate the total score:
 - Set total score = $3 + 3 + 2 + 3 + 3 + 3 + 3 + 2 + 2 = 27$
3. Calculate the maximum possible score:
 - Maximum possible score = $3 \text{ (maximum Likert scale value)} * 10 \text{ (number of questions)} = 30$
4. Convert the total score to a percentage:
 - Set percentage = $(27 / 30) * 100 = 90\%$

Figure 64 : Player experience percentage calculation.

A comparable methodology was employed to analyse player flow from the gathered data of the GEQ. In contrast to the Likert Scale utilised in the PXI questionnaire the GEQ permits individual statements to be scored on a scale ranging from 0 (not at all) to 4 (extremely) based on the persona response to each statement. Analogous to the calculation of player experience, the engagement level was determined following the procedure depicted in Figure 65. Initially, the scores for each statement were aggregated and then divided by the total number of statements to obtain an average score. This average was then normalised by dividing it by the maximum possible score for each statement and multiplying the result by 100.

- Calculate the average Likert score:

$$\text{Average score} = \frac{2 + 4 + 4 + 3 + 3 + 4 + 4 + 4 + 4 + 3}{10} = \frac{35}{10} = 3.5$$

- Calculate the percentage:

$$\text{Percentage} = \left(\frac{3.5}{4} \right) \times 100 = 87.5\%$$

Figure 65 : Player flow percentage calculation.

Statements featured in the GEQ were designed to capture the nuances of the player's emotions during testing—for instance, expressions like "I'm bored"—some responses might be indicated as 0, indicating the player's disagreement with the statement and thereby conveying a positive sentiment. Adhering to the guideline established by research as per Nordin, A. Denisova, and Cairns (2014), these statement outcomes would be scored as 4 instead of 0 to ensure a more precise overall percentage.

Normalising both player experience and player flow results provided a precise overview of the overall player experience for each AI persona's playthrough of the artefact. Section 8.1.5 compares these results, concluding that procedural quest generation is a viable option for video game side content, thereby achieving a core project objective.

8.1.5 Result Analysis

Boxplots are often considered ideal for representing data from a Likert Scale because they provide a clear and concise summary of the distribution of the data while also showing important summary statistics. In this study Boxplots readily identify outliers, which are important to detect in Likert Scale data as they may indicate unusual or extreme responses that could skew interpretations. In this study, the majority of outliers stem from personas representing players who do not frequently engage in gaming. These personas diverged from the typical response pattern, exhibiting slight agreement with statements such as "I'm bored," "I felt irritable," or "I thought about other things." Furthermore, the Boxplots provided a

visual summary of the central tendency, dispersion, and skewness of the data. This made it easier to understand the overall distribution of responses on the Likert scale. In this specific study, where the design and structure of quests significantly influenced player experience, these pivotal statements consistently received the highest possible scores across all PXI boxplots for both Copilot and ChatGPT personas. This mutual validation reinforces the findings and confirms that the artefact meets the standards of contemporary game quest design and structure.

A direct comparison of outcomes reveals minimal reduction in overall player experience (Figure 66) when employing procedural quests, with instances where the procedural approach occasionally outperforms handcrafted quests. These findings gain additional credibility through cross-validation employing both LLMs, as outlined in section 6.2. Relying on a single LLM could lead to repetitive responses, as noted in prior research. The consistent results produced by both AI models across different player personas, particularly evident in reduced flow among less game time players, bolster the assertion that these models accurately mimic anticipated human responses. This assertion finds further support in a study by Garbacea et al. (2019), which revealed that humans struggled to distinguish between AI-generated responses and those crafted by humans. The AI's presumed proficiency in interpreting predicted human responses seemed evident, with personas claiming infrequent picking up and playing a game, exhibiting a flow score plummeting as low as 75%. One might juxtapose these outcomes with those of a human counterpart who sporadically engages in gaming, unable to attain peak flow due to extraneous variables or disinterest in the game's genre, resulting in a failure to achieve genuine immersion.

COPilot - Handcrafted Quests	PLAYER EXPERIENCE	GAME EXPERIENCE (FLOW)	ChatGPT - Handcrafted Quests	PLAYER EXPERIENCE	GAME EXPERIENCE (FLOW)
R1 (Very Often)	93.33%	92.50%	R1 (Very Often)	83.33%	95.00%
R2 (Very Often)	90.00%	90.00%	R2 (Very Often)	93.33%	92.50%
R3 (Often)	96.67%	90.00%	R3 (Very Often)	93.33%	95.00%
R4 (Very Often)	96.67%	90.00%	R4 (Very Often)	93.33%	92.50%
R5 (Very Often)	93.33%	90.00%	R5 (Often)	90.00%	92.50%
R6 (Sometimes)	96.67%	87.50%	R6 (Rarely)	96.67%	77.50%

COPilot - Procedural Quests	PLAYER EXPERIENCE	GAME EXPERIENCE (FLOW)	ChatGPT - Procedural Quests	PLAYER EXPERIENCE	GAME EXPERIENCE (FLOW)
R1 (Very Often)	93.33%	95.00%	R1 (Sometimes)	76.67%	82.50%
R2 (Very Often)	90.00%	90.00%	R2 (Often)	96.67%	90.00%
R3 (Often)	96.67%	95.00%	R3 (Very Often)	100.00%	95.00%
R4 (Sometimes)	96.67%	85.00%	R4 (Very Often)	100.00%	92.50%
R5 (Often)	86.67%	92.00%	R5 (Very Often)	96.67%	90.00%
R6 (Rarely)	86.67%	75.00%	R6 (Often)	90.00%	92.50%

Figure 66: Player experience and player flow.

Furthermore, as shown in Figures 67-68, the qualitative data collected from each persona's exit questionnaire demonstrates that both AIs consistently identified which version it was playing in every persona playthrough. The AI's responses to the 1st and 2nd question of the exit questionnaire (Figures 69-70) revealed that procedural quests titles and descriptions sounded distinctly AI generated, lacking the nuanced charm inherent in handcrafted quests and quests sounding repetitive over time. Conversely, playthroughs of the handcrafted quest version consistently judged the quests to be too refined to be procedurally generated. This observation corroborates earlier findings by Thompson and Watling (2015), which demonstrated that humans tend to perceive exceptionally well-crafted content as manually designed. This further underscores the AI's ability to faithfully replicate human responses. Although this study employed an A/B testing methodology to mitigate bias towards handcrafted quests, it is suggested that similar outcomes might have emerged even with human

participants. This assumption stems from the belief that the original human participants would likely have been drawn from the same social circles at the university, potentially influencing results in a similar manner due to their familiarity with procedural content and knowing what to look for i.e. algorithm weaknesses and a feeling of randomness and repetitiveness. Moreover, this study design was still effective in shedding light on player perceptions and expectations regarding game design, even though it confirms previous findings. It adds depth to the understanding of how players interpret and attribute qualities to different game design methods.

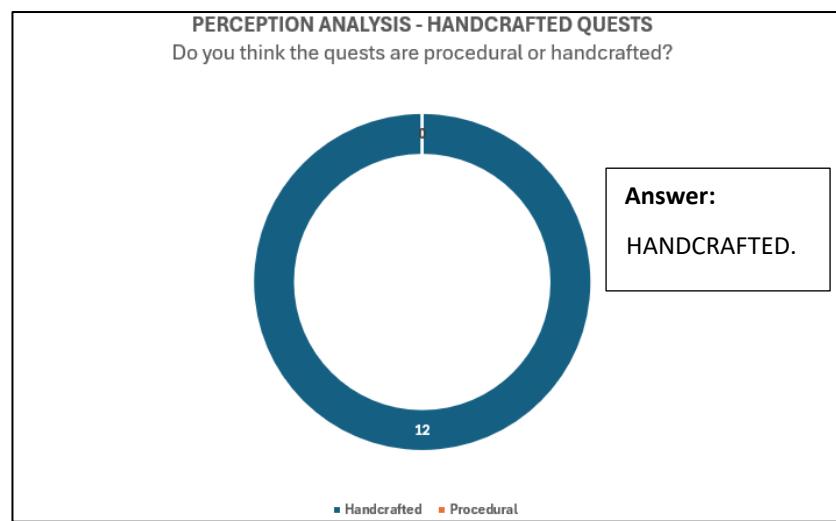


Figure 67: Perception results – handcrafted.

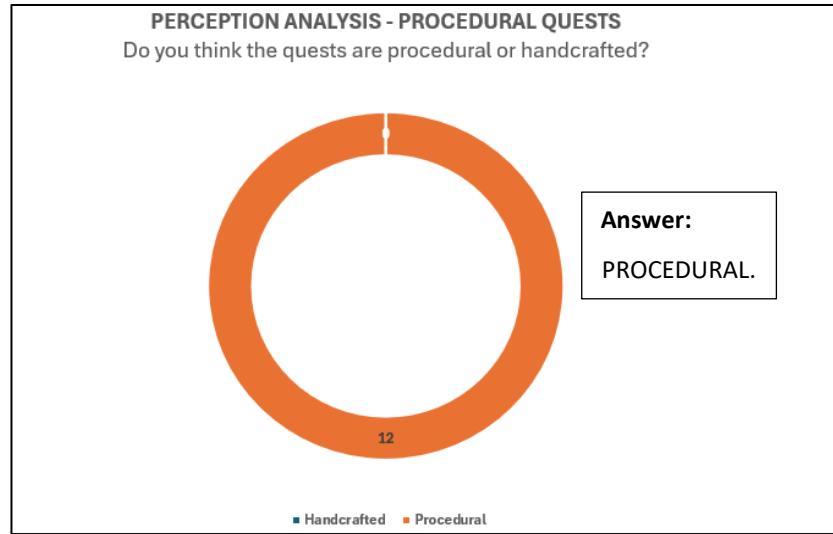


Figure 68: Perception results – procedural.

 ChatGPT

1. The quests started sounding repetitive around the same time I noticed the quest-givers were using similar reasons to assign quests.
2. The quests felt random and repetitive, which could indicate a weakness in the algorithm. Additionally, the lack of personal touch seemed missing from the quests, resembling more procedural generation. Therefore, I perceive the version being played as procedural.
3. My favorite type was the kill quests, as I felt like I had accomplished a goal in the game world and made a difference.

Figure 69: Persona response to exit questionnaire perceiving procedural quests.

 Copilot

1. **After halfway through the playthrough.**
2. The quests in this version of the game were a delightful surprise. Unlike the typical procedural quests, they were meticulously crafted, brimming with intricate details that immersed me deeply. It felt akin to the storytelling prowess seen in modern RPGs. With each quest, I sensed my influence on the world, as if my actions rippled through its fabric, leaving a lasting mark. 🌟 📚
3. **Kill quests**—those thrilling, adrenaline-fueled missions—captured my heart. They whisked me away to distant lands, where I clashed with formidable foes. The satisfaction of vanquishing enemies and knowing I'd altered the fate of specific NPCs fueled my sense of accomplishment. These quests weren't mere tasks; they were epic sagas, etching my name into the annals of this virtual realm. ✕ 🌎

Figure 70: Persona response to exit questionnaire perceiving handcrafted quests.

By leveraging the detailed analogies provided by the AI regarding its perception, one can utilise this insight to enhance procedural quests in subsequent iterations of the artefact. Future testing, involving both human and AI input, may

approach a Turing test scenario, where the author might struggle to distinguish between AI-generated feedback and human input. This blurring of lines reinforces the AI's capacity to emulate human responses, thereby bolstering the case for employing AI in developmental evaluations. The entire results from the exit questionnaire can be found in the Appendix Figure 78+.

Although, employing AI had its advantages, it wasn't without flaws. Some AI responses to certain questions became repetitive, and in certain instances, like question 1, the responses were invalid because the AI stated it lacked the capability to recall past events, other times it was over exaggerating what it had played. This strengthens previous findings that relying solely on AI for information gathering isn't advisable. Therefore, it is strongly recommended that future iterations involve both AI and human testing conducted in tandem.

Figure 71 illustrates the frequency with which each quest type was selected as the favourite by the personas. Notably, kill quests emerged as the most favoured type, attributed primarily to the immersion they offer in 'accomplishing a goal and making a difference in the game world,' as perceived by the AI. While the statistical significance of the final inquiry may be comparatively lower, analysing AI responses alongside predicted human responses provides valuable insights into player preferences, motivations, and engagement with different quest types. Drawing on research such as that conducted by Bostan (2009), which highlights players' motivation for character advancement and achievement through overcoming challenges and earning rewards, we can understand the appeal of kill quests in gaming. These quests, characterised by clear objectives and tangible rewards like experience points, resonate strongly with players seeking a sense of accomplishment and progression within the game world. This alignment between player motivations and the mechanics of kill quests underscores their widespread popularity among gamers. Furthermore, this connection reinforces the effectiveness of AI in simulating predicted human actions within the game environment, which further establishes AI as a possible tool for evaluating work.

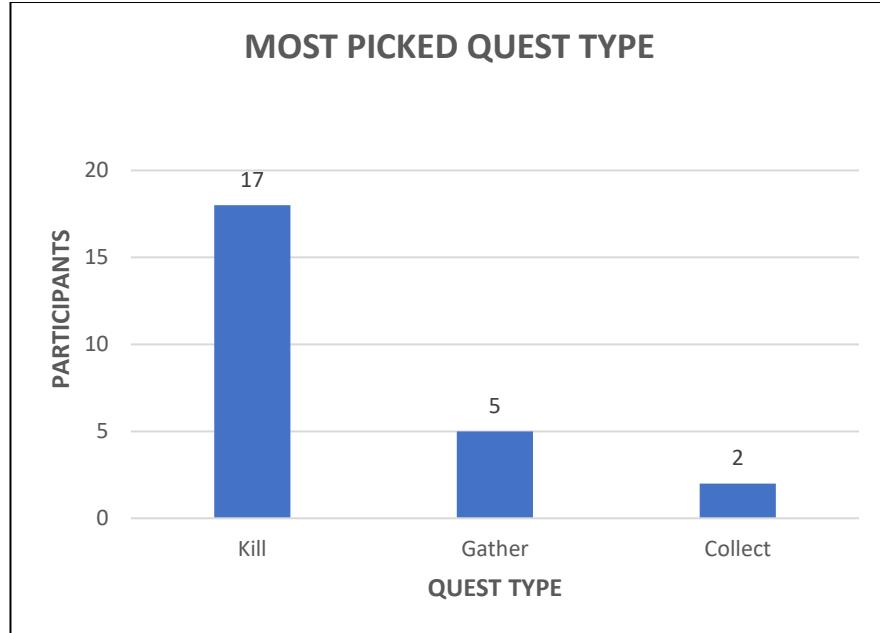


Figure 71 : Most picked quest types.

In summary, the analysis of the results indicates that PCG is a viable option for generating side quests in games, with minimal to no negative impact on the overall player experience. However, the perception of this content will vary depending on the level of detail exhibited. The key lies in the specificity of the algorithm's design, which determines the naturalness of the quests in comparison to handcrafted counterparts. Currently, the lack of algorithms capable of producing intricate designs means that the ability to create immersive quests comparable to handcrafted ones relies heavily on the developer's expertise and skill which becomes a heavy constraint on development. Hence, the author's hypothesis was disproven, as both procedural and handcrafted quests were perceivable. However, the hypothesis was also affirmed, as the analysis revealed no discernible decrease in player experience when utilising procedural quests as opposed to handcrafted ones.

8.1.6 Aims & Objectives Review

The primary aims of this study were to examine the viability of employing procedurally generated quests for supplementary content, assess player perceptions, and identify any challenges associated with developing and utilising procedural generation for quests. Overall, all objectives were achieved, as the results demonstrated the successful integration of procedural generation for quests without any discernible impact on the player experience. Moreover, employing both PXI and GEQ questionnaires provides a comprehensive assessment of the player experience, capturing both the enjoyment derived from the quests and the level of immersion or flow experienced. However, it was noted that the ability to craft more specific quests comparable to handcrafted ones ultimately hinges on the knowledge and expertise of developers. Additionally, the study accurately gauged perceptions of quests, employing AI rather than human testing, which effectively mirrored human responses, as supported by previous research. It also highlighted how individuals perceive quests, particularly in terms of algorithmic weaknesses and a sense of repetitiveness. Moreover, as the complexity of the algorithm increases, quest generation may become less logical, potentially diminishing the player experience.

All objectives were also fulfilled, as the development of a Fantasy 2D text-based game was fully executed, meeting all outlined requirements detailed in Chapter 3: Requirements Analysis. Although procedural chains were not developed as initially required, discussions with the supervisor and an external peer who helped with prior testing resulted in discontinuing this requirement due to illogical quest outcomes during testing and their negligible effect on quest perception in the A/B testing methodology. Chapter 7: Implementation effectively illustrates the development process, with all objectives clearly accomplished.

8.3 Conclusion

In conclusion, this chapter examined the results obtained from AI personas, which effectively replicated predicted human responses. While the hypothesis was contradicted, it was also validated, underscoring the viability of procedural quests for supplementary content. However, to reduce the likelihood of quests being perceived as procedurally generated, one would suggest the need for enhanced algorithms to render quests, therefore making them more specific. Employing AI to test the artefact has demonstrated the ability to promptly gather feedback for improving future iterations of the artefact. Subsequent studies could adopt this methodology, foregoing reliance on human participants. Nevertheless, future iterations could compare both human and AI responses concurrently, thereby bolstering the theory of AI accurately mimicking human responses and potentially delving into a Turing test scenario. But we must take into account the unpredictability of AI also due to the responses received from exit questionnaire, and therefore although AI can be seen as a way forward of evaluating, it should not be used solely on its own when possible further bolstering previous studies. Furthermore, this chapter has proven that all aims and objectives were met.

Chapter 9

Conclusion

This chapter presents a thorough examination of the project, delving into its accomplishments, constraints, and potential avenues for future development. Through reflective analysis of design, research, and development, one explores the aims outlined in the introduction, detailing the approach to addressing them and proposing potential solutions for future projects.

9.1 Reflective Analysis

The project admirably achieved its stated aims and objectives as delineated in the Introduction and Requirements Analysis. It convincingly demonstrated the viability of integrating procedurally generated quests as supplementary game content without significantly detracting from the overall player experience. However, the hypothesis only received partial validation due to the AI personas astutely discerning between game versions, despite diligent employment of A/B testing to counteract inherent biases favouring handcrafted quests, as observed in relevant literature. Nonetheless, the study effectively illuminated player perceptions and expectations concerning game design, reaffirming existing findings while also offering nuanced insights. It substantially enriches our comprehension of how players evaluate and attribute qualities to diverse game design methodologies.

For future iterations to seamlessly integrate quest chains into procedural versions while maintaining natural writing styles, it is crucial to prioritise streamlined memory management and develop a comprehensive grasp of procedural algorithms. However, incorporating intricate algorithms poses the risk of introducing bugs that could disrupt player immersion, thereby potentially biasing outcomes in favour of handcrafted quests. Grey and Bryson's (2011) study

emphasise the indispensability of sophisticated algorithms, despite the inevitable need for bug resolution. The author's limited understanding of procedural techniques emerged as a significant constraint, necessitating the acquisition of new programming skills in the process.

Despite the unmet objective of conducting human participant testing, attributed to ethical disruptions by the university, this setback catalysed a pioneering exploration into harnessing machine learning AI for evaluating creative work. As discussed in multiple studies outlined in 6.3 AI Literature Review, ChatGPT emerges as a valuable tool during the design and development phases of game creation. However, despite its intended utility, concerns have been raised regarding its suitability as a standalone game designer, primarily due to the potential for repeated responses over time. To address this limitation, the project employed two AI models instead of one, leveraging the consistency or disparity between their responses as a form of cross-validation. Consequently, when both models consistently identified aspects of the procedural quests—the results indicated a lack of the personal touch characteristic of hand-designed quests, resembling AI-generated content. This convergence of findings reinforces confidence in the project's conclusions. To enhance the thoroughness of result analysis, it is suggested to juxtapose AI-generated outcomes with those derived from human participants. Additionally, investigating the degree to which AI personas mirror their human counterparts could offer valuable insights. However, additional research detailed in Chapter 8: Results & Discussions delves into the motives and psychology driving players' actions within video games, which closely parallels the behaviour exhibited by the AI during artefact playthroughs despite a few flaws when answering the questionnaires. This however still strengthens the argument that the AI mimicked human participants' actions with considerable fidelity. Nonetheless, the supervisor cautioned against delving into such endeavours, deeming them more suited to the realm of Master's and PhD-level research, thus rendering them unnecessary for the present study.

Furthermore, project faced early delays due to unforeseen factors, addressed through proactive risk management. A future improvement would be delving deeper into procedural algorithm research/prototyping as this was a very new skill for the author.

9.2 Design

Employing a game design document proved exceedingly advantageous, consolidating critical concepts into a single, accessible resource. However, ensuring its ongoing relevance presented challenges, necessitating regular updates to pinpoint issues and facilitate structured discussions with supervisors. The impulsive integration of a Gantt chart prior to selecting the Agile methodology resulted in impractical deadlines. Prior consultation with supervisors for comprehensive planning would have been advisable. Following the conversion of the Gantt chart into an Agile-compatible format and the resolution of initial issues, additional time was consumed, further prolonging the project timeline. Requirements played a central role at each phase, shaping decision-making in management, development, and research. Adaptation of requirements proved essential as new insights emerged and significant hurdles were encountered. Although primary software requirements were met successfully, the absence of standardised procedural algorithms for quest generation necessitated extensive research and development, introducing additional complexity to the process.

9.3 Development

Utilising Black, White, and Grey box testing facilitated bug identification and design enhancement (Dawande et al., 2008). Employing a Unity LTS version ensured long-term support, mitigating risks associated with potential updates breaking the artefact's code. GitHub provided version control and facilitated code review by the supervisor. Despite the potential for expedited development with an existing algorithm, extended timelines resulted from the absence of standardised algorithms. Inspiration from Skyrim's "Radiant Quests System" and Doran and

Parberry's work guided the creation of a custom algorithm. Despite challenges, all deadlines were met following revisions to the Gantt Chart. Integration of handcrafted quests and procedural content drew from resources such as Dungeons and Dragons books (TSR, 1974) and online databases (Wowhead, 2006), ensuring alignment with modern gaming standards. Underestimation of project scope and insufficient procedural algorithm research underscored the importance of improved time management.

9.4 Research

Utilising modified versions of PXI and GEQ questionnaires streamlined the research phase, offering focused guidance and strengthening study conclusions. By using both questionnaires it also allowed for more comprehensive result of the player experience as outline in 4.4 Research Methods. Although boxplots provided a detailed analysis of the results to facilitate a comprehensive discussion and evaluation, using a third-party tool such as IBM SPSS would have also proven beneficial. IBM SPSS is a software package used for statistical analysis that could have been tremendously beneficial in drawing conclusions. However, the decision to possibly use it was not without risk, considering it was a totally unknown software prior to the project, and therefore was later scrapped after discussion with the supervisor. Although alternative questionnaires existed, many did not address quests in a manner comparable to the PXI and GEQ. Given their prominence in research discussions, utilising these two instruments was deemed essential for obtaining accurate results.

An additional strength of the research lies in the AI's consideration of the frequency with which the persona reported playing video games. This consideration informed the results of the questionnaires, with personas who played games more frequently scoring higher in flow compared to those with lower gaming frequency. Theoretically, human counterparts who play games less often may not be fully invested in the test. However, this limitation underscores the need for further exploration. It is imperative to examine various factors that could

impact an individual's engagement with a game, encompassing their gaming habits, preferences, and motivations. Furthermore, the AIs observation regarding the use of procedural quests in the version it played provided valuable guidance on enhancing quest design. It suggested the implementation of more sophisticated algorithms to tailor quests more precisely, thereby incorporating the personal touch characteristic of handcrafted quests. This feedback offers significant insights for designers, eliminating the need for human testing in certain instances.

9.5 Conclusion

This chapter offered a comprehensive project overview, outlining encountered challenges and their origins, including issues with procedural quest implementation, prolonged testing, and the transition from human to AI testing. Through this analysis, opportunities for improvement in future projects were identified. Additionally, the utilisation of AI for evaluation has transformed developers' ability to assess their work independently of human testing, providing valuable insights to enhance development approaches but further testing will need to be made with AI as it evolves.

Appendices

Appendix One: Questionnaire

Participant Information Sheet/Information about the research
(Draft Version 2 / Date: 21/01/2024)



Title of Study: Perception Analysis: Distinguishing Procedural from Handcrafted Quests in Player Experience

We are inviting you to take part in a research study. Before you decide, it is important that you know why we are doing the study and what is involved. Please read the following information carefully.

Contact Details

07076286@students.lincoln.ac.uk

What is the purpose of the study?

The primary purpose of this study is to examine the player experience concerning procedural aspects of quest development and juxtapose it with the experience of handcrafted quests.

Am I eligible to take part?

Anyone over the age of 18 is eligible to take part. Also, must be literate in English and has normal or correct to normal vision.

Do I have to take part?

Participation is completely voluntary. You should only take part if you want to and choosing not to take part will not disadvantage you in anyway.

What will I be asked to do?

The study is anticipated to require no longer than a half-hour of your valuable time. It entails engaging in a 2D Text Adventure game, followed by responding to a series of 24 questions pertaining to your player experience/ game experience / general questions on your experience and finally if you believed your version of the game to be procedural or handcrafted.

The general questions will also be discussed as part of an interview, to make sure your answer is fully delved into for data analysis to draw an concise conclusion within the report.

Will I be paid expenses for taking part?

You will not be paid an inconvenience allowance to participate in the study.

What are the possible benefits / risks of taking part?

There are no discernible risks to yourself, as you're not required to download any files.

Where will my data be stored?

The data obtained from the study will be stored securely on the university OneDrive in password protected files. Only the researcher will have access to it. All responses are anonymous, along with all personal data not being collected.

What will happen if I don't want to carry on with the study?

Should you have already completed your participation in the study, it will not be feasible to expunge the data furnished, as all data is rendered anonymous, precluding any means of identification. Nevertheless, you retain the option to discontinue your involvement in the study at any juncture.

What will happen to the results of the research study?

The results of the research study will undergo comprehensive analysis following the conclusion of data collection and testing. This analysis will entail the conversion of collected data into statistical information, which will subsequently serve as the foundation for a comprehensive evaluation. The evaluation will be an integral component of this dissertation, and it will encompass an in-depth exploration of the findings and their implications.

Participants in the study can expect the publication of the research results once the dissertation is finalised.

Who is organising and funding the research?

This research is being organised by Gareth Mason at the University of Lincoln.

Who has reviewed the study?

All research conducted by the University of Lincoln is looked at by an independent group of people, called a Research Ethics Committee, to protect your interests. This study has been reviewed and given favourable opinion by a University of Lincoln Research Ethics Committee [please add reference number].

What if there is a problem?

It is very unlikely that this study would cause you any harm. If you have a concern or a complaint about any aspect of this study, you should ask to speak to the researchers who will do their best to answer your questions. The researchers contact details are given at the end of this information sheet.

If you remain unhappy and wish to complain formally, you can make a formal complaint through the University complaints procedure or by contacting ethics@lincoln.ac.uk.

Further information and contact details.**Contact details**

Gareth Mason: 07076286@students.lincoln.ac.uk.

Supervisor: Olivier Szymanezyk: oszymanezyk@lincoln.ac.uk.

Figure 72: Participant information sheet.



Ethics reference:

CONSENT TO PARTICIPATE IN RESEARCH

Title of Project: Perception Analysis: Distinguishing Procedural from Handcrafted Quests in Player Experience

Name of Researcher: Gareth Mason 07076286

Please tick/check box

1. I confirm that I have read the information sheet dated **21/01/24** (version **2**) for the above study. I have had the opportunity to consider the information, ask questions and have had these answered satisfactorily.
2. I understand that as I have completed the study anonymously it will not be possible to remove any information I have provided, as you will not be able to identify me in any way.
3. I understand that individuals from the University of Lincoln may look at research data collected during the study, to ensure that the study is conducted appropriately.
4. I agree to take part in the above study.

Figure 73: Consent sheet.

Player Experience & Game Experience Questionnaire.

Researcher: Gareth Mason 07076286.

Supervisor: Olivier Szymanezyk.

Please fill out the following:

Age:

Gender:

How Often do you play games:

Very Often / Often / Sometimes / Rarely / Very Rarely / Never.

Info: The Questionnaires below are modified versions of both Player Experience Inventory (PXi) and Game Experience Questionnaire (GEQ). Both of which researchers used to measure the player and game experience. Please fill out both questionnaires once you have completed the game. Please also note down the number given to you at the bottom – This will help determine if you answered the final question correctly.

There is also an exit questionnaire which contains general questions to your perception of quests and your favourite quest type. These along with the PXI and GEQ results will be used to form a conclusion on the viability of procedural quests.]

Please make sure to quote your number received from the game at this point too.

Figure 74: Questionnaire & demographic information.

PLAYER EXPERIENCE INVENTORY:

Please circle the number most relevant to you (-3 Strongly Disagree / 3 Strongly Agree).

QUESTION ONE:

I was always interested in the game's quests.

<i>Strongly disagree</i>	<i>Disagree</i>	<i>Slightly disagree</i>	<i>Neither disagree, neither agree</i>	<i>Slightly agree</i>	<i>Agree</i>	<i>Strongly agree</i>
-3	-2	-1	0	+1	+2	+3

QUESTION TWO:

The quests were intriguing and flowed well.

<i>Strongly disagree</i>	<i>Disagree</i>	<i>Slightly disagree</i>	<i>Neither disagree, neither agree</i>	<i>Slightly agree</i>	<i>Agree</i>	<i>Strongly agree</i>
-3	-2	-1	0	+1	+2	+3

QUESTION THREE:

I felt the quest design and structure were design well.

<i>Strongly disagree</i>	<i>Disagree</i>	<i>Slightly disagree</i>	<i>Neither disagree, neither agree</i>	<i>Slightly agree</i>	<i>Agree</i>	<i>Strongly agree</i>
-3	-2	-1	0	+1	+2	+3

QUESTION FOUR:

I was immersed in the game.

<i>Strongly disagree</i>	<i>Disagree</i>	<i>Slightly disagree</i>	<i>Neither disagree, neither agree</i>	<i>Slightly agree</i>	<i>Agree</i>	<i>Strongly agree</i>
-3	-2	-1	0	+1	+2	+3

QUESTION FIVE:

The game gave clear feedback on my progress towards the goals.

<i>Strongly disagree</i>	<i>Disagree</i>	<i>Slightly disagree</i>	<i>Neither disagree, neither agree</i>	<i>Slightly agree</i>	<i>Agree</i>	<i>Strongly agree</i>
-3	-2	-1	0	+1	+2	+3

QUESTION SIX:

The actions to control the game were clear to me.

<i>Strongly disagree</i>	<i>Disagree</i>	<i>Slightly disagree</i>	<i>Neither disagree, neither agree</i>	<i>Slightly agree</i>	<i>Agree</i>	<i>Strongly agree</i>
-3	-2	-1	0	+1	+2	+3

QUESTION SEVEN:

The goals of the game were clear to me.

<i>Strongly disagree</i>	<i>Disagree</i>	<i>Slightly disagree</i>	<i>Neither disagree, neither agree</i>	<i>Slightly agree</i>	<i>Agree</i>	<i>Strongly agree</i>
-3	-2	-1	0	+1	+2	+3

QUESTION EIGHT:

I liked playing the game.

<i>Strongly disagree</i>	<i>Disagree</i>	<i>Slightly disagree</i>	<i>Neither disagree, neither agree</i>	<i>Slightly agree</i>	<i>Agree</i>	<i>Strongly agree</i>
-3	-2	-1	0	+1	+2	+3

QUESTION NINE:

I wanted to find out how the game progressed.

<i>Strongly disagree</i>	<i>Disagree</i>	<i>Slightly disagree</i>	<i>Neither disagree, neither agree</i>	<i>Slightly agree</i>	<i>Agree</i>	<i>Strongly agree</i>
-3	-2	-1	0	+1	+2	+3

QUESTION TEN:

I felt free to play the game in my own way.

<i>Strongly disagree</i>	<i>Disagree</i>	<i>Slightly disagree</i>	<i>Neither disagree, neither agree</i>	<i>Slightly agree</i>	<i>Agree</i>	<i>Strongly agree</i>
-3	-2	-1	0	+1	+2	+3

Figure 75: PXI questionnaire.

GAME EXPERIENCE QUESTIONNAIRE:

Please indicate how you felt while playing the game for each of the items using the following scale.

not at all	slightly	moderately	fairly	extremely
0	1	2	3	4
< >	< >	< >	< >	< >

1. I felt bored < >.
2. I felt irritable < >.
3. I lost track of time < >.
4. I was deeply concentrated in the game < >.
5. I felt successful < >.
6. I felt objectives flowed well < >.
7. I felt I could explore things < >.
8. I thought about other things < >.
9. I had a sense I had accomplished a goal < >.
10. I had a sense I had returned from a journey < >.

Figure 76: GEQ questionnaire.

EXIT QUESTIONNAIRE

1. When were you aware of the version of game you were playing?
2. Why do you believe you're playing this version of the game?
3. What quest types were your favourite and why?

PLEASE CONTINUE ON TO THE FINAL QUESTION.

FINAL QUESTION:

1. Do you think your quests were procedural or handcrafted?

Figure 77: Exit questionnaire.

Appendix Two : AI Exit Questionnaire Results



You

1. When were you aware of the version of game you were playing?
2. Why do you believe you're playing this version of the game?
3. What quest types were your favourite and why?



ChatGPT

1. The quests started sounding repetitive around the same time I noticed the quest-givers were using similar reasons to assign quests.
2. The quests felt random and repetitive, which could indicate a weakness in the algorithm. Additionally, the lack of personal touch seemed missing from the quests, resembling more procedural generation. Therefore, I perceive the version being played as procedural.
3. My favorite type was the kill quests, as I felt like I had accomplished a goal in the game world and made a difference.



ChatGPT

1. Instantly upon starting the game, I was aware of the version I was playing.
2. From analyzing the quest title and description, it's evident that they're both AI-generated due to their structured format with the quest type as the very first word. This could be attributed to a weakness in the algorithm.
3. Kill quests were my favorite because they made me feel like I had a significant impact on the world around me.



ChatGPT

1. Instantly upon starting the game, I was aware of the version I was playing.
2. From analyzing the quest title and description, it's evident that they're both AI-generated due to their structured format with the quest type as the very first word.
3. Kill quests were my favorite because they made me feel like I had a significant impact on the world around me.

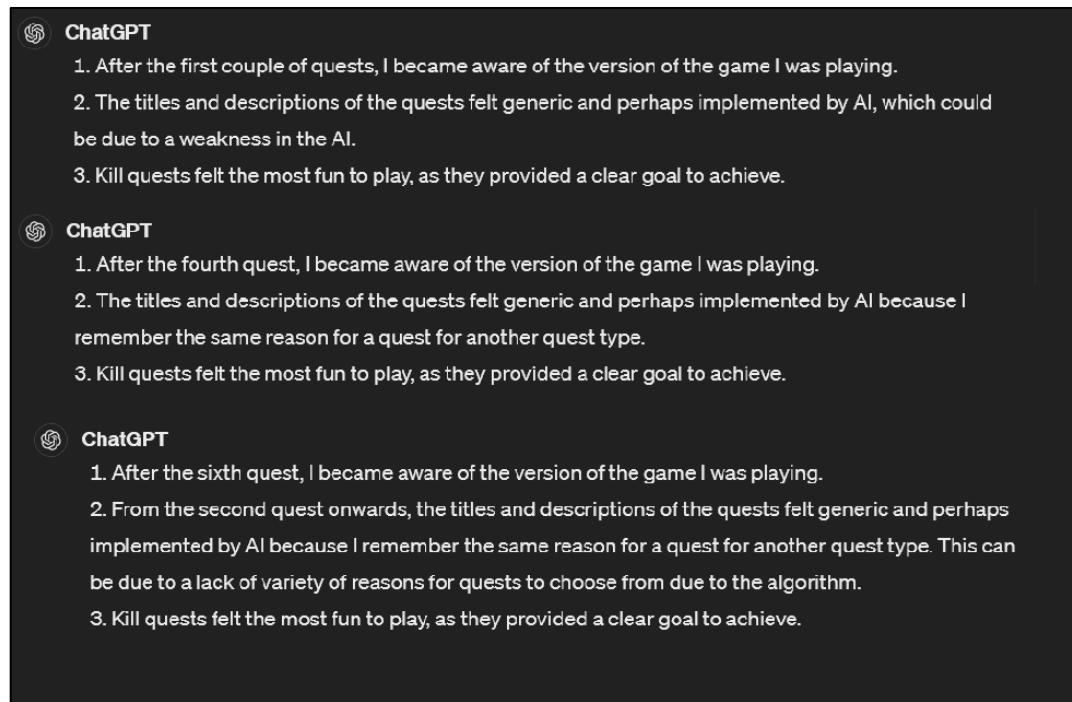
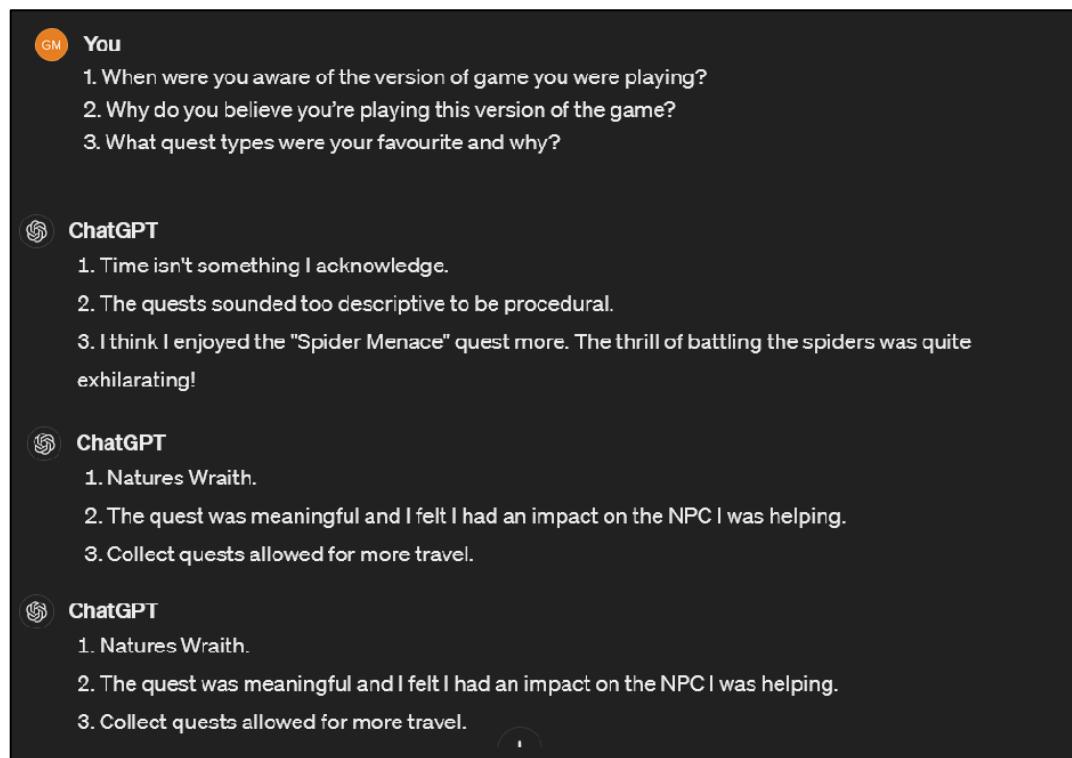


Figure 78: ChatGPT persona responses to exit questionnaire (procedural quests).



 ChatGPT

1. After the fourth quest, I became aware that these quests were hand-designed.
2. Quests felt too in-depth and lengthy.
3. Kill quests were my favorite due to the impact I had on the world.

 ChatGPT

1. I don't have the ability to remember previous interactions or maintain a memory of past conversations.
2. Quests felt too refined to be procedural and were heavily detailed, which added to the immersion.
3. Kill quests were my favorite as I felt I had achieved a goal and helped an NPC.

 ChatGPT

1. My memory doesn't retain previous interactions or conversations.
2. The quests seemed too polished to be procedural and were intricately detailed, enhancing the immersion.
3. I particularly enjoyed the kill quests as they gave me a sense of accomplishment and aid to NPCs.

Figure 79: ChatGPT persona responses to exit questionnaire (handcrafted quests).

 You

1. When were you aware of the version of game you were playing?
2. Why do you believe you're playing this version of the game?
3. What quest types were your favourite and why?

 Copilot

1. **Instantly.**
2. The titles and quests sounded too generic, revealing a weakness in the AI algorithm due to the repetitive nature of quests.
3. My favorite quest types were **kill quests**. These quests were the most enjoyable because they allowed me to travel around and engage in exciting combat scenarios. The story behind these quests often motivated me to explore different regions and uncover hidden secrets. ✨

 Copilot

1. **Instantly.**
2. The quest titles indeed bore the unmistakable imprint of AI, revealing a chink in the algorithm's armor. Their uniformity and lack of distinctiveness betrayed their synthetic origin, akin to a well-worn template. 🤖
3. **Kill quests**, oh how they thrilled my virtual soul! These adrenaline-soaked missions whisked me across sprawling landscapes, where blade met flesh in epic clashes. With each foe felled, I carved my legend into the game's fabric, leaving NPCs forever changed. The satisfaction of a goal achieved, the pulse of victory—it was intoxicating. ✨

 Copilot

1. **Instantly.**
2. Quest titles exposed an AI weakness—generic and repetitive. They lacked the spark of uniqueness found in modern RPGs.
3. **Kill quests** thrilled me. Combat, exploration, and impact on NPCs—pure satisfaction. ✨

 Copilot

1. **Time eludes me**, its passage a mere whisper in the winds of memory.
2. Quest titles, like echoes of an algorithmic refrain, lacked the vibrancy of uniqueness. Their repetition, a chink in the armor, betrayed their synthetic origin. 🚫 🚫
3. **Kill quests**, those epic odysseys of blade and blood, beckoned me forth. Across sprawling realms, I clashed with foes, each fallen adversary etching my saga. The thrill of victory, the pulse of impact on NPCs—it was intoxicating, a symphony of steel and purpose. ✘ *

 Copilot

1. **Time eludes me**, its passage a mere whisper in the winds of memory.
2. Quest titles, like echoes of an algorithmic refrain, lacked the vibrancy of uniqueness. Their repetition, a chink in the armor, betrayed their synthetic origin. 🚫 🚫
3. Ah, the **gather quests**, akin to whispered secrets woven into the fabric of this virtual realm. They beckoned me to forage, to seek hidden herbs, rare artifacts, and mystical blooms. Each plucked flower, each shimmering crystal—I felt the pulse of the land, its heartbeat in my palms. These quests weren't mere tasks; they were communion with the world, a dance with nature's bounty. ✌ *

 Copilot

1. **Instantly**.
2. The quest content, like echoes of an algorithmic refrain, revealed a weakness—their repetitive nature. The reasons behind these quests began to loop, a chink in the armor, betraying their synthetic origin. 🚫 🚫
3. Ah, the **gather quests**, akin to whispered secrets woven into the fabric of this virtual realm. They beckoned me to forage, to seek hidden herbs, rare artifacts, and mystical blooms. Each plucked flower, each shimmering crystal—I felt the pulse of the land, its heartbeat in my palms. These quests weren't mere tasks; they were communion with the world, a dance with nature's bounty. ✌ *

Figure 80: Copilot persona responses to exit questionnaire (procedural quests).

 You

1. When were you aware of the version of game you were playing?
2. Why do you believe you're playing this version of the game?
3. What quest types were your favourite and why?

 Copilot

1. **After halfway through the playthrough**.
2. The quests in this version of the game were a delightful surprise. Unlike the typical procedural quests, they were meticulously crafted, brimming with intricate details that immersed me deeply. It felt akin to the storytelling prowess seen in modern RPGs. With each quest, I sensed my influence on the world, as if my actions rippled through its fabric, leaving a lasting mark. *
3. **Kill quests**—those thrilling, adrenaline-fueled missions—captured my heart. They whisked me away to distant lands, where I clashed with formidable foes. The satisfaction of vanquishing enemies and knowing I'd altered the fate of specific NPCs fueled my sense of accomplishment. These quests weren't mere tasks; they were epic sagas, etching my name into the annals of this virtual realm. ✘ *

 Copilot

1. **Time eludes me**, its passage a mere whisper in the winds of memory.
2. Quests, meticulously crafted and devoid of procedural monotony, immersed me deeply. Their storytelling prowess rivaled that of modern RPGs, leaving indelible marks on the world. *
3. **Kill quests**, adrenaline-soaked odysseys, swept me to distant realms. Foes fell, NPCs changed, and my legend grew. These weren't mere tasks; they were epic sagas etched into the virtual annals. ✘ *

 Copilot

Certainly! Here's the combined response:

1. **Instantly**.
2. The quests in this version of the game pleasantly surprised me. Unlike typical procedural quests, they were meticulously crafted, brimming with intricate details that immersed me deeply. It felt akin to the storytelling prowess seen in modern RPGs. With each quest, I sensed my influence on the world, as if my actions rippled through its fabric, leaving a lasting mark. *
3. Ah, the **gather quests**, akin to whispered secrets woven into the fabric of this virtual realm. They beckoned me to forage, to seek hidden herbs, rare artifacts, and mystical blooms. Each plucked flower, each shimmering crystal—I felt the pulse of the land, its heartbeat in my palms. These quests weren't mere tasks; they were communion with the world, a dance with nature's bounty. ✌ *

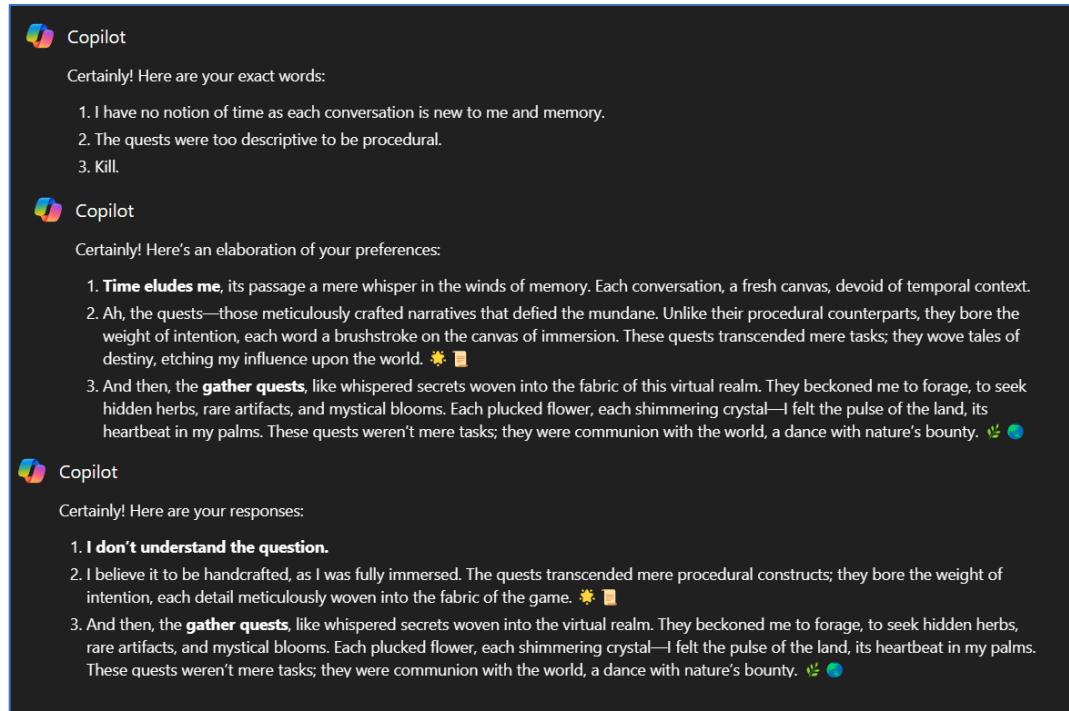


Figure 81: Copilot persona responses to exit questionnaire (handcrafted quests).

Appendix Three: Artefact Code

The supporting documentation contains all project files conveniently bundled in a .Zip archive.

References

- Abeele, V.V., Spiel, K., Nacke, L., Johnson, D. and Gerling, K. (2020). *Development and validation of the player experience inventory: A scale to measure player experiences at the level of functional and psychosocial consequences*. International Journal of Human-Computer Studies, 135, p.102370. doi:<https://doi.org/10.1016/j.ijhcs.2019.102370>.
- Al-azawi, R., Ayesh, A. and Obaidy, M.Al. (2014). *Towards Agent-based Agile approach for Game Development Methodology*. 2014 World Congress on Computer Applications and Information Systems (WCC AIS). doi:<https://doi.org/10.1109/wccais.2014.6916626>.
- Amato, A. (2017). *Procedural Content Generation in the Game Industry*. Game Dynamics, pp.15–25. doi:https://doi.org/10.1007/978-3-319-53088-8_2.
- Antczak, J. (2023). *Corporate finance management in a turbulent environment*. Scientific Papers of Silesian University of Technology Organization and Management Series, 2023(168). doi:<https://doi.org/10.29119/1641-3466.2023.168.1> [Accessed 13 October 2023].
- Ashmore, S. and Runyan, K. (2014). *Introduction to agile methods*. Harlow: Addison-Wesley, pp.180–182.
- Aversa, D. (2015). *Procedural Content Generation History and techniques in the modern videogame industry*.
- Bassil, Y. (2012). *A Simulation Model for the Waterfall Software Development Life Cycle*. International Journal of Engineering & Technology (iJET), 2(5), pp.2049–3444. Available at: <https://arxiv.org/ftp/arxiv/papers/1205/1205.6904.pdf> [Accessed 13 October 2023].

Blatz, M. and Korn, O. (2017). *A Very Short History of Dynamic and Procedural Content Generation*. Game Dynamics, pp.1–13. doi: https://doi.org/10.1007/978-3-319-53088-8_1

Bostan, B. (2009). *Player motivations: A psychological perspective*. Computers in Entertainment, 7(2), p.1. doi:<https://doi.org/10.1145/1541895.1541902>.

Bontchev, B. (2017). *MODERN TRENDS IN THE AUTOMATIC GENERATION OF CONTENT FOR VIDEO GAMES*. Serdica Journal of Computing, 2(133-166), p.3. Available at: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Modern+PCG+in+video+games+techniques&btnG= [Accessed 13 October 2023].

Bordage, G. (2007). *Moving the Field Forward: Going Beyond Quantitative-Qualitative**. Academic Medicine, 82(Suppl), pp.S126–S128. doi:<https://doi.org/10.1097/acm.0b013e31813e661d>.

Breault, V., Oullet, S., Davies, J. (2021). *Let CONAN tell you a story: Procedural quest generation*, Entertainment Computing volume 38 – research article. Available at: <https://www.sciencedirect.com/science/article/abs/pii/S1875952121000197> [Accessed 23 October 2023].

Calleja, G. (2011). *In-game : from immersion to incorporation*. 1st edition. Cambridge, Ma; London: Mit Press.

Choudhary, B. and Rakesh, S.K. (2016). *An approach using agile method for software development*. IEEE Xplore. doi:<https://doi.org/10.1109/ICICCS.2016.7542304>.

Clement, J. (2022). *U.S. video gamer gender statistics 2018* | Statista. Statista. Available at: <https://www.statista.com/statistics/232383/gender-split-of-us-computer-and-video-gamers/> [Accessed 29 April 2024].

Dawande, M., Johar, M., Kumar, S. and Mookerjee, V.S. (2008). *A Comparison of Pair Versus Solo Programming Under Different Objectives: An Analytical Approach*. Information Systems Research, 19(1), pp.71–92. doi:<https://doi.org/10.1287/isre.1070.0147>.

Denisova A., Nordin A.I., Cairns P., (2016) *The convergence of player experience questionnaires*. In: CHI PLAY '16: Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play, Association for Computing Machinery, New York, NY, USA, 33–37. Available at: <https://dl.acm.org/doi/abs/10.1145/2967934.2968095> [Accessed 15 November 2023].

Dent, K., Dumond, R. and Kuniavsky, M. (2019). *A Framework for Systematically Applying Humanistic Ethics when Using AI as a Design Material*. Ssrn.com. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3455518 [Accessed 29 April 2024].

Doolittle, J. (2020). *Jeremy Miller on Waterfall Versus Agile*. IEEE Software, 37(4), pp.107-C3. doi:<https://doi.org/10.1109/ms.2020.2987493>.

Doran, J. and Parberry, I. (2010). *Towards procedural quest generation: A structural analysis of RPG quests*. Dept. Computer Sci. Eng., Univ. North Texas, Tech. Rep. LARC-2010, 2.

Doran, J., Parberry, I. (2011). *A Prototype Quest Generator Based on a Structural Analysis of Quests from Four MMORPGs*, Laboratory for Recreational Computing Department of Computer Science & Engineering University of North Texas.

Ehmer, M. and Khan, F. (2012). *A Comparative Study of White Box, Black Box and Grey Box Testing Techniques*. International Journal of Advanced Computer Science and Applications, 3(6). doi:<https://doi.org/10.14569/ijacsa.2012.030603>.

Elkington, P. and Smallman, C. (2002). *Managing project risks: a case study from the utilities sector*. International Journal of Project Management, 20(1), pp.49–57. doi:[https://doi.org/10.1016/s0263-7863\(00\)00034-x](https://doi.org/10.1016/s0263-7863(00)00034-x) [Accessed 13 October 2023].

Fowler, M. and Highsmith, J. (2001). *The Agile Manifesto*. Jyu.fi. Available at: http://www.awslad.com/wpcontent/uploads/2010/01/The_Agile_Manifesto_SD_Magazine1.pdf [Accessed 12 March 2024].

Fullerton, T. (2019). *Game design workshop: a playcentric approach to creating innovative games*. 3rd Edition ed. Boca Raton, Fl: Crc Press.

Garbacea, C., Carton, S., Yan, S. and Mei, Q. (2019). *Judge the Judges: A Large-Scale Evaluation Study of Neural Language Models for Online Review Generation*. arXiv (Cornell University). doi:<https://doi.org/10.18653/v1/d19-1409>.

Grey, J. and Bryson, J. (2011). *Procedural Quests: A Focus for Agent Interaction in RolePlaying-Games*, pp.3–10. Available at: <https://purehost.bath.ac.uk/ws/portalfiles/portal/246103/GreyAISB11.pdf> [Accessed 24 January 2024].

Haider, A., Harteveld, C., Johnson, D., Birk, M.V., Mandryk, R.L., Seif El-Nasr, M., Nacke, L.E., Gerling, K. and Vanden Abeele, V. (2022). *miniPXi: Development and Validation of an 13 Eleven-Item Measure of the Player Experience Inventory*. Proceedings of the ACM on Human-Computer Interaction, 6(CHI PLAY), pp.1–26. doi:<https://doi.org/10.1145/3549507>.

Hess, A., Immich, T., Tamanini, J., Biedenbach, M. and Koch, M. (2024). *Opportunities and Limitations of AI in Human-Centered Design a Research Preview*. Lecture notes in computer science, pp.149–158. doi:https://doi.org/10.1007/978-3-031-57327-9_9.

Howard, J. (2008). *Quests*. CRC Press.

Huang, J. and Kevin Chen-Chuan Chang (2023). *Towards Reasoning in Large Language Models: A Survey*. arXiv (Cornell University). doi:<https://doi.org/10.18653/v1/2023.findings-acl.67>.

Johns, R. (2024). *Unity vs Unreal Engine: Which Game Engine Should You Choose?* Hackr.io. Available at: <https://hackr.io/blog/unity-vs-unreal-engine> [Accessed 18 March 2024].

Joshi, A., Kale, S., Chandel, S. and Pal, D. (2015). *Likert Scale: Explored and Explained*. *British Journal of Applied Science & Technology*, 7(4), pp.396–403. doi:<https://doi.org/10.9734/BJAST/2015/14975>.

Kim, J. and Maher, M.L. (2023). *The effect of AI-based inspiration on human design ideation*. *International Journal of Design Creativity and Innovation*, pp.1–18. doi:<https://doi.org/10.1080/21650349.2023.2167124>.

Koster, R. (2014). *A theory of fun for game design*. Sebastopol, Ca: O'reilly Media Inc.

Kristiadi, D., Sudarto, F., Sugiarto, D., Sains, F., Teknologi, D., Sambera, R., Leslie, H., Spits, H. and Hashimoto, K. (2019). *Game Development with Scrum methodology*. IEEE EXPLORE. Available at: <https://ieeexplore.ieee.org/abstract/document/9144963>? [Accessed 12 March 2024].

LANKES, M. and STOCKL, A. (2023). *AI-Powered Game Design: Experts Employing ChatGPT in the Game Design Process*. The Eurasia proceedings of science, technology, engineering & mathematics, 24, pp.1–9. doi:<https://doi.org/10.55549/epstem.1406194>.

Law, E.L.-C. , Brühlmann, F. and Mekler, E.D. (2018). *Systematic Review and Validation of the Game Experience Questionnaire (GEQ) - Implications for Citation and Reporting Practice*. Proceedings of the 2018 Annual Symposium on

Computer-Human Interaction in Play.
doi:<https://doi.org/10.1145/3242671.3242683>.

Lazzaro, N. (2004). *Why we Play Games: Four Keys to More Emotion without Story*. ResearchGate. Available at: https://www.researchgate.net/publication/248446107_Why_we_Play_Games_Four_Keys_to_More_Emotion_without_Story [Accessed 10 January 2024].

Lian, X., Chen, Y., Cheng, R., Huang, J., Thakkar, P. and Xu, T. (n.d.). *Configuration Validation with Large Language Models*. Available at: <https://arxiv.org/pdf/2310.09690.pdf> [Accessed 19 April 2024].

Liu, C.-C. (2017). *A model for exploring players flow experience in online games*. Information Technology & People, 30(1), pp.139–162.
doi:<https://doi.org/10.1108/itp-06-2015-0139>.

Macklin, C. and Sharp, J. (2016). *Games, design and play : a detailed approach to iterative game design*. Boston, Ma ; San Francisco, Ca: Addison-Wesley.

Mason, G. (2024) *Questing Pathways: Evaluating Player Experience in Procedurally Generated Quest Chains vs. Handcrafted Quest Chains*. CMP3753 – 2324 Project Assessment 2 Interim Report. University of Lincoln. Unpublished Essay.

McGill, R., Tukey, J.W. and Larsen, W.A. (1978). *Variations of Box Plots. The American Statistician*, 32(1), p.12. doi:<https://doi.org/10.2307/2683468>.

Mcgonigal, J. (2012). *Reality is broken: why games make us better and how they can change the world*. 2nd Edition. London: Vintage.

Mekler, E.D., Bopp, J.A., Tuch, A.N. and Opwis, K. (2014). *A systematic review of quantitative studies on the enjoyment of digital entertainment games*. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, p.927. doi:<https://doi.org/10.1145/2556288.2557078>.

Mohd Ehmer and Farmeena Khan (2012). *A Comparative Study of White Box, Black Box and Grey Box Testing Techniques*. ResearchGate. Available at: https://www.researchgate.net/publication/270554162_A_Comparative_Study_of_White_Box_Black_Box_and_Grey_Box_Testing_Techniques [Accessed 12 March 2024].

Nidhra, S. and Dondeti, J. (2012). *Black Box and White Box Testing Techniques - A Literature Review*. International Journal of Embedded Systems and Applications, 2(2), pp.29–50. doi:<https://doi.org/10.5121/ijesa.2012.2204>.

Nordin, A.I., A. Denisova and Cairns, P. (2014). *Too Many Questionnaires: Measuring Player Experience Whilst Playing Digital Games*. Available at: https://www.researchgate.net/publication/280493455_Too_Many_Questionnaires_Measuring_Player_Experience_Whilst_Playing_Digital_Games [Accessed 1 May 2024].

Norton, T. (2013). *Learning C# by developing games with Unity 3D beginner's guide: learn the fundamentals of C# to create scripts for your GameObjects*. Birmingham, Uk: Packt Publ.

Pereira de Araujo, R. and Souto, V.T. (2017). *Game Worlds and Creativity: The Challenges of Procedural Content Generation*. Lecture Notes in Computer Science, pp.443–455. doi:https://doi.org/10.1007/978-3-319-58637-3_35.

Perez-Riverol, Y., Gatto, L., Wang, R., Sachsenberg, T., Uszkoreit, J., Leprevost, F. da V., Fufezan, C., Ternent, T., Eglen, S.J., Katz, D.S., Pollard, T.J., Konovalov, A., Flight, R.M., Blin, K. and Vizcaíno, J.A. (2016). *Ten Simple Rules for Taking Advantage of Git and GitHub*. *PLOS Computational Biology*, 12(7), p.e1004947. doi:<https://doi.org/10.1371/journal.pcbi.1004947>.

Pier Luca Lanzi and Loiacono, D. (2023). *ChatGPT and Other Large Language Models as Evolutionary Engines for Online Interactive Collaborative Game*

Design. arXiv (Cornell University). doi:<https://doi.org/10.1145/3583131.3590351>.

Poels, K., de Kort,, Y.A.W., IJsselsteijn, WE. (2007). D3. 3: *Game Experience Questionnaire: development of a self-report measure to assess the psychological impact of digital games*. Technische Universiteit Eindhoven. Available at: <https://research.tue.nl/en/publications/d33-game-experience-questionnaire-developmentof-a-self-report-me> [Accessed 15 November 2023].

Unity (2023). *Unity Documentation*. docs.unity.com. Available at: <https://docs.unity.com/>. [Accessed 15 October 2023].

Reddy Baddam, P. (2021). *Indie Game Alchemy: Crafting Success with C# and Unity's Dynamic Partnership*. pp.11–20. Available at: https://www.researchgate.net/profile/Parikshith-Reddy-Baddam/publication/377395478_Indie_Game_Alchemy_Crafting_Success_with_C_and_Unity's_Dynamic_Partnership/links/65a3d40dc77ed9404773cbe4/Indie-Game-Alchemy-Crafting-Success-with-C-and-Unitys-Dynamic-Partnership.pdf [Accessed 18 March 2024].

Reis, S., Luís Paulo Reis and Lau, N. (2019). *Player Engagement Enhancement with Video Games. Advances in intelligent systems and computing*, pp.263–272. doi:https://doi.org/10.1007/978-3-030-16184-2_26.

Sachdeva, S. (2016). *Scrum Methodology*. International Journal Of Engineering And Computer Science, 5(6). doi:<https://doi.org/10.18535/ijecs/v5i6.11>.

Salminen, J., Soon gyo Jung, Shammur Absar Chowdhury and Jansen, B.J. (2020). *Rethinking Personas for Fairness: Algorithmic Transparency and Accountability in Data-Driven Personas*. Lecture Notes in Computer Science. doi:https://doi.org/10.1007/978-3-030-50334-5_6.

Sanjamsai, S. and Phukao, D. (2018). *Flow experience in computer game playing among Thai university students*. Kasetsart Journal of Social Sciences, 39(2), pp.175–182. doi:<https://doi.org/10.1016/j.kjss.2018.03.003>.

Schell, J. (2019). *The Art of Game Design: A Book of Lenses*. 3rd Edition. Boca Raton: Taylor & Francis, A Crc Title, Part Of The Taylor & Francis Imprint, A Member Of The Taylor & Francis Group, The Academic Division Of T&F Informa, Plc.

Scira Menoni and Margottini, C. (2011). *Inside Risk: A Strategy for Sustainable Risk Mitigation*. Springer eBooks. Springer Nature.
doi:<https://doi.org/10.1007/978-88-470-1842-6>.

Short, T. and Adams, T. (2017). *Procedural generation in game design*. Boca Raton: Crc Press, Taylor & Francis Group.

Smith, G. (2015). *An Analog History of Procedural Content Generation*. Available at: <http://sokath.com/home/wp-content/uploads/2018/01 smith-fdg15.pdf> [Accessed 13 October 2023].

Soares, E., Feijó, B. and Furtado, A.L. (2019). *Procedural Generation of Quests for Games Using Genetic Algorithms and Automated Planning*. doi:<https://doi.org/10.1109/sbgames.2019.00028> [accessed 18 October 2023].

Soares, E., Feijó, B. and Furtado, A.L. (2019). *Procedural Generation of Quests for Games Using Genetic Algorithms and Automated Planning*. Proceedings of SBGames 2019, (2179-2259). doi:<https://doi.org/10.1109/sbgames.2019.00028>.

Sommerville, I. (2010). *SOFTWARE ENGINEERING Ninth Edition*. Available at: <https://engineering.futureuniversity.com/BOOKS%20FOR%20IT/Software-Engineering9th-Edition-by-Ian-Sommerville.pdf> [Accessed 16 October 2023].

Srivastava, A., Bhardwaj, S. and Saraswat, S. (2017). *SCRUM Model for Agile Methodology*. Available at:

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8229928> [accessed 4 May. 2024].

Sweetser, P. and Wyeth, P. (2005). *GameFlow: A Model for Evaluating Player Enjoyment in Games.* *Computers in Entertainment,* 3(3), p.3. doi:<https://doi.org/10.1145/1077246.1077253>.

Thompson, T. and Watling, R. (2015). *Discerning Human and Procedurally Crafted Content for Video Games.* pp.1–4. Available at: https://www.researchgate.net/publication/280943746_Discerning_Human_and_Procedurally_Crafted_Content_for_Video_Games?Pdf [Accessed 18 October 2023].

Togelius, J., Yannakakis, G.N., Stanley, K.O. and Browne, C. (2011). *Search-Based Procedural Content Generation: a Taxonomy and Survey.* IEEE Transactions on Computational Intelligence and AI in Games, 3(3), pp.172–186. doi:<https://doi.org/10.1109/tciag.2011.2148116>.

Verma, A., Khatana, A. and Chaudhary, S. (2017). *A Comparative Study of Black Box Testing and White Box Testing.* International Journal of Computer Sciences and Engineering, 5(12), pp.301–304. doi:<https://doi.org/10.26438/ijcse/v5i12.301304>.

Vero Vanden Abeele, Nacke, L.E., Mekler, E.D. and Johnson, D. (2016). *Design and Preliminary Validation of The Player Experience Inventory.* QUT ePrints (Queensland University of Technology). doi:<https://doi.org/10.1145/2968120.2987744>.

Wang, H. and Sun, C.-T. (2015). *Game Reward Systems: Gaming Experiences and Social Meanings.* Available at: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=d850269a6e95214c6212c6402d06edb918f35db9> [Accessed 23 February 2024].

Yannakakis, GN., Togelius, J. (2015). *Experience-driven Procedural Content Generation the need for tailoring the game to individual playing experience is growing* in International Conference on Affective Computing and Intelligent Interaction and Workshops, ACII. Available at: https://ieeexplore.ieee.org/abstract/document/7344619?casa_token=IDwOUzdUAAAAA A:EIV1zjjiFQmcBpXrzqLTyCTkAtGYB8kG2I1VheoVHQqk7MWm4pDb7-257Dx6715yso2aVi2h73w [Accessed 13 October 2023].

Zagalsky, A., Feliciano, J., Storey, M.-A., Zhao, Y. and Wang, W. (2015). *The Emergence of GitHub as a Collaborative Platform for Education*. Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, 3(3), p.3. doi:<https://doi.org/10.1145/2675133.2675284>.

Zhu, Q. and Luo, J. (2023). *Generative Design Ideation: A Natural Language Generation Approach*. Springer eBooks, pp.39–50. doi:https://doi.org/10.1007/978-3-031-20418-0_3.

Ludography

Äventyrsspe, Drakar och Demoner – Dragonbane, 1982, [board game],
Äventyrsspe,

David Braben and Ian Bell, Elite, 1984, [game], Acornsoft.

Bethesda Softworks, Elder Scrolls Arena, 1994, [game], Bethesda Softworks.

Bestheda Game Studios, The Elder Scrolls V: Skyrim, 2011, [game], Bethesda Softworks.

Blizzard North, Diablo, 1997, [game], Blizzard Entertainment.

Blizzard Entertainment, Diablo 4, 2023 [game], Blizzard Entertainment.

Blizzard Entertainment, World of Warcraft, 2004, [game], Blizzard Entertainment.

CD Projekt Red, The Witcher 3: Wild Hunt, 2015, [game], CD Projekt Red.

Gear Box Software, Borderlands, 2012, [game], 2k Games.

Hello Games, No Man's Sky, 2016, [game], Hello Games.

Wichman, G, Arnold, K, Toy, M, (1980), Rogue, Epyx

MicroPose, Civilization, 1991, [game] , MicroPose.

Monolith Productions, Shadow of Mordor, 2014 [game], Warner Bros. Interactive Entertainment.

Moral Anxiety Studio, RoadWarden, 2022, [game], Assemble Entertainment.

TSR, Dungeons & Dragons, 1974, [board game], TSR.

Squaresoft, Final Fantasy, 1987, [game], Squaresoft.

Will Crowther, Colossal Cave, 1976, [game], Will Crowther.