

Project UMBRAE

Group Project Retrospective

By James (Jamie) Marland - 23100010
PR6001 - May 2026 - Word count: 4644



Figure 1. Project UMBRAE Beauty Shot 1

1. Introduction

1.1. Overview

This project retrospective covers my experience on the PR6001 Module, Professional Studio Project. The aim of this module is to simulate a real industry environment by working as a multidisciplinary team to deliver a vertical slice of a game in Unreal Engine 5 (Epic Games, 2026). This was our largest group project as Escape Studios and was made up of 15 students across Technical Art, Game Art, Game Design, Animation, and Character Creation. The team chose to make Project UMBRAE, a first-person psychological horror game set in a 2009 American suburb, where the player character, Alexis, is house-sitting for their late aunt. Solving Puzzles and exploring the house to discover what happened and progress to the end.

The game draws heavily on existing games in the genre, the most important of which are the Resident Evil franchise, specifically RE7 (CAPCOM, 2017), Visage (SadSquare Studio, 2020), and Silent Hill's P.T., a cultural icon (The A.V. Club, 2018).

The project ran for 18 scheduled weeks across six months (Figure 2). I split this into the following sections:

- Four weeks of pre-production to flesh out ideas, create proof of concepts, a blockout, an art bible, a GDD and an MVP.
- 11 weeks of production to work on milestones, iterate on feedback and create a complete game.
- Three weeks of free time to have as a buffer or to polish the game, focusing on fixing bugs, updating visuals, and preparing other deliverables for submission.

Karl Wooley, Global Head of Immersive at Framestore, served as our industry mentor, and I also organised a secondary external industry showcase focused on profiling.

Project Timeline					
★ = Milestone					
WK	DATE RANGE	TYPE	STAGE	PRODUCT VERSION	IMPORTANT DATES
01	24 Nov – 1 Dec	Uni	Preproduction	Planning	
02	2 Dec – 8 Dec	Uni	Preproduction	MVP	
03	9 Dec – 15 Dec	Uni	Preproduction	MVP	Mentors: Fri 12 Dec
04	16 Dec – 22 Dec	Uni	Preproduction	MVP ★	
05	23 Dec – 29 Dec	Break	Production	Alpha	
06	30 Dec – 5 Jan	Break	Production	Alpha	
07	6 Jan – 12 Jan	Reading Week	Production	Alpha	
08	13 Jan – 19 Jan	Uni	Production	Alpha	
09	20 Jan – 26 Jan	Uni	Production	Alpha	
10	27 Jan – 2 Feb	Uni	Production	Alpha ★	
11	3 Feb – 9 Feb	Uni	Production	Beta	
12	10 Feb – 16 Feb	Uni	Production	Beta	
13	17 Feb – 23 Feb	Uni	Production	Beta	
14	24 Feb – 2 Mar	Reading Week	Production	Beta ★	
15	3 Mar – 9 Mar	Uni	Production	Gamma	Mentors: Fri 6 Mar
16	10 Mar – 16 Mar	Uni	Production	Gamma	
17	17 Mar – 23 Mar	Uni	Production	Gamma	
18	24 Mar – 30 Mar	Uni	Production	Gamma	
19	31 Mar – 7 Apr	Uni	Production	Gamma ★	
20	7 Apr – 13 Apr	Break	Postproduction	Gold	
21	14 Apr – 20 Apr	Break	Postproduction	Gold	
22	21 Apr – 27 Apr	Uni	Postproduction	Gold	
23	28 Apr – 4 May	Uni	Postproduction	Gold	
24	5 May – 11 May	Uni	Postproduction	Gold ★	Project Sub: Thu 7 May
25	12 May – 18 May	Uni	Postproduction	Retrospective	
26	19 May – 22 May	Uni	Postproduction	Retrospective ★	Retro Sub: Fri 22 May

Figure 2. Timeline and milestones

The brief was deliberately structured to simulate a studio simulation rather than a traditional coursework assignment. The point was not only to create a game together but also to prove we could practise the patterns of a real studio, with source control, Agile development, external reviews, playtesting, quality assurance (QA), secondary deliverables, and project management, naturally growing an understanding of why these practices exist in the industry to begin with. The final game is playable and published on itch.io (Figure 3).

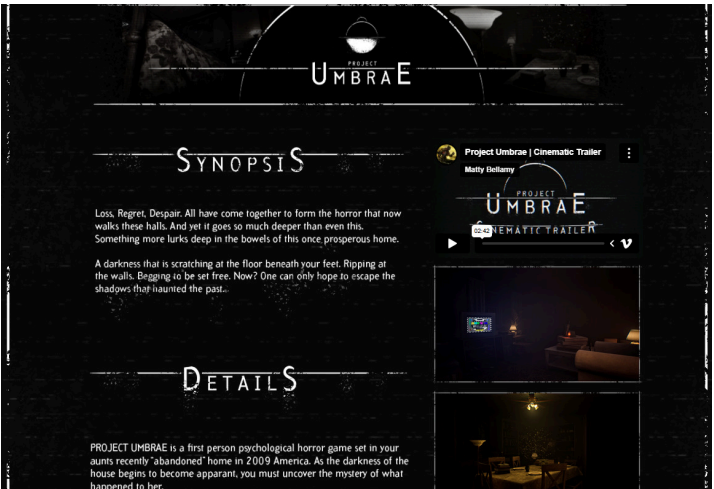


Figure 3. Published on Itch.io (Team Penumbra, 2026a)

1.2 Personal Goals

I came into this module with two specific goals. Firstly, I wanted to develop my tool and pipeline skills in technical art, as it is an area I want to highlight and specialise in, and it was also specifically noted in my recruiter's feedback as the reason I did not get the internship last summer at Rebellion. To do this, I wanted to build a real-world use-case tool that would benefit my team and showcase my C++ knowledge by creating a custom Unreal plugin.

Secondly, I wanted to develop additional soft skills, such as communication, organisation, and project management, which are also important for technical art (Certified Digital Artist, 2022).

1.3 Deliverables

Alongside the game project and build, the module has other secondary deliverables:

- **Game Design Document (GDD)** - A documentation of the game's mechanical specifications (Ashby, 2024).
- **Art Bible** - The visual style and direction across the game's props, environment, gameplay, lighting, VFX, and characters (Figure 4) (Muñoz, 2025).



Figure 4. Project Art Bible

- **Gameplay footage** - A recorded playthrough of the game.
- **Cinematic trailer** - A pre-rendered trailer showcasing the game in a couple of minutes (Figure 5).

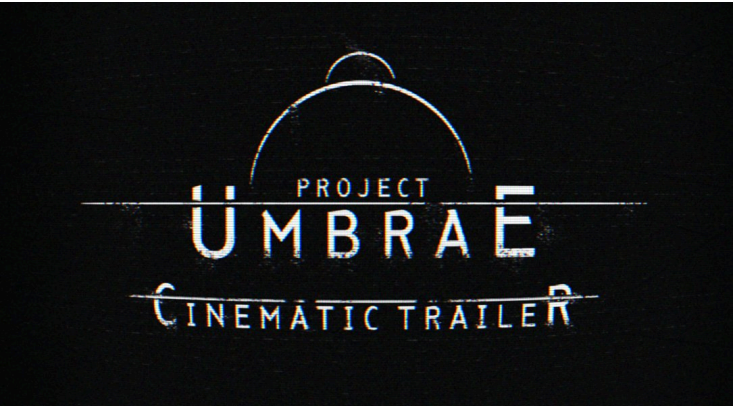


Figure 5. Cinematic Trailer (Team Penumbra, 2026b)

- **Beauty shots** - High-quality images showcasing the environment.
- **Individual Production Log** - A week-by-week breakdown of what you individually did (Figure 6).

Professional Practice - Research Document
James Marland - 23100010

Contents	
1. Introduction	Page 1
2. Role Research/Requirements	Page 2
3. Company Research	Page 4
4. Short-Term Plan	Page 9
5. Long-Term Plan	Page 10
6. Networking	Page 11
7. Portfolio Research	Page 14
8. Online Presence	Page 16
9. Cover Letter	Page 18
10. CV	Page 19
11. Self-Evaluation	Page 20
12. Bibliography	Page 21

1. Introduction
1.1. Personal Summary
My name is James Marland, and I am in my final year of the BSc (Hons) Technical Art for Games & VFX course at Escape Studios (Escape Studios, 2023). Before this, I completed UAL Extended Diploma in Creative Media and Technology (Games Design) at Chelmsford College, and received a Distinction. I learned the fundamentals of game development, and, combined with my existing programming knowledge, I had a well-rounded skill set that prepared me well for Technical Art. Something I wanted to pursue even before Escape Studios announced the course.
My studies at Escape Studios continued this passion. Throughout the first year, I built the knowledge needed to communicate effectively across roles, an important aspect of being a technical artist. In the second year, I continued as a technical art generalist, learning shaders, tools, and POG. I also led as producer, a 12-person group project, with game design and art students. Borderline Racing (Figure 1) would go on to receive Finalist at The Rookies in 2025 in the Game of the Year - Console & PC category.



Figure 6. My Individual Production Log

- **Individual Contribution List** - A list of every asset you contributed to across development and what contributions you made (Figure 7).

INDIVIDUAL CONTRIBUTION LIST
James Marland - 23100010 - Team Penumbra

Asset Name	Screenshots	Summary
Builds		I packaged every project build and released them. Any errors during packaging I fixed.
Art Bible		Set up the skeleton layout, helped refine some slides, and presented it. Gave Matty advice on appearance.
Itch.io		Set up the initial page and upload builds. Gave Matty advice on appearance.
Sound List		Created the whole document of all sounds required with categories, looping, status, 3D, length, modulate, owner and handed it to Jamie Steveson, our external sound designer.

Figure 7. My Individual Contribution List

- **Additional deliverables** - Any supporting documentation that helped aid in development, concept art, moodboards, playtesting feedback, asset lists and more (Figure 8).

Possible Documentation for Interactive Projects	
→ Game Design Document ◆ Design Pillars ◆ Core Loop ◆ Mechanics	→ Technical Documentation → Concept Art → Task Lists → Draft Plan/Schedule → Level/Environment Block Outs → Proof-of-Concept Mechanics → etc...
→ Art Bible	
→ Moodboards	
→ MVP & Milestones	
→ Minimum Viable Product	

Figure 8. Possible Documentation provided by Escape

1.4 Technical Terms

Technical terms will repeatedly be used throughout this document and are defined up front below:

- **Vertical Slice** - A short but polished section of a game that demonstrates the final visual and gameplay quality a studio would aim for a game to achieve (Gianty, 2024).
- **Technical Artist** - A person who works between art and programming, communicating and organising development between them. As well as typically being involved in pipeline, tools, shaders, VFX, technical animation (TA), optimisation, R&D and educating other roles (Tech Art Aid, 2022).
- **Agile** - A project management philosophy that focuses on adapting to changes (Slack, 2024).
- **SCRUM** - A framework for putting Agile development into practice. Development is organised into weekly sprints, with regular evaluation of progress and task priorities (Schwaber and Sutherland, 2020).
- **MVP** - The simplest version of a product to test an idea as early as possible (Raj, 2024).
- **QA** - testing, bug-fixing and polishing a product to fit standards (Into Games, 2019).
- **Playtesting** - Getting players to test out a game to identify problems or receive feedback (Kwalee, 2024).
- **Source Control** - Tracking and managing changes to a file system or code, allowing for reversing or backups as well as letting multiple developers work at the same time (Gehman, 2019).
- **Build** - A compiled version of a game at a moment in development (Catness Game Studios, 2023).
- **Pipeline** - A structured game development process to manage the flow of work (Kraj, 2025).
- **MetaHuman, Lumen, Megalights, Nanite, MetaSounds, Level Streaming** - A list of proprietary engine features and systems developed for the Unreal Engine 5 framework (Figure 9).

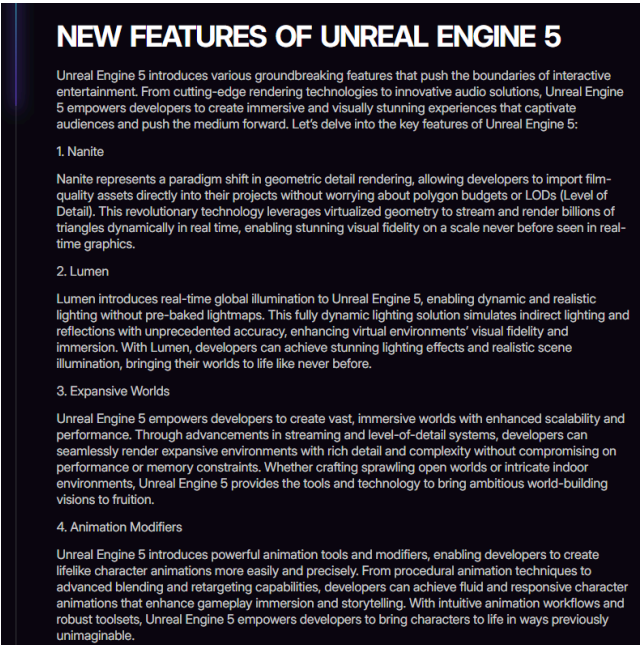


Figure 9. New UE 5 features (Porokh, 2023)

2. Engagement & Approach

2.1. Role

Given my past experience with the role, the team agreed to appoint me as the Producer, which put me in a good position to achieve my personal goals for this project. So my role on Project UMBRAE was split between Producer and Technical Artist, the same combination I held in my second-year group project, PR5001. Across both roles, I consulted with and collaborated with relevant teammates, keeping them informed of changes and how to use systems. Collaborating on so many aspects of the project meant that I had to manage both team and individual priorities carefully.

As a Producer, I was in charge of managing and organising the project and coordinating all my teammates' ideas. I worked on the following:

- **Itch.io** - Setup, outlined, guidance and distributed builds.
- **Art Bible** - Setup, outlined and guidance.
- **YouTrack** - Organised and structured the documentation, some pre-production like the timeline, milestones, stretch goals, pre-prod level layout, references, MVP, gameplay
- **Sound List** - Created and organised formatting all for the external sound designer.
- **Asset List** - Formatted the final version by room/role and cleaned up duplicate assets.
- **Presentations** - For tutor/industry showcases, I often presented on behalf of the team.
- **Project Schedule** - Organised the project's scale, timeline, milestones and stretch goals.
- **SCRUM Master** - I organised the Scrum process with weekly sprints and daily stand-ups, making sure tasks were assigned and prioritised based on current objectives (Figure 10).

Schedule

The team consistently organised throughout development into meetings organised by James Marland as the Scrum master. Though it took time to find the best day, we eventually landed on the following setup:

Monday	Start of sprint at 2 pm	What was done the previous week, plans for this week.
Tuesday	In University	Group work together, easy communication.
Wednesday	SCRUM Daily at 2 pm	What people are up to and anything that needs to be resolved.
Thursday	In University	Group work together, easy communication.
Friday	Online tutor session SCRUM Daily at 2 pm	What people are up to and anything that needs to be resolved.

While we tried YouTrack at first, we found it had too much friction, and people did not check or update their cards, so we resorted to more frequent, in-depth discussions during meeting time. Future plans and current project scale were discussed at the start of week sprints and evaluated as a team, with milestones set as goals.

These online calls were recorded via Teams and can be viewed at the bottom.

Disclaimer: This information is based only on the online meetings recorded as part of our regular schedule. It does not account for any off-record or in-class sessions, so it is not an accurate reflection of involvement, but it does show evidence of our team's communication.

Online Meetings statistics pulled from Team's attendance reports:
Total online sessions: 47
Average online meeting time: 1h 5m 11s

As a Technical Artist, my responsibilities covered a wide variety of development tasks, focusing on enabling other roles to operate smoothly together, a key aspect of Technical Art. I worked on the following:

- **Game builds** - I compiled the weekly builds, making sure they were not broken and fixing any bugs.
- **Internal tutorial videos** - Created helpful video guides for common processes for artists and animators.
- **Source control** - Organised the project Perforce structure and managed any conflict or checkout problems, rolling back if necessary for the whole team.
- **Asset Guidelines** - Created constraints and a formatting guide for artists to follow based on asset types (Figure 11).

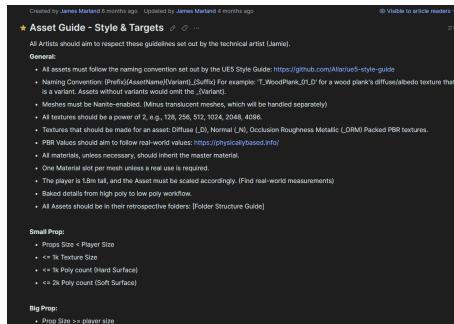


Figure 11. YouTrack Asset Guidelines

- **R&D** - Tested and implemented any advanced technical features like the portal concept, VATs for VFX and Houdini Engine for HDAs (SideFX, 2013).
- **Bug fixing/Polish** - Frequently, before builds and showcases, playtested, fixing and polishing to ensure the game worked.
- **Project setup/organisation** - Set up the initial project, settings, files, outliner, and sub-levels (Figure 12).

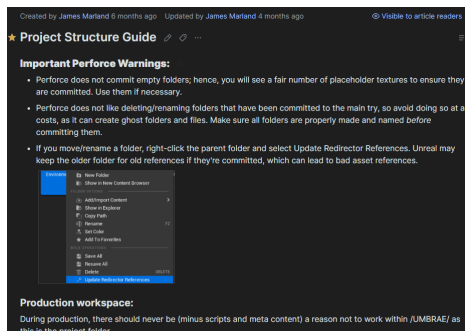


Figure 12. YouTrack Structure Guide

- **TV system** - Interactive TV with dynamic lighting, audio, video, shader effect, as well as showing the main menu UI with the same shader and transitioning from the menu to the game start.
- **Event system** - Created a parent and four event types that handled staging, conditions, execution, level sequence events and manager. I also explained how to use the user-friendly side for animators to use.
- **Cutscene system** - Handled cutscene execution and finishing. In addition to player and object states, I handled the technical aspects of cutscenes.
- **Light system** - Built a unified light system so that all interactable lights could be easily used and adjusted by the lighting artist.

- **Generic openable system** - Created a generic openable letting any openable assets, like a cupboard, support any number of drawers and doors with open/closed states.
- **Interact icon system** - Created the icon system that would show what was interactable to the player, cleanly and updated with interaction state.
- **Audio** - Set up most attenuation settings and implemented a lot of audio. Implemented MetaSounds for pitch and volume randomisation.
- **Animation** - Set up the metahuman system for the player and creature; assisted animators in getting the player movement animations working; and set up post-process animation, Foot-IK, player movement, and camera sway.
- **Development Progress Log** - Set up an automatic level-sequence render for progress cameras using a Python script that runs on project startup, tracking and burning in ID, Date, and Version (Figure 13).

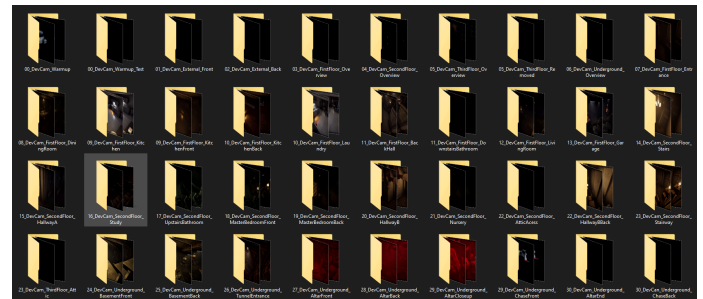


Figure 13. Development screenshots

- **Profiling/Optimisation** - Created a deterministic free-cam profiling mode and organised an industry profiling showcase with Jan Mroz. Made custom occlusion with LSVs and consistently ran performance passes on the game, optimising as needed.
- **Escape Asset Helper** - I developed a custom C++ plugin that was feature-complete and configurable, allowing for custom naming conventions for prefixes, bases, suffixes and variants. Defaulted to Allar's Style guide (Allar, 2023). Asset validation has custom rules with queries and actions from either presets or blueprints, and it includes an import dialogue with auto-fix and a content browser status overlay. It also worked with source control systems. Then it was released on GitHub with documentation and supported with updates (Figure 14). The stylised beat-em-up project, Ecstasy, also used it.

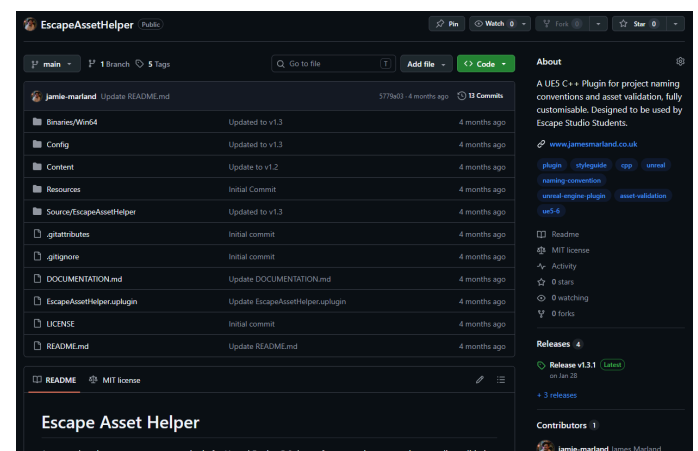


Figure 14. Escape Asset Helper Github (Marland, 2026)

2.2. Workflow

I approached this project with a strong focus on adhering to industry practice. This meant creating high-quality pre-production deliverables to organise and plan the project, culminating in the MVP (Figure 15), and then iterating continuously with target dates for development milestones. These mimic the software development life cycle, with Alpha, Beta, Gamma, Gold, and Release Candidate stages (QATestLab, 2018). This ensured that the game remained playable throughout and that features were not left half-finished.



Figure 15. Snapshot of the MVP version

Our weekly SCRUM via Teams (Figure 16) arranged our week. Our Sprint on Monday focuses on summarising the previous week's tasks and project feedback, assessing team-wide and individual priorities for each team member, and planning the following week's tasks for everyone. Daily standups throughout the week addressed any problems people were stuck on and the status of tasks. Thursday marks the playtesting day to gather player feedback, and Friday is tutor/industry feedback, compiled for the week after. This setup allows us to regularly address concerns, which is recommended specifically for SCRUM-based game development (Godoy and Barbosa, 2010).

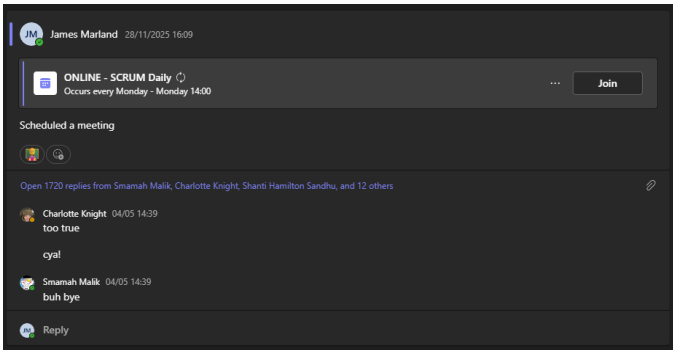


Figure 16. Teams Online SCRUM Daily

The rest of our development pipeline used YouTrack for project documentation in some cases; Google Sheets, Docs, and Slides were used and linked instead when a document required particularly asynchronous work or formatting that YouTrack did not support. Perforce was used for source control in Unreal Engine and for final submission. To address a problem I have seen in past projects, we enforced a 'One asset, one person' approach as much as possible to reduce people stepping on each other's work, and to treat asset lists as the ground truth for who was assigned what (Figure 17).

Another thing I carried over from previous modules, such as PR5001, was to include buffer time in the schedule (Mladenovic, 2024). This held up well whenever teammates could not contribute, or external factors limited us.

Asset Name	Asset ID	Type	Reference	Priority	Assignee	Status	Current Stage	Assigned To	Notes
Background	Background_01	Img		High	James Marland	Completed	In Progress	Adam Honey	Character Knight
Character	Character_01	Model		High	Charlotte Knight	Completed	In Progress	Adam Honey	Character Knight
Environment	Environment_01	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Environment
Gameplay	Gameplay_01	Script		High	Adam Honey	Completed	In Progress	Adam Honey	Gameplay
UI	UI_01	Image		High	Adam Honey	Completed	In Progress	Adam Honey	UI
Weapon	Weapon_01	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Weapon
Character	Character_02	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Character Knight
Environment	Environment_02	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Environment
Gameplay	Gameplay_02	Script		High	Adam Honey	Completed	In Progress	Adam Honey	Gameplay
UI	UI_02	Image		High	Adam Honey	Completed	In Progress	Adam Honey	UI
Weapon	Weapon_02	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Weapon
Character	Character_03	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Character Knight
Environment	Environment_03	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Environment
Gameplay	Gameplay_03	Script		High	Adam Honey	Completed	In Progress	Adam Honey	Gameplay
UI	UI_03	Image		High	Adam Honey	Completed	In Progress	Adam Honey	UI
Weapon	Weapon_03	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Weapon
Character	Character_04	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Character Knight
Environment	Environment_04	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Environment
Gameplay	Gameplay_04	Script		High	Adam Honey	Completed	In Progress	Adam Honey	Gameplay
UI	UI_04	Image		High	Adam Honey	Completed	In Progress	Adam Honey	UI
Weapon	Weapon_04	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Weapon
Character	Character_05	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Character Knight
Environment	Environment_05	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Environment
Gameplay	Gameplay_05	Script		High	Adam Honey	Completed	In Progress	Adam Honey	Gameplay
UI	UI_05	Image		High	Adam Honey	Completed	In Progress	Adam Honey	UI
Weapon	Weapon_05	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Weapon
Character	Character_06	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Character Knight
Environment	Environment_06	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Environment
Gameplay	Gameplay_06	Script		High	Adam Honey	Completed	In Progress	Adam Honey	Gameplay
UI	UI_06	Image		High	Adam Honey	Completed	In Progress	Adam Honey	UI
Weapon	Weapon_06	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Weapon
Character	Character_07	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Character Knight
Environment	Environment_07	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Environment
Gameplay	Gameplay_07	Script		High	Adam Honey	Completed	In Progress	Adam Honey	Gameplay
UI	UI_07	Image		High	Adam Honey	Completed	In Progress	Adam Honey	UI
Weapon	Weapon_07	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Weapon
Character	Character_08	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Character Knight
Environment	Environment_08	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Environment
Gameplay	Gameplay_08	Script		High	Adam Honey	Completed	In Progress	Adam Honey	Gameplay
UI	UI_08	Image		High	Adam Honey	Completed	In Progress	Adam Honey	UI
Weapon	Weapon_08	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Weapon
Character	Character_09	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Character Knight
Environment	Environment_09	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Environment
Gameplay	Gameplay_09	Script		High	Adam Honey	Completed	In Progress	Adam Honey	Gameplay
UI	UI_09	Image		High	Adam Honey	Completed	In Progress	Adam Honey	UI
Weapon	Weapon_09	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Weapon
Character	Character_10	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Character Knight
Environment	Environment_10	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Environment
Gameplay	Gameplay_10	Script		High	Adam Honey	Completed	In Progress	Adam Honey	Gameplay
UI	UI_10	Image		High	Adam Honey	Completed	In Progress	Adam Honey	UI
Weapon	Weapon_10	Model		High	Adam Honey	Completed	In Progress	Adam Honey	Weapon

Figure 17. Asset list

Setting up project structures, asset guidelines, and asset lists early helped us significantly when planning ahead and paid off when faced with later changes. I also set up the development progress log system to be automated in Python this time, and I embedded the room ID, date, and version in the screenshot (Figure 18). A problem I encountered in PR5001 was that it was difficult to recapture prior progress and achieve a high capture frequency because the process was manual. This time, it was addressed early, which meant we had a healthy dataset to use when referencing earlier stages of development or making time-lapses.



Figure 18. An example of a development screenshot.

Overall, the workflow we created helped us and allowed us to address feedback quickly and absorb time lost to pivots into higher-priority tasks without losing time on polishing or creating secondary deliverables. I did not anticipate that scaling from a team of 9 in PR5001 to 15 (Figure 19) in PR6001 would be nonlinear but rather exponential, as every task required an interconnected web of communication and bureaucracy to address, a well-known phenomenon at workplaces (Moken Digital, 2026).



Figure 19. All team members

2.3. Feedback

We gathered feedback frequently throughout development and addressed any issues that arose. Feedback came from four sources:

- **Playtesting** - Let us test for bugs and moment-to-moment gameplay issues with a wide range of players, each with different gaming experiences and knowledge of the project. This let us see whether there were any bugs or visual issues to address and whether the gameplay was intuitive. This was our most frequent and quantitative source of feedback.
- **Tutors' reviews** - These typically address the project-wide scope, provide feedback on the state of assets and our weaker areas for development, and include suggestions for improvement.
- **Industry showcases** - In weeks 3 and 13, we presented our game to Karl Wooley, who advised on the overall game, reviewed any secondary deliverables we had, and offered industry insights and suggestions for improvement (Figure 20). Both times, the project was very well received, with only a few areas pointed out as needing improvement, and instead suggestions for how its full potential could be realised by delving deeper into its puzzles and art direction.



Figure 20. Karl Wooley's industry showcase session.

- **Profiling Showcase** - Outside of Escape, I had been in regular contact with Jan Mroz, a graphics programmer from The Knights of U, who organised weekly calls for a community to discuss optimisation and profiling. As part of this, I proposed a community event to showcase our game as a case study. Over two weeks, other members analysed the performance using the deterministic profiling free-cam and suggested improvements and general game feedback. The feedback was incredibly useful, and I learned many techniques and features. This took place between weeks 10 and 12 (Figure 21).

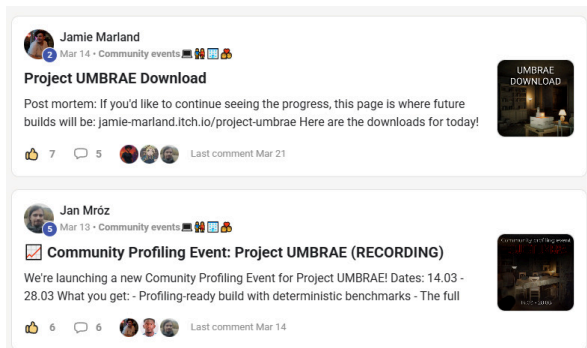


Figure 21. Project UMBRAE Community Event post

While we handled most feedback in the following week, some proved more challenging and repeatedly came up during development and needed to be iteratively addressed:

- **Keeping Tension** - As we transitioned from no scares to a placeholder, then to the creature, we faced an unusual problem. Players found the game scarier before anything appeared, and even after the creature replaced it, this feeling remained. To address this, we worked to make the creature's animation and appearance scarier, but what worked best was adding fog to significantly reduce what the player could see ahead or in their peripheral vision. Combined with sound, this got us back on track to creating a psychological horror game.
- **Game feel and juice** - Puzzles were frequently said to feel lacking in polish and were described as hard or unintuitive to understand early on. Once they were modelled and textured and some game-feel polish was added, they felt better, but they still required several iterations.
- **Game depth** - While not a problem to begin with due to early development expectations, the lack of depth in gameplay became more apparent later, as players' only interaction was with puzzles. This was solved by adding inspectable objects, notes, puzzle keys, and interactable doors and drawers to allow for more exploration.

3. Evaluate

3.1. Team Decisions

The project stayed aligned with industry standards by using Agile project management and Scrum. This worked well for the team, making clear what the coming week's focus and priorities were and evaluating areas we could scrap from the scope. Most issues were solved in these meetings rather than in extended chats, as we could have nuance and iterate quickly in calls. The 'One asset, one person' approach also helped prevent overlap or redundant work, with the master project tracker (Figure 22) and individual asset lists serving as easy ways to analyse scale and redistribute workload.

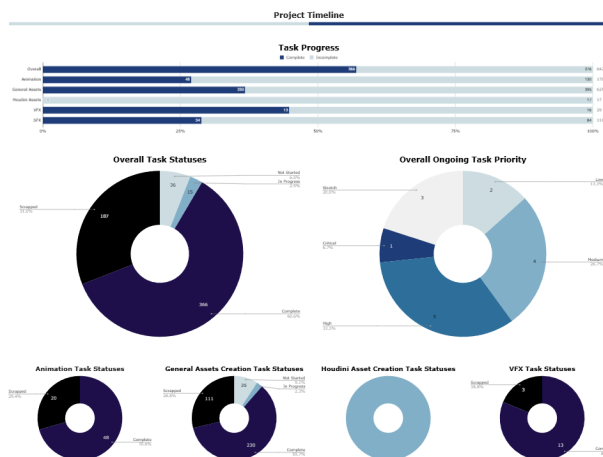


Figure 22. Master project tracker

Having buffer time again absorbed any unexpected delays and reduced team-wide stress. The weekly sprints provided structure and ended each one with a showcase, or, at a minimum, playtesting, giving us steady external pressure to keep ourselves organised and prepared with playable builds.

One of our biggest decisions as a group was the decision to include third-party assets. When it became apparent, thanks to our trackers, that we would not be able to achieve the desired level of visual clutter for art direction, we used external fab assets to pad out the set dressing. This decision allowed team members to focus on higher-priority gameplay and unique assets that could not be compromised without reducing scope.

Another big decision was that we collectively decided we would not be able to complete two original handmade characters by the deadline, and pivoted to using MetaHumans, scrapping the work so far, but freed our character creator to customise and iterate quickly and without much compromise to the overall scale. It had the benefit of doing heavy lifting of meeting the realistic appearance required by the art direction, and it was closely integrated into Unreal and, with some retargeting and setup, worked seamlessly with the existing placeholder. We consulted with another team, Outlast Trails, as well as one of the character artist tutors, to ensure we were prepared and informed before implementing this decision (Figure 23).

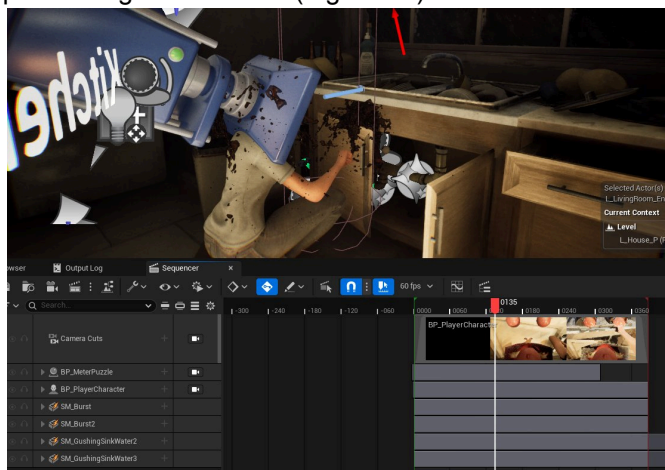


Figure 23. Metahuman player character in cutscene

These key decisions, along with a focus on gameplay and unique assets, were implemented in time for our second industry showcase and were successfully completed. Showing an understanding of cost, risk assessment and flexibility seen in industry.

3.2. Individual Decisions

My own decisions were dominated by one key factor: Time helping others with their work vs working on my own tasks. I continually assessed this from start to finish, primarily based on weekly priorities, and it did not cause any major problems for the team or me. Working on a system as early as possible resulted in a rewarding ease of adding content later. It would not have been possible without following industry standards such as Object-Oriented Design (OOD), coding standards, and file/folder/code organisation. For others, I continually helped them debug issues they had, made guides, and trained artists, animators, and game designers on the system or pipeline processes used in the project. This can be most clearly seen in the animators, as none of them had prior Unreal experience, but they ended up developing animations inside level sequences, importing their own animations, retargeting them, and using systems like Events.

Continuing industry-level work not typically seen at University was optimisation, I followed the practices laid out by Jan Mróz et al (Mróz, 2026) as well as online educational technical artist I have learned from to profile using Unreal Insights or NSights and read frame breakdowns (Figure 23) to figure out high-cost performance problems and implement a custom culling system setup with light blockers, texture caps, removing unnecessary overhead like Nanite and simulate hair, changing quality setting for volumetric fog and even fixing performance caused by other like the rendering target for inspectables.

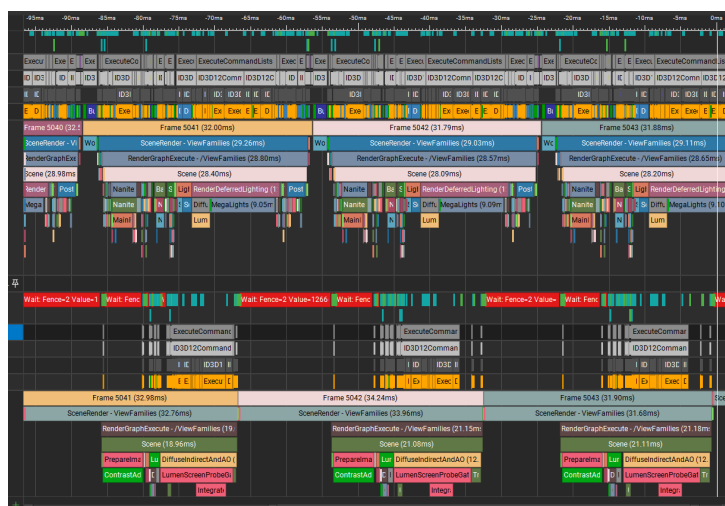


Figure 23. NSight frame breakdown

3.3. Outcome vs Industry

Our team's final outcome, the playable demo, meets all requirements and expectations laid out by the module. We were able to do this without compromising much on our vision. We reassessed consistently, scrapping aspects that were unnecessary or had failed in feedback, such as the attic, laundry, downstairs bathroom, VHS camera item, non-Euclidean portals, and sanity. This remained in line with the core development pillars established in the GDD and the art direction laid out in the art bible. This proved we could successfully follow industry standards. The final game features multiple complete characters, events, cutscenes, items, puzzles, and rooms in a fully set-dressed, atmospherically lit house. The gameplay features a complete story and goal, meets the mood and expectations of the psychological horror genre, and has a playtime of ~20 minutes (Figure 24).



Figure 24. Screenshot from final gameplay recording

Individually, I feel I met the industry standard. I was able to work on a wide range of technical art aspects and also focus on and release a plugin. My soft skills as a producer also improved, as the team had no real conflicts or problems. Together, this also means I met both my personal goals. I was able to work on polishing more than expected in this project, optimising performance, fixing bugs, and visual inconsistencies. Having a range of light health modes, interaction icons with smooth fade-in and fade-out based on status, and multiple complete cutscenes (Figure 25) shows a level above the standard, making it a clear case for industry-ready work.



Figure 25. Cutscene from final gameplay recording

I think the aspects that met industry standards the least were gameplay. Unfortunately, much like last year's group project, it lacks the depth I would expect in an AAA game; this time, however, it is much less of a concern. The final two aspects are sound and UI; importantly, these were not requirements for the module but were to be added later and not at an industry-standard level of polish. These gripes are small, however and do not detract much from the quality of the game.

Overall, Project UMBRAE compares favourably to industry practices. The technical foundation of the team was able to stay organised throughout a full development cycle, scaling scope to deadlines, iterating on feedback, addressing and resolving any issues, and collaborating across disciplines; these are core to industry practice (RGS, 2024). I also individually created unified OOD systems with fleshed-out feature sets (Figure 26) (GeeksforGeeks, 2018). Organised asset pipelines, asset guidelines and standards with documentation (Technical Artists Forum, Inc, 2011) which also pertain to the best practices set out by my industry.

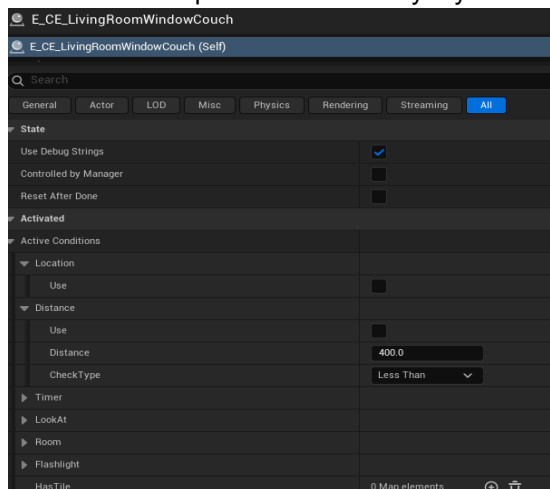


Figure 26. Event system with OOD state conditions

4. Reflection

4.1. Effectiveness

My effectiveness on Project UMBRAE was mainly contributed to a couple of factors:

- **Early planning** - I led the team's organisation, especially during pre-production, getting tasks started and assigned, and tracking their progress. I also outlined the project structure and made guides to help the project and my team. Before developing pipelines, I worked with artists to understand what they were already doing to integrate getting it together with feedback and change it if needed.
- **Communication** - I constantly talked with teammates and organised SCRUMs to make group decisions, updating out-of-the-loop teammates on changes and teaching anyone who needed it about custom systems or Unreal technology.
- **Consistency** - I created a set schedule for everyone to follow each week and organised a pattern that established expectations for what would be done each week.
- **Knowledge** - I was able to use my knowledge to work on complex aspects of development and teach teammates.
- **Not holding on** - The expression in game design is "Kill you darling" (Horowitz, 2023), but importantly, it means I never treated any ideas as necessary and could quickly adapt to new ones, especially with my teams.

4.2. Improvements

If I were to approach this project again, despite its success, there are two main aspects I would like to do differently.

Firstly, despite my attempts to be more focused on Technical Art over Producer this group project and having more teammates dedicated to aspects like lighting, vfx, materials, and game design compared to last year, I still fell into the same producer-heavy workload where I spent a significant portion of my time focused on overall scale, getting builds polished, helping others and presentation. This was worth it, but it meant I took a lot more of my day-to-day to keep up the Technical Art role, feeling perpetually guilty for not doing enough. Next time, I will distribute the workload more evenly to maintain a better balance across my roles and avoid spending all my free time on the project, which caused burnout this time.

Secondly, assigning a clear art director and lead for animation at the beginning of development would have helped take the sole project management pressure off me during the first nine weeks. While development worked well after this point, these being set from the start would have been beneficial (Figure 27).

Determine roles and skills needed

A typical game development project has many different roles. You'll need a programmer, an artist, and a level designer. You will also likely need a sound engineer and a tester. The right mix of skills is crucial for your project's success.

Structure the team

Think about the strengths and weaknesses of each person and assign roles that best suit their skills. Delegate tasks so everyone can work harmoniously and collaborate on ideas with ease.

Figure 27. Managing a game dev team (Pavliuk, 2023)

4.3. Ethical and Legal

As Project UMBRAE is a non-commercial educational work, it was important that we followed copyright laws and had correct licenses for assets used:

- Unreal Engine aspects like FAB, Metahuman, and Megascans EULA are all covered by the same Epic Content License, which means that as long as profit is under \$ 1 million and developed inside Unreal Engine, it is free for commercial use (Epic Games, 2017).
- We used CC0 licenses for many audio assets during development; Creative Commons 0 waives all copyright, so it would not be an issue (Creative Commons, 2020).
- External off-platform 3rd-party assets, such as David Gruwier's flashlight pack, were purchased with an educational license (Gruwier, 2026).
- Escape Studios provided SideFX's Houdini with an Educational Apprentice License (SideFX, 2026).

Some of these would require a more expensive license if the project were released commercially, such as Gruwier's Flashlight Textures and SideFX's Houdini.

Another ethical and legal aspect was the use of AI; Claude Opus, an LLM, was used to create the plugin. I followed Escape Studios' Code of Conduct, which outlines its use and purpose to assist development. The broader debate over AI training data and ownership remains unresolved (Figure 28), and the legal position on whether AI-assisted output is copyrightable depends on the human contribution being substantive.

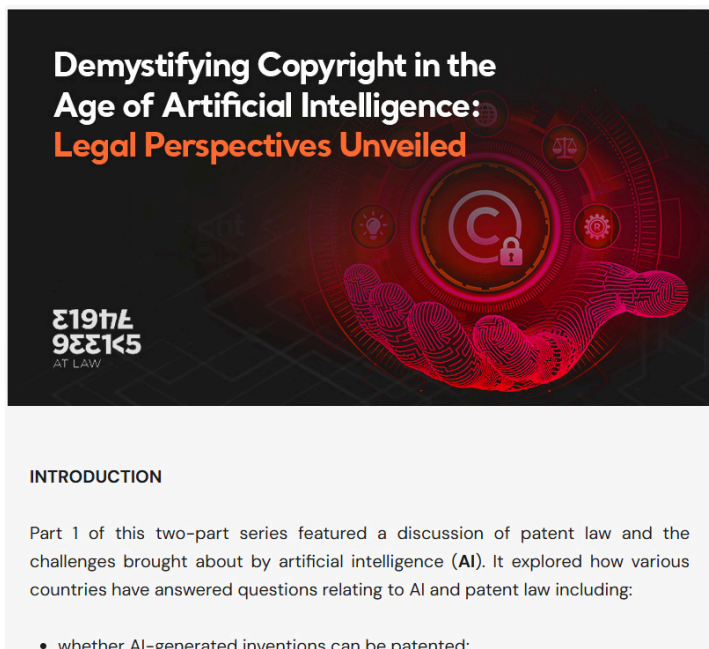


Figure 28. AI and Copyright (Eight Geeks Law, 2024).

4.4. Learning

My goal for this project was to showcase my skills in general Technical art and to create a solid portfolio piece in tools. Escape Asset Helper achieved this well, creating a feature-complete tool. I completed this using my existing knowledge and skills, and I had theorised about it since my last group project. The effort was rewarded, as it was key to securing my internship with Epic Games.

I explored one new area of Technical Art: technical animation. Prior to this, I had limited experience with animation and working with animators beyond technical terminology and interest. Setting up the technical animation pipeline with Animation Blueprints, Level Sequences, Cutscenes, Notifiers, Skeleton sockets, and retargeting were all areas I needed to learn and successfully introduced to animators (Figure 29). I am more confident in this area, and I do not want to focus on technical animation, as animation is a fragile system and requires some of the most dedicated bug-fixing across development.



Figure 29. A broken attempt at post-process IK

4.5. Future Goals

I did not get to explore as much Houdini Engine pipeline integration as I had hoped. I would like to use more advanced tools with HDA and integrate them into Unreal Engine, as I see that this is typical in technical artist portfolios (Figure 30).

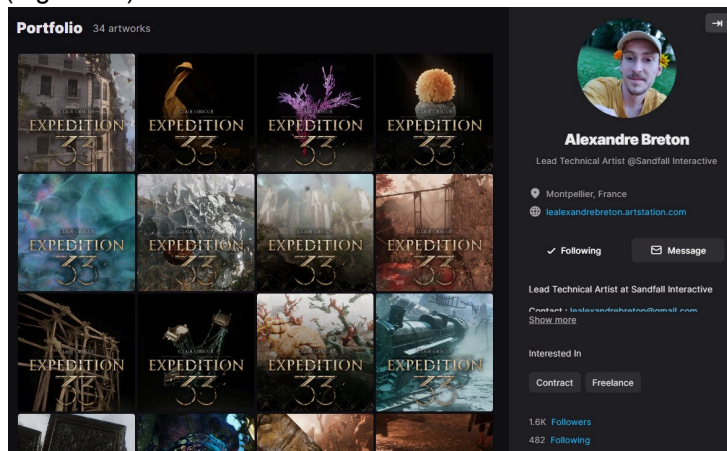


Figure 30. Technical Art Tool Portfolio (Breton, 2025)

I would also have liked to create a more dynamic post-process IK system to handle some of the more static animations, allowing them to connect procedurally to surfaces in Unreal with targeting and repositioning. This kind of polish is a key aspect I will continue working on for the project before our Rookies' submission.

Outside of this, however, I will focus on my Epic Games Quixel internship, as that is key to my next steps.

5. Conclusion

Project UMBRAE was a 15-person group project for the PR6001 Professional Studio Project, which aimed to simulate a real studio. Our team developed a polished first-person psychological horror game across 18 weeks. I was the team's technical artist and producer. The team was able to deliver a playable demo of a fully decorated house, featuring cutscenes, events, puzzles, items, audio, and UI, creating a vertical slice. I successfully pushed the team and the game into a continually playable state, receiving consistent feedback and iterating on it with weekly playtesting and tutor showcases. We had official Industry showcases with Karl Wooley on two occasions and a profiling-focused showcase with Jan Mroz, both of which provided industry insights and feedback. To follow industry practice, we adopted Agile development with SCRUM Sprints, used source control and asset pipelines, and collaborated cross-discipline. This was proven in how we handled changes in the plan, absorbed with buffer time, and were still able to go beyond by completing polish like sound and UI. We were also able to scrap and rescale as required without losing direction.

I completed two personal goals across development: first, creating a feature-complete custom C++ plugin called Escape Asset Helper, a naming-convention and asset-validation tool used by our team and Ecstasy's team. Secondly, I further developed soft skills as a producer in communication, organisation, and planning. By outlining asset budgets and creating guided videos, I also properly expanded into technical animation for the first time.

I learnt the challenges that come with a team as large as 15: communication and changes become exponentially slower when coordinating development. I am working on balancing my dual roles so I can focus on both equally to reduce burnout and workload going forward. As well as having clearly defined Art director and Leads to distribute production responsibilities with teammates early on.

The experience on this project has already shaped my future. Through my tool development on Escape Asset Helper, I landed an internship at Epic Games as a Tech Art Intern with Quixel. Alongside this, I will continue with this specialisation and work on polishing the showcase of this work for the Rookies.



Figure 31. Project UMBRAE Beauty Shot.

6. Bibliography

- **CAPCOM (2017).** *Resident Evil 7 Biohazard*. [online] store.steampowered.com. Available at: https://store.steampowered.com/app/418370/Resident_Evil_7_Biohazard/ [Accessed 22 May 2026].
- **Allar, M. (2023).** *UE4 Style Guide()* {. [online] GitHub. Available at: <https://github.com/Allar/ue5-style-guide> [Accessed 22 May 2026].
- **Ashby, S. (2024).** *How to Write a Game Design Document*. [online] www.gitbook.com. Available at: <https://www.gitbook.com/blog/how-to-write-a-game-design-document> [Accessed 22 May 2026].
- **Breton, A. (2025).** *Alexandre Breton Portfolio*. [online] ArtStation. Available at: <https://www.artstation.com/lealexandrebreton> [Accessed 22 May 2026].
- **Catness Game Studios (2023).** *Catness Game Studios*. [online] Catness Game Studios. Available at: <https://catnessgames.com/wikigame/build/> [Accessed 22 May 2026].
- **Certified Digital Artist (2022).** *Soft Skills Are the New Power Skills for Creative Industries*. [online] Certified Digital Artist. Available at: <https://certifieddigitalartist.com/blog/soft-skills-are-the-new-power-skills-for-creative-industries/> [Accessed 22 May 2026].
- **Creative Commons (2020).** *Creative Commons - CC0 1.0 Universal*. [online] Creativecommons.org. Available at: <https://creativecommons.org/publicdomain/zero/1.0/deed.en> [Accessed 22 May 2026].
- **Eight Geeks Law (2024).** *Eight Geeks*. [online] Theeightgeeks.com . Available at: <https://www.theeightgeeks.com/news/details-5> [Accessed 22 May 2026].
- **Epic Games (2017).** *Epic Content License*. [online] Unreal Engine. Available at: <https://www.unrealengine.com/eula/content> [Accessed 22 May 2026].
- **Epic Games (2026).** *Unreal Engine 5*. [online] Unreal Engine. Available at: <https://www.unrealengine.com/unreal-engine-5> [Accessed 22 May 2026].
- **GeeksforGeeks (2018).** *ObjectOriented Design (OOD) System Design*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/system-design/oops-object-oriented-design/> [Accessed 22 May 2026].
- **Gehman, C. (2019).** *What Is Source Control?* [online] Perforce Software. Available at: <https://www.perforce.com/blog/vcs/what-source-control> [Accessed 22 May 2026].
- **Gianty (2024).** *What Is a Vertical Slice? Exploring Key Concepts and Benefits* | GIANTY. [online] GIANTY. Available at:

<https://www.gianty.com/vertical-slice-game-development/>
[Accessed 22 May 2026].

- **Godoy, A. and Barbosa, E. (2010).** *Game-Scrum: an Approach to Agile Game Development*. [online] SB Games. Available at: https://www.sbgames.org/papers/sbgames10/computing/s hort/Computing_short19.pdf. [Accessed 22 May 2026].
- **Gruwier, D. (2026).** *DGruwier Flashlight Textures*. [online] Gumroad.com. Available at: <https://dgruwier.gumroad.com//dgruwier-flashlights?layout =profile> [Accessed 22 May 2026].
- **Horowitz, Y. (2023).** *Killing Your Whats?* [online] Game Developer. Available at: <https://www.gamedeveloper.com/design/killing-your-whats-> [Accessed 22 May 2026].
- **Into Games (2019).** *Quality Assurance Tester*. [online] Intogames.org. Available at: <https://intogames.org/careers/role/quality-assurance-tester -games> [Accessed 22 May 2026].
- **Kraj, N. (2025).** *Game Development Process: Keys for a Better Pipeline*. [online] Gdkeys.com. Available at: <https://gdkeys.com/game-development-process/> [Accessed 22 May 2026].
- **Kwalee (2024).** *LinkedIn*. [online] LinkedIn.com. Available at: <https://www.linkedin.com/pulse/what-playtesting-why-impor tant-game-development-kwalee-zb4uf/> [Accessed 22 May 2026].
- **Marland, J. (2026).** *GitHub - jamie-marland/EscapeAssetHelper* [online] GitHub. Available at: <https://github.com/jamie-marland/EscapeAssetHelper> [Accessed 22 May 2026].
- **Mladenovic, A. (2024).** *What Is Buffer Time & How to Find Your Sweet Spot | Memtime*. [online] Memtime. Available at: <https://www.memtime.com/blog/buffer-time> [Accessed 22 May 2026].
- **Moken Digital (2026).** *Why Growing Teams Feel Slower despite Having More People*. [online] Moken.digital. Available at: <https://www.moken.digital/post/why-growing-teams-feel-slo wer-despite-having-more-people> [Accessed 22 May 2026].
- **Mróz, J. (2026).** *GPU Optimization for Unity*. [online] Skool.com. Available at: <https://www.skool.com/gpu-optimization-for-unity> [Accessed 22 May 2026].
- **Muñoz, K. (2025).** *Art Bible*. [online] Art_Bible. Available at: https://dusthandler.github.io/Art_Bible/ [Accessed 22 May 2026].
- **Pavliuk, M. (2023).** *How to Manage a Game Development Team?* | Newxel. [online] Newxel. Available at: <https://newxel.com/blog/managing-a-game-development-te am/> [Accessed 22 May 2026].
- **Porokh, A. (2023).** *The Difference between Unreal Engine 4 & 5: Sneak Peek with Kevuru Games -*. [online] Kevuru Games. Available at:

<https://kevurugames.com/blog/the-difference-between-unr eal-engine-4-5-sneak-peek/> [Accessed 22 May 2026].

- **QATestLab (2018).** *Difference between Alpha, Beta, and Gamma Testing*. [online] qatestlab.com. Available at: <https://qatestlab.com/resources/knowledge-center/alpha-b eta-gamma/> [Accessed 22 May 2026].
- **Raj, N. (2024).** *Minimum Viable Product (MVP) - What Is It & How to Start*. [online] Atlassian. Available at: <https://www.atlassian.com/agile/product-management/mini mum-viable-product> [Accessed 22 May 2026].
- **RGS (2024).** *Role of Full Cycle Game Development in the Gaming Industry*. [online] Ropstam Game Studio. Available at: <https://ropstamgames.com/role-of-full-cycle-game-develop ment-in-the-gaming-industry/> [Accessed 22 May 2026].
- **SadSquare Studio (2018).** *Visage*. [online] store.steampowered.com. Available at: <https://store.steampowered.com/app/594330/Visage/> [Accessed 22 May 2026].
- **Schwaber, K. and Sutherland, J. (2020).** *The Scrum Guide*. [online] Scrum Guides. Available at: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scr um-Guide-US.pdf> [Accessed 22 May 2026].
- **SideFX (2023).** *Houdini Engine | SideFX*. [online] www.sidefx.com. Available at: <https://www.sidefx.com/products/houdini-engine/> [Accessed 22 May 2026].
- **SideFX (2026).** *Houdini Apprentice | SideFX*. [online] www.sidefx.com. Available at: <https://www.sidefx.com/products/houdini-apprentice/> [Accessed 22 May 2026].
- **Slack (2024).** *Agile vs. Scrum: Best Approach for Collaborative Teams | Slack*. [online] Slack. Available at: <https://slack.com/intl/en-gb/blog/collaboration/agile-vs-scru m>.
- **Team Penumbra (2026a).** *Project UMBRAE*. [online] itch.io. Available at: <https://jamie-marland.itch.io/project-umbrae> [Accessed 22 May 2026].
- **Team Penumbra (2026b).** *Project Umbrae | Cinematic Trailer*. [online] Vimeo. Available at: <https://vimeo.com/1190412394?fl=pl&fe=cm> [Accessed 22 May 2026].
- **Tech Art Aid (2022).** *Applying for a Technical Artist Job // Skills, Offers, CV*. [online] YouTube. Available at: <https://www.youtube.com/watch?v=KPbgNX75E0w> [Accessed 22 May 2026].
- **Technical Artists Forum, Inc (2011).** *Technical Artist, Best Practices*. [online] Tech-Artists.org. Available at: <https://www.tech-artists.org/t/technical-artist-best-practices /1436/5> [Accessed 22 May 2026].
- **The A.V. Club (2018).** *The 35 Greatest Horror Games of All Time*. [online] The A.V. Club. Available at: <https://web.archive.org/web/20181103000942/https://www. avclub.com/the-35-greatest-horror-games-of-all-time-1830 082156> [Accessed 22 May 2026].