Games Project Proposal

Extermination – "Procedurally generated doom in a fantasy setting"

Steven Holmes 1007041746

University of Derby

Table of contents:

1. Introduction

This project features a procedurally generated environment inspired by *The Binding of Isaac*, *Doom*, *Gears of War*, *Warhammer 40k*, and *Scorn*. Players take on the role of an exterminator tasked with investigating a once-thriving settlement overrun by mutated inhabitants.

Narrative and Setting
The journey begins on the surface and leads to a grimdark subterranean research facility, gradually revealing the source of a deadly alien pathogen.

Level Design and Structure
Levels follow a linear, room-based structure enhanced by procedural generation for replay ability. Each level ends with a boss fight and access to an elevator that deepens the descent.

Gameplay Mechanics and Progression
Inspired by *The Binding of Isaac*, each level includes an upgrade room and a randomized boss. Players collect upgrades and progress through increasingly challenging encounters.

Replay ability and Meta-Progression
Procedural systems ensure variety in enemies, items, and bosses. A meta-progression system unlocks new content over time, encouraging repeated runs and strategic experimentation.

There will be 5 levels, four of those with lower tier bosses and the 5th level being the final boss area.

2.Aims

The aim of this project is to make a three-dimensional rogue like which remains relatively underexplored within the games industry, particularly in the context of first-person shooters. By merging the replay ability and procedural unpredictability of traditional roguelikes with immersive first-person gunplay, the game seeks to deliver a compelling and engaging experience that prioritises both mechanical depth and narrative atmosphere.

Procedural generation will be utilised not only as a tool to enhance replay ability but also as a practical development strategy to streamline content creation. This approach enables the team to significantly expand the game's variety and longevity while maintaining high production efficiency. The end goal is to create a rich, repayable experience with dynamic combat, diverse encounters, and emergent storytelling, all within a cohesive and atmospheric sci-fi setting.

3.Objectives

- Design a modular set of rooms for procedurally generated levels.
- Create a variety of smaller enemy types with distinct behaviours.
- Develop multiple unique boss encounters, including a final boss.
- Implement a replayable gameplay loop with meta-progression, rewarding players with unlockable content such as new enemies, weapons, and upgrades.
- Develop a range of weapons and upgrades to support varied playstyles.
- Develop a combat system involving melee and fire arm mechanics

- Public beta or commercial release by the end of the year.
- Develop AI pathing and implementation of hordes for enemies

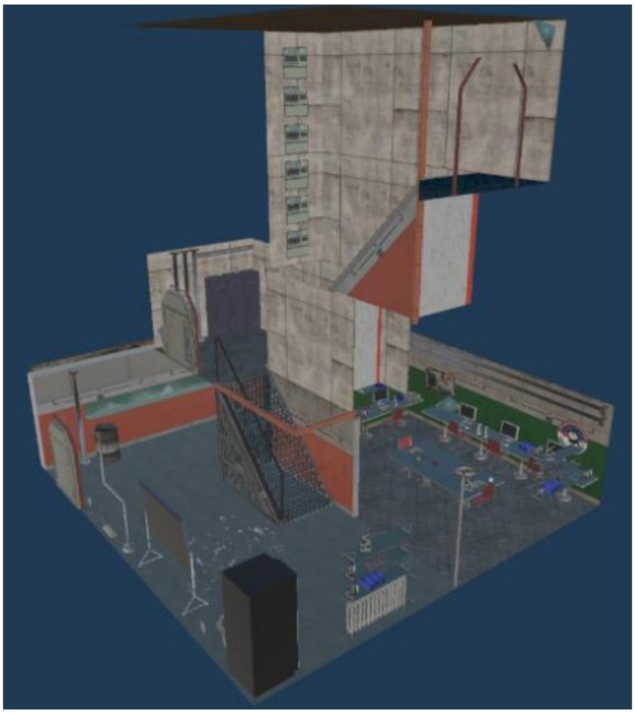
4. Initial Research and literature review

The procedurally generation tools created last year will be suggested for use in this year's project and will be open to adaptation for the teams needs, research has been conducted into procedural generation previously already, information on this will be in the appendix. Ther are videos documenting Its uses and a written report.

. "*The Binding of Isaac*" applies procedural generation to its dungeon layouts, items, trinkets, and collectibles. The same design ethos will be followed in the creation of this games project. Just as Isaac does there will be preconstructed rooms and lay outs, we can look at the game Repo for how it handles 3d environments,

Repo uses procedural generation for its levels on runtime.

The developers use preconstructed prefab rooms, corridors, and stairways instantiated at predefined positions. This node-based method ensures unique level configurations each playthrough. "Unlike traditional maps, these aren't just shuffled; they are built differently each time." (**GamerBlurb 2025**)



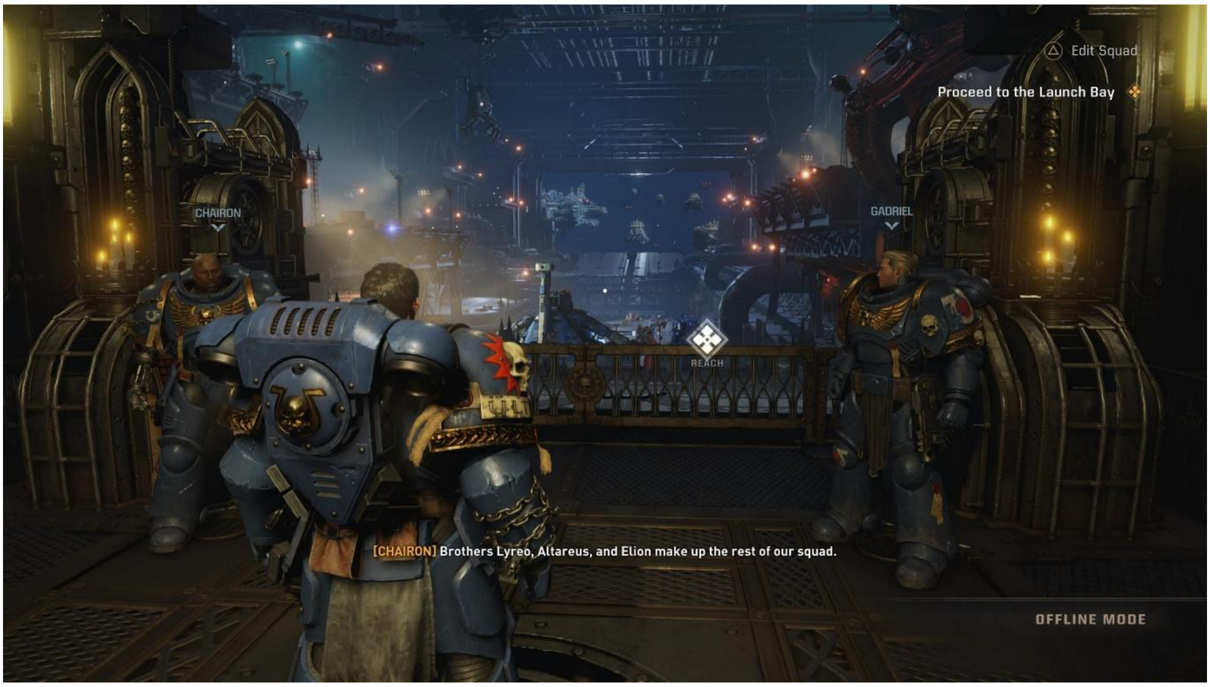
Environment design research conducted thus far:

Above ground: level 1 a dark and dreary place:

Clearly once a bustling town but not abandoned, the above ground setting looks under development in terms of technology, the residence here would have stood no chance of surviving or putting up a defence against the pathogen lying below. The environment story telling needs to lead the player to the entry to take you below, it should be clear that corpses and blood trails lead in this direction as if the mutants and pathogen have been dragging down food for their survival.

Transition level design ideas thus far:

Each level will end with entering an clear exit, portal,staircase or elevator and going down to load the next area it’s at this point we can make use of level streaming, un load the old and load in the new, whilst on the elevator we are generating the next level below, each time delved down lower the environment becomes more organic, you are slowly entering the belly of the beast. Space marine and gears of war use this same elevator system for their games to load new areas here’s an example:



[CHAIRON] Brothers Lyreo, Altareus, and Elion make up the rest of our squad.

The design changes for this game would be a more enclosed look so the procedural generation can be hidden. This way there can be seamless gameplay and no need for loading screens.

5. Proposed method

1. Assess Team Skills and Assign Roles
Evaluate each team member’s skills, experience, and personal interests to assign roles and workloads that best match their strengths and preferences. Ensure every member feels capable and comfortable with their responsibilities. Encourage early identification and open discussion of any challenges to maintain team cohesion and promote efficient collaboration.

2. Set Up Version Control and Project Management
Create a GitHub repository for version control to efficiently track changes and manage project files. Utilize GitHub’s project management tools to coordinate tasks, maintain transparency, and keep all planning documents publicly accessible to the entire team.

3. Establish Communication Channels
Create a dedicated Discord server with multiple voice channels and organized text channels tailored to different project aspects. This will facilitate clear and efficient communication within the team.

4. Implement Scheduling and Meetings
Schedule a mandatory weekly team meeting, with attendance required unless a valid

reason is provided in advance. Team members should mark their availability, and a consistent weekly day will be set for a scrum meeting to review progress, address challenges, and plan upcoming tasks.

5. Development journal
Develop a marketing plan including social media presence, trailers, demos, and community outreach. Start building a player community to generate buzz and gather pre-release feedback. Video Logs.

6. Develop Style Guides
Begin by creating comprehensive style guides for both the programming and art departments. These guides will set cohesive standards and best practices to ensure consistency across all assets and code.

7. Create and Review Test Logs
After finalizing style guides, team members will create detailed test logs for their work. All test logs must be thoroughly completed and reviewed by at least two team members before merging into the main project repository. Assigning three reviewers per submission is recommended to improve efficiency.

8. Begin Work and Set Milestones
Once roles are assigned and workloads are comfortable for everyone, active development begins. It is expected that within two weeks; the team delivers a full game loop and a block out of at least the first level.

9. Iterative Development and Regular Playtesting
Continue development with an iterative approach, building features incrementally while conducting regular playtesting sessions. Collect and analyse player feedback to inform design refinements, bug fixes, and gameplay balancing. Encourage open communication between teams to address issues quickly.

10. Content Expansion and Polishing
Develop additional game content such as levels, enemies, weapons, and bosses, building upon the initial block out and game loop. Focus on polishing art assets, animations, UI, and audio to elevate the player experience. Ensure procedural generation systems are robust and deliver variety as intended.

11. Quality Assurance and Bug Fixing
It is imperative that the team QAs as often as possible. Initiate thorough QA cycles with dedicated bug tracking. Prioritize fixing critical issues, optimizing performance, and ensuring cross-platform compatibility if applicable. Involve both internal testers and external beta testers where possible.

12. Beta Testing and Feedback Integration
Prepare and release a closed or open beta to gather broader player feedback. Monitor

gameplay data and community input to identify further improvements and final adjustments. Maintain active communication channels with testers.

13. Games Expo Preparation
In preparation for the university games expo, the team will coordinate a strong and professional public appearance. This includes creating high-quality marketing materials such as standees, printed t-shirts, business cards, and potentially small merchandise items. The team will also ensure a polished and stable version of the game is ready to demo, with a setup plan for the booth and a clear rota for team representation during the event.

14. Final Release Preparation
Finalize all game assets and code, conduct final testing, and prepare distribution channels (e.g., Steam, Epic, console stores). Ensure all legal requirements and certifications are met.

15. Launch and Post-Launch Support
Release the game to the public. Monitor player reception and address any urgent issues with patches or hotfixes.

6.Game References

Scorn, Vermintide, Warhammer 40k, gears of war, doom, resident evil village,

the binding of Isaac,enter the gungeon, Hell divers.