

Project Proposal Final

by Jacob James-MacHin

Submission date: 31-Oct-2025 04:30PM (UTC+0000)

Submission ID: 266451281

File name: 50542_Jacob_James-MacHin_Project_Proposal_Final_695044_414445743.docx (34.23K)

Word count: 1672

Character count: 8898

SOURCE ENGINE IMMERSIVE SIM PROJECT PROPOSAL

By Jacob James-
Machin Games
Development Year
3 BSC

Supervisor: Nathan
Morgan

Date: 39/10/2025

Project Overview

This is the Project Proposal of the Source Engine [1] project which will utilize immersive sim design elements [2] and an algorithm that would change its levels to account for the player's playstyle. This project will utilize puzzle elements for the player to solve and combat elements. All of which can change and adjust slightly that would accommodate the player on how they approach solving problems.

For example, if the player decides to play this game using stealth the game would change its levels in terms of lighting, and it will also change and revamp enemy and item placements that would also change whatever if the player decided to keep with stealth or decides to use a combat heavy playstyle.

The intention behind this game is to be completely different every time the player replays it and offer a new experience each time.

The Problem to Solve

To create a procedurally generated immersive sim within the Source Engine. I want to create one that uses a procedurally generated algorithm [3] that would change itself for the player while also using in engine assets. Not to just generate levels but to generate lighting and enemy placement. Something the Source Engine is capable of through the Left 4 Dead games with its "AI Director" [4]

The Aims

The aim of this project overall is to design procedurally generated levels/items that would change itself to suit the player's playstyle. For example, if the player wants to go all in using a combat heavy playstyle the game would add in more enemies and ammo for bigger guns and if the player wants to sneak around and use stealth the game would spawn in items that would help the player sneak around and spawn in less enemies. To note also this concept could be used to change lighting so the players can use the darkness to sneak around.

To create a project like this is to understand how procedural generation works in level editing and how that could be applied to the existing toolsets that you have access to already. For example, using c++ and the official Hammer editor.

The use of lighting

Lighting is a necessary tool in a game that helps create visual depth and evoke a certain mood. However, it could also be used as a gameplay mechanic. As demonstrated in the aims I want to use lighting as a gameplay mechanic that will change depending on your playstyle. Like in this game if the player decides to use combat as part of their playstyle the lighting would be used to foreshadow enemy placement or possible strategies and flanks whereas if the player opts in for stealth it could highlight puzzle elements, make areas darker or suggest areas to explore. [5] [6]

The use of Algorithms

The use of these kinds of algorithms that dynamically alter the game comes down to a set of rules and instructions that a game would follow and, in this instance, it will be used to

dynamically change and alter the levels in a game. Not just relying on NPC interaction. For example, procedural generation. This is method that creates data algorithmically without the need for human input. This will be determined by the playstyle of the player. Encouraging a dynamic difficulty curve and allow for the game to feel different each time a player interacts with the game. [7][8]

The Objectives

- Deliver a compelling design-oriented experience that would feel fresh and unique every single time you play it in any way you like.
- Develop an algorithm that can adjust how the levels are laid out through lighting and object placement.
- Develop a narrative that would tie it all together.
- Build a demo that show cases this technology using one of the levels listed in my design document and mind map.

The Methods

There will be a controlled and carefully conducted questionnaire where it will see what kind of playstyle players would go for and what kind of solutions they would be able to come up with. This questionnaire will be conducted in a way that would place out a scenario for the player (by asking a question) and see how they would respond. E.g. how would the user approach an obstacle.

Project Background

The project is based around the Games Development module and the process of creating and designing a game.

Project Timeline

November

- Create a design document detailing what will be in the game

December

- Project development will begin in December
- Create a daily log of everything that will be added to the development pipeline. This will help me address any bugs

January

- Create the algorithm that will be used to generate gameplay elements.
- Design a test level that will use this technology

February

- The start of the design phase. I will begin building the necessary levels

March

- Work on a demo release that will include a portion of the finalized game

April

- Bug fixing and polishing

May

- Demo release

Resources

The game will be on the source engine, so you won't need a crazy powerful computer to run it. However, a technical know-how is needed to understand what you are working with so there will need to learn how to use the Source Engine and use the available documentation that Valve has provided over the years. [9]

The tool that will be used to develop this mod/game is the Hammer++ editor and another tool that will be utilized is Visual Studio and C++. [10]

To understand the scale of this project a technical specification must be outlined with the necessary equipment and tools.

Also to further note this everything will be documented for bug testing which will include comments in the code and a daily dev log. Detailing every change made in the development pipeline

Technical Specifications

The biggest requirement for a project this scale is a computer that can manage game development processes, and a computer is also required to test the finished game. As for software requirements. [11] As mentioned, Visual Studio is required alongside hammer++ as a source engine project with scripted sequences will be developed. However more software is needed such as:

Office

Use of Microsoft office will help in constructing a game design document and a potential story script that will detail everything the game will have.

Notepad

This will allow the use to take notes during development of this game. This will help in a potential final evaluation and bug testing. Also to keep track of the development of this game.

Microsoft Forms/Google Forms

The use of these questionnaire software programs would help take real time feedback for this game while also help with bug testing. It could also be used in the initial design phase of the game allowing users to answer questions that is related to the design.

Notepad++

Notepad++ will be used for additional coding and to create config files. Source Engine Games use these config files aka “cfg” files to determine the state of the game.

Blender

The use of blender will come in the use of potential modelling is required. Even though this project will use pre created assets from Half-Life 2, but potentially new models will be created. So, the use of Blender is required.

7Zip

7zip will be used to help package and will be used in the release files as users will download the zip file.

GitHub

GitHub will be a repository for the code for the game. It could also be used as a helpful resource which could help solve potential problems down the line.

ModDB

ModDB will be the platform for the project's final release.

Literature resources

The Literature Resources required for this project is both books and online resources. One of these online resources is a website known as the “Valve Developer Wiki” Articles will also be utilized as there's a ton of information about game design and using the source engine online. As such articles or presentations that covers replayability and procedural generation through AI. [12]

Backup Plan

In case that the Source Engine won't be as of use. There are alternatives that could be used such as Unreal or Unity.

Project proposal

Analysis

To make sure this project will have a smooth release. There will be a coordinated playtesting with the supervisor and developer to make sure the game runs well and all the features work as intended

Platforms/Release plan

This project will have an initial demo release that will showcase a part of the final game. With a planned full release on moddb alongside a demo release through OneDrive or Google Drive.

Bibliography

- [1] Anon, "Source Engine," Wikipedia, 12 August 2025. [Online]. Available: https://en.wikipedia.org/wiki/Source_game_engine. [Accessed 23 October 2025].
- [2] rosodude, "'Immersive Sim' is a Design Philosophy, not a Genre," Word Press, 14 December 2020. [Online]. Available: <https://rosodudemods.wordpress.com/2020/12/14/immersive-sim-is-a-design-philosophy-not-a-genre/>. [Accessed 23 October 2025].
- [3] Anon, "Procedural Generation," Wikipedia, 6th September 2025. [Online]. Available: https://en.wikipedia.org/wiki/Procedural_generation. [Accessed 23 October 2025].
- [4] A. J. Champandard, "11 Secrets about LEFT 4 DEAD's AI Director and its Procedural Zombie Population," 7 October 2009. [Online]. Available: <chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.cs.drexel.edu/~so367/teaching/2012/CS680/papers/11%20Secrets%20about%20LEFT%20%20DEAD%e2%80%99s%20AI%20Director%20and%20its%20Procedural%20Zombie%20Population%20%7c%20AiGameDev.com.pdf>. [Accessed 23 October 2025].
- [5] M. Jenssen, "Functional Lighting," 2012. [Online]. Available: https://magnarj.net/article_funclight.html. [Accessed 28 10 2025].
- [6] A. Y. Robert Yang, "The Level Design Book Lighting," 2025. [Online]. Available: <https://book.leveldesignbook.com/process/lighting>. [Accessed 28 10 2025].
- [7] T. L. Fort, "CONTROLLING RANDOMNESS: USING PROCEDURAL GENERATION TO INFLUENCE PLAYER UNCERTAINTY IN VIDEO GAMES," University of Central Florida, Orlando Florida The School of Visual Design, 2015.
- [8] Anon, "AI Game Algorithm (What You Need To Know)," HELIKA, 2025. [Online]. Available: <https://www.helika.io/ai-game-algorithm-what-you-need-to-know/>. [Accessed 28 10 2025].
- [9] D. Crespo, "The Evolution of Valve's Source Engine," Deus In Machina, 28 12 2023. [Online]. Available: <https://www.deusinmachina.net/p/the-evolution-of-valves-source-engine>. [Accessed 30 10 2025].
- [10] A. M. Julia Case Bradley, Programming in Visual C# with Visual Studio Professional Edition Software, United States: McGraw-Hill, Inc., 2007.
- [11] S. Madhav, Game Programming in C++: Creating 3D Games, Addison-Wesley Professional, 2018.
- [12] M. Booth, "Replayable Cooperative Game Design: Left 4 Dead," Valve Software, Bellevue Washington, 2009.



Project Proposal Final

GRADEMARK REPORT

FINAL GRADE

31 / 100

GENERAL COMMENTS

Introduction / Aims & Objectives: 10/20

Project concept is interesting and technically ambitious

Clear focus on procedural generation in Source Engine

All four objectives fail SMART criteria. They lack measurable targets, specific metrics, and clear success thresholds.

Something else to consider is the formality, we're aiming to write it in an "Academic Style" so we aim to avoid an informal style such as ("This is the Project Proposal...")

Background / Literature Review: 2/20

Project

Design / Methodology: 8/30

Section severely underdeveloped (approximately 69 words vs 500 required)

No development methodology stated (Agile? Waterfall? Iterative? Prototyping?)

No algorithm design methodology specified

A Development methodology selection with justification would have been great to see.

This would have been a prime location to discuss the algorithm design and testing approach, i.e. your ideas on how you would have made it work

You mention including users to test it, but what would they be looking for? A detailed user study protocol would be used to capture this information.

Resources / Planning: 7/15

Clear monthly timeline showing progression but timeline lacks specific dates (only months).

Comprehensive list of required software and tools.

Mentions documentation strategy (daily dev log, code comments).

Timeline doesn't align with methodology (no user testing phase shown)

Document

Presentation: 4/15

Generally clear writing in most sections

Logical structure attempting to follow required format

Technical terminology used appropriately

Spelling and grammar mostly correct

Key issues are that the proposal completely lacks a literature review section and that a bibliography is a little different from a reference list, but I can see here the word has been used interchangeably.

Ideas on how we can take this further:

Comparative analysis: How does your adaptive approach compare to static level design in terms of player engagement and completion rates?

Perception study: At what point do players notice the procedural adaptation? Is subtlety important?

Classification methodology: What metrics/thresholds define "stealth" vs "combat" playstyle?

Replayability metrics: How to quantitatively measure "different experience each time"?

PAGE 1

PAGE 2



Comment 1

Like the yellow paint approach but with light?

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7
