Tampere University of Applied Sciences

Creating PSX-Style Isometric Environments in Godot Engine

Karin Aimonen

BACHELOR'S THESIS

May 2025

Degree Program in Media and Arts
Interactive Media

ABSTRACT

Tampereen Ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Media and Arts
Interactive Media

AIMONEN, KARIN
Creating PSX-Style Isometric Environments in Godot Engine

Bachelor's thesis 60 pages
May 2025

In the recent years, the gaming industry has experienced a surge of nostalgia through the revival of retro graphics, embraced by both major companies and independent studios, reminiscent of the iconic aesthetic of early 3D computer graphics. This visual style not only suits a variety of game genres but also aligns with the limited resources of smalls development teams. These benefits led to the exploration of retro-style graphics inspired by the PlayStation-era to create the visual direction of the game Project Survival Academia, an isometric survival game with management elements developed by the author’s team.

This thesis covers the process of producing PSX-inspired graphics using modern software, highlighting the benefits and challenges of using an isometric projection. The theoretical part is accompanied by a prototype scene for Project Survival Academia, first modelled in Blender, then implemented and stylized in the Godot Engine.

The results of the project demonstrated that the workflow needed to produce retro-inspired graphics was easily accessible by adhering to technical limitations of the 1990’s, such as low-resolution textures, low-poly models and baked lightmaps, all facilitated by a modern engine like Godot.

Key words: retro graphics, PSX, aesthetic, isometric, art style, Godot

CONTENTS

ABBREVIATIONS AND TERMS

PSX	PlayStation X, original name of the PlayStation
Godot	An open-source game engine
Blender	A 3D-modelling software
Polygon	A plane geometry divided by line segments
PC	Personal Computer
CD-ROM	Compact disc read-only memory, discs that contain data
Integer	Data type used in programming that represents whole numbers and doesn’t support fractional values
Float	Data type used in programming that represents a wide range of values, including fractional values
FPU	Floating-Point Unit, a component of a computer system designed to execute operations on floats.
CPU	Central Processing Unit
RAM	Random Access Memory, a type of memory storage in a computer
VRAM	Video Random Access Memory, a computer memory that stores the data of the graphics that will be rendered on screen
Projection	A technique to simulate perspective in computer graphics
Orthographic	Parallel projection type in which projection lines are perpendicular to the projection plane
Isometric	Projection type in which the angle between coordinate axes is 120 degrees.
Bits	Determine the amount of memory and the precision of data that can be processed in a game console
Z-buffer	Determines depth information of objects in a 3D scene
Color depth	Determines the number of colors that can be displayed, also defined by bits
Texture CD	A disc containing texture libraries
Krita	A drawing and painting software

1 INTRODUCTION

The media field has been experiencing a surge of nostalgia in the recent years, with the revival of 2000s pop culture trends in music and fashion, as well as re-boots of successful movies from that time like Jumanji or Mean Girls (Doschinescu 2024). It is no different for the video game industry, seeing how big companies are re-releasing beloved classic games and indie developers inspiring themselves from early 2000's graphics.

Isometric projection has been used since the early days of the game industry. Today, it remains a popular type of view among players and is adopted by numerous new releases, as it fits various genres of gameplay and offers great advantages for developers in terms of efficiency.

This thesis will research both topics of PSX-style graphics and isometric projection. The text will discuss the qualities, techniques and limitations of PSX-era graphics, as well as theory of graphical projections and the benefits of using isometric projection in games. As the topic of this thesis is about creating isometric environments, the content will also partly delve into environment design in games. The findings will then be implemented in a practical project, where the intention is to create PSX-inspired graphics using modern tools.

The practical project will include a prototype scene for a game developed by the author in collaboration with a game designer from the Interactive Media path at TAMK. The game in question is a survival game with aspects of action and management.
The scene will be modelled in Blender and implemented in Godot Engine. The practical section of the thesis will introduce the reader to Godot in general, while exploring the engine's tools to create retro-style visuals.
The aim of the project is not to re-create authentic PSX-era graphics, but to find a suitable visual style for the game while experimenting with and drawing inspiration from PSX-games.

This thesis is written with an expectation of the reader understanding the basics of video game development. The intention is to shed light on the process of creating retro-style visuals. Additionally, this thesis hopes to offer a better understanding of Godot engine.

2 PSX-AESTHETHIC

2.1 History

Video games have been around for decades, with platforms ranging from arcade systems to home consoles, to handheld and mobile devices. Bringing video games from local arcades to living rooms marked the beginning of console game history. One of the most significant consoles of all time is the PlayStation (PICTURE 1.), Sony's entry into the video game console market.

PICTURE 1. The Sony PlayStation, also known as PS1 or PSX.

Between the years 1989 and 1991, Sony was waiting for its chance to break into the video game industry through its anticipated collaboration with the gaming giant, Nintendo. Sony was expected to develop a CD-ROM expansion for Nintendo's SNES console and would later be allowed to create its own entertainment hardware, the 'Nintendo PlayStation', which would run both SNES cartridges and CD-ROM games (McFerran, 2015, p.9). However, Nintendo ended up striking a deal with Dutch company Philips instead, leaving Sony to pursue its own path. Fortunately, the PlayStation prototype was already being developed at this point, along with possible launch game titles.

To separate itself from the previous project with Nintendo, Sony branded its new console 'PlayStation-X', although the 'X' was eventually dropped from the name at the official launch in 1994 (McFerran, 2015, p.11). The name gave rise to the

abbreviation ‘PSX’, which is still used today to describe a significant era of video game history.

2.1.1 The rise of 3D

The PlayStation had managed to join two of the biggest advancements of the 1990’s console game technology: CD storage and cutting-edge 3D graphics (Donovan, 2010, p.215). While the launch of the console had a successful start, the rest of the industry was doubtful, as 3D had yet to make its breakthrough in games.

3D game graphics had already taken a step in the right direction on the other side of the world, in Texas. Id Software were pioneers of textured 3D models. Their PC games Wolfenstein 3D, released in 1992, and Doom, a year after, shook the industry with the most detailed 3D environments of that time (PICTURE 2.). But it was Sega that took game characters to a whole new level.

PICTURE 2. Doom (1993), Id Software.

The gaming field had not seen successfully animated 3D characters, until the release of Virtua Fighter for the Sega AM2. In this first polygon-based fighting game, the characters’ movements were realistic and fluid, and looked more alive than previously unmatched 2D sprites ever had (Sponsel 2010).

Virtua Fighter was rushed to release in the US in 1995 as the launch title of Sega Saturn. But the Saturn port proved to be defective, and around the time Sega was shipping a re-issue, Sony swept in with well-regarded titles such as Battle Arena Toshinden (PICTURE 3.) of the same genre, and the racing game Ridge Racer (Campbell 2012). Sega had convinced the industry of the potential of 3D, so Sony could truly take off and dominate the 3D field in gaming for years to come.

There were considerable technical aspects setting PlayStation apart from previous consoles. For one, the PlayStation was a 32-bit system, a natural step up from its 16-bit predecessors SNES and Sega Mega Drive, and in practice meant more complex gameplay.

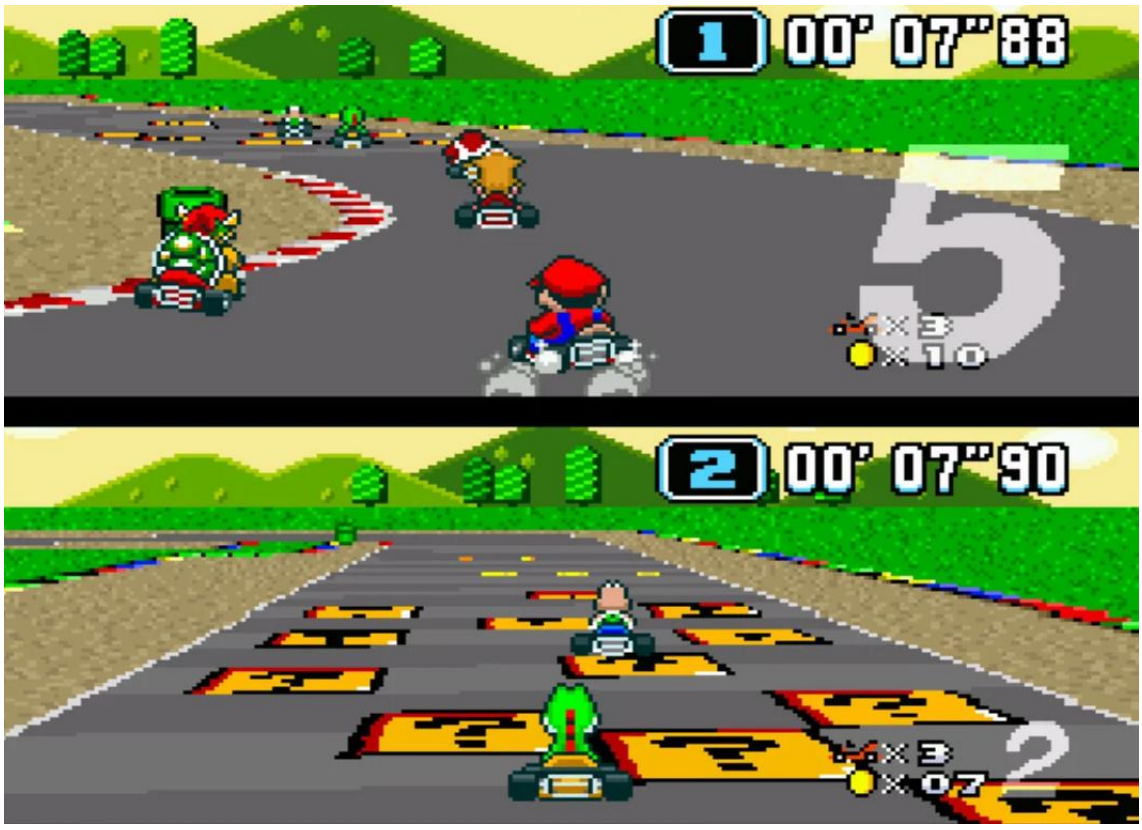
PICTURE 3. Battle Arena Toshinden (1995), Tamsoft.

The switch from cartridges to CD's changed everything. CD-ROMs became the new generation of data storage, as they were significantly cheaper and faster to produce than cartridges. However, as the name states, CD-ROM (compact disc read-only memory) couldn't overwrite data, meaning save files couldn't be written or erased on the disc. The saving functionality came instead from external memory cards, along with an increased amount of data storage thanks to the PlayStation being able to support multiple-disc games. When swapping discs, the

memory card could carry your progress over to the next game disc, extending the duration and complexity of video games. Transferring data in this way was not possible with cartridges.

Improved storage space meant not only more complex games with larger worlds, but more detailed graphics and higher-quality audio. Basically, CDs offered a more immersive gaming experience on consoles, as already proven by early PC games.

The jump to more defined graphics created opportunities for polygonal graphics. Simple polygon graphics were available on some consoles before the PSX, but they were textureless, rigid objects such as race cars and spaceships. Some SNES games utilized faux 3D backgrounds, which were essentially 2D planes rotated and scaled to simulate perspective. This was done for Mario Kart with the graphics mode called Mode 7 (PICTURE 4.). The PlayStation on the other hand relied on a vector and matrix math coprocessor (GTE) to provide the high speed needed to achieve extensive 3D graphics: lighting, geometry and polygon and coordinate transformations (Pezzi 2024).



PICTURE 4. Super Mario Kart (1992), Nintendo.

2.1.2 Graphic Specifications

Despite its achievements in 3D graphics, Sony soon found a competitor in its field with Nintendo's N64 console, released in 1996, only two years after the PlayStation. When comparing the two, especially on any screen more modern than the old CRT's, there is a noticeable difference (PICTURE 5.).

PICTURE 5. A comparison between Rayman 2 on PlayStation (left) and Nintendo 64 (right). Rayman 2: The Great Escape (1999), Ubisoft.

As mentioned earlier, 3D games hadn't quite broken into the market when Sony released their console, in 1994. PC was slowly making headway for consoles, but consumer-focused dedicated graphic cards only came around in 1997 (PC Plus 2010). Before that, even the most popular PC games were integer-based engines.

Game programming of the past was widely based on integer calculations because of the limitations in hardware performance. Some Floating-Point Unit (FPU) designs were on the market since before the PlayStation, but they were expensive and there still wasn't enough expertise in programming games for them. In addition, Sony was already producing their graphic and audio chips in-house, and only needed a secondary CPU supplier to manage their components (Copetti 2019). For those reasons, Sony had opted out of using an FPU and chosen an older CPU, pressing on to produce their machine.

Meanwhile, Nintendo was looking to commission the processor architecture of the N64. They partnered up with a company called Silicon Graphics (SGI), a leading name in the field of computer graphics at the time, to bring state-of-the-art graphics systems into home consoles. The CPU that came from the collaboration made the N64 a powerhouse of consoles, performing operations with 64-bit precision including floating-point arithmetic for high-performance graphics. (Peddie 2020).
As a result, the N64 had the advantage in 3D rendering capabilities, as it was capable of floating-point calculations and was able to handle more polygons on screen at once, as well as physics. The absence of FPU on the PlayStation caused some unwanted visual effects, like the famous polygon jittering, which is why animations in N64 games look smoother than those of PSX games.

On the other hand, the shared memory system of the N64, where the GPU and CPU share the same pool of RAM, prevented the console from fully utilizing its advantage in processing power. The PlayStation's efficient design, which relied on a dedicated VRAM, allowed for more immediate access to graphics data. This is why even if the N64 theoretically possessed more processing power, it didn't necessarily outdo the PlayStation in terms of performance (Retrolize 2025).

When it came to texture quality, the N64's custom graphic chip was capable of drawing Z-buffer, but the console lacked sufficient storage for textures, resulting in high texture compression and therefore blurry graphics (Retrolize 2025). On the PlayStation, the lack of Z-buffer caused texture warping and flickering effects in games. However, its VRAM provided more texture storage space over the Nintendo 64, meaning the PSX could process higher resolution textures.

The consoles also differed in terms of color depth, or how many colors can be displayed on screen. The PlayStation could output 4-bit, 8-bit and 16-bit colors, albeit the higher color depth was mostly used for still 2D images (User PolyHertz 2021). This is what gave rise to the trend of pre-rendered backgrounds in games of the 1990's. The N64 on the other hand had a maximum output of 24-bit colors, but most games would use lower color depth to conserve resources (Carlson 2020).

Ultimately, Sony sacrificed lower grade graphics including jittery polygons and distorted textures, in exchange for better speed and lower cost of building its hardware. Making games for the PlayStation was also cheap both in cost and performance, not to mention developer friendly, because of Sony using standard programming language and offering support for developers through development toolkits.

The comparison between the two consoles demonstrates how the efficiency of Sony's hardware design made up for the technical gap between the PSX and N64. Thus, Sony managed to convince both customers and developers alike of the potential of the PlayStation as the new generation entertainment system, and ended up selling over 100 million units, surpassing N64's sales of about 33 million (Sirani, 2024).

2.2 Retro Graphics in the Modern Day

Video game technology is constantly evolving. New game titles emerge every year, each one more immersive, visually impressive and technically sophisticated. The rise of virtual reality and AI in the recent years have shown us a glimpse of the direction gaming is most likely headed towards. How is it that, at the same time, a wave of nostalgia is sweeping over the industry?

There are a few reasons why retro gaming has been gaining popularity over the recent years, one being accessibility. Emulators make it easy for anyone to get their hands on old classics. The hobbyists who develop emulators often contribute to creating and supporting online retro gaming communities, as most of them provide their software for free with additional features such as options to integrate cheat codes and create save and load states. One example is the ePSXe emulator, written by three authors under the aliases "calb", "_Demo_", and "Galtor", and available to download for free online.

The emulating trend has not gone unnoticed by major companies such as Nintendo, cashing in on the phenomenon with the release of their own emulators, NES Classic and SNES Classic.

2.2.1 Nostalgia and Simplicity

Most gamers these days are returning to old classics because it provides them with an effective form of escapism. Madigan (2013) brings up Tim Wildschut and Constantine Sedikides' study on how a person in a bad mood or stress is more inclined to recall fond memories. In this way, nostalgia can provide us with comfort and higher self-esteem. (Madigan 2013).

According to McCarthy (2021), SohoMD cofounder Jacques Jospite Jr. delves more into the nostalgia of old games, as he explains the intrinsic and extrinsic properties of retro gaming. He states that the intrinsic aspect is the timeless game design, comparable to chess. The extrinsic qualities on the other hand are the experiences, places and people associated with the game, which evoke positive feelings. (McCarthy 2021).

Besides the emotions tied to a certain gameplay, visuals and audio are a significant factor in evoking nostalgia. The pixel graphics and 8-bit tunes of retro classics are easily recognizable and iconic. The advantages of such simplistic visuals and audio are not limited to old games, as proven by well-received modern games like Celeste (Maddy Makes Games, Extremely OK Games) and Stardew Valley (ConcernedApe).
Pixel art or low-poly graphics are a natural choice for indie studios with limited resources. Some modern titles using retro-style polygonal graphics are Crow Country (SFB Games), which embraces N64-style visuals (PICTURE 6.), and Dusk (New Blood Interactive, David Szymanski), inspired by 90's first person shooters.

PICTURE 6. Crow Country (2024), SFB Games.

Essentially, the simplicity is what draws people in. In the modern world, where everything is rapidly becoming more complex and advanced, many players find comfort in the simplicity of old games. A similar response against the digital age has been seen in other industries too, applying more emphasis on slowness, craftmanship and re-using material from the past (Cream Magazine 2024).

3 ISOMETRY

3.1 Projection

Perspective is a method used in art to give an impression of depth and space. It allows an artist to convey their perception and to draw the viewer's attention to a specific element of their piece (Larochelle 2013).
In computer graphics, projection is the technique to simulate perspective on our 2D screens. When it comes to video games, the choice of projection can be based on how realistic the game should feel, or the desire to express a certain artistic perception. The definitions for several different types of projection are a result of centuries of studies in science and arts.

To date, there is a multitude of video game genres, all using various types of projection. Most video games though, fall into one of two categories, **perspective** or **parallel projection.**

Perspective projection makes objects on screen look the way they would in the real world. This type of projection relies on vanishing points, which means that the lines that create the geometry of a scene all vanish into one or more points. Based on the position of the vertices that make up a 3D object, the object will appear large up close, and small as it gets farther from the viewer and closer to the vanishing point (Pezzi 2022). Perspective projection is used in first- and third-person games (PICTURE 7.).

Parallel projection on the other hand is the group of projection types which can often be seen in old games and indie titles. It means that the projection lines run parallel to each other, rather than recede into the distance. The relative dimensions of the objects are preserved, but the projection view lacks depth.

From parallel projections can be drawn two categories, **orthographic** and **oblique projection** (PICTURE 8.). In Orthographic view, projection lines are aligned perpendicularly to the projection plane (McReynolds, Blythe 2005). Orthographic projection will typically display one side of the object completely. In

oblique projection, projection lines do run parallel to each other but aren't perpendicular to the projection plane. An oblique projection distorts proportions to show depth in any one axis.

PICTURE 7. Perspective projection in Spyro the Dragon (1998), Insomniac Games.

PICTURE 8. Comparison between oblique projection in Earthbound (left) and Orthographic projection in Final Fantasy VI (right). Earthbound (1994), Ape Inc., HAL Laboratory. Final Fantasy VI (1994), Square. Edited.

Axonometric projection, which is a type of orthographic projection, means that besides the lines being perpendicular to the plane, they are also foreshortened with varying angles. This method rotates an object to display multiple sides. There are three categories of axonometric projection: **isometric**, **dimetric** and **trimetric**. In this paper we are focusing on isometric projection.

In isometric projection, proportions of an object are preserved, and the angle between each coordinate axis remains at 120 degrees (PICTURE 9.). The y-axis is always vertical, making the x- and y- axis slant 30 degrees from the horizon (McReynolds, Blythe 2005).

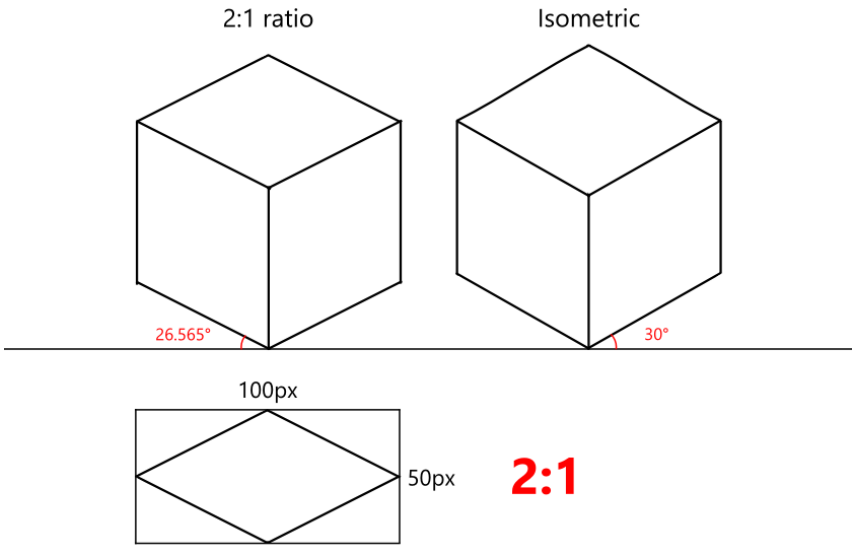
PICTURE 9. Isometric view in Sonic 3D Blast on Sega Saturn (1996), Sega.

In video games, isometric projection became popular between the 80's and 90's, before 3D graphics were common. Early isometric game worlds were built with 2D sprites, carefully positioned and textured to create an isometric projection. Using pre-drawn images freed up computing power for animation (Krikke 2000). Players could quickly grasp the surrounding game world, brought to life by multi-

ple animations playing simultaneously. It was a way to develop impactful experiences in a time where computers had limited power and developing complex games was expensive.

The first popular game to use the isometric projection is likely to be 1982's Zaxxon, Sega's scrolling shooter arcade game. However, the game that is considered to have established the isometric genre is the action-adventure game Knight Lore, released in 1984. This game brought isometric view to home PC's and has been a significant influence on games decades after its release. (Maglio 2023.)

The idea behind using isometric projection in the past was to simulate 3D graphics, and as isometric games consisted entirely of 2D sprites between the 80's and 90's, pixel artists relied on visual tricks to create depth, such as shading, size differences and overlapping. Most isometric games didn't strictly follow the rules of isometry, as they were programmed to use angles between axes that didn't equal to an exact 120 degrees. This was done to achieve a perfect 2:1 pixel ratio (PICTURE 10.), which simplified the work of artists and programmers alike (Pezzi 2022). This projection type, however, is closer to dimetric projection, where only two of the axes have to be equally foreshortened and angled. As proven by the artists that used the popular 2:1 ratio, dimetric projection allows for greater flexibility in adjusting the view by increasing or decreasing the foreshortening along the vertical axis (McReynolds, Blythe 2005).



PICTURE 10. A visual representation of a projection with 2:1 pixel ratio compared to true isometric projection

This means that, from the beginning, most games labelling themselves as isometric, should have been labelled dimetric instead. This fact will be revisited in a moment.

As said, most isometric games of this time were made of 2D sprites entirely, but some games had begun combining isometric 2D tiles with pre-rendered 3D graphics, something that was inspired by Donkey Kong Country. This sparked a significant era of PC games, including Diablo (PICTURE 11.) and Baldur's gate. Of course, technology was evolving rapidly, and after the arrival of dedicated 3D cards for home computers, isometric games with true 3D graphics started to emerge. This changed the isometric genre, as using a fixed-perspective view was no longer the only option for games, and the player suddenly had the ability to not only smoothly zoom in and out of view but even rotate the map around.

PICTURE 11. Diablo (1996), Blizzard North.

This, combined with the fact that even early isometric games were actually dimetric, is why the meaning of isometric projection has blurred, and the term has

evolved into an umbrella term describing any game that has an overhead view appearing “isometric”. Regardless, dimetric projection, including the classic 2:1 pixel ratio, is close to isometric projection, and dimetric games even on the largest platforms are still being published under the isometric label. Because of those reasons, and for the sake of clarity, this thesis will include dimetric games when talking about isometric games.

The next sections will explain more about the effects of an isometric projection and what kind of project it’s suitable for.

3.1.1 Benefits and Downsides

There are several advantages to an isometric projection in games. As stated previously, there was a particular need to use this projection in the past. While 3D graphics were out of reach, developers needed an alternative to pure top-down and side-view projections, which sometimes caused problems in differentiating between elements in the game world. Isometric view was the solution for these problems.

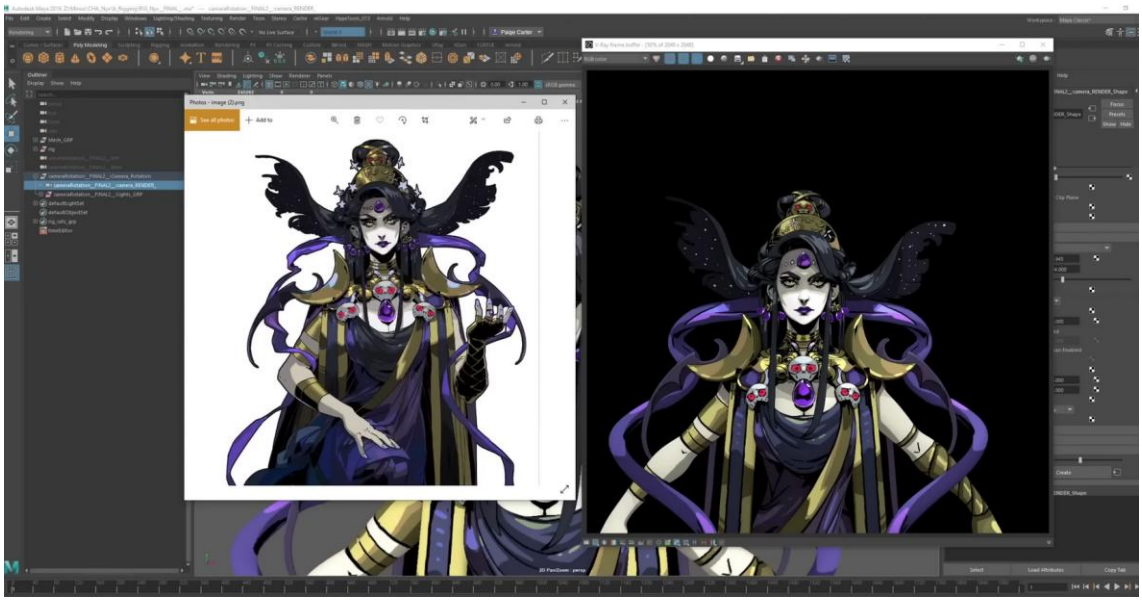
Nowadays, computing power is not nearly as much of an issue compared to the past, but considerable savings still come with isometric projection. For one, parallel projection doesn’t cause perspective scaling or distortion, so the computer isn’t required to calculate certain size changes of game objects (Rocket Brush Studio 2022). This improves the loading time of 3D game environments.

There are various types of isometric games, games that use purely 2D graphics, ones that use 3D graphics entirely, and some titles that employ a mixture of both. For games that use 3D graphics completely, a clear advantage comes from the fact that environmental assets are traditionally only viewed from one angle.
In 3D graphics, the more complex the model, the heavier it is to process. Minimizing vertex amount is therefore usually the most efficient way of optimizing models, but most of the time it means sacrificing quality and detail. Isometric projection makes this easier, since vertices on the back and bottom sides of the

objects can safely be reduced or even removed — they will never be seen by the camera.

An isometric projection can speed up the process of asset creation. It's easier for 2D artists to draw more accurate lines and proportions along the principal axes (Rocket Brush Studio 2022). In 3D modelling, not having to perfect each side of an object is a great advantage and frees up working time for constructing environments faster.

A title that has crafted iconic visuals within an isometric projection while taking advantage of its efficiency and lower cost is Supergiant's Hades. Although Hades appears at least partly 3D, everything from environments to characters and weapons are 2D sprites. The characters are modelled and rendered from 3D (PICTURE 12.) and then generated into animation frames, while the environments are hand-painted with intricate detail (Hades Developer 2019).



PICTURE 12. Nyx's character design for Hades in 2D (left) and 3D (right) during production. (2020, "Inside Hades - 3D Modeling & Rigging" from Supergiant Games' Youtube Channel**)**

A pre-rendered graphics style such as Hades's comes with its own risks. This method may result in display issues, as 2D graphics need to be adjusted and rendered according to the ever-changing display resolutions and aspect ratios of modern devices (Nejam 2022).

Another common problem with isometric 2D games is sorting order. Sorting sprites is the most important method of creating an illusion of depth in 2D games (Sunny Valley Studio n.d). An isometric view poses more challenges in terms of sorting order, since the readability of this projection type allows for more object overlapping compared to other views. Beyond that, many isometric game environments are built with 2D tiles, adding to the burden of sorting.

Looking at isometric projection through the perspective of visual design and readability, it has clear advantages over other parallel projection types. A more overhead view such as the 3/4 view used by many pixel art games, for example older Pokémon games, makes it difficult to distinguish differences in height between objects. On the other hand, the distorted proportions and angles of an oblique projection like cabinet are often visually unclear and confusing, which may affect gameplay. Isometry is a great middle ground of parallel projections, displaying objects in the correct proportions and with more accurate height variation.

To summarize, although isometric projection has its own disadvantages, the positive effects are too great to ignore. Isometric projection can offer a more efficient and sustainable workflow for developers, which is why many modern indie studios opt for isometric graphics. In the end, any chosen projection type depends on a game’s particular setting and game mechanics.

4 ENVIRONMENT DESIGN

4.1 About Environment Design

Environment design in games is a key factor in creating immersive experiences through visual, auditory and interactive elements. In general, it's the whole process of guiding the player with functional and narrative methods (Porokh 2023).
As this paper focuses on the visual aspects of game environments, the next subchapters will study visual environmental elements in games more carefully, to understand the process behind creating them. This will include not only environmental props, but visual effects and overall atmosphere.
Studying these environments requires knowledge in art theory as well as some principal understanding of game design. This base will help us analyse what works well in both isometric and PSX-style graphics.

4.2 Environment Design in Isometric Games

Environmental design depends heavily on the projection type used. Keeping in mind the previous chapter's findings on the positive and negative effects of using an isometric projection, the following section will study an isometric game based on its environmental design. The reference is Hades, a game that has heavily influenced the creation of Project Survival Academia.

4.2.1 Hades

Hades is a roguelike action role-playing game released in 2020 by Supergiant Games. Ever since its release it's taken the industry by storm, receiving endless praise over its narrative, gameplay, music and of course, art style. The story follows Zagreus, prince of the underworld, who's trying to escape Tartarus.

In the game, the player battles their way through randomly generated rooms inhabited by a various selection of enemies. The camera distance from the player

character allows for thorough scale variation of background assets and makes the dungeons and halls seem grand. The distance is mainly important for the action part of the game, as there are usually multiple enemies attacking the player simultaneously from all sides.

Upon closer inspection, it seems Hades uses the same classic angle of projection as older isometric games, to achieve 2:1 pixel ratio.

The environments of Hades are dark and moody dungeons, usually contrasted by bright visual effects (PICTURE 13.). The game favours red, green and purple as its main colours, and the assets are all highly stylized with a cel-shading technique. Cel shading and black outline is an efficient way to give 2D games a distinctive and high-quality look that ages better than most other art styles. Cel shading works well with isometric projection as well, since the outlined edges make it easier to distinguish between overlapping objects.

PICTURE 13. Bright red lava contrasting the otherwise dark environment. Hades (2020), Supergiant Games.

In his GDC talk from 2015, Capy Games' Dan Cox presents the importance of order in environment design; environments need to be arranged in such a way that it prevents players from getting lost, confused or frustrated. According to him, a method of doing this is to apply perceptible patterns inside the space, i.e. the

reuse of colour, material and shape. (Cox 2015). As human beings, we find comfort in patterns, because they help us identify and understand our surroundings. This is also known as Gestalt Theory, which states that the whole is greater than the sum of its parts.

In Hades's case, the rooms must be arranged in a way as to not disrupt the flow of action, while keeping them interesting and varied with random generation. Perceptible patterns are created through decorative patterns, gaps in the floors, and even interactable elements, such as breakable pillars and vases. The intricate hand-painted style of the assets prevents the repeating, pre-made environments from dulling the gameplay experience.

There is another point in Cox's talk where he discusses enrichment, the suggestion of mood and narrative within a space (Cox 2015). Environmental storytelling is a key factor for keeping the player interested and immersed in the game. In Hades, narrative emits from each one of the many environments; various ornaments decorating the floors and walls, the cracking stone architecture, and the candles scattered around with wax dripping over (PICTURE 14.). The gaps in the floors reveal either lava or a mystical cloud river. Looking at these elements, the theme of hell and Greek Gods is certainly consistent in each biome.

PICTURE 14. Detailed environment in Hades (2020), Supergiant Games.

To summarise, the environment design of Hades is consistent, memorable and elevated. It complements the gameplay and keeps the narrative alive.

4.3 Environmental Design in PSX-style Games

The next paragraphs will involve games from the PSX-era as well as modern instalments inspired by the aesthetic.

4.3.1 Resident Evil 2

While the first instalment of the franchise was one of the games to define the genre of survival horror, Resident Evil 2 is considered to have established the series' place as one of the best gaming franchises of all time (West 2024). The story begins with the outbreak of a zombie virus in Raccoon city, trapping the protagonists inside the police department located within an eerie old art museum (PICTURE 15.).

PICTURE 15. The museum in Resident Evil 2 (1998), Capcom.

As the game is known for its claustrophobic level design and uncanny camera angles, the player often ends up in very up-close and personal interactions with the enemy. Designing environments for a horror game as such is quite different

compared to the environmental design of other genres. This topic will be returned to in the coming sections.

The previous subchapter briefly referred to Gestalt Theory, which describes how humans perceive objects and surroundings. It's the foundation of any visual designer's work, because it helps to understand the psychological effects on how the artwork is viewed. One of the principles of Gestalt is defined by Gestalt psychologists as "Good Form". Essentially it means that humans prefer to look at things that are simple and organized, since they are easier to process and therefore feel safer (Bradley 2014).

In most game genres, especially games that have linear gameplay, environments should follow good form. This is a task shared between level designers and environment artists. It's a level designer's job to guide the player without making the game feel too easy or too difficult. But good level design can go completely unnoticed without proper environmental design. If environmental assets lack familiarity or variety, the player will easily become lost or confused. The assets that guide the player's path should be organized in a way that makes sense for the worldbuilding but also in the way that is visually clear and easy to understand.

But level design for horror games serves a different purpose. Horror games are designed to make the player feel alone, powerless, and anxious. Environmental design plays a major part in this. As the protagonists in Resident Evil 2 find their way through the narrow hallways of the museum, it is apparent that clutter is welcome. Various objects are scattered around to signify chaos and unrest (PICTURE 16.), such as books, documents, and wooden planks that used to barre broken windows. Pieces of broken glass or concrete cover the floors and walls are stained with blood. The bigger areas are much more simplistic, but the emptiness and grandeur make the player feel particularly small. Everywhere you go, lighting is scarce, and textures look dirty, even muddy. The environments in Resident Evil 2, or RE2, may not lead the player astray as much as some other horror games, such as the foggy areas of Silent Hill 2, but they are not exactly demonstrating good form either. Together, horror game level design and environment art walk a very thin line between order and chaos.

PICTURE 16. Various props scattered around the room. Resident Evil 2 (1998), Capcom.

Mystery is undoubtedly a part of horror. Fahs explains that fear is not what you see, it's what you don't see (Fahs 2012). It's the subtle suggestion of something looming in the darkness. It's safe to assume that the unusual choice of housing a police station within an old museum was made for the sake of mystery. There is a sense of history, not only in the items that are usually found in museums like paintings, statues and taxidermy, but in the architecture and furnishing. The mystery resides partly in the past of this location; the way this museum feels lived in and used.

There is also mystery in the illusion of space. Looking at the map, the museum seems huge, yet the corridors and rooms are quite tight. The claustrophobia is made worse by the surrounding darkness, leaving the player to wonder when a figure might emerge from the shadows.

4.3.2 Signalis

Released in 2022, Signalis is the debut game of Rose-Engine, a two-person indie studio based in Germany. This survival horror game draws significant inspiration

from Resident Evil for its gameplay and clearly channels the visual style of the PSX-era. In terms of projection, Signalis uses a 3/4 view, which is close to the group of axonometric projections.

In the game, you play as an android named Elster, exploring the abandoned facilities of a remote planet. As the story proceeds, the player ventures deeper below the surface of the planet, facing off with horrendous monsters and experiencing surreal flashbacks and visions.

The world of Signalis is dystopian and isolated. Propaganda posters and surveillance cameras display signs of a totalitarian regime, while the handwritten notes, bloodstained floors and the disorder and malfunction of the facilities describe the distress and struggle of the people. Both Elster's individual mission as well as the mystery of what happened to the local android population is what drives the player to keep going deeper below the depths.

The environments are mostly enshrouded by darkness with weak, foggy lighting illuminating the most important objects, and small, animated lights emanating from different kinds of machines. (PICTURE 17.) Despite appearing 2D, Signalis is a full 3D game. The pixelated graphics are reminiscent of the PSX-era, and there is even an optional CRT-mode included in the game. This paper will next delve into what makes the graphics of Signalis resemble early retro horror games.

In his GDC talk of 2019, Mason Smith, creator of the indie horror game Faith, recalled playing Doom in his childhood and being terrified of something he couldn't identify. He found that, to create that fear of the unknown, he could inspire himself by the limitations of retro hardware. He also achieved juxtaposition by shocking the player with a sudden switch of art style; the rigid, thick pixel graphics were replaced by higher resolution, fluid rotoscope animation. (Smith 2019). This kind of unexpected change has the potential to shake the player's experience entirely and enforce the uneasy feelings of dread and vulnerability.

PICTURE 17. Dark, foggy room with blood-stained floors. Signalis (2022), Rose Engine.

Like Mason Smith, the developers at Rose-Engine inspired themselves by the constraints of retro graphics to define their visual style. Signalis uses a resolution of 640 x 360 pixels to render game scenes, before being upscaled to the target screen resolution. In the 2018 interview developer Alex Zwerger explains that a reason for rendering at a low resolution is to preserve the game's original stylized look from its early development days, when it was built entirely with 2D assets. (Indie Graze 2018). Additionally, the textures and visual effects appear to be drawn in a pixel art style, reinforcing the 2D aesthetic.

The retro style graphics also support the game's world building and characters. Despite the futuristic theme revolving around androids and spaceships, the devices and machinery within the game resemble technology of the 1990's (PICTURE 18.). Furthermore, there is something uncanny in the stark, pixelated edges and textures of otherwise fluid 3D characters. Looking at the game first hand, it's easy to mistake the graphics for pure 2D, which is why the smooth, realistic animation of the pixelated and partly abstract monsters might make them seem even more intimidating. In this way, Signalis' graphical choices achieve a similar sense of Juxtaposition as in the game Faith.

PICTURE 18. Technology that looks outdated. Signalis (2022), Rose Engine.

5 ENVIRONMENT DEVELOPMENT

This chapter will focus on the concepting process, while taking into account findings from previous chapters. The scene that will be designed is a classroom environment for the survival game which in this thesis will be titled “Project Survival Academia”. The project, while on hiatus at the time of publishing this thesis, is developed by a two-person team, and will eventually be released on the Steam platform. In the game, the player ventures through the ruins of the school in search of missing students, meanwhile collecting resources to bring back to the shelter. The shelter is the hub area of the game, where surviving students have set up camp.

The main locations of the game are typical school environments such as classrooms, offices and hallways. The twist to the game is that, as a result of a weather phenomenon, the school ruins are floating in the sky, which makes for an interesting challenge when designing the environments. The prototype scene for this thesis is made within a limited timeframe, and as such any necessary techniques to speed up the process were used.

5.1 Producing Environments

Designing environments for a game requires extensive research and skills in multiple different areas. For concept drawings, this includes the ability to draw in perspective, knowledge in color theory and shading techniques, and research in various topics such as architecture, materials and history. Besides the visual aspects, an environmental artist needs to be aware of the game design, story, and worldbuilding. What the game is supposed to convey and feel like has a big impact on the visual style of the environments.

5.1.1 Concepting

As explained earlier, the locations of Project Survival Academia are ruined, severed areas of a big college campus. Since the story is set in the 1990s, I needed

to research classrooms and school buildings set in that time (PICTURE 19.). More specifically, the school in the game is supposed to be an older building, originating from the beginning of the 19th or 20th century. I also had to take into account the parts of the building's interior or furniture that would realistically have been renewed to fit into the 90's era. The technology should fit the time period, as should any other tools found in the school.

PICTURE 19. Some of the reference images gathered into a moodboard.

As the setting is built around catastrophic events, the atmosphere needs to convey that. The students are stranded in the ruins of their school, trapped in high altitude, with no way to escape. On top of that, they're surrounded by hostile creatures and a raging storm. The colors, lighting and composition of the final scene all need to contribute to this collective feeling of uncertainty and anxiety.

Besides worldbuilding, I needed to be aware of the game design. The environments created through this project wouldn't necessarily become the final game levels, but rather function as concepts indicative of the overall atmosphere and style of the game, as well as a blueprint for how the required game assets could be used.

The game has a resource management system. Various assets around the school can be destroyed and their materials repurposed for the shelter, including

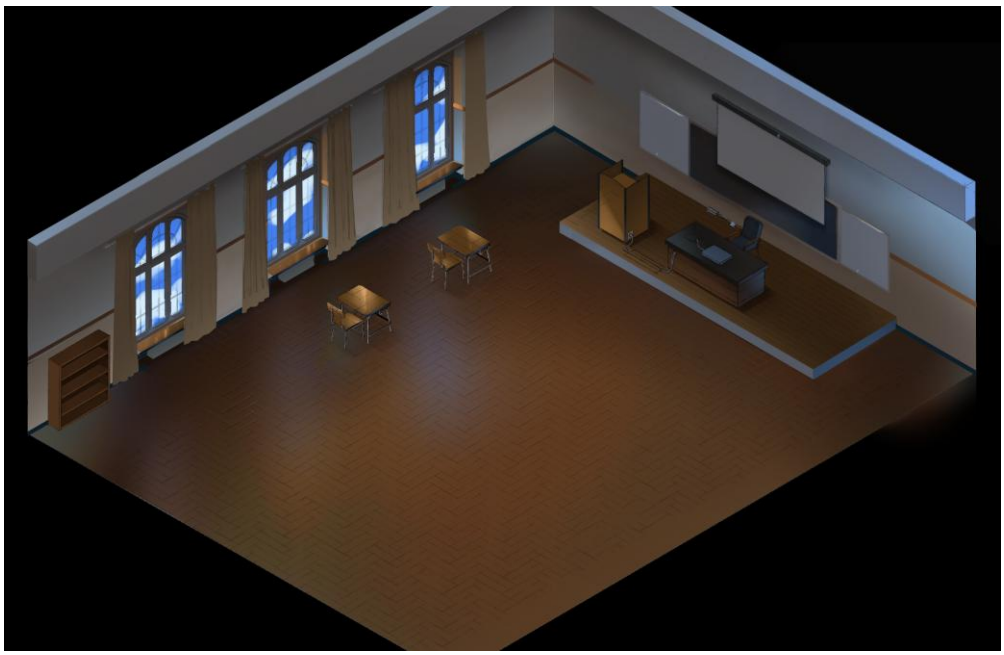
desk, lockers, statues, computers, curtains and so on. Therefore, some interactable items should be demonstrated in the environmental concepts as well. (PICTURE 20.).

Since the game uses an isometric projection, it's important for the concept to be drawn in the right angle. For the concept, I used a mockup blender scene with an isometric projection as a base to draw on top of.

PICTURE 20. Variations of the student desk and chair.

First, a version of the classroom in its normal state, meaning before the catastrophic events, was created (PICTURE 21.). It was important to do this to ease the process of both drawing and modelling the broken and ruptured assets later. Using Pinterest, a lot of references were gathered for the concept art, featuring European and American universities built mostly in the 1700 or 1800's, as well as various classrooms pictured from 1990 and earlier. The aim was to create an aesthetic that felt traditional and upper-class.

A key aspect of the environment was the sense of scale. The school in Project Survival Academia is a prestigious college for the nation's most promising young individuals, which should be reflected in its architecture and interior design. The environment design must strike a balance between making the player feel small within the environment and maintaining the sensible proportions and layout of a classroom.



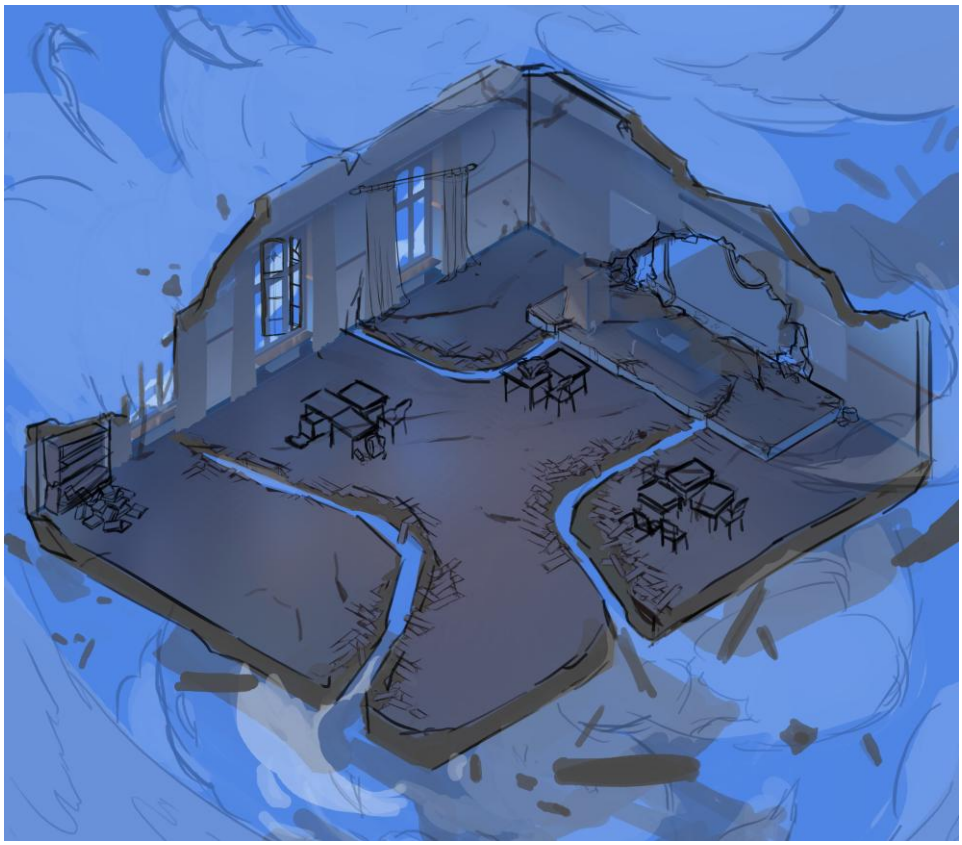
PICTURE 21. Normal state of the classroom.

Making an environment feel grand using isometric projection can be challenging, since the player can't experience the accurate proportions of the environment, like they would in perspective projection. Therefore, the sense of scale of the classroom must be conveyed through, for example window size and wall height, to create an illusion of high ceilings. Long curtains flowing down from the high ceiling to touch the floor can help support this illusion.

A central aspect of narrative revolving elite institutions and school settings in different kinds of media is usually a hierarchy system. In the school of Survival Academia, teachers have absolute authority, and students are expected to respect that.
While this hierarchy is difficult to express in just one room concept, I wanted to include at least a small detail recognizing that. Hence why the front of the classroom, which included teachers' equipment, was slightly elevated to create some symbolic distance between students and teachers. This platform couldn't be too high, as varying terrain is not the best fit for action-focused gameplay.

As said earlier, the impact of the weather phenomenon has transformed the school building into ruins floating in the sky. The storm has ravaged the floors, forcing the player to move by dashing from one floor piece to another, and thus having to plan their route more carefully. The cracks in the floors can't extend into

areas near walls where interactable objects are going to be placed, such as curtains and shelves. The cracked patterns don't have to be realistic, but they do need to be readable in terms of showing the player where they can and cannot go. Game Designer Jesse Schell makes a point about designers and artists being able to control where a person is about to go (Schell, 2008, p.287). This means that by drawing the player's focus to certain parts of the environment, you can affect which route they are going to take. In this case, the cracks in the floor need to guide the player through the classroom in a certain flow. Whatever the order in which the player visits locations of the room, they need to dash over the cracks to gather some of the resources. Besides gameplay, the cracks and other kinds of damage in the environment serve the aesthetic of the game by displaying sky through floors and walls and offering breathing room between various, but monotonous locations of the school. Picture 22 highlights the broken and ruptured parts of the classroom.



PICTURE 22. Initial sketch of the broken version of the classroom.

As there needed to be an exit through the north facing part of the room, a big gaping hole was made into the wall, taking advantage of the theme of ruined

environment. Wires were hung from inside of the broken walls, and slabs of concrete were used to fill up empty corners of the room. Referring to Cox's point about perceptible patterns, these kinds of re-used details help arrange environments in a consistent manner to maintain composition and visual theme, but also create familiarity and routine for the player travelling between different kinds of locations.

Students' desks and chairs were assembled into small clusters around the room, to make it clear that they are interactable assets. From the player's perspective, it's more satisfying to break groups of assets at once, rather than individual objects. This is because the sound effects and visual effects from interacting with many objects at once as opposed to a single object can create sort of a domino effect of sounds and particle effects

As seen in Hades, small details in the environment contribute to narrative enrichment. In Project Survival Academia, the player scavenges for resources around the school to maintain the shelter. Scattered around the clusters of desks and chairs, backpacks and smaller items like shoes and flashlights can be found. This kind of memorabilia is meant to paint a backstory of lost students trying to survive through the events, possibly taking shelter under the desks.

All stages of the concepting phase were shared and discussed with the project's game designer.

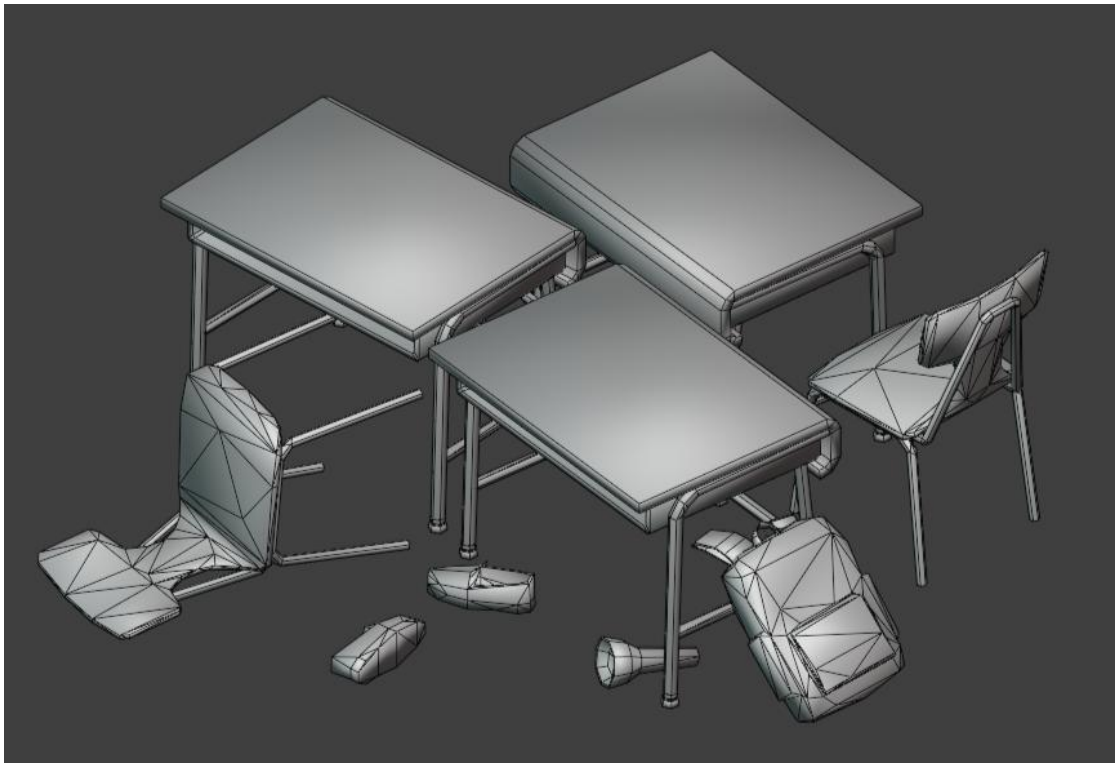
5.1.2 Modelling

Before starting to model an isometric environment, the correct view needs to be set up in the modelling software. An isometric projection in Blender can be achieved by rotating the camera 54.7 degrees in the x-axis and switching the camera mode to orthographic.
When modelling isometric assets, it's important to keep a fixed isometric view of the scene on one side of the viewport during the whole process. Additionally, concept art can be set up as a reference image, angled perpendicular to the cam-

era. The modelling process began by blocking out the main shapes of the classroom, walls, floors and windows. Once the base shapes had been laid out approximately in the right places, I began to define some of the smaller props as they were the easiest to model.

The one important thing to keep in mind when modelling assets in the style of PSX is the polygon count, as models in PlayStation games were always low poly. Other than that, there is no clear blueprint on how to model assets the way 3D artists did back in the day, but there are a few free resources online that provide downloadable assets from various popular games, old and new. One of these is The Models Resource, which was used for this project as a reference to analyze the topology and textures of game assets from early PlayStation and PC games.

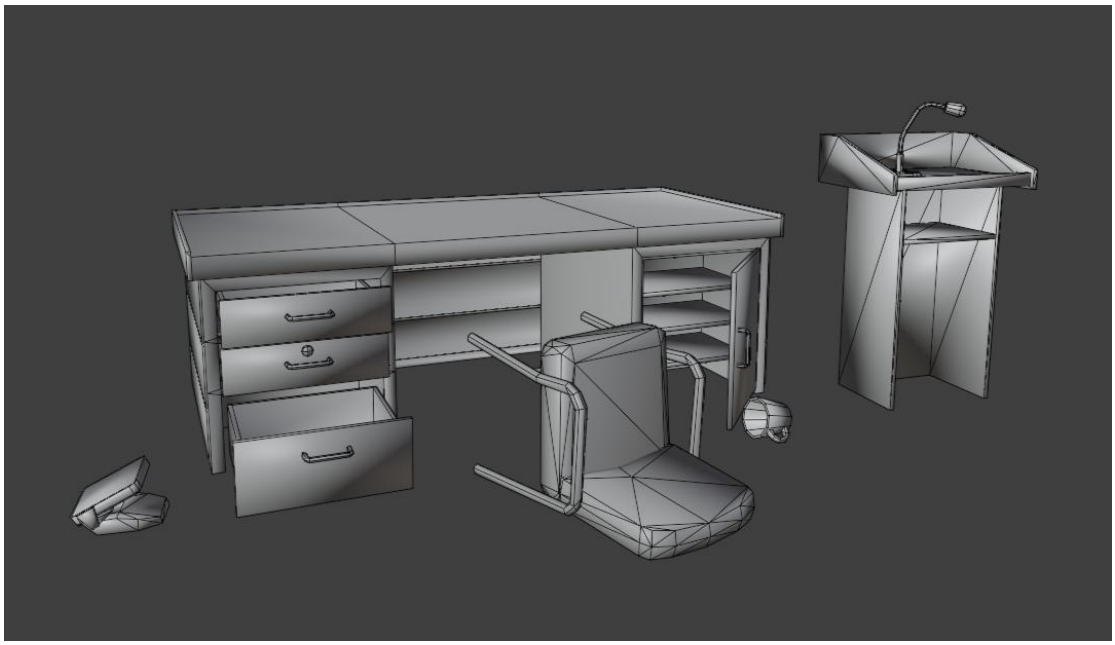
After some of the smaller assets were finished, such as chairs, desks (PICTURE 23; PICTURE 24.) and a bookshelf, it was time to model the floor.



PICTURE 23. Students' desks.

As mentioned earlier, the process of creating such a scene requires consideration of the game design, during both concepting and modelling phases. As concept

drawings often don’t perfectly portray accurate proportions and angles of the environment, it was important to reconsider the game design when separating the floor into regions. Since the player is able to dash over cracks in the floor, there needs to be enough space on each floor piece to run around and fight enemies if necessary. There also needs to be space for destroyable assets, for example students’ desks and book piles. A basic character model made earlier was used to help maintain correct proportions.



PICTURE 24. Teacher’s equipment.

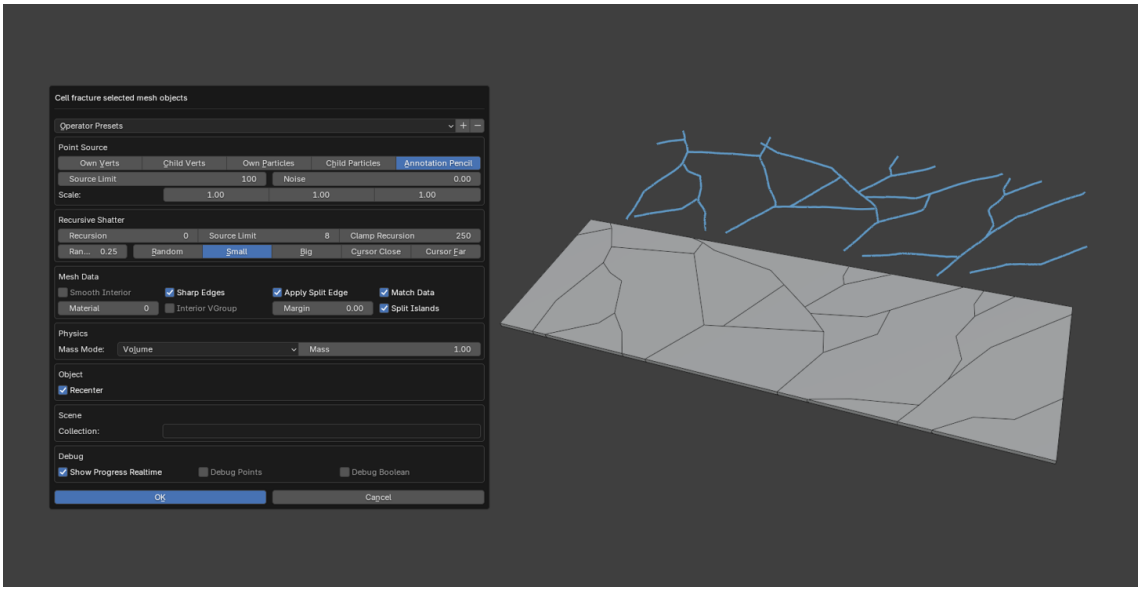
In later development, the game will have many kinds of rooms, and as this was the first prototype for one of those rooms and the game design was not yet finished, I decided that while keeping the gameplay in mind, I would primarily focus on the visual look and composition of the floor segments.

For separating the floor into pieces, I experimented with Blender’s cell fracture addon using the annotation option (PICTURE 25.). I quickly realized that while this addon would surely be useful with other kinds of assets in the game, creating the floor pieces would require a more exact tool. Blender’s bisect and knife tools worked well for that purpose.

The topology of the broken edges of the concrete pieces was defined using the displacement modifier and a custom concrete texture. The pieces were subdivided a bit in order for the modifier to have more to work with. After applying the

modifier, the model was simplified using the decimate modifier, to achieve lower topology. The same method was used for the walls.

Next were the hardwood floors. Modelling a large object consisting of small repetitive objects and patterns is never an easy task, especially when it comes to making it look natural and varied.

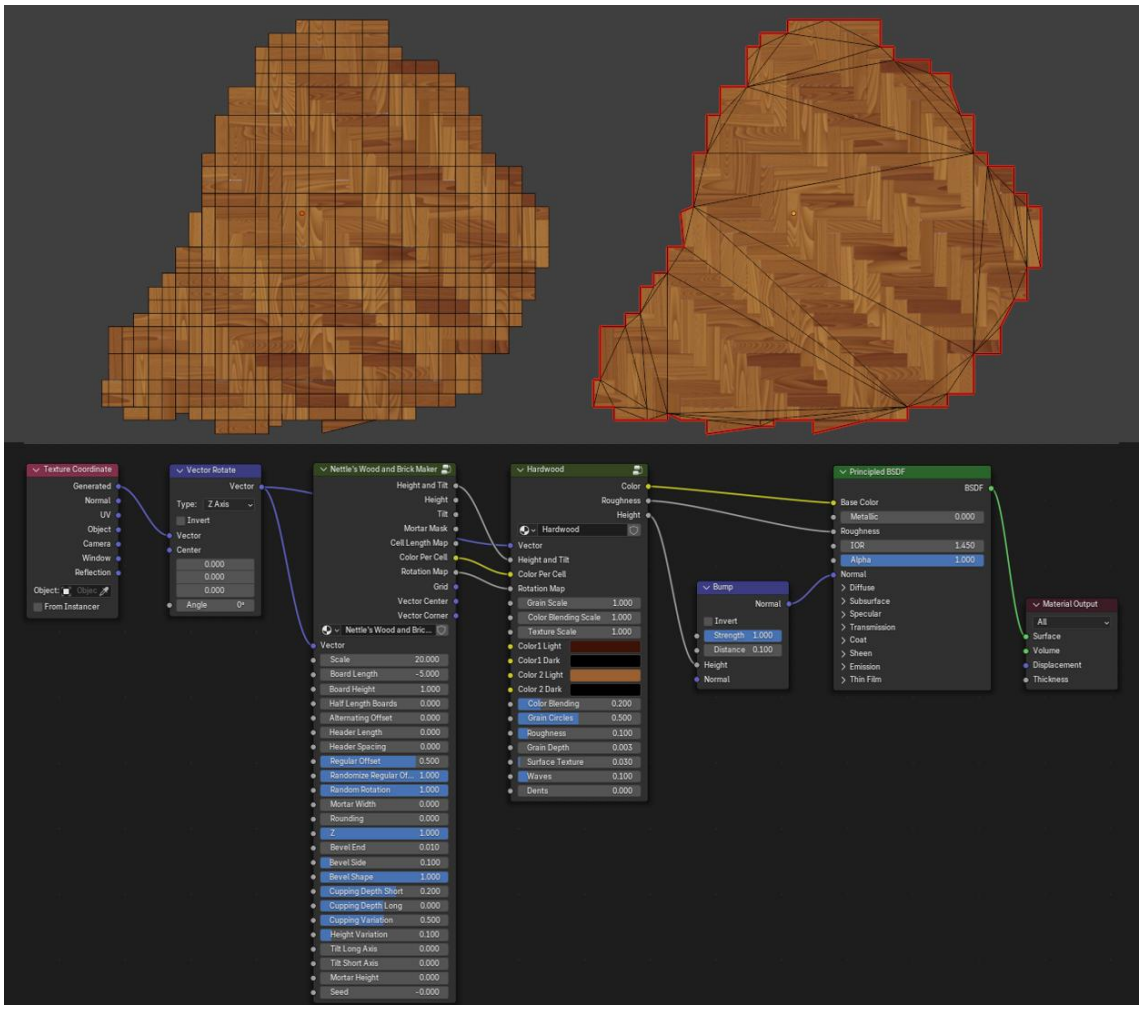


(PICTURE 25.) Blender's Cell Fracture addon using the annotation tool.

I had plans to use Blender's array modifier to create the floor planks. I came a long way in this process until I understood the difficulty of texturing such a pattern. Although I utilized the UV offset option in the modifier, and experimented with free textures found online, the workflow was simply too slow and tedious. So, I opted for a more complex procedural texture, where the planks would be generated in in the correct positions and textured, all in the same shader. For this I used Youtuber Nettle Ada's procedural herringbone floor and adjusted it to my liking. In the future I would have to recreate this from scratch, but considering the experimental nature and limited timeline of this project, I decided to utilize any free resources available. The procedural material was applied onto a grid, from which the shapes of the wooden flooring were cut out (PICTURE 26.). The final steps required some more detailed work by hand, i.e. separating individual pieces from the grid to create loose planks and randomizing their positions and rotations. Some of those

were broken with the help of the Boolean modifier, followed by the decimate modifier to decrease the number of vertices, giving the planks a splintered look. The same technique was used on other wooden assets.

The wooden chair railing was an easy way to make the classroom look more traditional. The railing was made with curves, using the bevel setting in the curve properties.



(PICTURE 26.) Floor piece cut out from a grid and decimated into a low poly model. Nettle Ada's procedural herringbone floor material below.

The process of creating low poly assets typically also includes creating normal maps. This would have been the case if the graphics of Project Survival Academia weren't inspired by PSX-style visuals.

It's unclear whether the PS 1 was able to process bump maps or not. It's assumed however that no PlayStation game, or even early PlayStation 2 games actually

used bump maps, as there wasn't a lot of expertise on the topic, as well as the heavy toll on performance.
For Survival Academia, it was decided to avoid using normal maps, partly to maintain a level of authenticity. The more weighing reason was that, considering the distance of the camera and more importantly, the pixelation effect that would later be applied to the scene, creating normal maps for the assets would've been a waste of resources. Instead, the game would rely solely on diffuse textures to provide the needed detail in assets.

One aspect to look out for when modelling isometric assets is that models are not going to look good from all angles, particularly when objects are facing the camera. The wrong rotation angle in especially long, narrow objects like the floor planks will end up looking disproportional.
Another thing to remember during the modelling process, when modelling for a game using a pixelation effect, is the level of detail. In such a game, most smaller assets are going to be completely unrecognizable. In the end, some of the props modelled for this project are too small to be seen in the full game view, but they will be used for the management system later in development, as the player will be able to view and rotate collected items in a specific 3D view in the user interface.

5.1.3 Texturing

The goal of the texturing process was to crunch down higher resolution textures into a low resolution mimicking the PSX-era. The PlayStation hardware supported up to 256 by 256 pixels, so the average texture size for models was generally 128x128 or 256x256 pixels (Colson 2021) (PICTURE 27.).
Youtuber Kid Leaves Stoop shares in his video about how most PSX and N64-era games used real images for their textures acquired by texture CDs, texture libraries stored on CD-ROMs, which were the standard texturing method of the 1990's 3D modelling industry (Kid Leaves Stoop 2021).
In this project, multiple techniques were used for the texturing process. Staying true to the era, some textures used were real photos found from online texture libraries, like Texture Ninja and Duion. The textures for the books in the class

were books found at home, photographed by the author (PICTURE 28.) and edited in Krita painting software.

PICTURE 27. Comparison between high resolution textures (top) and the same textures scaled down to 128x128 pixels (bottom).

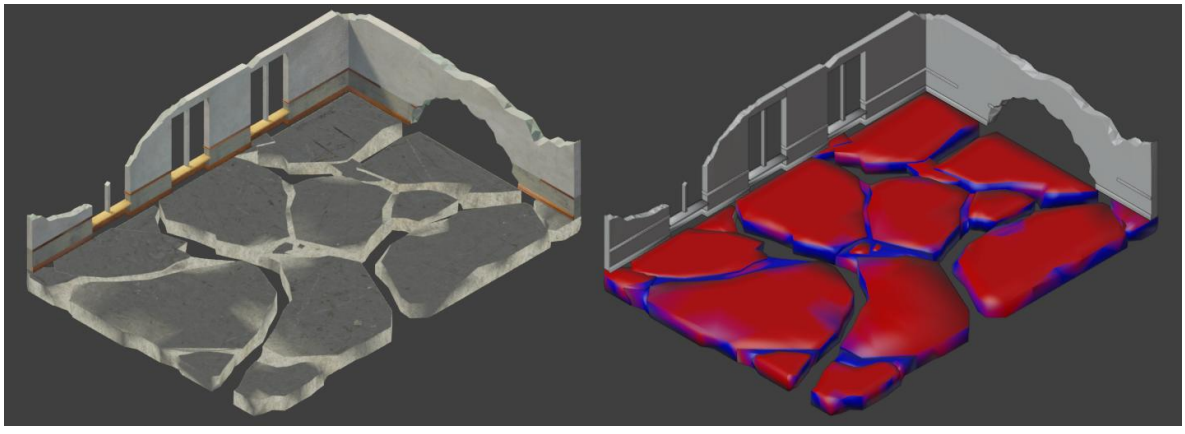
PICTURE 28. Some of the book textures used in the project.

Some of the image textures needed to have their colors adjusted, using Krita. This was done to maintain a muted color palette for the environment. The aim was to not stick to any pre-made palette, but to explore using image textures and finding a good balance of colors by adjusting hue and saturation.

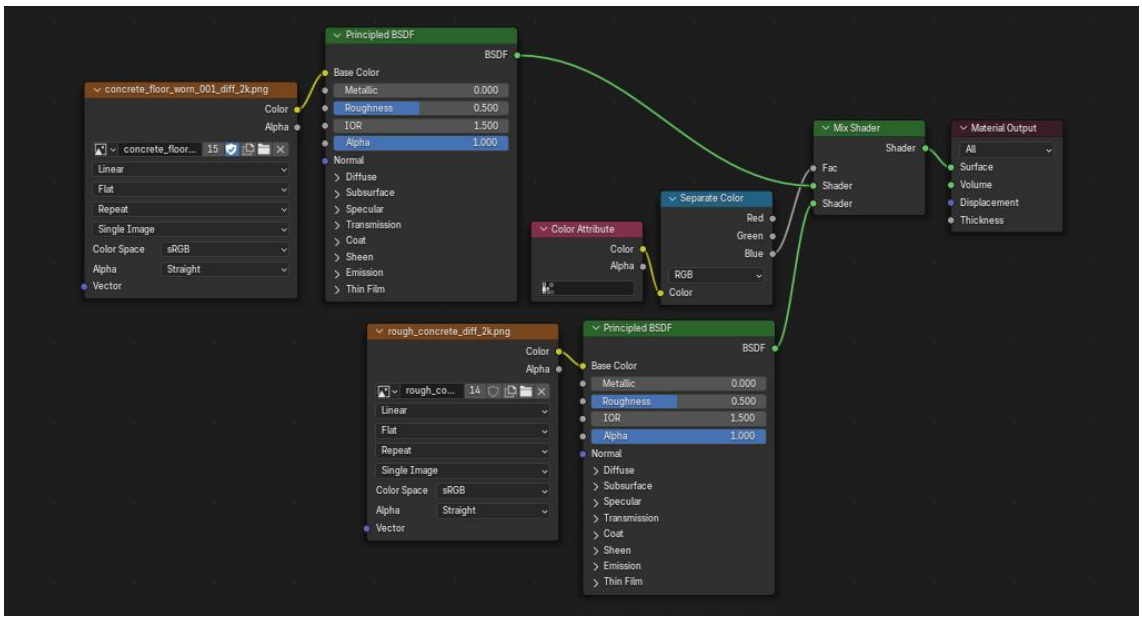
Whenever the image textures didn't fulfill the project's needs, textures were generated procedurally with Blender, or hand painted using Krita.
As stated earlier, the herringbone hardwood floor was one of the materials created procedurally.

Each texture had a weathered counterpart, for example images of materials like fiberboard or concrete rubble. These textures were applied to the chipped and cracked details of the models. To make the materials look even more worn and dirtier, some of the textures were blended together using Blender's vertex painting mode. This can be seen for example in the concrete material (PICTURE 29; PICTURE 30.).

Along with decreasing texture resolution, posterizing textures is recommended to mimic the dithering effect of the PlayStation, which can be done with photo editing software like Adobe Photoshop or Krita. Alternatively, dithering can be produced through post-processing in engine, which was the plan for this project.



PICTURE 29. Vertex paint on the concrete material.



PICTURE 30. Blending the two concrete textures.

6 GODOT ENGINE

Given that Godot is a relatively young engine competing against AAA-engines like Unity and Unreal Engine, it has evolved rapidly, gaining significant popularity over recent years. Much of this success is due to it being a free, open-source engine, which anyone can contribute to. For a long time, Godot was largely regarded as a 2D engine, lacking the support and rendering capabilities of other 3D game engines. But with the release of Godot 3.0 and onward, it has been consistently improving its 3D features including upgrading to PBR renderers, offering built-in principled BSDF, and providing real-time GI workflow (Linietsky 2018). Some well-known games that are made with Godot are The Case of the Golden Idol (Color Gray Games, 2022), Cassette Beasts (Bytten Studio, 2023) and Buckshot Roulette (Blobfish, Seaven Studio 2022) (PICTURE 31.).

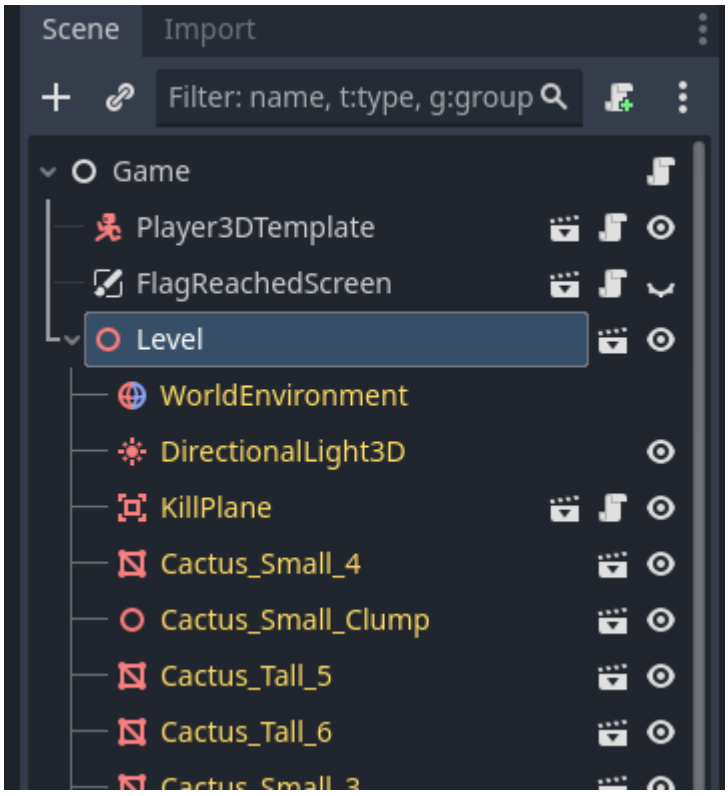
PICTURE 31. Buckshot Roulette (2023), Mike Klubnika.

6.1 Working in Godot Engine

Godot uses a node-based interface, providing a smooth workflow. Nodes make it easy especially for new users to find the related tools to each object in scene. A scene is essentially nodes organized into a tree, and once saved, it can be

added as a child to an existing node of another scene (PICTURE 32.). This is similar to Unity's prefabs.

The recommended format for importing models to Godot is glTF, as it is integrated in the editor and better supported in open-source 3D software like Blender. After importing the classroom models and creating a scene, the first priority was to configure camera view.



PICTURE 32. Godot's nodes, with the scene "Level" added as a child of the scene "Game". The color yellow signifies that the children of "Level" have been made editable within "Game", whereas by default they would be hidden.

6.1.1 Projection and Resolution

Chapter 3.1 discusses the inconsistencies regarding projection angles of isometric games. As Hades was visually the greatest inspiration for Project Survival Academia, the team debated between using the same dimetric projection or using a true isometric projection. In the end, changing between these projections wouldn't cause any trouble this early in production, so the team opted for true

isometric projection for now, while remaining open to trying dimetric projection later.
To achieve true isometric perspective in Godot, the camera's projection type first needs to be changed to orthographic. The camera should then be rotated 45 degrees in the Y-axis, and -35.3 degrees in the X-axis. The rotation value of the X-axis is the result of the formula atan(1/scrt(2)), used in programming to calculate the graphical projection values for an isometric projection.

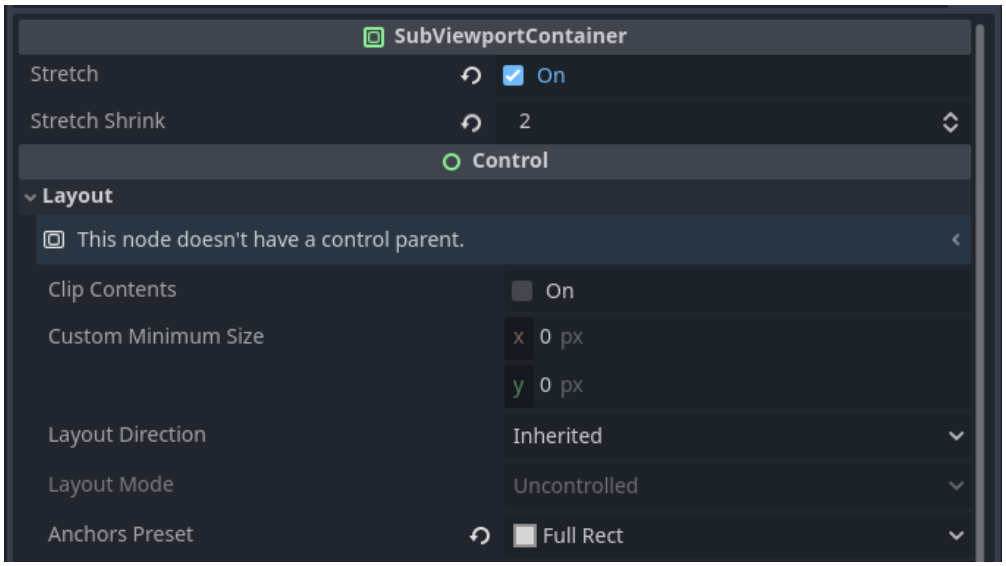
It seems necessary to point out the difference in rotation values between Godot and Blender. To have an isometric view in Blender, the camera rotation should be set to 54.7 degrees in the x-axis. I couldn't find a definitive answer as to why this is, but assumably the two softwares use different measurements.

After the first necessary steps to achieve isometric view, any additional changes to the camera properties depend on the game's needs. The camera's position was adjusted until it was roughly centred to the scene and the orthogonal size value was increased, as the player needs to have sufficient field of view to move around in the game.

The easiest way to achieve something close to the crisp pixelated look of PSX-graphics is to decrease the game's resolution. The PlayStation supported resolutions from 256x224 to 640x240 pixels (progressive) and 256x448 to 640x480 (interlaced).
In Godot, viewport resolution can be changed either via the project settings or by using Subviewports. According to Godot's documentation, a subviewport is an interface to the game that doesn't draw directly onto the screen (Godot Docs). Subviewports can therefore be used to display contents independently from the game scene, such as UI, or as a render pass to combine the scene with post processing effects, among other things. The main reason for using the subviewport in this project was to be able to easier change the resolution and see the pixelation effect without having to turn on Play mode. The other reason was to be able to use post-processing effects if needed.

After adding the subviewport parented by a SubviewportContainer node into the scene, there are a few settings to change. The subviewportcontainer's layout

needs to be set to ‘Full Rect’, and the stretch mode turned on (PICTURE 33.). This ensures that the subviewportcontainer will use up the whole viewport area, and that the subviewport will automatically scale along with it. The ‘Stretch shrink’ value can then be adjusted to alter the resolution while preserving the viewport’s scale. With the container’s size set to 1280x720 pixels, and setting the stretch shrink to 2, the subviewport will be rendered at 640x360 pixels, which was the right amount of pixelation to resemble PSX graphics and suit this project.



PICTURE 33. Inspector view of the SubviewportContainer.

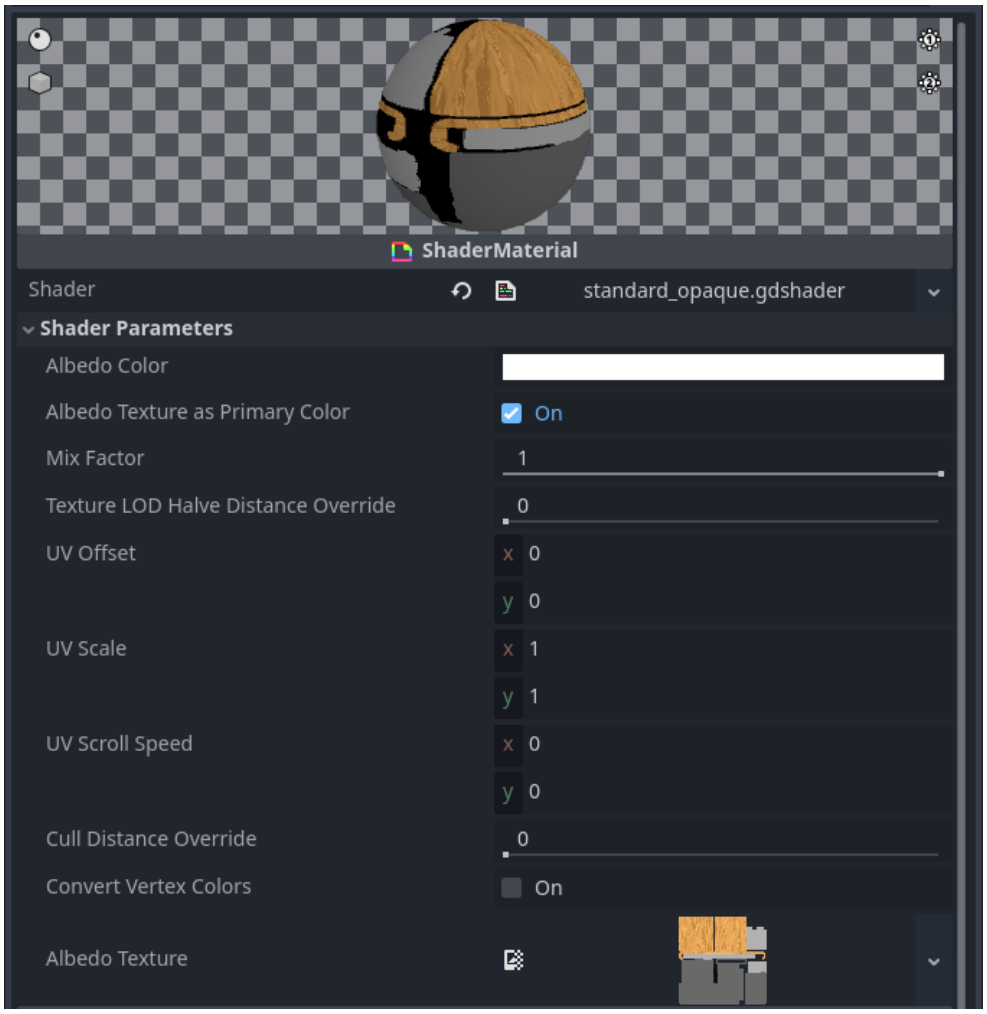
Some other recommended configurations for PSX-graphics are, in project settings, turning on Vertex Shading via the Rendering tab, and from the Texture 2D section under Import Defaults, turning off Mipmaps and disabling 3D detection.

6.1.2 Retro Shader

As discussed in earlier chapters, there are distinct visual effects that contribute to the iconic look of PlayStation games. All of those effects were not necessary to implement into this project, but some preferred to be replicated were, the low resolution or pixelation, vertex snapping or texture jittering, and vertex lighting. There is a big online community of developers making PSX-inspired games, so finding the needed resources to produce these components was not an issue.
A particular shader was found that already included all aforementioned effects, namely a large repository of shaders and tutorials called Ultimate Retro Shader,

compiled by Github user Zorochase. To utilize these shaders, they only have to be applied into a material, along with a texture, creating a custom shader material (PICTURE 34.). The result was very efficient PSX-style materials, with jittering polygons, and using per-vertex shading (PICTURE 35.). The texture filter option was also set to nearest through the script, which is an important setting when creating PSX-style textures, as the PlayStation didn't use texture filtering. Texture filtering determines the color of a pixel based on the texture mapped onto a model, while also affecting texture quality. Without the shader, the texture filtering setting could have been changed through individual material properties.

Included in this shader bundle was also a dithering canvas_item shader. The other shaders are spatial shaders, applied to materials, while canvas_item shaders are applied to the SubviewportContainer. Using the shader parameters, the amount of dithering can be adjusted (PICTURE 36.).



PICTURE 34. Custom shader material using the standard opaque shader included in the Ultimate Retro Shader.

PICTURE 35. Custom shader materials applied to environment.

PICTURE 36. Scene with dithering applied, on the left color depth is set to a value of 16, on the right to a value of 8, which results in heavier posterizing.

6.1.3 Lighting

With the limitations of the PlayStation hardware, lighting was another area of development where developers had to cut corners. As the console wasn't capable of running a scene with multiple lights at once, artists had to come up with new techniques to replicate realistic lighting. The most common of these was vertex lighting. This method didn't use real lighting, but rather colour painted on top of pre-existing vertices. Vertex lighting mimicked basic lighting effects; lit areas were

painted with a bright colour, including reflections, and areas outside of the light were painted with a darker colour.

Another technique was lightmap baking, utilized especially in some of the later games of the PlayStation. This is still a commonly used technique in modern games. Baking lights into textures was a way to achieve a photorealistic look while effectively saving on precious processing power. The downside of baked lighting is that it cannot be used on any moving or otherwise reactive lights, such as flashlights or muzzle flash.

Developers combined these two techniques to create the illusion of cinematic lighting, shadows and reflections. Some notable PlayStation games that effectively used these lighting methods to establish memorable atmospheres are Alien Resurrection, Metal Gear Solid and Vagrant Story (PICTURE 37.).

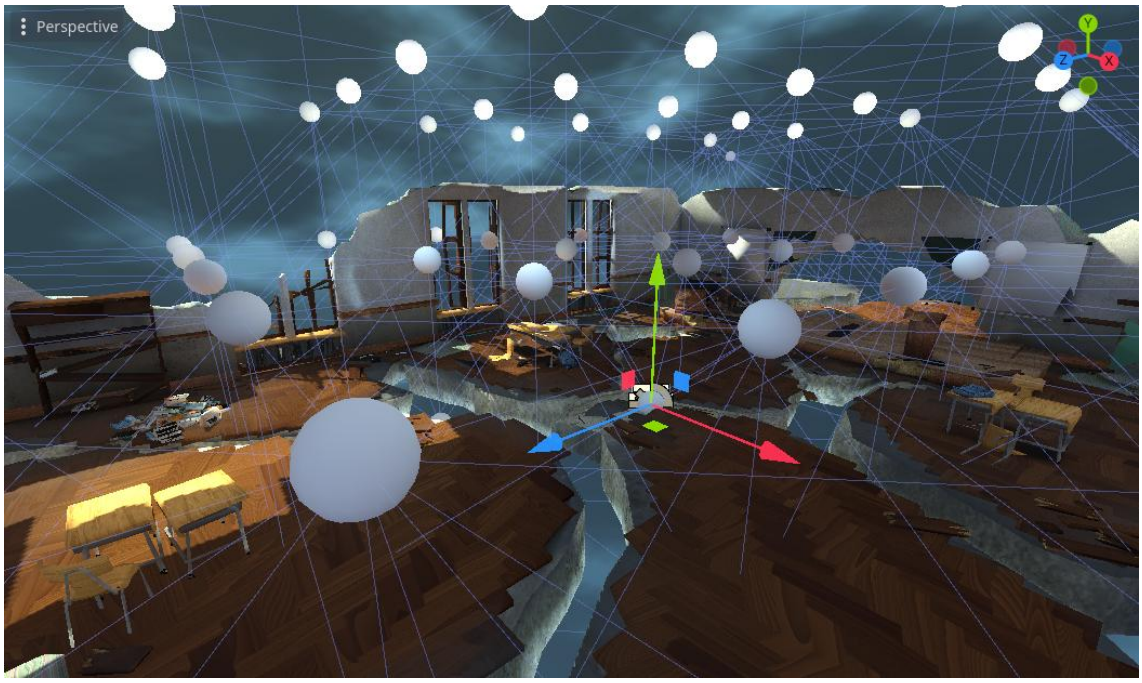
PICTURE 37. Alien Resurrection (2000), Argonaut Games.

Creating vertex lighting in Godot is very easy to do, thanks to a single setting in the material properties, allowing to switch from per-pixel shading to per-vertex shading. Using the custom shader materials, this step was unnecessary, as calculations for vertex lighting were already made through script.

In order to get even closer to PSX-era lighting, lighting should be baked directly into the vertex colors of a mesh, before importing the mesh into the game engine.

However, to save time in this project, lightmaps were created in-engine. This meant that shadows needed to be enabled, which through the shader script were disabled by default to stay true to PSX- and N64-era visuals, as consoles of that era didn't support real-time lighting and shadows.

To bake lights in Godot, after setting up the light sources, a LightmapGI node needs to be added into the scene. Adding a LightmapProbe node along with it is optional, but recommended to improve lighting details in areas that are often affected by dynamic objects (Godot Docs) (PICTURE 38.).



PICTURE 38. Godot's LightmapGI.

After adding these components, the most important step is to unwrap meshes into lightmaps. This setting can be found in the 'Mesh' dropdown menu above the viewport. Regarding the LightmapGI properties, the most significant ones in terms of the PSX-look are the 'quality' and 'max texture size' – settings. The quality can be set to low or medium and max texture size should be turned down as low as possible since the goal is low resolution textures and lighting.

After baking the lightmap for the scene, the lights were disabled, and particle effects and post processing fog were added. The result can be viewed below (PICTURE 39).

PICTURE 39. Prototype scene with baked lighting.

7 DISCUSSION

The outcome of the project is a visual style reminiscent of the PlayStation-era. While the low poly assets combined with a pixelation effect already go a long way, the convincing result is most likely thanks to the dithering and polygon jittering effects.
What could be improved in the future is the sharpness of the image, as a typical feature of PSX-era visuals is a certain amount of blur.

Creating a retro-inspired look in Godot didn't require too much effort. Regardless of the shaders being used in this project, Godot's standard project and material settings make it easy for anyone to experiment with PSX- or N64-style graphics. The dithering effect from the shader can quite easily be replaced by posterizing textures before importing them into the project.

The initial setup of an isometric projection is easy in modern engines, but using an orthographic camera in the long run can cause some issues, as the setting isn't compatible with some features of Godot. Therefore, isometric projection might require some workarounds through programming, depending on the project's needs.
Using an isometric projection does save time during the modelling and texturing process, as the level of detail can be significantly reduced from the back sides of certain assets. This effect will surely be more noticeable as the project expands and more assets are made. It's also easier to achieve a pleasing composition using an isometric projection, compared to for example perspective projection.

Creating isometric visuals inspired by retro graphics is a great option for small teams or developers who are aiming to minimize the costs of production. As old classic games are being remastered for modern consoles, and retro graphics are resurfacing within the indie community, it's as good a time as any to explore retro-style aesthetics.

Regarding engines, Godot is an efficient platform suitable for various kinds of projects. It’s beginner-friendly, versatile, and free of charge, which makes it a good choice for indie developers and bigger projects as well.

REFERENCES

Doschinescu, Michaela. 2024. The Nostalgia Industry: Why We Feel the Need to Relieve the Past. Published on 03.12.2024. Read on 23.03.2025. https://traileoni.it/2024/12/the-nostalgia-industry-why-we-feel-the-need-to-relive-the-past/

McFerran, Damien. 2015. The PlayStation Book. BourneMouth: Image Publishing.

Sponsel, Sebastian. 2010. History of: Virtua Fighter. SEGA-16. Published on 03.02.2010. Read on 30.08.2024. https://www.sega-16.com/2010/02/history-of-virtua-fighter/

Campbell, Colin. 2012. How SEGA Saved PlayStation. IGN. Published on 06.09.2012. Read on 29.08.2024. https://www.ign.com/articles/2012/09/06/how-sega-saved-playstation

Pezzi, Gustavo. 2024. How PlayStation Graphics & Visual Artefacts Work. Pikuma. Published on 08.08.2024. Read on 08.10.2024. https://pikuma.com/blog/how-to-make-ps1-graphics

PC Plus. 2010. The Evolution of 3D Games. Published on 11.07.2010. Read on 08.10.2024. https://www.techradar.com/news/gaming/the-evolution-of-3d-games-700995/2

Copetti, Rodrigo. 2019. PlayStation Architecture - A Practical Analysis. Copetti.org. Published on 08.08.2019. Read on 11.05.2025. https://www.copetti.org/writings/consoles/playstation/

Peddie, Jon. 2020. Famous Graphics Chips: Nintendo 64. IEEE Computer Society. Published on 29.02.2020. Read on 09.10.2024. https://www.computer.org/publications/tech-news/nintendo-64

C, Steve. 2025. Nintendo 64 vs. Sony PlayStation: The 3D Console Wars. Retrolize. Published on 26.01.2025. Read on 11.05.2025. https://retrolize.co.uk/blogs/retro-blog/nintendo-64-vs-sony-playstation-the-3d-console-wars

User PolyHertz. 2021. Retro 3D Art FAQ – Everything you need to know to create PS1/N64/Dreamcast/etc. 3D art. Posted in the technical talk category of Polycount's forum. Posted in April 2021. Read on 06.05.2025. https://polycount.com/discussion/226167/retro-3d-art-faq-everything-you-need-to-know-to-create-ps1-n64-dreamcast-etc-3d-art

Carlson, Matthew. 2020. Nintendo 64: Architecture and History. Medium. Published on 07.09.2020. Read on 13.05.2025. https://medium.com/@matthew-fcarlson/nintendo-64-architecture-and-history-8a01cf503a6a

Sirani, Jordan. 2024. The Best-Selling Video Game Consoles of All Time. IGN Nordic. Published on 13.05.2024. Read on 15.09.2024. https://nordic.ign.com/nintendo-switch-1/62861/news/where-switch-ps5-rank-among-the-best-selling-video-game-consoles-of-all-time

Madigan, Jamie. 2013. The Psychology of Video Game Nostalgia. Published on 06.11.2013. Read on 23.03.2025. https://www.psychologyofgames.com/2013/11/the-psychology-of-video-game-nostalgia/

McCarthy, Anne. 2021. Why Retro-Looking Games Get So Much Love. Wired. Published on 21.03.2021. Read on 28.10.2024. https://www.wired.com/story/why-retro-looking-games-get-so-much-love/

Cream Magazine. 2024. This Is Why Retro Games Are Trending Again in the Digital Age. Published on 18.09.2024. Read on 28.10.2024. https://creammagazine.com/2024/09/18/this-is-why-retro-games-are-trending-again-in-the-digital-age/

Larochelle, Audrey. 2013. A New Angle on Parallel Languages: The Contribution of Visual Arts to a Vocabulary of Graphical Projection in Video Games. GAME, The Italian Journal of Game Studies 2 (1), 33.

Pezzi, Gustavo. 2022. Isometric Projection in Game Development. Pikuma. Published on 26.06.2022. Read on 22.10.2024. https://pikuma.com/blog/isometric-projection-in-games

McReynolds, T. & Blythe, D. 2005. Advanced Graphics Programming Using OpenGL. 1st edition. Morgan Kaufmann.

Krikke, J. 2000. Axonometry: A Matter of Perspective. IEEE Computer Graphics and Applications 20 (4), 10.

Maglio, Alyx. 2023. A History of Isometric RPGs. CBR. Published on 19.06.2023. Read on 15.09.2024. https://www.cbr.com/isometric-rpg-history/

Rocket Brush Studio. 2022. Isometric Video Games and How Isometry Benefits Game Developers. Published on 18.03.2022. Read on 24.10.2024. https://rocketbrush.com/blog/isometric-games-how-isometry-benefits-game-developers

User kid_zomb, Hades developer, posted in the Discussion board of Hades's Steam page. Posted on 11.12.2019. Read on 28.08.2024. https://steamcommunity.com/app/1145360/discussions/0/1738883810796606862/

Nejam, Abderrahemane. 2022. Why Isometric Graphics are Crucial for Video Games. CBR. Published on 02.10.2022. Read on 24.10.2024. https://www.cbr.com/isometric-graphics-games-hades-league-diablo/

Sunny Valley Studio, n.d. How to Sort Sprites by Y-axis in Unity 2D. Read on 25.10.2024. https://www.sunnyvalleystudio.com/blog/how-to-sort-sprites-in-unity

Porokh, Alena. 2023. Introduction to Game Design Environment and Initiating Your Journey. Published on 20.12.2023. Read on 31.10.2024. https://kevuru-games.com/blog/understanding-and-initiating-game-environment-design/

Cox, Dan. GDC Vault. 2015. Interior Design and Environment Art: Mastering Space, Mastering Place. GDC. Watched on 04.11.2024. https://gdcvault.com/play/1022089/Interior-Design-and-Environment-Art

West, Josh. 2025. The 10 best Resident Evil games of all time. Read on 04.11.2024. https://www.gamesradar.com/best-resident-evil-games/

Bradley, Steven. 2014. Design Principles: Visual Perception and the Principles of Gestalt. Smashing Magazine. Published on 29.03.2014. Read on 6.11.2024. https://www.smashingmagazine.com/2014/03/design-principles-visual-perception-and-the-principles-of-gestalt/

Smith, Mason. GDC Vault. 2019. The Art of 'FAITH': Horror at 192 x 160 Pixels. Watched on 11.11.2024. https://gdcvault.com/play/1026045/The-Art-of-FAITH-Horror

Indie Graze. 2018. Interview: Signalis' Alez Zwerger. Published on 18.05.2018. Read on 05.11.2024. https://www.indiegraze.com/2018/05/18/interview-signalis-alex-zwerger/

Schell, Jesse. 2008. The art of Game Design: A Book of Lenses. 3rd Edition. Elsevier/Morgan Kaufmann.

Colson, David. 2021. Building a PS1 Style Retro 3D Renderer. Published on 30.11.2021. Read on 28.08.2024. https://www.david-colson.com/2021/11/30/ps1-style-renderer.html

Kid Leaves Stoop. Youtube Channel. 2021. Published on 19.03.2021. Watched on 25.02.2024. https://www.youtube.com/watch?v=bsCN0Yx2Vbs&ab_channel=KidLeavesStoop

Linietsky, Juan. 2018. Godot 3.0 is Out and Ready For the Big Leagues. Published on 29.01.2018. Read on 24.02.2024. https://godotengine.org/article/godot-3-0-released/

Godot Docs n.d. SubViewport. https://docs.godotengine.org/en/stable/classes/class_subviewport.html

Godot Docs. n.d. LightmapGI. https://docs.godotengine.org/en/stable/classes/class_lightmapgi.html