3D Environment Design For 2D Pixel Art Game "Missing Melody"

Kresnandito Riko Pambudi.[1] and Tito Ari Pratama*[2]

[1,2] Multimedia Nusantara Polytechnic, Tangerang, Indonesia
tito.ari@mnp.ac.id

Abstract. This study examines a 2.5D game development approach that integrates two-dimensional pixel art assets within a three-dimensional environment, implemented in the game Missing Melody, a psychological side-scrolling title developed using Unreal Engine 5. Because conventional 3D environments are not inherently compatible with the visual characteristics of 2D pixel art, the research explores design strategies and technical workflows to achieve stylistic cohesion between both asset types. The methodology combines experimental exploration—focused on identifying effective techniques for constructing 3D environments that preserve a pixel-art aesthetic—and applied research aimed at formulating practical design solutions. The production pipeline consists of three primary stages: 3D modeling and UV mapping using Blender, texturing with Substance Painter, and environmental layout assembly in Unreal Engine 5. To evaluate visual compatibility, an alpha test involving 18 respondents was conducted, assessing clarity, coherence, and overall user perception of the blended asset styles. The findings indicate that the 3D environment successfully supports and enhances the 2D pixel art elements, with most respondents affirming that the 3D backgrounds did not disrupt the visual experience. The majority also expressed that both asset types blended cohesively, although some noted minor inconsistencies in texture rendering. Overall, the study concludes that a carefully structured 3D environment can effectively complement 2D pixel art and strengthen visual integration in 2.5D game development.

Keywords: *3D Environment, Pixel art, 2.5D game, retro style, unreal engine 5*

1 INTRODUCTION

Since its initial emergence in the 1950s, the game industry has undergone significant transformation [1]. Early games relied on limited two-dimensional (2D) graphics characterized by simple visuals and basic mechanics. However, advances in computer technology and hardware enabled the adoption of three-dimensional (3D) graphics, fundamentally changing the player experience. This technological progress also creates opportunities for Indonesia to actively contribute to the global game industry [6].

One example of this evolution can be observed in the development of the *Super Mario Bros* franchise by Nintendo. In 1985, Mario appeared in a simple pixelated 2D form, while later titles introduced more complex visual representations supported by 3D graphics technology [14]. The transition from 2D to 3D graphics not only affected

T. E. Widiasih et al. (eds.), *Proceedings of the International Moving Image Cultures Conference (IMOVICCON 2025)*, Advances in Social Science, Education and Humanities Research 1023,
https://doi.org/10.2991/978-2-38476-603-1_15

character design but also expanded the development of virtual environments that provide spatial depth and interactive elements. These environments enhance immersion by enabling players to explore digital worlds that simulate real spatial experiences.

The 3D environment has become a crucial component in modern game development. Contemporary games rely on visual realism, lighting systems, and environmental interaction to create immersive experiences [9]. Compared to 2D environments, 3D environments allow developers to implement perspective, depth, and spatial interaction more effectively. However, the production of 3D environments also requires more complex pipelines, including modeling, UV mapping, texturing, lighting, and rendering processes [10]. Therefore, developers often seek alternative approaches that balance visual quality and production efficiency.

One of these approaches is the hybrid method known as 2.5D, which combines 2D and 3D elements [7]. In this method, characters are typically presented in 2D format, while environments use 3D assets. The 2.5D approach offers several advantages. First, 2D characters are generally faster to produce compared to fully rigged 3D characters, as they do not require complex modeling or animation processes [12]. Second, combining 2D visuals with 3D environments allows developers to maintain nostalgic visual characteristics while still benefiting from modern lighting and rendering technologies. This balance can improve production efficiency while maintaining artistic uniqueness. However, the 2.5D approach also presents several challenges. Visual consistency between 2D and 3D elements must be carefully managed to prevent stylistic mismatch. Lighting adjustments must be optimized to ensure that 3D environments do not appear too realistic compared to stylized 2D characters. Additionally, camera positioning and perspective design must be carefully controlled to maintain the illusion of depth without breaking the visual coherence of the pixel art style.

Pixel art is a digital art style that uses visible pixels as the main visual element [15]. Historically, pixel art emerged due to hardware limitations that restricted screen resolution and color palette [3]. Although modern hardware no longer imposes such limitations, pixel art continues to be widely used due to its distinctive aesthetic value and nostalgic appeal. Pixel art provides visual clarity, stylistic identity, and relatively efficient production workflows compared to high-resolution 3D assets.

Based on these considerations, the author develops a game project titled *Missing Melody*, a psychological side-scrolling game that combines 2D pixel art characters with a 3D environment. This approach aims to integrate retro aesthetics with modern rendering technology to produce a distinctive visual experience. The research problem is formulated as follows: “How can a 3D environment be designed to support the pixel art visual style in the game *Missing Melody*?”.

This study focuses specifically on designing a 3D environment that supports pixel art aesthetics and does not cover gameplay mechanics, narrative structure, or character design. The production pipeline utilizes Blender for modeling and UV mapping, Substance Painter for texture creation, Photoshop for texture resampling using Nearest Neighbor interpolation, and Unreal Engine 5 for final visualization. The use of Nearest Neighbor resampling is intended to maintain sharp pixel edges without anti-aliasing effects, ensuring visual consistency with pixel art characteristics. Compared to manual pixel texturing, this workflow offers faster production time while maintaining stylistic consistency [4]. However, this method requires careful resolution adjustment to prevent texture distortion when applied to 3D surfaces. Unreal Engine 5 is used to control texture filtering settings to maintain sharp visual output despite low-resolution textures. This demonstrates that optimization techniques can maintain visual quality while reducing computational requirements. The side-scrolling perspective simplifies navigation while allowing depth perception through layered 3D environments. This approach enables efficient lighting adjustments to support narrative mood without requiring fully realistic assets. Furthermore, the combination of 2D characters and 3D environments reduces computational complexity compared to fully 3D games, making the project more accessible for devices with lower specifications.

By combining pixel art aesthetics with modern rendering techniques, this research aims to provide insights into how stylized visual approaches can be integrated with contemporary game engines. The results are expected to contribute to the development of efficient visual production methods that maintain artistic identity while optimizing technical performance.

2 METHODS

This research focuses on designing a 3D environment for a pixel art-style game titled *Missing Melody*. The scope of this study is limited to environmental visual design and does not include other aspects of game development such as gameplay mechanics, narrative structure, or character design. The primary objective is to develop a 3D environment capable of visually supporting the 2D pixel art style without reducing aesthetic consistency. To achieve this objective, the researcher applied an experimental approach through technical exploration. This method allows iterative testing of visual and technical adjustments in order to identify the most effective workflow for integrating 2D pixel art elements within a 3D environment. The study adopts an Applied Research methodology, which focuses on solving practical problems and producing outcomes that can be directly implemented in real production contexts [11]. In this case, the practical problem lies in maintaining pixel art visual characteristics within a 3D rendering system that typically produces smooth and anti-aliased textures.

The production pipeline utilizes four main software tools: Blender, Substance Painter, Adobe Photoshop, and Unreal Engine 5. These tools were selected due to their widespread industry use and accessibility for practitioners, allowing the workflow to remain replicable and relevant to common game development practices. Blender was used for 3D modeling and UV mapping, Substance Painter for base texture generation, Adobe Photoshop for pixel resampling using Nearest Neighbor interpolation, and Unreal Engine 5 for rendering and visual integration. The experimental process began with the creation of a conventional 3D environment, which was then modified to support pixel art characteristics based on theoretical references. This step highlights a critical challenge in 2.5D visual design: 3D rendering systems are generally optimized for smooth gradients and realistic shading, while pixel art emphasizes sharp edges, limited color palettes, and visible pixel structure. Therefore, adjustments in texture filtering, resolution scaling, and lighting setup are necessary to maintain stylistic consistency.

To evaluate the effectiveness of the visual integration, an alpha testing phase was conducted to gather feedback from respondents. This evaluation aimed to identify whether the 3D environment successfully preserved the pixel art aesthetic and whether the visual output remained coherent between 2D and 3D elements. The theoretical foundation used in this research consists of two primary areas: 3D modeling theory and 2.5D game design theory. Supporting technical references were also utilized, including the study titled "*Rancang Bangun Pixel Art Converter Menggunakan Segmentasi berbasis K-means Clustering"* by Yuna Sugianela, which provides insight into pixel simplification and color segmentation processes. This reference supports the technical approach used to maintain pixel clarity when adapting textures from high-resolution formats into pixel-based visual structures.

Overall, the selected methodology enables the researcher to critically examine how conventional 3D production workflows can be adapted to preserve pixel art aesthetics. The approach also demonstrates that stylistic consistency in 2.5D games requires not only artistic consideration but also technical adjustments in rendering parameters and texture processing.

2.1 3D Environment for Games by Arianto (2024)

Three-dimensional (3D) games have significantly transformed user experience by enabling more spatially complex and interactive virtual environments that enhance player immersion [2]. The 3D environment is not only a visual component but also a structural element that influences how players interpret space, atmosphere, and narrative context. However, the pursuit of realism in 3D graphics often contrasts with stylized visual approaches such as pixel art, which emphasize simplified forms, limited color palettes, and visible pixel structures. This creates a technical and aesthetic challenge: while 3D environments rely on smooth shading and high-resolution textures, pixel art intentionally preserves sharp edges and minimal visual complexity. Therefore, adapting 3D environments to support pixel art aesthetics requires careful modification of conventional production workflows to maintain stylistic coherence without sacrificing spatial depth and interactivity.

Arianto (2024) explains that 3D asset production generally involves structured stages such as planning, pre-production, modeling, UV mapping, texturing, rigging, and Level of Detail (LOD) optimization [2]. Although these stages are primarily discussed in the context of character production, they are also applicable to environment creation. Nevertheless, not all stages are equally relevant for stylized 2.5D development. In this study, the focus is limited to modeling, UV mapping, and texturing, as these stages directly influence visual style translation from high-resolution 3D assets into pixel-based aesthetics. UV mapping plays a critical role in controlling how textures are distributed across 3D surfaces, particularly when low-resolution textures are used to simulate pixel art characteristics. Without careful UV planning, textures may appear stretched or inconsistent, reducing the visual clarity essential to pixel art style.

Consequently, adapting conventional 3D workflows requires selective modification, especially in texture resolution, filtering methods, and shading adjustments, in order to preserve stylistic consistency between 2D pixel art elements and 3D environments. This highlights that the challenge in 2.5D production is not merely technical execution but also the reconciliation of two different visual paradigms: realism-driven 3D rendering and abstraction-driven pixel art design. Arianto (2024) defines 3D modeling as a technique used to produce three-dimensional digital images that realistically represent objects, whether animate or inanimate. This process entails designing and arranging objects to appear lifelike and possess volume within three-dimensional space. A 3D model is fundamentally composed of elements such as polygons, edges, and vertices, which are structurally interconnected to define the object's surface.

2.2 Game 2.5D by Hartono & Kurniawan (2022)

2.5D games represent a hybrid visual format that combines two-dimensional (2D) assets with spatial arrangements that simulate a three-dimensional (3D) environment [5]. Although gameplay mechanics generally operate along a limited axis, such as horizontal or vertical movement, the visual presentation is designed to create depth perception through layered composition. This approach allows developers to retain the simplicity and readability of 2D interaction while improving visual richness through spatial arrangement. As a pseudo-3D format, 2.5D does not fully rely on geometric depth but instead constructs the perception of depth through visual structuring of assets such as sprites, textures, and environmental layers.

A key technique commonly applied in 2.5D games is parallax scrolling, in which foreground, midground, and background layers move at different speeds relative to camera movement [5]. This differential motion simulates spatial distance and enhances environmental immersion without requiring fully modeled 3D geometry. From a production perspective, this layered system offers efficiency advantages because assets can be reused and rearranged without significantly increasing computational cost. However, the visual result strongly depends on the accuracy of layer spacing, scale proportion, and camera positioning. If the speed ratio between layers is not carefully calibrated, the illusion of depth may appear artificial or visually inconsistent.

Critically, while parallax scrolling improves the perception of spatial depth, the technique also introduces limitations in environmental interaction and visual realism. Because depth is simulated rather than physically constructed, lighting behavior and object perspective may not fully correspond to real 3D spatial logic. As a result, inconsistencies between visual layers may reduce immersion if not carefully controlled. In pixel art-based 2.5D games, these challenges become more pronounced because low-resolution textures emphasize edge clarity and limit gradient transitions. Therefore, achieving a coherent visual result requires balancing stylistic consistency, technical optimization, and perceptual depth, ensuring that the final output maintains the pixel art aesthetic while still providing a convincing sense of space.

In the context of *Missing Melody*, the 2.5D approach is applied to balance stylistic consistency with technical efficiency. The use of layered environments and controlled parallax scrolling enables the 3D environment to support the visual characteristics of 2D pixel art characters without introducing excessive geometric complexity (see Fig. 1). This approach is particularly relevant for maintaining pixel sharpness while still achieving spatial depth through lighting and perspective control within Unreal Engine 5. However, the effectiveness of this method depends on how well the visual contrast between low-resolution pixel textures and real-time 3D lighting can be harmonized.

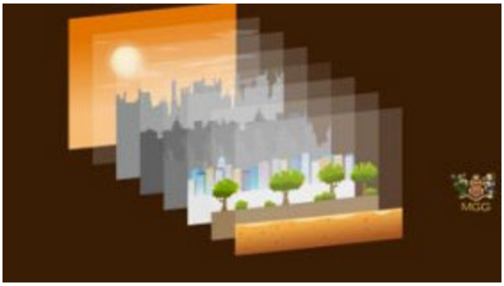

Fig. 1. Parallax Scrolling by Hartono & Kurniawan (Personal Documentation)

2.3 Pixel Art in Game by Sugianela (2017)

Pixel art is a form of digital visual design produced using raster graphics software, in which images are constructed by directly manipulating individual pixels. This visual style was widely used in early video games from the 1990s, such as *Pac-Man*, *Super Mario Bros.*, and *Galaga*, and has recently regained popularity among independent (indie) developers due to its nostalgic aesthetic and production efficiency. According to Sugianela et al. (2018, p. 3), pixel art is characterized by block-based visual structures, visible pixel edges, and the use of limited color palettes, typically associated with 8-bit or 16-bit visual limitations. From a production perspective, pixel art offers both artistic and technical advantages. The limited color palette and simplified visual complexity result in smaller file sizes and faster asset creation compared to high-resolution graphics. This efficiency is particularly beneficial for small studios with limited resources. For example, Maulidan Games, a game development studio based in Surabaya, applies pixel art techniques to accelerate asset production workflows within their projects. Highlight the need for automated tools capable of converting high-resolution images into pixel-based visuals, reducing the need for manual redrawing and shortening development time [13].

However, the pixel art approach also presents limitations. The restricted resolution and simplified visual detail may reduce realism and limit stylistic flexibility, especially for projects aiming for cinematic or highly immersive visual experiences. Therefore, the decision to adopt pixel art should be considered strategically, balancing production efficiency, artistic direction, and the intended player experience. In this context, pixel art is not only a stylistic choice but also a production method that reflects technical constraints and creative priorities within the game development pipeline.

3 RESULTS & DISCUSSION

3.1 Pre-Production

Following discussions with the production team, a total of nine scenes were identified as requiring 3D environment development to support the overall game experience. These scenes include Arka's room, the residential street, the alleyway, the school exterior, the school corridor, the classroom, the music club room, and the meadow environment. Each environment plays a significant role in shaping the narrative atmosphere and visual continuity of the game world. The environments were designed to reflect the emotional tone of the story, which emphasizes a nostalgic and stylized visual direction that combines pixel art characters with simplified 3D spatial design.

From the nine scenes, the author selected one representative environment as the primary production example within this documentation. This selected scene demonstrates the core workflow and technical considerations that are consistently applied across all environments in the project. The chosen scene effectively represents the artistic and technical challenges encountered during development, including optimization for performance, stylistic consistency, and compatibility with the game engine pipeline.

During the pre-production stage, blocking techniques were utilized to determine spatial composition, object placement, and environmental proportions. Blocking plays a crucial role in establishing a clear visual hierarchy before entering the detailed modeling stage. By defining object scale early in the workflow, the production team was able to maintain consistency across scenes while ensuring that all environments remain compatible with the side-scrolling camera perspective. This approach reduces the likelihood of scale inconsistencies and allows for faster iteration when adjustments are required.

From an analytical perspective, the use of blocking as an early visualization method provides several advantages. It allows designers to test spatial readability and gameplay navigation before committing to high-detail modeling. However, one limitation of blocking is that it may oversimplify spatial complexity, which can lead to additional revisions during later production stages. Despite this limitation, the blocking method remains efficient because it minimizes unnecessary detail work in areas that may not be visible within the camera framing.

3.2 Production Progress

The production process focuses on the application of theoretical principles related to 3D asset development, particularly in the stages of modeling, UV mapping, and texturing. The modeling process was conducted using Blender 4.2, utilizing the box modeling technique. This method involves modifying primitive geometric shapes and progressively refining them into the desired forms. Box modeling was selected because it provides a structured workflow that is well-suited for low-poly game environments and allows for efficient topology control.

One of the primary technical considerations during the modeling stage is polygon count optimization. Poly count directly affects hardware performance, particularly in real-time rendering environments such as game engines. A higher number of polygons increases the computational load on the GPU, which can negatively impact frame rate stability. Therefore, maintaining a balanced poly count is essential to ensure that the game remains accessible across various hardware specifications.

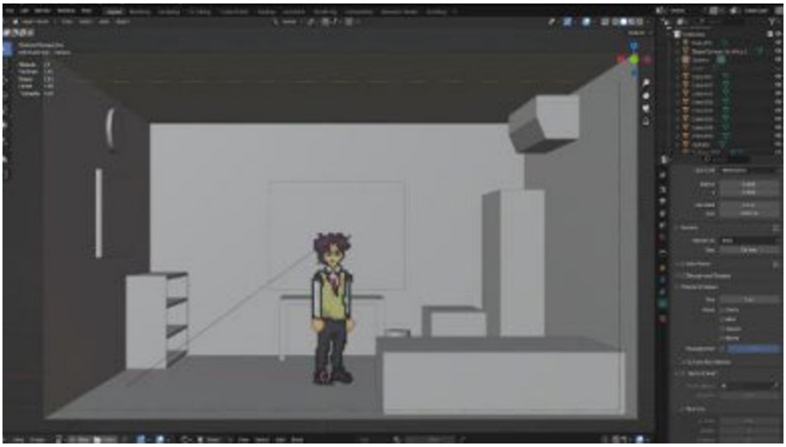

Fig. 2. Modeling and Blocking Progress (Personal Documentation)

The previously created blocking layout served as a proportional guideline during the modeling stage (see Fig. 2). Since the game adopts a side-scrolling perspective, certain sections of the environment, particularly the front-facing wall of the room, were intentionally omitted. This design decision prevents the environment from obstructing the camera's view of the character. From a critical standpoint, this approach demonstrates how technical constraints influence artistic decisions. While omitting geometry may reduce realism, it significantly improves gameplay readability and camera clarity.

To further optimize topology efficiency, the author separated objects requiring higher detail from simpler background elements. This selective detailing strategy ensures that additional edge loops are applied only to areas requiring visual emphasis. The advantage of this method lies in maintaining visual clarity while preventing unnecessary geometric complexity. However, the limitation of this technique is that inconsistent topology density across objects may create visible contrast if not carefully balanced.

Face orientation was also carefully monitored throughout the modeling process. In game engines, meshes typically render only one visible side, meaning inverted normals can cause objects to appear transparent. This technical constraint requires careful inspection of surface orientation before exporting assets. Failure to address face orientation can lead to rendering artifacts that disrupt immersion and require time-consuming corrections during later stages.

The texturing process began by exporting the 3D model in FBX format and importing it into Substance Painter (see Fig. 3). Prior to export, the model was configured to use a single material in order to prevent the automatic generation of multiple texture sets. This decision simplifies texture management and ensures consistency across asset surfaces. Within Substance Painter, the project template was configured for Unreal Engine compatibility, ensuring that the generated textures align with the rendering pipeline.

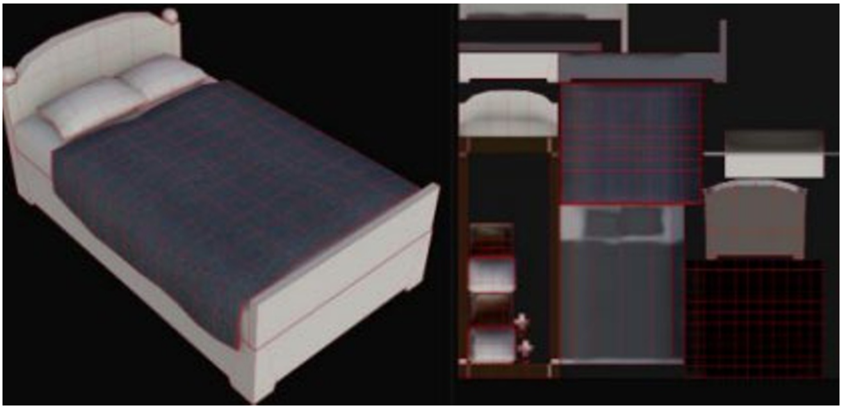

Fig. 3. Texturing in Substance Painter Progress (Personal Documentation)

Texture resolution selection was determined based on object scale. Medium to small assets utilized 1024x1024 textures, while larger assets used 2048x2048 resolution. This approach balances visual fidelity with memory efficiency. From an analytical perspective, higher resolution textures provide more detailed surfaces but increase memory usage. Therefore, careful resolution selection is necessary to prevent excessive resource consumption.

An important technical configuration involves selecting the DirectX normal map format. Using OpenGL format would invert the normal direction, producing incorrect lighting behavior within the Unreal Engine. Baking was performed within Substance Painter to generate ambient occlusion and curvature information. These maps enhance visual depth without increasing polygon count. Because the project adopts a low-poly modeling approach, texture detail plays a crucial role in conveying surface complexity.

Height maps and masking techniques were used to simulate surface depth, such as fabric folds on the bed blanket. This approach demonstrates how visual richness can be achieved without increasing geometry complexity. The advantage of this method is improved rendering efficiency, while the disadvantage is that simulated depth may appear less convincing when viewed from extreme angles.

After texturing, the assets initially appeared visually inconsistent with the 2D pixel art characters. As a result, texture resizing was performed using Photoshop. The image resolution was reduced, for example from 1024x1024 to 512x512, using the "Nearest Neighbour" resampling method (see Fig. 4). This method preserves hard pixel edges by avoiding smooth interpolation. The result produces jagged edges characteristic of pixel art aesthetics.

Fig. 4. Before and After Using Filter "Nearest Neighbour" Progress (Personal Documentation)

Critically, converting realistic textures into pixel art style requires balancing readability with stylistic coherence. While pixelated textures enhance stylistic unity with 2D characters, excessive pixelation may reduce environmental clarity. Therefore, resolution reduction must be carefully tested to maintain gameplay readability (see Fig. 5).

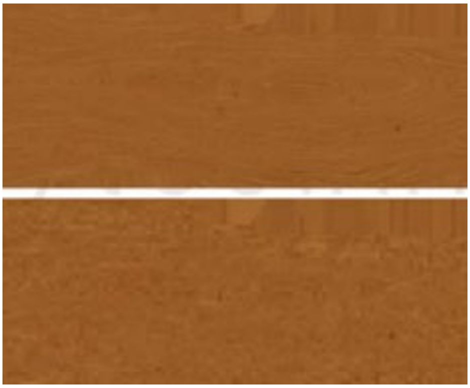

Fig. 5. Resizing Result (Personal Documentation)

Upon re-importing textures into Blender, interpolation settings were changed from Linear to Closest to preserve pixel sharpness. Assets were then imported into Unreal Engine 5. Materials were constructed by connecting texture maps according to their respective channels, including Occlusion, Roughness, and Metallic values. Texture filtering was adjusted to Nearest to prevent automatic smoothing.

The implementation of 2.5D theory is demonstrated through parallax scrolling elements, such as grass sprites and tree sprites (see Fig. 6). These layered visual components create depth illusion while maintaining performance efficiency. The advantage of parallax techniques is increased visual richness without significantly increasing computational cost. However, improper layer speed calibration may cause visual discomfort or reduce spatial coherence.

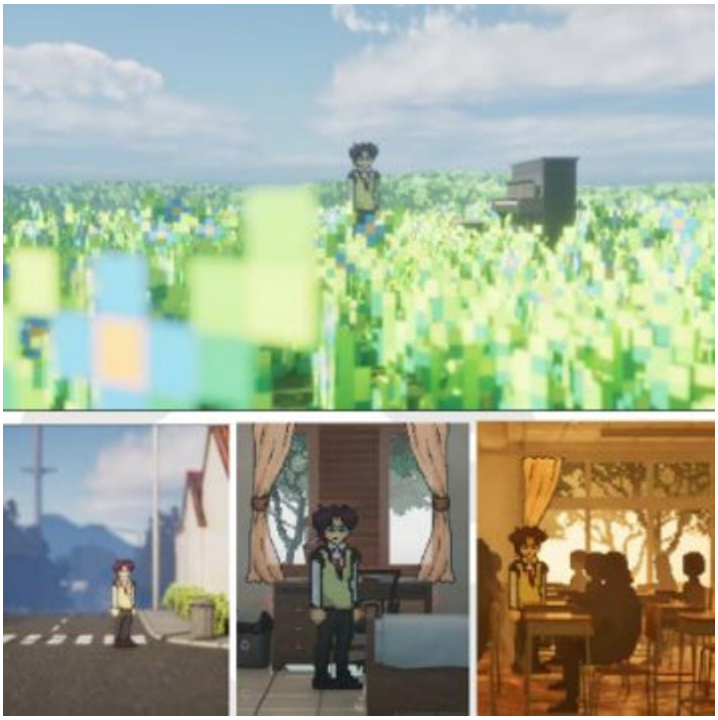

Fig. 6. Parallax Scrolling Implementation (Personal Documentation)

3.3 Post-Production Progress

During the post-production stage, the 3D environments were prepared for alpha testing (see Fig. 7; see Fig. 8). The testing session was conducted in collaboration with the Mugen Student Activity Unit (UKM), a student organization at Multimedia Nusantara Polytechnic focusing on game development. A total of 40 participants were invited to evaluate the game “Missing Melody.”

The purpose of alpha testing was to identify technical issues, including collision errors, visual inconsistencies, and performance instability. Additionally, feedback was collected regarding difficulty balancing and user experience clarity. Early-stage testing provides valuable insights because it reveals usability issues that may not be apparent during internal development.

From an analytical perspective, involving external participants increases evaluation objectivity. Developers often become accustomed to their own design decisions, which may reduce their ability to detect usability problems. However, alpha testing also presents limitations, as feedback may vary depending on player experience level. Beginner players may focus on usability clarity, while experienced players may prioritize gameplay depth.

Overall, the alpha testing process contributes to iterative improvement by identifying areas requiring refinement before entering the beta stage. The integration of technical optimization and player feedback demonstrates the importance of combining artistic vision with user-centered design principles.

Fig. 7. Arka's Classroom Result in Missing Melody (Personal Documentation)

Fig. 8. Arka's Classroom Layout in Missing Melody (Personal Documentation)

4 CONCLUSION

This research aims to design a 3D environment capable of supporting the visual and aesthetic quality of 2D pixel art assets in the game "Missing Melody." The study applies experimental and applied research methodology by integrating theories of 3D game production, 2.5D visual concepts, and texture processing techniques adapted for pixel art conversion. The results indicate that the use of 3D objects as environmental backgrounds significantly enhances spatial perception, depth, and visual variation compared to static 2D backgrounds. This improvement contributes to a more immersive experience while maintaining the stylistic identity of pixel art visuals.

The production pipeline employed low-poly modeling, optimized UV mapping, and texture conversion techniques using the nearest neighbor resizing method. This approach ensures that pixel sharpness is preserved while maintaining efficient performance across different hardware specifications. From a technical perspective, low-poly modeling reduces computational load, while pixel-based texturing ensures stylistic consistency with 2D character assets. Additionally, the implementation of parallax scrolling using layered 2D sprites enhances the illusion of depth without significantly increasing rendering complexity. This demonstrates that visual richness can be achieved without compromising system efficiency.

Based on alpha testing conducted with 18 respondents from the game development community, most participants agreed that the 3D environment integrates effectively with the 2D pixel art characters and supports gameplay readability. Nevertheless, feedback suggested improvements in pixel scale consistency and stronger environmental storytelling elements. These insights were used to refine visual adjustments and improve cohesion between assets. Overall, this research demonstrates that combining 3D environments with pixel art aesthetics can support the development of visually engaging, performance-efficient 2.5D games suitable for multi-platform deployment.

REFERENCES

1. Aprilia, S. (2023). Industri Game: Revolusi Digital dan Potensi yang Tak Terbatas. [Online post]. Marketeers. https://www.marketeers.com/industri-game-revolusi-digital-dan-potensi-yang-tak-terbatas/
2. Arianto, A. R., Kusnadi, & Parman, S. (2024). Pembuatan 3D karakter desain menggunakan teknik LOD untuk optimasi pada game mobile. Jurnal TIKomSiN, 12(2). https://doi.org/10.30646/tikomsin.v12i2.870
3. Binus University. (2017). "Pixel Art" sebagai Gaya Visual dalam Video Game Modern. [Online post]. https://binus.ac.id/malang/2017/09/pixel-art- sebagai-gaya-visualdalam-video-game-modern/
4. Graphie. (2024). Menciptakan Dunia Imersif: Bagaimana Desain 3D Mengubah Pengalaman Pengguna di Ruang Digital [Online post]. https://www.graphie.co.id/blog/280/menciptakan-dunia-imersif-bagaimana-desain3d-mengubah-pengalaman-pengguna-di-ruang-digita l.

5. Hartono, V., & Kurniawan, D. (2022). Perancangan Parallax Background pada Game MotoX Hill Racing. Jurnal Tanra, 9(2). https://doi.org/10.26858/tanra.v9i2.34393
6. Jamalullail, D. (2023). Industri Game: Kemajuan Industri Game di Indonesia. [Online post]. https://kumparan.com/dziqry-j/kemajuan-industri-game-di-indonesia20jrZml5WCa
7. Khan, M. S., & Mankar, V. H. (2016). A Review on 2d, 2.5d and 3d Image visualization Techniques. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 5(3).
8. Kuswana, R. (2023). Unity vs Unreal Engine, Software Mana yang Harus Dipilih? [Online post]. Alan Creative. https://alan.co.id/en/unity-vs-unreal-engine-softwaremana- yang-harus-dipilih /
9. Mačković, D., & Bernik, A. (2018). UNREAL ENGINE 4 ENVIRONMENT WORKFLOW. Polytechnic and Design, 6(4), 234–244. https://doi.org/10.19279 /TVZ.PD.2018-6-4-15
10. Nurhayati, N., Azizul, M., Kurmazi, R., Firdania, M. S., & Sitompul, N. (2023). Pembuatan Film Animasi 3D "Buah Tangan Berbuah Kenangan" dengan Metode MDLC. Pixel :Jurnal Ilmiah Komputer Grafis, 16(2), 153–164. https://doi.org/10.51903/ pixel.v16i2.1316
11. Pratama, T. A. (2024). Wacana Kecepatan Perkembangan Teknologi Vision Pro lewat Pandangan Paul Virilio. Jurnal Penelitian Humaniora, 29(2). https://doi.org/10.21831/ hum.v29i2.72706
12. Pratama, T. A., & Rosita, E. (2023). Implementation of Augmented Reality for Isometric works at Animation Art Exhibition (Animaxtion). Ultimart: Jurnal Komunikasi Visual, 16(1), 12–25. https://doi.org/10.31937/ultimart.v16i1.3111
13. Sugianela, Y., Suciati, N., & Bagus A.R, M. (2018). Rancang Bangun Pixel Art Converter Menggunakan Segmentasi berbasis K-means Clustering. Jurnal Teknik ITS, 6(2), A655-660. https://doi.org/10.12962/j23373539.v6i2.24409
14. Tania, M. A. (2021). Pengaruh 3D pada Perkembangan Dunia Game [Online post]. https://www.gamelab.id/news/748-pengaruh-3d-pada-perkembangan-dunia-game
15. Wei, X., & Wang, Y. (2021). Research on the Application of Pixel Style in UI (User Interface) Design: Proceedings of the 6th International Conference on Arts, Design and Contemporary Education (ICADCE 2020). The 6th International Conference on Arts, Design and Contemporary Education (ICADCE 2020), Moscow, Russia. https://doi.org/10.2991/assehr.k.210106.109