

Lost Forest : A Game Developed Using Unity

A project

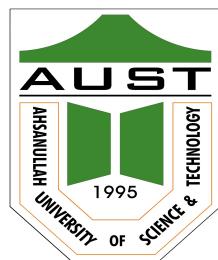
Submitted in partial fulfillment of the requirements for the Degree of
Bachelor of Science in Computer Science and Engineering

Submitted by

| | |
|-----------------------------|------------------|
| Ahmed Saad Tanim | 180204015 |
| Sanjay Kumar Mandal | 180204039 |
| Shayla Sharmin Niha | 180204095 |
| MD. Mahin Zaman Khan | 170204042 |

Supervised by

Mr. Md. Aminur Rahman



Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Dhaka, Bangladesh

May 2023

CANDIDATES' DECLARATION

We, hereby, declare that the project presented in this report is the outcome of the investigation performed by us under the supervision of Mr. Md. Aminur Rahman, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. The work was spread over two final year courses, CSE4100: Project and Thesis I and CSE4250: Project and Thesis II, in accordance with the course curriculum of the Department for the Bachelor of Science in Computer Science and Engineering program.

It is also declared that neither this project nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Ahmed Saad Tanim
180204015

Sanjay Kumar Mandal
180204039

Shayla Sharmin Niha
180204095

MD. Mahin Zaman Khan
170204042

CERTIFICATION

This project titled, "**Lost Forest : A Game Developed Using Unity**", submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in May 2023.

Group Members:

| | |
|-----------------------------|------------------|
| Ahmed Saad Tanim | 180204015 |
| Sanjay Kumar Mandal | 180204039 |
| Shayla Sharmin Niha | 180204095 |
| MD. Mahin Zaman Khan | 170204042 |

Mr. Md. Aminur Rahman
Assistant Professor & Supervisor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Professor Dr. Md. Shahriar Mahbub
Professor & Head
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

ACKNOWLEDGEMENT

We want to express our deepest gratitude to our honorable project supervisor Mr. Md. Aminur Rahman, Assistant Professor, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, for his guidance, support and feedback in this game development project. We would also like to thank Professor Dr. Md. Shahriar Mahbub, the honorable Head of the Department of Computer Science and Engineering of Ahsanullah University of Science and Technology, and all our respected teachers, lab assistants, and friends for their moral support and helpful discussions. Finally, we would like to thank our parents for their constant support and encouragement throughout our study.

Dhaka

May 2023

Ahmed Saad Tanim

Sanjay Kumar Mandal

Shayla Sharmin Niha

MD. Mahin Zaman Khan

ABSTRACT

Digital games have become a ubiquitous part of our society. Game development is the process of creating games for various platforms, including consoles, computers, mobile devices, and virtual reality devices. In many countries, game development is a substantial and important industry as it is the most valuable entertainment industry. Despite the economic instability and crisis deeply affecting the world, the growth of this industry is at a rate of 57 percent surprisingly. So, game development is an important field to make a career in as it is both challenging and exciting. This report provides an overview of Unity game development, including its features, benefits, and challenges. Unity is a cross-platform game engine that is widely used all over the world for developing 2D and 3D games for all gaming platforms. It highlights all the steps that are required to develop our game "Lost Forest". It is an immersive and captivating 2D action-adventure game with 3D effects that transports players on an exciting trip through a mysterious and magical woodland. The game combines interesting gameplay, amazing aesthetics, and a fascinating plot to present players with a great gaming experience. It was developed with complete originality. In this game, players take on the role of a brave protagonist who must navigate perilous terrain, solve difficult riddles, and battle fearsome foes. The game's complex plot and dynamic gameplay provide a wonderful combination of action, exploration, and problem-solving, keeping players interested and delighted throughout their journey. The visual presentation of Lost Forest is one of its most notable elements. The game brings the forest to life with the smart use of the parallax effect and finely produced sprite animations, immersing players in a lively and aesthetically fascinating environment. The Lost Forest development process was created on a strong basis, leveraging the adaptable Unity game engine and a defined software process architecture. This method allowed us to combine numerous assets rapidly, create complicated gameplay dynamics, and assure a polished and high-quality end product. Lost Forest is a tribute to the Unity game engine's capability and the ingenuity of a determined development team regardless of prior experience and relatively small team size.

Contents

| | |
|---|-----|
| CANDIDATES' DECLARATION | i |
| CERTIFICATION | ii |
| ACKNOWLEDGEMENT | iii |
| ABSTRACT | iv |
| List of Figures | vii |
| 1 Introduction | 1 |
| 1.1 Overview | 1 |
| 1.2 Motivation | 2 |
| 1.3 Objective | 3 |
| 2 Comparison between games | 4 |
| 2.1 Similarity between the games | 4 |
| 2.2 Dissimilarity between the games | 5 |
| 3 Development Procedure | 7 |
| 3.1 Sprite Acquisition | 9 |
| 3.2 Sprite Preparation | 9 |
| 3.3 Front-end Implementation | 10 |
| 3.4 Back-end Coding | 10 |
| 3.5 Game Polishing | 11 |
| 3.6 Publishing | 11 |
| 4 Game Engine | 12 |
| 4.1 Framework | 12 |
| 4.2 Unity vs Other game engines | 13 |
| 5 Game Environment and effects | 15 |
| 5.1 Render Pipeline | 15 |
| 5.2 Camera Settings | 16 |

| | | |
|----------|-------------------------------------|-----------|
| 5.3 | Layer order | 16 |
| 5.4 | Parallax Effect | 17 |
| 5.5 | Visual Effects | 17 |
| 5.6 | Modular Characters | 18 |
| 5.7 | World Bosses | 19 |
| 5.8 | World Map | 20 |
| 6 | Game Assets | 23 |
| 7 | Limitations and Future Works | 25 |
| 7.1 | Limitations | 25 |
| 7.2 | Future Works | 26 |
| 8 | Conclusion | 27 |
| | References | 28 |

List of Figures

| | | |
|-----|---|----|
| 3.1 | Flowchart of Development Procedure | 8 |
| 4.1 | Comparison between Unity and several other popular game engines | 14 |
| 5.1 | First image consists of 16 layers to give parallax effect. Second image consists of only 4 layers to give the same effect | 17 |
| 5.2 | Base player model and 4 different hero modes | 18 |
| 5.3 | World Bosses | 20 |
| 5.4 | World Map | 21 |
| 6.1 | Sample of Assets that were purchased from store | 23 |

Chapter 1

Introduction

1.1 Overview

Unity [1] is a popular game engine that provides its users with an essential game development environment to produce high quality games that can run on various platforms. The goal of this project is to create a 2D action-adventure game that provides players with an exciting and memorable gaming experience. Unity has revolutionized game industry since its release in 2005. Because of its versatility, scalability and ease of use it is used by many professionals as well as aspiring new learners. It is a very versatile platform for developing games.

The decision to make this game in Unity was made with regard to the platform having many in-built tools and providing a good framework for us to get started with the development as fast as possible. The fact that Unity 2D also used C# as a development language was also in consideration since we wanted to learn the C# language for it being relatively similar to C language.

Lost Forest is a 2D platformer action-adventure game developed using Unity. It will be available for PC and Android platform. Lost Forest has rich story, action, puzzle etc. to attract the players attention. Although the game is 2D game, it is built using Parallax effect which gives us the illusion of 3D in a 2D space. The visuals and the game mechanics are top priority for us as these plays a crucial role for a game's success. As for the animation, we are using bone animation. The characters of our game are simply just still images called sprites. Each sprites are then divided into multiple parts which are then moved independently according to the animators choice and then recorded for a limited time. There are multiple animation for a single character to make visuals more appealing.

During the research to find out what 2D modeling program to use, we found that we could use different studios to create models that could later import into our game project. The

complete game contains models made in both Unity and Krita [2]. With these choices made, we progressively made our development environment, set to use Unity 2D supporting the framework, and Unity and Krita for modeling the graphical components. For some of the sound effects, we also made use of Adobe Audition 2.0

1.2 Motivation

The project is not only concerned with developing an engaging game, but also with the development team's learning experience. The choice of Unity and C# was decided partly to give a chance to learn a new programming language while also creating a game. We were inspired and learned a lot from the tutorials made by Brackeys [3], Code Monkey [4] and Thomas Brush [5]. The usage of several modeling programs and sound effects software also gives an opportunity to get knowledge with a variety of game production technologies. The usage of the C# language for game development with Unity brings several advantages. For starters, C# is reasonably simple to learn for people with prior programming knowledge because it is similar to other popular programming languages like Java and C++. Second, C# is a highly versatile programming language that may be used to create complicated games with a variety of functions. Finally, C# is a popular language with a large community that provides plenty of resources for developers to learn from and leverage.

We are making the game with the Unity game engine because of its intuitive user experience will help us to get start with the game development that we are passionate about and to provide a high-quality 2D action-adventure game that provides players with an interesting and unforgettable experience. The project us gives a chance to study and get expertise with various game development tools and applications, such as Unity, Krita, and Adobe Audition 2.0. We are inspired by the challenge of creating a game from the ground up as well as the gratification of watching the end product come to life. In addition, we wish to motivate budding game developers to follow their ambitions by sharing our enthusiasm for game production.

We chose to make a 2D action-adventure game rather than a 3D mainstream scroller or idle clicker game since we noticed there was a shortage of significant and memorable games in today's game publishing outlets. While 3D mainstream scrollers and idle clicker games have their own appeal, we believe that 2D action-adventure games provide players with a more engaging and immersive experience. 2D action-adventure games are noted for their deep tales, challenging puzzles, and thrilling action sequences that frequently leave gamers with a lasting impression. Many players associate these games with a sense of nostalgia, and they are frequently regarded as gaming classics. Many 3D mainstream scrollers and idle clicker games, on the other hand, are more concerned with gameplay mechanics and

repetitive actions than with story or character development.

1.3 Objective

This project's objective is to build an entertaining and unforgettable 2D action-adventure game utilizing the famous game engine Unity. Since its debut in 2005, Unity has been a game changer in the industry, providing a robust and scalable framework for game production. The project intends to use Unity's built-in tools such as its physics engine and the simplicity of its C# development language to build a high-quality game in a relatively short amount of time.

One of the aims of our initiative is to help make game creation a viable career option in our nation. In recent years, the video game business has grown significantly and has the potential to become a big contributor to the economy. By creating a high-quality game with Unity, we intend to demonstrate the industry's potential and encourage more individuals to consider it as a career option.

Furthermore, we believe that games have the ability to educate, inform, and inspire. We aspire to contribute to the development of a better society by developing meaningful and moral games. Games can be an effective tool for learning, and we want to capitalize on this by including educational features into our game. This can help gamers acquire new abilities or ideas while still having fun. We also believe that games can be used to promote positive social values as well as empathy and compassion for others. We seek to motivate players to become more compassionate and aware of diverse viewpoints by designing games that offer essential life lessons or investigate societal concerns. This can contribute to the development of a more inclusive and accepting society.

Chapter 2

Comparison between games

Lost Forest is a 2D action-adventure game created using Unity that strives to give players with an engaging and unforgettable gaming experience. The game features gorgeous graphics, difficult puzzles, and a captivating plot that keeps players engaged. Other popular 2D action-adventure games on the market include **Skull** [6], **Hollow Knight** [7], and **Ori and the Blind Forest** [8]. In this comparison, we will evaluate Lost Forest against these three popular games, comparing and contrasting their gameplay, aesthetics, sound, plot, and overall gaming experience. By the end of this comparison, we hope to have provided insights into what distinguishes Lost Forest from these games and how it compares to some of the most popular 2D games.

2.1 Similarity between the games

Ori and the Blind Forest, Hollow Knight, and Skull all qualify as highly thought of 2D action-adventure games that have acquired popularity among players.

Ori and the Blind Forest is an aesthetically attractive and emotionally engaging 2D action-adventure game praised for its art style, plot, and gameplay. The game, developed by Moon Studios and launched in 2015, tells the narrative of Ori, a little spirit being entrusted with defending Nibel's woodland from the darkness that threatens to swallow it. Players may explore a vast, colourful world filled with fascinating animals and locations in the game. The images are hand-painted and animated in a stunning and realistic manner that has been lauded for its originality and attention to detail. The music, written by Gareth Coker, is also a prominent aspect that adds to the game's immersive experience. Ori and the Blind Forest has a gripping plot that is both emotional and engrossing, in addition to its gorgeous graphics. As players guide Ori through a journey filled with heartbreak moments and difficult obstacles, the game's narrative explores themes of sacrifice, loss, and redemption. Ori and

the Blind Forest's gameplay combines platforming, puzzles, and combat, requiring players to use their brains and reflexes to proceed through the game. The platforming components are especially notable, since they need exact timing and expert navigation to progress through the game's stages. The problems are also skillfully crafted and frequently need imaginative methods to be solved.

Hollow Knight, a 2017 game created by Team Cherry, is set in the strange underground land of Hallownest. Players assume the character of a small bug warrior on a mission to learn the ancient kingdom's mysteries. The game is well-known for its difficult gameplay, complicated level design, and stunning hand-drawn graphics. On the other hand, Skull is a dark and atmospheric 2D action-adventure game set in a disturbing world of bones and skulls, developed by Studio Plumeau and expected to be published in 2020. The game has non-linear gameplay, difficult combat, and a disturbing music that contributes to the gloomy mood.

All three games have received praise for their distinct art styles, engaging gameplay, and immersive worlds in which players can lose themselves. They have a devoted fan base and have received several awards for their originality and inventiveness in the gaming industry. Lost Forest, like the other games mentioned, is a 2D platformer action-adventure game that attempts to provide players with a memorable and immersive experience. Lost Forest, like Ori and the Blind Forest and Hollow Knight, features an engaging story that is complemented with a stunning visual presentation and an interesting gaming experience. Lost Forest, like Skull, concentrates on action features while also including puzzle-solving, exploration, and character advancement. The purpose of Lost Forest is to present players with an immersive and visually gorgeous game experience that is both tough and rewarding. All of these games have successfully captured the essence of the 2D world, demonstrating the gaming industry's reliance on nostalgia. Lost Forest is no exception, seeking to present gamers with a memorable and enjoyable gaming experience that will linger with them long after the game has been played. We also set out to build a game that not only retains the spirit of 2D gaming, but also gives gamers with something fresh and distinctive by learning from the accomplishments of earlier titles.

2.2 Dissimilarity between the games

The size of the development team is an important component that can impact the game's features and development process. Since our team is small in comparison to the studios behind Hollow Knight, Skull, and Ori and the Blind Forest, we were forced to make some compromises in terms of game features or the time it took to develop them. Despite our tiny team, we have worked hard to create a game that retains the essence of the 2D action-

adventure genre.

Another distinction between our game and Hollow Knight and Skull is the feeling we hope to instill in our players. Both games feature a darker, gloomier ambiance that sets the tone for their respective tales. Our game, on the other hand, is designed to create a more intense and exciting atmosphere. Our primary goal is to guarantee that the game mechanics, stories, and aesthetics all add to a sense of adventure and excitement that keeps players involved and wanting to explore further. While there may be some differences between our game and these other titles, we hope that Lost Forest will stand out as a unique and engaging 2D action-adventure game in its own right.

Last but not the least, Lost Forest lacks unique character and environment creation as compared to Ori and the Blind Forest, Hollow Knight, and Skull. These three games are noted for their remarkable and distinctive characters, each with their own narrative and personality. These three games' game landscapes are beautifully crafted, with attention devoted to every detail. On the other hand, Lost Forest, falls short in these areas. The character design is not as unique as those accomplished games and it lacks the inventiveness exhibited in the mentioned games. It's lack of creative design may be a detriment to fans searching for a visually attractive and engaging gaming experience.

Chapter 3

Development Procedure

Even without an appropriate team or sufficient personnel, a well-planned and controlled growth process is essential. This can help guarantee that the finished product meets the required quality standards and is delivered on time and within budget. In this case, one method that may be done is to use a lightweight software development process, such as Agile. This technique places an emphasis on flexibility, adaptation, and cooperation, making it ideal for small teams or projects with minimal resources. The team works in short iterations or sprints in an Agile development approach, focused on providing tiny bits of functionality that can be swiftly tested and assessed. This enables continual feedback and improvement, lowering the likelihood of mistakes or delays. Furthermore, Agile procedures frequently encourage communication and cooperation among team members and stakeholders. Despite the minimal manpower, this can assist guarantee that everyone is on the same page with the project's goals and progress.

Overall, while working with a small team or a few workers might create obstacles in software development, using a lightweight approach such as Agile can assist alleviate these challenges and raise the chance of a successful project conclusion.

The figure 3.1 represents the step by step flowchart of our development process. It consists of mainly 6 phases which are described as below:

The step by step process of our game development starts with sprite acquisition. It is then followed by the phase sprite preparation which consists of two phases, character building and world building, both are critical requirement as they are served to provide good quality user experience. Afterwards, the phases front-end implementation and back-end coding are like two sides of a coin and is complimentary to each other. Finally, the step game polishing is to fix any undesired mistakes or disliked features of our game so that we can complete the final step which is publishing.

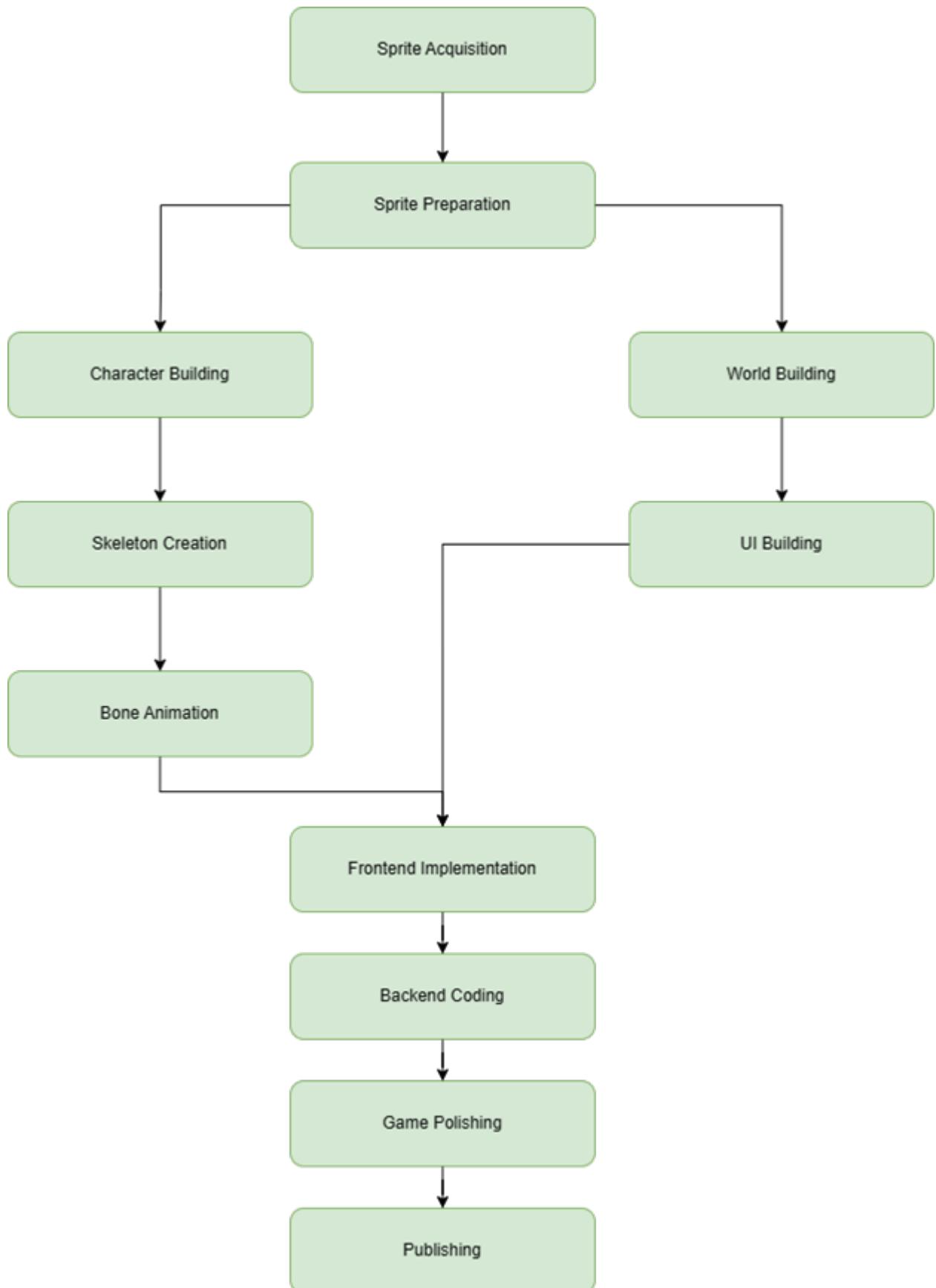


Figure 3.1: Flowchart of Development Procedure

3.1 Sprite Acquisition

We investigated a variety of possibilities for getting sprites or graphics elements during this period. Sometimes, we produced these assets ourselves using specialized tools, other times we collaborated with third-party sources to buy or license pre-existing assets that match the graphic style and general concept of our game. This stage is crucial for defining the game's visual identity and setting the tone for the player's experience. The sprites or graphical elements obtained will form the basis for the game's characters, environments, and user interface.

3.2 Sprite Preparation

The following phase in our game development procedure was to prepare the sprites we had bought and created for application in the game. This included tasks like scaling, cropping, and optimizing the sprites to ensure they performed well in our gaming context. This part of the development process consists of two phases, such as:

Phase I (Character Building): Our objective in this phase is to design the main character and other non-playable characters (NPC) that will feature in the game. Creating a skeleton structure also known as Bone or Skeletal animation, which acts as the basis for the character's movement and animation, is part of this procedure. Then we added the skeleton's sprites to bring the figure to life. The sprites are chosen and organized with care to portray the proper character design, personality, and motions. After building the character, the team tested it to confirm that it moves and animates as to our liking.

Phase II (World Building): In this phase, our task was to create the game environment, which includes the game world, objects, and user interface. The game world consists of a global map, which acts as a guide for level creation and game mechanics. The world map is populated with objects such as platforms, obstacles, and items that the player can interact with. Also, another part of world building is to create the user interface. It is meant to be intuitive and user-friendly, giving the player with the required information to continue through the game. We also focused on the visual design of the game world, producing a unified and immersive setting that complements the game's plot and ideas. This phase's overall purpose is to build an engaging gaming environment that attracts the user in and keeps them interested.

3.3 Front-end Implementation

The front-end implementation stage is critical in the game development process because it brings together all of the collected and prepared materials to produce the game's graphical user interface (GUI) and general feel. It entails creating an immersive and engaging user experience by developing and integrating numerous game elements such as character sprites, backdrop visuals, and UI components.

During this phase, the game world is constructed by positioning the characters, objects, and environment pieces. The game menus are meant to give players with a user-friendly interface for navigating through various stages, settings, and choices. Sound effects and background music are also used to make the game more immersive and interesting for the players.

Front-end development is an iterative process that requires ongoing testing and revision to ensure that the game's user interface is aesthetically appealing, easy to use, and immersive. The goal is to provide players a smooth and immersive experience that immerses them in the game environment.

3.4 Back-end Coding

The back-end development phase entails writing code that implements numerous game dynamics and capabilities in the C# programming language. This involves specifying how characters move and interact with the game environment, as well as the game's rules and how various game items interact with one another.

For example, we may develop code to govern the protagonist's mobility, such as how quickly the character walks, how high they can leap, and how they interact with various objects in the game environment. We would also build code to define how the enemy NPC act, such as how they move and attack the player.

Furthermore, we would develop code to govern the game's rules, such as the number of lives the player has, how the game is scored, and when the game finishes. In addition, we would develop code to handle events like player inputs, collisions, and game states.

The back-end coding step is critical for developing a playable and interesting game since it determines how the game mechanics and rules work and how players interact with the game environment.

3.5 Game Polishing

Our team concentrated on enhancing the game's quality throughout the game polishing phase. This involves testing the game for flaws or problems that were overlooked earlier in the production process. Any faults that are discovered are then addressed to ensure that the game performs properly and plays smoothly.

In addition, we optimize the game's performance to guarantee that it works smoothly on a variety of devices and operating systems. This involves tweaking the graphical settings, music effects, and animations to increase game performance without sacrificing overall quality.

Finally, we are working to improve the overall appearance and feel of the game. This might involve changing the colors, lighting, and textures to make the gaming world more immersive. We also strive to improve the user experience by making the game controls and interface more intuitive and user-friendly. Overall, the polishing process of the game is crucial in ensuring that the final product satisfies our quality requirements and is ready for publication.

3.6 Publishing

After the game has been developed and polished, the final step is to publish it to the relevant platforms. This involves preparing the game for distribution, including compiling the game code, packaging all necessary assets, and ensuring that the game meets all necessary requirements and regulations for the platform. For example, if the game is to be published on Steam, it must be submitted for review and meet the platform's guidelines for content and performance. Similarly, if the game is to be published on mobile platforms, it must be optimized for the specific devices and operating systems. Once the game is approved and released, the development team may continue to monitor and update the game as needed to ensure its success. Currently our primary focus is to release our game in Steam and Google Play Store.

Chapter 4

Game Engine

The world of game production has never been more accessible, owing in part to the multiple game engines on the market today. There is a game engine for every style of game developer, from open-source engines that are free to use to sophisticated engines with great graphical capabilities. These game engines include a number of tools and features that can assist creators in creating games quickly and effectively, enabling them to concentrate on the creative parts of game production.

Many of these game engines are tailored to certain platforms, such as mobile devices or desktop computers, whilst others are cross-platform and may be used to develop games for numerous platforms. Some game engines have easy visual editors that allow developers to design and change game levels and assets without writing any code, but others include strong scripting languages that enable advanced game dynamics and interactions. Aside from technical features, game engines frequently include active communities of developers and users that provide assistance, tutorials, and tools to help new creators get started. This is especially useful for small game producers who may not have the same resources as larger game production firms.

Thus, as among many of these game engines Unity has cross-platform compatibility, simplicity of use, rich feature set, active community, and Asset Store, it is a popular choice for efficiency and adaptability. Its versatility and strength make it perfect for both seasoned and inexperienced game producers.

4.1 Framework

Unity is a popular game engine for developing interactive 2D and 3D games for a range of platforms such as mobile, desktop, and console. The Unity game engine framework comprises many main components that work together to make game production easier:

Scene Editor: Unity includes a visual scene editor that enables game creators to design and edit game levels and environments, as well as add and position game objects and configure lighting and other visual effects.

Asset Store: The Asset Store in Unity is a marketplace where game creators may buy pre-built game components including character models, sound effects, and ambient pieces.

Scripting: Unity allows developers to manage game behavior and interactions by writing scripts in C# or Unity Script (a modified version of JavaScript).

Physics: Unity's physics engine simulates realistic physics for game objects, allowing developers to create dynamic worlds that respond to human actions.

Cross-Platform Deployment: With a single code base, Unity lets game creators to create games for numerous platforms, including iOS, Android, Windows, and Mac OS.

Overall, Unity's architecture offers a complete collection of tools and features that make it a popular choice for game production, providing efficiency as well as adaptability. Its adaptable design, strong physics engine, cross-platform deployment features, and active development community make it an excellent choice for both seasoned and inexperienced game makers.

4.2 Unity vs Other game engines

Comparing Unity to other game engines may be a difficult task because it is dependent on several elements such as the type of game, the platform, and the developer's own demands and preferences. Here are some basic comparisons between Unity and other major game engines:

Unreal Engine: Unreal Engine is a well-known gaming engine with high visual capabilities and comprehensive development tools. While Unity is more user-friendly and simple to use, Unreal Engine has more complex capabilities like as real-time ray tracing, dynamic lighting, and extensive physics simulation. Unreal Engine is also more focused on AAA game creation, whereas Unity is more adaptable and can be used for both independent and AAA game development.

Godot: Godot is an open-source gaming engine with a lightweight and efficient architecture. Godot has a lower learning curve than Unity, and its simplicity makes it suitable for independent developers looking to create 2D games. Unity, on the other hand, offers more extensive functionality and is better suited to building complex 3D games with realistic physics, visuals, and animations.

Defold: Unity and Defold are two game engines with distinct advantages and disadvan-

| | Unity | Unreal | Godot | Defold |
|----------------|-------|--------|-------|--------|
| Engine | | | | |
| Open source | ✗ | ✓ | ✓ | ✓ |
| Free to use | 🚧 (1) | 🚧 (1) | ✓ | ✓ |
| Learning curve | ✓ (2) | 🚧 (2) | ✓ (2) | ✓ (2) |
| 3D | ✓ | ✓ | 🚧 (3) | ✗ |
| 2D | ✓ | 🚧 (4) | ✓ | ✓ |
| Editor | | | | |
| Windows | ✓ | ✓ | ✓ | ✓ |

Figure 4.1: Comparison between Unity and several other popular game engines

tages. Unity has a larger community and a more extensive feature set, making it suited for producing complicated 2D and 3D games, but Defold has a simplified feature set and outstanding performance, making it perfect for creating small 2D games and prototypes, particularly for mobile platforms. The UI of Unity is more intuitive, however the interface of Defold is simpler and easier to use. Both engines have learning curves, however Unity's larger feature set can cause performance concerns, whereas Defold's lightweight architecture works effectively on low-end hardware.

To summarize, each game engine has strengths and disadvantages, and the choice of game engine is determined by the developer's individual demands and the sort of game they want to create. Unity's flexibility, ease of use, and vibrant community make it a fantastic choice for independent and AAA game developers looking to produce games for a variety of platforms.

Chapter 5

Game Environment and effects

Unity offers a comprehensive range of tools and capabilities for developing immersive game worlds and spectacular effects. Its terrain editor lets developers to construct realistic settings, while the physics engine simulates objects and people accurately. The particle system in Unity makes it simple to build beautiful visual effects, such as realistic fire and smoke, magical spells, and explosions. It also supports dynamic lighting, global illumination, and post-processing effects, which may give gaming landscapes more depth and realism. Overall, Unity's game environment and effects capabilities enable developers to create aesthetically appealing and engaging games that engage players.

5.1 Render Pipeline

The Scriptable Render Pipeline (SRP) in Unity is a useful tool for constructing custom rendering pipelines that can improve game visual quality. It gives developers greater control over the rendering process and enables them to generate one-of-a-kind, high-quality visuals. The Universal Render Pipeline (URP) and the High Definition Render Pipeline (HDRP) are components of Unity's SRP. The URP is intended for cross-platform development and has been tuned for mobile and low-end devices. The HDRP is intended for high-end systems, and it offers sophisticated capabilities like ray tracing, volumetric lighting, and complex shader effects. Developers may use SRP to generate gorgeous, dramatic images that increase user immersion and engagement.

In our game we are using URP as it is able to produce high-quality, performant visuals across a range of devices. By improving the rendering process, it is intended to create a balance of quality and performance. It also has forward and delayed rendering routes, adjustable post-processing effects, and support for both 2D and 3D rendering. It also supports a number of rendering methods, including as lighting, shadows, and reflections, which may be tailored

to the demands of a particular project. One of the primary benefits of adopting URP is that it enables developers to produce aesthetically attractive games while keeping decent performance on a variety of hardware. This is accomplished by employing technologies like dynamic batching and GPU instancing, which minimize the amount of draw calls while improving rendering speed.

5.2 Camera Settings

The Cinemachine camera technology is utilized throughout Lost Forest to provide smooth and seamless camera motions. The camera follows the player's motions as they walk around the game area, and it may pan and zoom to emphasize crucial features or items. Cinemachine is a camera system created by Unity Technologies that offers a collection of tools for producing dynamic and dramatic camera views in games. It is intended to allow developers to simply and intuitively build complicated camera setups and motions without having to write a lot of custom code. It supports a broad range of camera effects and behaviors, such as tracking, damping, noise, and others. These effects may be blended and tweaked to provide sophisticated and dynamic camera angles that add depth and realism to the game environment.

5.3 Layer order

Unity utilizes a layer system approach for both graphics and physics. The layer system enables game creators to classify various game elements into separate levels depending on their purposes or features.

For rendering, Unity's layer system is utilized for rendering to control which items are visible to the camera and how they are arranged in the scene. A layer can be assigned to a game object, and the camera can be configured to only render particular layers. This gives you more control over which game items appear in the scene and can assist increase speed by decreasing the amount of unneeded objects generated.

For physics, to govern which objects may interact with each other in physics such as collision, gravity etc, Unity's layer system is employed. Each Unity game object may be assigned to a physics layer, and the physics engine can be set to enable only collisions or interactions between certain layers. This is important for developing sophisticated game dynamics, such as opponents that can only be damaged by specific weapons or platforms that can only clash with specific items.

The layer system in Unity is a useful tool for game creators, allowing them to organize their

game objects and regulate how they interact aesthetically and physically with one another.

5.4 Parallax Effect

The parallax effect is a popular and common visual method in game development for adding depth and movement to a 2D gaming world. The parallax effect is created in Unity by moving numerous layers of backdrop images or sprites at varying rates related to the camera's movement or it can be also done by creating a number of layers and stack them on top of each other varying their Z axis value then use a Perspective camera to render them. This offers the user a sensation of motion as they navigate around the game area and generates the illusion of depth.

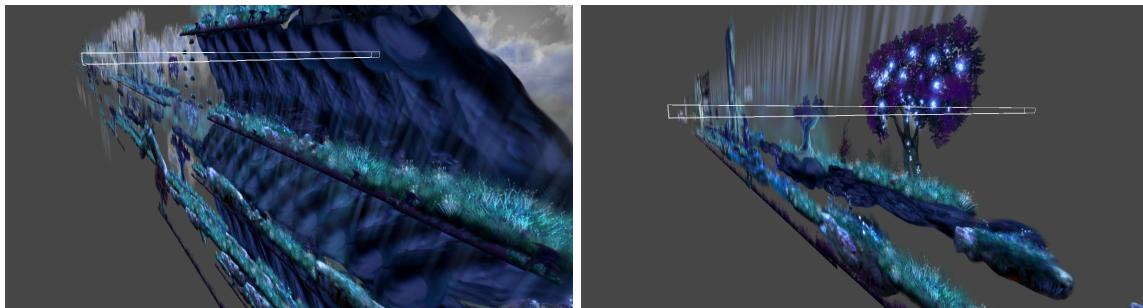


Figure 5.1: First image consists of 16 layers to give parallax effect. Second image consists of only 4 layers to give the same effect

In our game at-first we tried to use perspective camera to create the parallax effect. Although it worked, it also raised some performance issues for android devices. Thus, we switched into writing custom scripts for the camera to move different layers at different speed to give off the illusion of parallax effect as shown in figure 5.1. This not only optimized the game's performances as it requires less layers and it also resulted in making the management of the scene layers much more easier.

5.5 Visual Effects

To improve visual effects and develop custom shaders in our game, we used Unity's Shader Graph and VFX Tool. Shader Graph is a node-based visual tool that allows us to design and edit shaders without having to write code. By linking nodes that represent various shader attributes and processes, we could quickly construct complex and cool materials and effects.

In contrast, the VFX Tool allowed us to build and simulate particle effects and other visual components in real-time. By altering factors such as size, color, behavior, and emission patterns, we could construct and personalize particle systems. This program simplified the

process of producing amazing and dynamic visual effects that brought depth and authenticity to our game.

5.6 Modular Characters

We used AI technology in our game to create our primary character model and its four various power versions as shown in figure 5.2. We could only generate a picture of our player model which was then imported to Krita a photo editing software and then each of the model's body parts are separated into different sprites which then further exported to Unity and then rigged and readied to be able to animate it. Using AI for character development had various advantages. When compared to traditional manual modeling, it allowed us to effectively explore a wide variety of design alternatives while saving time and money. While keeping to the desired appearance and style of our game, the AI model developed unique versions. Furthermore, AI-based character generation allowed us to introduce our main character's four various power versions. We constructed multiple versions that depict the character's varied skills or power levels by modifying the created basic model, such as changing dimensions, adding or removing components, or applying new textures.

In our game, each power variants are unlocked after defeating each respected attributed zone bosses. These power versions of the main character not only has its own set of powers, but it also strengthens the fundamental hero abilities, making them even more dangerous. For example:



Figure 5.2: Base player model and 4 different hero modes

The Fire variant is built to survive high temperatures and flourish in lava conditions without suffering health reduction or any damage. Aside from the fire resistance, the character's fundamental abilities, such as strength and damage, are improved. The character's hands are able to produce flames, allowing them to hurl fire missiles or conduct strong close-range assaults. This variation excels at providing fire-based damage, making them especially useful against heat-sensitive foes.

The Water variant of the base model gives the character the capacity to breathe underwater indefinitely. This not only lets them to freely explore aquatic areas, but it also improves their swimming speed and maneuverability. Water-based attacks and defensive skills can be added to the character's fundamental powers. They may be able to call water barriers or missiles to assault foes, allowing them to control the course of battle.

The Earth variant of the base character can climb walls, giving them a major edge while crossing vertical situations allowing the player to reach previously unexplorable areas. Furthermore, their fundamental physical strength and endurance are improved. The character might learn to conjure earth-based structures or launch rock missiles. These talents allow them to be competent in both defense and offense, since they can erect barriers and perform devastating ground-based assaults.

The Thunder variant of the base model gives the player the ability to super leap or floating ability, allowing them to reach higher platforms or hover in the air. Their fundamental qualities, such as speed and reflexes, are strengthened, allowing them to be fast and elusive in battle. This variation may be capable of using electric-based attacks, such as unleashing bolts of lightning or creating electric fields that injure and paralyze enemies.

We hoped to give players with a diverse and exciting gameplay experience by introducing these distinct abilities into each power type. Each variety not only thrives in unique places or situations, but also strengthens the original character's primary talents, transforming them into a powerful force to be reckoned with in their own realms.

5.7 World Bosses

As mentioned earlier, The power versions of the main character in our game are unlocked by conquering their corresponding ascribed zone bosses, often known as world bosses shown in figure 5.3. These world bosses are tough opponents, posing incredibly demanding and intense engagements that put players' skills and strategic talents to the test. Each zone boss is custom-made to fit the theme and qualities of the power variety it protects. These bosses have great power, unique skills, and sophisticated attack patterns, necessitating the study of their actions and the development of successful techniques to destroy them.



Figure 5.3: World Bosses

Defeating a world boss is a difficult job that marks a key milestone in the game's growth. As these will unlock the specific power variants of a player, it will allow them to reach previously inaccessible areas. Players must master their character's talents while also utilizing their own skills and expertise of the game mechanics to overcome these challenging encounters.

As already mentioned, successfully defeating a zone boss grants the player the associated power variation, allowing them to access a new range of skills and gameplay options. This progression system adds complexity and replay value to the game by motivating players to take on more difficult tasks, better their abilities, and acquire all possible power types. The toughness of these world bosses is purposefully designed to deliver a high degree of challenge as well as a sense of success upon victory. Because of its intimidating character, players are encouraged to think, adjust their play style, and perhaps think of the order of defeating each zone boss to acquire specific power variants to counter contrasting attributes.

We hope to create a gratifying and interesting experience for gamers who prefer hard encounters and want to explore the full depth of our game by including tough world bosses as gatekeepers for obtaining power variations and various unreachable areas that are unexplorable without specific abilities which are acquired from those different power variants.

5.8 World Map

Our game has a large and immersive world map that provides players with a varied and compelling gameplay experience. The globe map has been painstakingly constructed to include a range of varied places, each with its own particular environment, landmarks, and

problems. Players will discover beautiful woods alive with species, towering mountains cloaked in mist, wide lava zones with deadly volcanoes, and fascinating underground caverns packed with ancient mysteries as they travel the world map. Beautiful visual elements, dynamic weather systems, and day-night cycles bring the landscapes to life, adding depth and authenticity to the universe. Exploration is important in the game, and the global layout is meant to reward daring explorers. Those that wander off the usual route will find hidden passageways, secret locations, and rich supplies. Players can find secret ruins or powerful items that improve their character's skills.

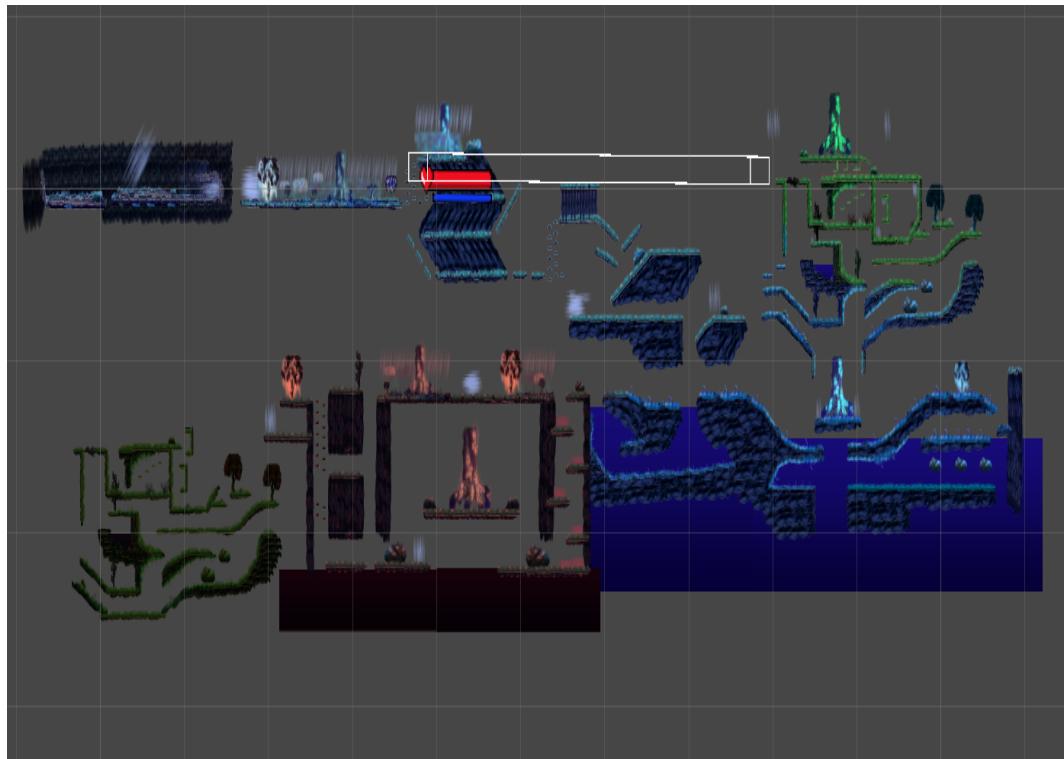


Figure 5.4: World Map

As shown in figure 5.4 our vast world map is divided into numerous diverse zones, each with its own set of difficulties, aesthetics, and exploring chances. These zones are meticulously designed to give gamers with a variety and engaging gameplay experience. Here are some quick summaries of the previously stated zones:

The Fire Zone is characterized by volcanic scenery, flowing lava, and tremendous heat. Extreme environmental risks, such as molten rock rivers and exploding geysers, will befall the players. It is populated by fire-based animals and is guarded by a powerful fire world boss. Players may only obtain the fire power variety by overcoming the difficulties of this zone and conquering the boss.

The Water Zone transports players to a spectacular aquatic environment complete with immense oceans, underground tunnels, and vivid marine life. Players may explore the depths, uncovering secret treasures and meeting unusual aquatic species. To reach the water world

boss, who controls the secrets of the water power variety, they must endure tremendous currents and negotiate complicated underwater mazes.

The Earth Zone is characterized by harsh terrain, steep cliffs, lush woods, and twisting routes. Players will confront difficulties such as steep inclines and perilous terrain. Earth elements and other nature-based species live in the earth zone. At the center of this zone, the earth world boss awaits, challenging players with their great power and earth-shattering skills. Victory against this monster grants access to the earth power variation.

The Thunder Zone consists of a stormy sky, craggy peaks, and crackling energy describe the thunder zone. Players will have to traverse through violent thunderstorms while using lightning-charged platforms and overcoming electric obstacles. Thunder-based opponents represent a significant danger. The thunder world boss, located at the apex of this zone, is a fearsome opponent whose defeat gives players the thunder power variant.

In summary, each zone has its own visual appeal, environmental dangers, and monster kinds, providing players with a different and difficult experience as they continue through the game. Navigating these zones, overcoming challenges, and killing their formidable world bosses are critical steps towards unlocking and unleashing the full potential of the main character's several power variations and also reaching the end story of our game.

Chapter 6

Game Assets

Our assets in our game are divided into three categories: items purchased from retailers, objects generated by AI, and assets created by ourselves. This combination enables us to retain a varied choice of high-quality content while balancing efficiency and creative control. Assets purchased from shops are pre-made digital assets obtained through third-party markets or asset stores such as Unity Asset Store [9] or itch.io [10]. 3D models, textures, sound effects, music tracks, and other items that improve the visual and aural experience of the game are examples of assets. We save time and money by utilizing the knowledge of professional artists and musicians by acquiring these assets. Such as assets shown in figure 6.1 are bought from various assets stores online.

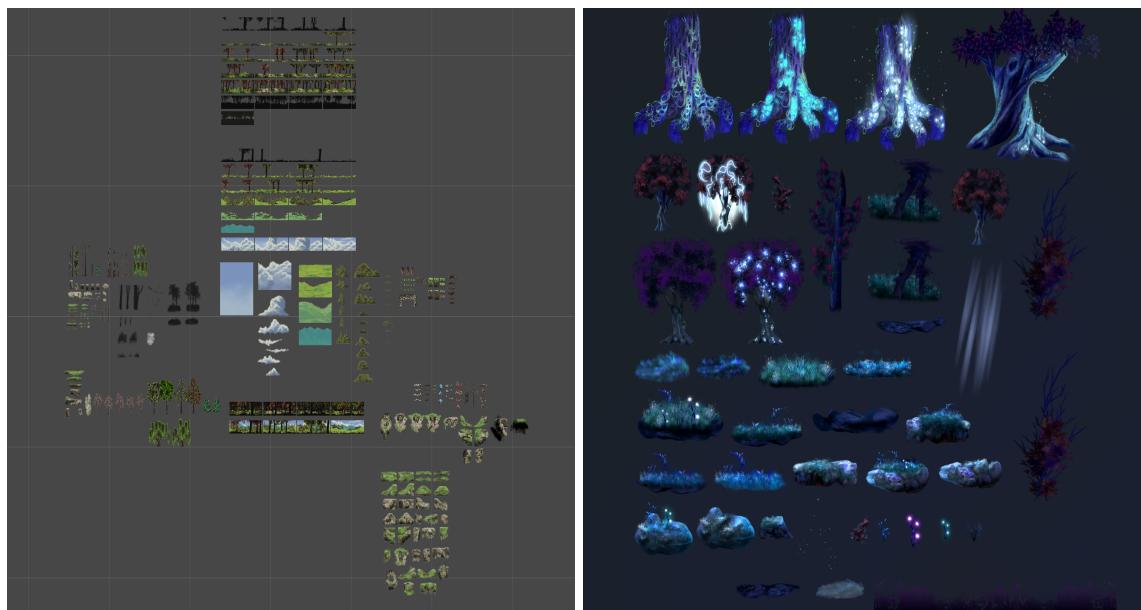


Figure 6.1: Sample of Assets that were purchased from store

AI generated assets entail the use of artificial intelligence algorithms and technologies to build specific character models as shown in 5.2. AI-generated character models, landscape

texturing, and procedural level generation are examples of this. AI-based asset production enables us to swiftly develop a wide range of material with distinct variants and styles, allowing us to engage in creative exploration and efficiency that traditional manual creation may not provide. In our game however we only used AI generated character models. Our team's created assets are generally used in UI and in level designing and character customization. It is the product of our own aesthetic and technological abilities. This also contains environment creations, animations, and other visual assets. These materials provide us complete control over the game's creative vision, ensuring that it matches our intended aesthetics and gameplay mechanics.

We tried to create a well-rounded and visually appealing gaming experience by mixing content from these three areas. Assets purchased from retailers provide us with professional-quality material, assets generated by AI gives us efficiency and unique variants, and assets made by our team allow us to incorporate our own creativity and vision into the game. This mix of asset sources enables us to provide our users with a fascinating and immersive gaming experience.

Chapter 7

Limitations and Future Works

The development of any game project is not without constraints and hurdles. In this chapter, we look at the limits we faced while developing our game, Lost Forest, as well as possible areas for future improvement and growth.

7.1 Limitations

During the making of our game, we faced several constraints throughout the whole development process, which influenced the development process and the final output. The limitations are discussed as followed:

Limited Resources: We had limited resources as a smaller team in terms of labor, time and funding compared to any game development studios. This limited our capacity to tackle complicated features or tasks to achieve a level of polish similar to larger and more established studios.

Lack of Expertise: Our team was made up mostly of people without any prior knowledge in game development and Unity. We were still learning and gaining expertise in various elements of game development using Unity. Due to a lack of competence in certain areas, such as character and environment design, obtaining the needed degree of inventiveness and aesthetic appeal proved difficult.

Performance Optimization: We had to prioritize speed optimization because we wanted to distribute the game on various platforms, including Android. This entailed carefully monitoring resource utilization, simplifying key game components, and optimizing code to ensure seamless gameplay experiences even in low-end devices.

Time Constraints: Developing a detailed game takes long time even for well established studios that have earned their stripes in this industry, and thus our team encountered time

limits owing to a variety of issues. Also, balancing development needs with other obligations caused difficulties in terms of achieving deadlines and completely exploring all creative potential.

Quality Assurance: Conducting extensive quality assurance and testing is more difficult with a smaller staff. Due to limited resources, it was impossible to discover and resolve all potential flaws, which may have resulted in certain faults being included in the final product.

7.2 Future Works

We have the opportunity to collect vital feedback from players following the initial release of our game and the early alpha access phase. This input becomes an invaluable resource for identifying areas for development and potential future work.

Level design is one area that may be improved depending on user feedback. We may gather insights into which levels resonate best with players and which may require more modification by carefully listening to their experiences and evaluating their input. This feedback can help us improve the entire gameplay experience by fine-tuning current levels, adding new challenges, and introducing novel gameplay concepts.

Furthermore, the post-publishing process allows players to fix any technical or performance concerns they may have experienced. We can optimize the game's performance, provide a smooth gameplay experience, and improve overall stability by thoroughly studying and fixing these issues.

Chapter 8

Conclusion

Throughout the production process, we focused on producing a fascinating plot, exhilarating action scenes, hard puzzles, complex game mechanics and aesthetically appealing visuals throughout the production process. The parallax effect was used to create depth to the 2D world, giving players a feeling of immersion and improving the entire visual experience. While the game differs from well-known games such as Ori and the Blind Forest, Hollow Knight, and Skull, it is a monument to the work and passion of a small team of young game creators. From obtaining and preparing sprites to character and environment construction, frontend and backend implementation, game polishing, and eventually publication, the development process followed a disciplined manner.

Although we admit that there have been areas where we lacked experience, such as character and environment design, nevertheless we worked hard to provide gamers with a captivating and exhilarating gaming experience. Despite its tiny size in comparison to renowned companies, Lost Forest demonstrates the our dedication to developing a meaningful and engaging game. It is still a game demanding continuous updates and feed-backs to smooth out any issues or to introduce a player demand, as we know good post publishing service also impacts a lot for reputation building of a game development company.

Ultimately, Lost Forest demonstrates how Unity as a game engine, when paired with the team's creativity and tenacity, may result in the construction of a gripping 2D action-adventure game despite the size of the development team. In other words, it's highlighted how easily Unity provides anyone with the capability to create something creative and set their mark on this world. The release of the game intends to delight players, immerse them in a rich gaming world, and leave a lasting impression by providing a unique experience that ranks with other renowned products in the genre.

References

- [1] “Unity real-time development.” <https://unity.com/>. Accessed: 2023-05-17.
- [2] “Krita | digital painting. creative freedom..” <https://krita.org/en/>. Accessed: 2022-12-22.
- [3] “Brackeys.” <https://brackeys.com/>. Accessed: 2023-05-01.
- [4] “Code monkey | learn unity game development.” <https://unitycodemonkey.com/>. Accessed: 2022-08-11.
- [5] “Thomas brush | atmos games.” <http://atmosgames.com/>. Accessed: 2023-02-07.
- [6] “Skull: The hero slayer on steam.” https://store.steampowered.com/app/1147560/Skul_The_Hero_Slayer/. Accessed: 2022-08-20.
- [7] “Hollow knight.” <https://www.hollowknight.com/>. Accessed: 2023-08-20.
- [8] “Ori and the blind forest.” <https://www.orithegame.com/blind-forest/>. Accessed: 2022-08-20.
- [9] “Unity asset store.” <https://assetstore.unity.com/>. Accessed: 2023-05-17.
- [10] “itch.io | indie game asset store.” <https://itch.io/game-assets/free/genre-platformer/tag-unity>. Accessed: 2023-05-17.

Generated using Undegraduate Thesis L^AT_EX Template, Version 1.4. Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh.

This thesis was generated on Wednesday 24th May, 2023 at 5:31pm.