EcoVR: A Virtual Ecosystem Simulator

Toh Jing Hua (U2121032L)

Noel Wee Ee Cher (U2121856B)

Leong Mininn Miko (U2223270F)

Word Count: 4974

SC4060: Virtual and Augmented Reality

College of Computing and Data Science

April 4, 2025

Contents

Contents i

1 Introduction 1

2 Literature Review 3

2.1 Benefits of Simulations 3
2.2 Real-Life Examples of Virtual Reality Simulation 4
2.3 Gap in Existing Solution 5

3 Implementation 6

3.1 Tech Stack 6
3.2 Assets Used 7

4 Game Features 8

4.1 Main Menu Scene 8
4.2 Game Scene 9
4.3 Tutorial System 12
4.4 Natural Birth and Death (Conway's Game of Life) 13
4.5 Predator–Prey Dynamics (Animal Intelligence) 16
4.6 Hunger and Hydration System 20
4.7 Detection Radius Visualisation 22
4.8 Weather System 24
4.9 Binoculars and Magnifying Glasses 26
4.10 Feeding and Water Interaction 28

5 Analytics and Visualization 30

5.1 Analytics Scene 30

5.2 Ecosystem Data Analytics 32

6 Survey Results **36**

7 Conclusion **39**

7.1 Challenges and Solutions 39

7.2 Limitations and Future Work 40

7.3 Conclusion 42

Chapter 1

Introduction

Figure 1.1: A person exploring a forest ecosystem in virtual reality. Image generated using OpenAI's GPT-4o [1].

Imagine stepping into a vibrant jungle, witnessing the intricate dance between predator and prey, and observing the cascading effects of introducing a new species — all within a virtual world. This is the power of ecosystem simulation models, especially when enhanced by the immersive capabilities of virtual reality (VR) technology. These tools are transforming the way we study and teach the complexities of ecological communities, making abstract concepts tangible and engaging.

Ecosystem simulations play a crucial role in understanding the dynamic interactions that define ecological systems. By replicating real-world environments, they allow us to examine patterns such as predator-prey relationships and the consequences of environmental changes or species introductions. With the rapid advancement of VR, these simulations have become more accessible, interactive, and impactful across educational and research contexts.

This report aims to:

1. Explore the benefits and limitations of ecosystem simulations,
2. Highlight how VR can be integrated to enhance their design and effectiveness, and
3. Introduce EcoVR — our immersive VR tool that demonstrates the potential of these simulations as powerful educational experiences.

Chapter 2

Literature Review

This section will delve into the benefits of utilising Virtual Reality simulations for ecosystem research, as well as identify gaps in current solutions which EcoVR aims to resolve.

2.1 Benefits of Simulations

Time and Cost Efficiency: Simulations offer greater time and cost efficiency compared to their real-world experimental counterparts, especially when studying long-term or large-scale ecological processes [2, 3]. For instance, researchers would be able to simulate the effects of climate change on a forest ecosystem over decades, all in a fraction of the time it would take to observe these changes in the real world.

Furthermore, conducting real-world experiments may result in detrimental and irreversible impacts on the environment. However, simulation mitigates these concerns by allowing experimentation in a controlled, reversible and environmentally-safe manner.

Forecasting Ability: Simulations are an excellent tool for predicting the effects of environmental changes, human activities, and ecological management strategies, which helps to drive decision-making and conservation efforts. Keane and Karau [4] leveraged on simulation for wildfire prediction, enhancing fire management within ecosystems in turn. Moreover, Pitcher, Buchary, and Hutton [5] were successful in forecasting resource and fishery responses to artificial reefs deployed within no-take marine protected areas.

Improve Student Learning Initiative Carey et al. [6] highlighted that simulations, when used as a platform for hypothesis-driven scientific inquiry, allowed students to gain a greater understanding of ecosystem modeling. This was possible as the simulation enabled them to test predictions and observe outcomes within a controlled setting.

2.2 Real-Life Examples of Virtual Reality Simulation

Soft Skills and Professional Training: According to Talespin [7], VR is effective in professional training involving human interaction and soft skills development. In simulated scenarios, users interact with emotionally responsive virtual characters, which enhances realism and engagement. This emotional authenticity encourages users to fully participate without fear of judgment, making it easier to reflect on their decisions and receive real-time feedback. As a result, training sessions become more meaningful and impactful, helping users build confidence and internalize lessons more effectively.

Immersive Environment Simulation: HigherEchelon [8] highlights how VR simulations offer highly immersive and customizable environments that replicate real-world conditions with accuracy. These simulations are not only cost-effective but also accessible, they can be conducted from a single room while maintaining safety and control. Every interaction within the simulation can be recorded and analyzed to refine future training programs, allowing for continuous improvement and measurable learning outcomes.

Additional Use Cases:

- Zhu and Han [9] explores how VR is used to monitor marine environments.
- Thornley [10] investigates how VR is used to study grassland dynamics.
- Chandler et al. [11] discusses how simulations are used for woodland exploration.

App	Environment	VR?	Animal Interaction?
Zhu et al. [9]	Marine	Yes	No
Thornley [10]	Grasslands	No	No
Chandler et al. [11]	Woodlands	Yes	No
EcoVR	**Forest, Marine, Snow**	**Yes**	**Yes**

Table 2.1: Comparison of our solution compared to other existing simulators

2.3 Gap in Existing Solution

Our research identified a notable gap in the current landscape of ecosystem simulation tools. While existing VR simulations focus on marine environments [9] and woodland exploration [11], and non-VR simulations have been developed for grassland ecosystems [10], there is a significant lack of VR-based simulations specifically designed to study animal interactions within forest ecosystems.

This absence highlights a missed opportunity to leverage immersive technologies for ecological education and behavioral observation in forest settings. To address this, we developed *EcoVR*—a VR-based forest ecosystem simulator that enables users to observe and analyse animal behavior, species interaction, and environmental dynamics in a rich, interactive forest environment.

Chapter 3

Implementation

3.1 Tech Stack

EcoVR was developed using the Unity game engine and C#. Details are as follows:

- **Unity:** Used as the core development platform for creating and managing the VR scenes and assets.
- **C#:** Used to implement the core logic of EcoVR, including features such as climate, animal intelligence and analytics.
- **XR Device Simulator:** Used during development for testing and debugging VR interactions without needing a physical headset at all times. This helped accelerate the iteration process.
- **Meta Quest 3 (via Meta Quest Link):** Served as the primary target hardware for testing and deploying EcoVR.
- **Git:** Used for version control and collaboration. Git enabled efficient parallel development, change tracking, and team coordination throughout the project.

3.2 Assets Used

To enhance the visual quality and immersion of our VR application, we incorporated a variety of high-quality 3D assets from the Unity Asset Store across different scenes:

- **Sci-Fi Styled Modular Pack** [12]: Utilised in both the ANALYTICS SCENE and the MAIN MENU SCENE to construct a sci-fi laboratory environment. This asset helped convey the futuristic theme of a research facility dedicated to animal behavior analysis.
- **Stylised Nature Kit Lite** [13]: Integrated into the ANALYTICS SCENE to enrich the visual narrative with natural elements such as stylized trees, rocks, and foliage. These elements create the impression of returning from the forest to the lab, bridging outdoor exploration and indoor analysis experiences.
- **Dreamscape Nature Mountains** [14]: Employed in the GAME SCENE to create a rich and realistic forest setting where the ecosystem simulation unfolds.
- **Tenkoku Dynamic Sky** [15]: Leveraged in the GAME SCENE to simulate dynamic weather patterns and realistic skies, contributing to the overall immersion and variability in the ecosystem environment.
- **Animal Models** [16, 17]: A collection of animal models implemented in the GAME SCENE to populate the virtual ecosystem. These models interact with each other based on scripted behaviors, allowing users to observe emergent patterns and simulate how different species coexist within the environment.
- **Low Poly Food Lite** [18]: Used in GAME SCENE to feed the animals to reduce their hunger level and increase their hydration level, ensuring survivability.

Chapter 4

Game Features

4.1 Main Menu Scene

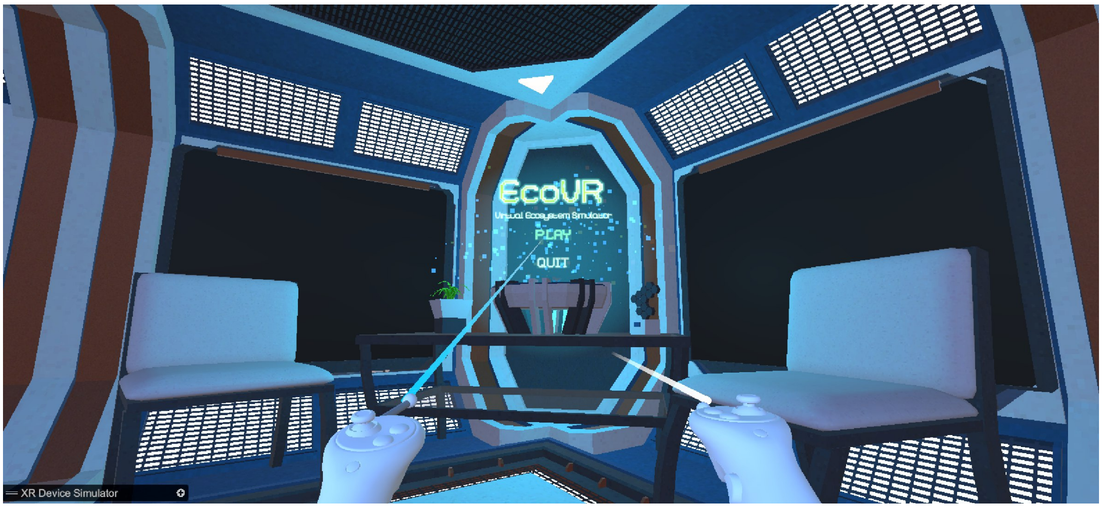

Figure 4.1: Main Menu Scene.

The Main Menu Scene establishes the tone for the simulation by placing users inside an immersive, sleek and high-tech pod set in a stylised desert landscape. The dramatic visual shift from the Main Menu's barren desert to the lush forest setting of the main Game Scene, was implemented to enhance the user's feeling of discovery as he transitions between different virtual environments.

Illustrated through a holographic projection at the centre of the pod is EcoVR's Main Menu. The Main Menu contains a start button which features proximity-based prompts that will serve to guide users closer for interaction. EcoVR's Menu interface with its intuitive and visually appealing design ensures it remains easy-to-use, while value-adding to the overall user experience.

4.2 Game Scene

The Game Scene offers the user two mode options: *Bird's Eye View* and *First-Person View*. Each mode has its own user interface and interaction model.

Bird's Eye View Mode

Figure 4.2: User interface of the bird's eye view mode.

In the bird's eye view (Figure 4.2), users are placed in a high-altitude position, allowing them to observe the entire ecosystem from above. UI elements are overlaid on the screen to enable direct interaction with the simulation parameters. The set of said elements consist of:

- **Spawn Buttons:** Buttons are provided to spawn new predator or prey animals into the scene, giving users the ability to test how population changes affect ecosystem dynamics.
- **Weather Control:** Users are given the option to instantly alter the current

weather, influencing the environment and animal behavior in turn, through a set of dedicated buttons (e.g. sunny, rainy, foggy).

- **Time Speed Slider:** A time scale slider lets users adjust the speed at which time progresses in the simulation, ranging from real-time to accelerated rates. This feature allows for quicker observation of long-term population trends and behaviors.
- **Zoom Slider:** Additionally, the camera zoom slider enables users to smoothly zoom in and out while in bird's eye view, allowing them to either focus on specific regions of the terrain or get a broader overview of the simulation.

This mode is ideal for macro-level ecosystem observation, experimentation, and rapid scenario testing.

First-Person View Mode

Figure 4.3: User interface of the first-person mode.

In the immersive first-person mode, users are placed at ground level within the forest environment, assuming the role of an observer (Figure 4.3). The UI is minimal to preserve immersion, and interaction is primarily through physical movement and gestures.

In this mode, users are able to:

- **Move:** Navigate the terrain freely using Unity's locomotion system to explore different areas of the forest.
- **Use Tools:** Equip and use binoculars or magnifying glasses to observe animals from afar or up close, respectively.
- **Feed and Hydrate Animals:** Interact directly with animals by offering them food or water, influencing their hunger and hydration levels and observing the behavioral outcomes.

This mode provides a more personal and exploratory perspective, allowing users to engage directly with the ecosystem and observe animal behavior in real-time.

4.3 Tutorial System

Figure 4.4: Tutorial system.

To guide users through the initial process, we implemented an interactive tutorial that appears the first time a user enters the GAME SCENE in the *Bird's Eye View Mode* (Figure 4.4). The tutorial introduces the user to the core UI elements and interaction tools available in this mode, ensuring they can effectively control and experiment with the ecosystem from a top-down perspective.

The tutorial teaches users to:

- Spawn predators and prey using the dedicated buttons.
- Change the weather with the weather control buttons.
- Adjust the time speed using the slider at the bottom.
- Zoom in and out using the vertical slider on the left.

Each feature is visually highlighted, and users can advance through the tutorial at their own pace.

4.4 Natural Birth and Death (Conway's Game of Life)

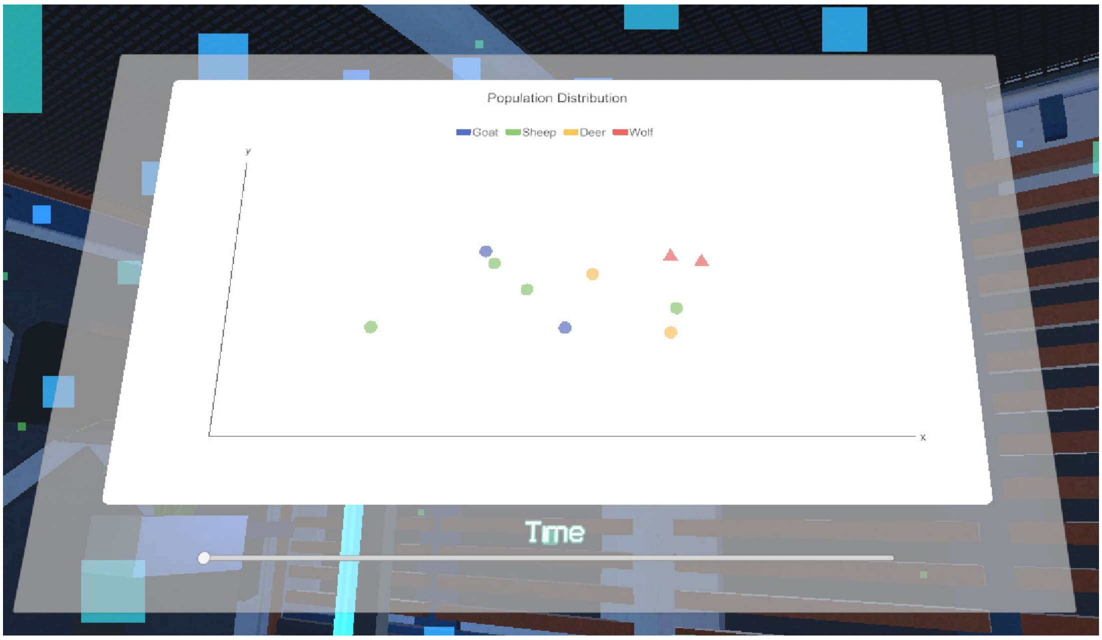

Figure 4.5: Population distribution heatmap. Circle indicates prey, triangle indicates predator.

Inspired by the rules governing Conway's Game of Life, we implemented a population control system that models natural birth and death processes based on spatial and social dynamics. This system is encapsulated in the `ConwayPopulationGrowth` script, which governs both prey and predator populations in the simulation. Every 10 seconds, the system evaluates the current state of all animals in the scene and applies biologically inspired rules to determine whether an animal should die or reproduce.

Death Conditions

Animals can die due to either overpopulation or underpopulation:

- **Death by Loneliness (Underpopulation):** Animals with fewer than a threshold number of nearby companions (within a defined radius) are considered too isolated to survive. If this condition is met, the animal has a high probability (80%) of dying due to loneliness. This models real-world phenomena such as social dependence and vulnerability in isolation.

- **Death by Competition (Overpopulation):** If too many animals of the same species are present within a consumption radius, it is assumed they are competing for the same limited resources (e.g., grass or prey). When the number of competitors exceeds a preset threshold, the animal dies from overpopulation pressure.

Birth Conditions

Animals may reproduce when a sufficient number of reproduction mates are nearby:

- **Reproduction by Proximity:** If animals of the same species are found within a reproduction radius and their count exceeds a defined minimum, reproduction may occur with a given probability (50%). This spatial requirement ensures that reproduction only occurs in localized population clusters.

Species-Specific Controls

The system handles prey and predator species separately, with different parameters tuned for each:

- **Prey:** Smaller competition and reproduction radii reflect the more localized behavior of herbivores. The competition threshold is higher, modeling the abundance of plant resources.
- **Predators:** Larger radii and lower competition thresholds represent the territorial nature and resource scarcity among carnivores.

All spatial computations are handled using Unity's physics engine through `Physics.OverlapSphere`, which performs an optimized spatial query to return all nearby colliders within a defined radius. This approach is more efficient than iterating over all animals in the scene and computing pairwise distances to determine proximity. While the latter has a time complexity of $O(n^2)$ for n animals, the physics-based spatial query scales more favorably, especially with Unity's built-in spatial partitioning (e.g., bounding volume hierarchies), making it ideal for real-time simulations with potentially large populations.

This system adds a layer of emergent behavior to the simulation, enabling natural population dynamics without hard-coded lifespans or scripted events. It also provides a foundation for observing how environmental conditions and animal attributes influence survival and reproduction.

4.5 Predator–Prey Dynamics (Animal Intelligence)

Figure 4.6: An intense moment when a wolf hunts a sheep.

To simulate realistic ecosystem behavior, we implemented predator–prey interactions through custom AI systems for both predators and prey (Figure 4.6). These agents operate autonomously using a finite state machine (FSM) structure that governs their behavior, perception, movement, and interaction.

Predator Behavior

The `PredatorAI` script models predatory behavior through a sequence of states: *Idle*, *Walking*, *Hunting*, *Attacking*, and *Dead* (Figure 4.7). Predators regularly scan their surroundings using a configurable detection radius. When a prey is detected within range, the predator transitions from passive states into the *Hunting* state and pursues the prey using Unity's NavMesh system. Upon reaching attack range and clearing a cooldown, the predator transitions into the *Attacking* state to engage its target.

Key mechanics include:

- **Hunger and Thirst Simulation:** Predators have a satiety and hydration bar that will deplete over time. The rate of depletion can be affected through user

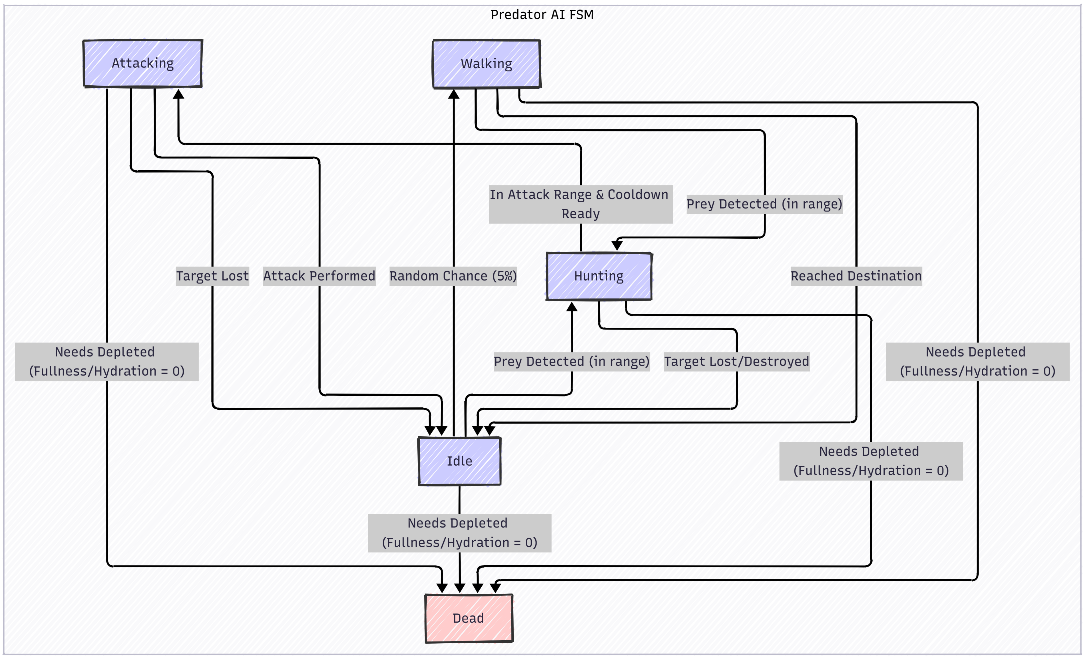

Figure 4.7: Finite state machine diagram of Predator AI.

determined weather conditions as well. If left unfed for several in-game days (configurable), the bar will fully deplete, resulting in death from starvation or dehydration. This feature introduces a natural population control mechanic.

- **Attack Logic:** When a predator successfully attacks, the prey is removed from the simulation, and the predator's hunger state is reduced accordingly. This resets the hunger and survival timer, mimicking real-world feeding behavior.
- **Target Selection:** Predators dynamically choose the nearest prey within their detection range using a spatial distance check.

Prey Behavior

Prey agents, implemented via the `PreyAI` script, exhibit autonomous roaming and foraging behavior as well as evasive responses to nearby predators. Their state machine includes: *Idle*, *Walking*, *Eating*, *Running*, and *Dead* (Figure 4.8).

Key features of prey behavior include:

- **Exploration and Foraging:** In passive states, prey choose random destinations to walk or rest, sometimes transitioning into an *Eating* state to simulate grazing.

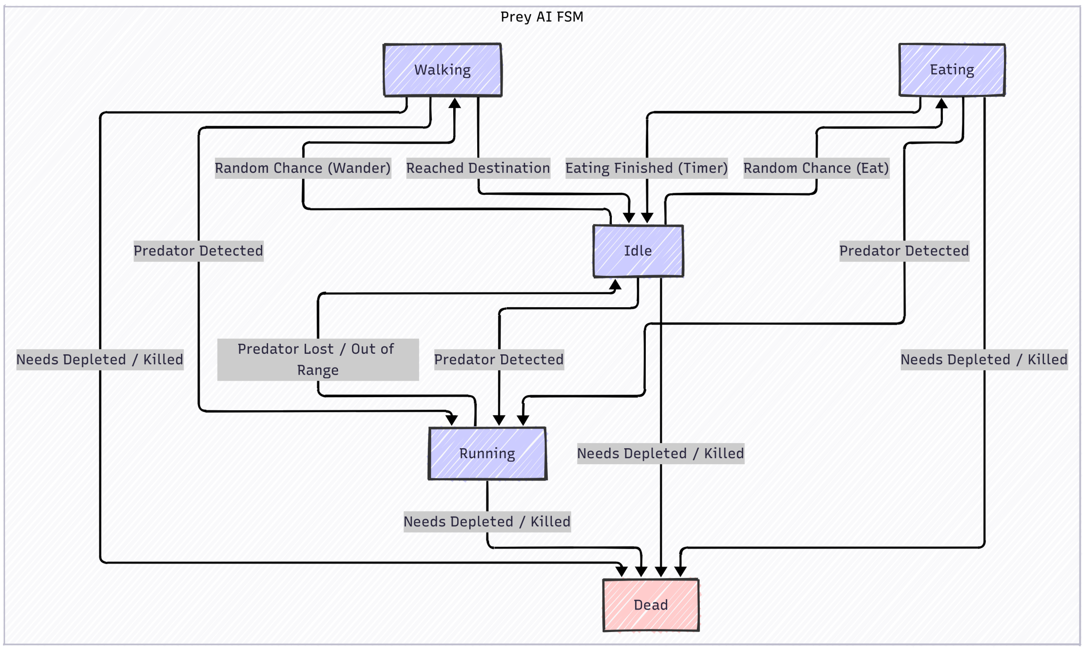

Figure 4.8: Finite state machine diagram of Prey AI.

- **Predator Detection and Escape:** When a predator enters the prey's detection radius, the prey enters the *Running* state and flees in the opposite direction, using a directional vector away from the threat to compute a flee destination.
- **Animation Variability:** To enhance realism, multiple idle and locomotion animations are selected randomly, contributing to diverse and lifelike movement patterns.

Emergent Ecosystem Behavior

The predator-prey agents interaction represent a dynamic feedback loop:

- **Predators rely on prey for survival**, and must successfully hunt at regular intervals to avoid starvation.
- **Prey must avoid predators** by maintaining awareness and fleeing strategically, which in turn affects their survival rates and opportunities to forage or reproduce.
- **Population equilibrium naturally emerges** as high predator density increases hunting pressure, while low prey availability leads to predator starvation and die-off.

This system introduces ecologically grounded dynamics into the simulation, where population trends and spatial movement patterns are not pre-scripted but arise from simple, rule-based agent interactions. The result is a rich, self-balancing ecosystem that reflects basic principles of predator–prey relationships in nature.

4.6 Hunger and Hydration System

Figure 4.9: An in-game showcase of the hunger and hydration system interface.

To further enhance the realism of animal behavior and survival dynamics, we implemented a hunger and hydration system that models basic biological needs. These systems introduce continuous pressure on animals to forage and seek shelter, contributing to more lifelike and emergent interactions.

Hunger Mechanism

Each animal tracks its hunger level over time. If the animal does not eat within a certain duration, its hunger level increases progressively through predefined stages. In the case of predators, failing to successfully hunt and consume prey causes them to eventually starve and die, as managed by a hunger timer in the `PredatorAI` script. This creates natural consequences for failed hunts and helps regulate predator population size.

The hunger system also impacts behavior:

- Predators become more aggressive and are more likely to enter the `Hunting` state as hunger levels rise.
- Herbivores prioritize grazing behavior and may shift away from roaming or idle

states if hunger increases.

Hydration Mechanism

In addition to hunger, animals are affected by environmental conditions—specifically, weather and sunlight. When the environment becomes sunny and hot, their hydration levels decrease.

If hydration drops too low and is not replenished, the animal will eventually die from dehydration. This mechanic introduces a link between climate conditions and survival, reinforcing the importance of environmental balance in the ecosystem.

Emergent Effects

Together, hunger and hydration systems:

- Encourage animals to move and act purposefully within their environment.
- Regulate population levels through natural consequences like starvation and dehydration.
- Create deeper interdependence between animals and their surroundings, particularly in relation to food availability and weather systems.

These mechanics add a layer of complexity that aligns with real-world biological constraints, reinforcing the ecological focus of the simulation.

4.7 Detection Radius Visualisation

Figure 4.10: A top-down view of predators and preys with their detection radius.

To help users better understand how animals perceive and respond to their environment, we introduced a detection radius visualisation system in the GAME SCENE (Figure 4.10). This feature allows users to see the perception range of each animal, providing insight into predator–prey interactions and how proximity influences behavior.

When activated, the simulation overlays transparent colored zones around each animal to represent its detection radius:

- **Red** indicates the detection radius of predators.
- **Blue** indicates the detection radius of prey.

These visual cues help users observe how close animals need to be before reacting—for example, a predator starting to chase, or prey beginning to flee. It also makes it easier to see how environmental factors like fog or weather reduce visibility and affect interaction dynamics.

The detection radius visualisation can be toggled on and off using a button in the top-center section of the UI. This keeps the interface clean while allowing users to turn on the feature when needed for closer inspection.

This tool is especially useful in Bird's Eye View Mode, where users are able to observe the spatial relationships between multiple animals simultaneously. It enhances both the educational and gameplay aspects of EcoVR by turning an otherwise invisible mechanic into an observable and interactive learning opportunity.

4.8 Weather System

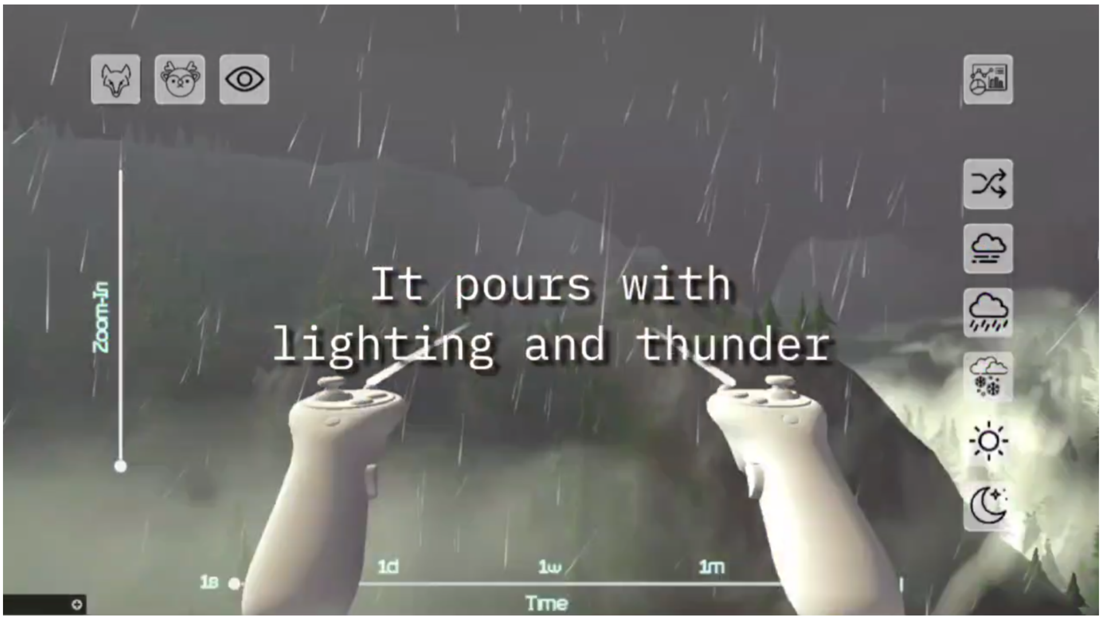

Figure 4.11: In-game showcase of rain and lighting.

To simulate environmental variability and its influence on animal behavior, we implemented a dynamic weather system using the **Tenkoku Dynamic Sky** asset. This system supports a range of weather conditions and time-of-day transitions, introducing new challenges and survival pressures within the ecosystem.

Weather Types

EcoVR includes the following weather conditions:

- **Sunny:** Common during daytime. Under sunny conditions, animals experience an increased rate of dehydration.
- **Stormy:** Increases hydration levels for all animals, simulating natural water replenishment. However, the rain also reduces animals' detection radius and movement speed due to visual and auditory interference.
- **Foggy:** Significantly lowers detection range and speed, affecting both predator hunting efficiency and prey evasion capabilities.
- **Snowy:** Reduces detection radius and movement speed due to cold and difficult

terrain. Snow also decreases the animals' hunger level as they expend more energy.

- **Night:** Reduces detection radius for all animals due to low visibility.
- **Default:** The default option allows users to revert the weather back to its original state, similar to a forest weather setting.

Overall, the weather system adds critical environmental context to the simulation, influencing animal decision-making and survival in ways that mirror real-world ecological challenges.

4.9 Binoculars and Magnifying Glasses

Figure 4.12: In-game showcase of binoculars and magnifying glass.

This feature simulates the use of binoculars and magnifying glasses in VR, enabling users to view distant or small objects with realistic magnification (Figure 5.2), further enhancing the first-person experience. The tools can be toggled using the primary buttons: **A** (right controller) for binoculars and **X** (left controller) for the magnifying glass.

To achieve this, we used a render texture setup powered by a secondary camera.

1. **Secondary Camera:** Positioned in front of the lens and parented to the object to follow hand movement.
2. **Render Texture:** Receives the camera output and projects it onto the lens surface.
3. **Lens Display:** A flat cylindrical mesh on the lens displays the render texture using an unlit material.
4. **Zoom via FOV:** The secondary camera's FOV is adjusted (e.g., 15° for binoculars, 7° for magnifying glass).
5. **Toggle Control:** Devices can be toggled on/off via controller input using Unity's Input System. A `ToggleItemsController` script manages input actions and object activation.

6. **Hand Attachment:** Devices are parented to the VR controllers to simulate holding.

4.10 Feeding and Water Interaction

Figure 4.13: Feeding a sheep an apple and water.

To enrich interactivity and reinforce the biological needs of animals within the simulation, we introduced a feature that allows users to directly feed animals or provide them with water (Figure 4.13). This mechanic complements the hunger and hydration systems, creating a more hands-on and engaging way to influence animal survival and behavior.

User Interaction Mechanics

In the GAME scene, users can now pick up food or water items and deliver them to nearby animals. When an animal consumes food or drinks water:

- Their hunger level is reduced.
- Their hydration level is increased.

Implementation Details

Each food and water object contains a collider and trigger zone that detects when an animal interacts with it. When the animal is within range, it automatically consumes the item, and the associated hunger or hydration values are updated in its script.

These items can be toggled using the secondary buttons: **B** (right controller) for food and **Y** (left controller) for water.

Educational Value

This feature encourages active participation while in-simulation and teaches users, more pertinently children, the value of empathy and care. Users are not only passive observers but can now:

- Test how feeding or hydrating animals influences survival rates.
- Observe changes in behavior when basic needs are met.
- Reflect on the human role in wildlife care and ecological impact.

Chapter 5

Analytics and Visualization

5.1 Analytics Scene

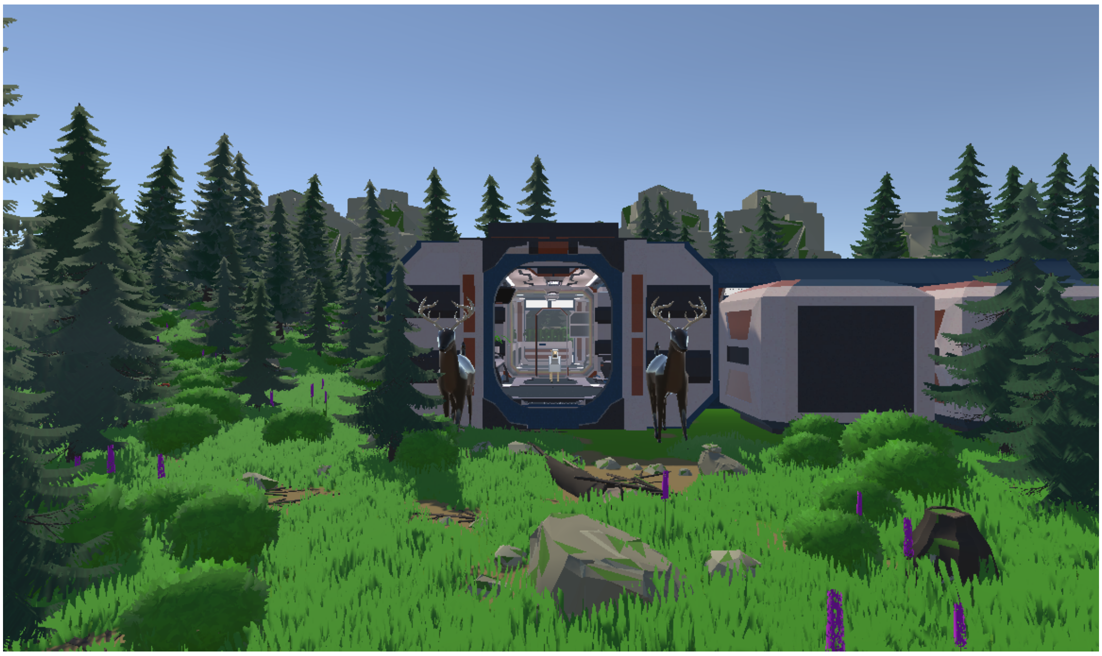

Figure 5.1: User's spawn location in the Analytics Scene.

The Analytics Scene is designed to visualize and interpret animal and population data collected from the simulation. To enhance immersion and maintain EcoVR's strong narrative, the scene is located within a futuristic research environment that blends natural surroundings with advanced data interfaces.

Forest Environment with a Central Pod

Users are spawned within a stylized forest environment, directly facing a high-tech observation pod. This setup serves as a natural transition from the GAME SCENE to the ANALYTICS SCENE by maintaining thematic continuity.

Animals Inside the Pod

Inside the pod, various animal models are placed within visible areas, simulating the interior of a wildlife research facility. These animals represent the species encountered in the simulation, and their inclusion reinforces the perception that users are now studying the subjects in a more scientifically controlled setting, enhancing authenticity and engagement in the process.

Futuristic Lab with Analytics Panels

The pod's interior contains futuristic lab, designed with clean metallic surfaces, illuminated accents, and interactive analytics panels arranged along the walls. Each panel displays a specific visualisation such as population distribution and predator-prey ratios. The user can move around the pod to view and interact with different panels, promoting an exploratory approach to understanding the data.

5.2 Ecosystem Data Analytics

Figure 5.2: Analytics lab.

To support data-driven exploration of animal behavior and population dynamics, we developed an in-game analytics system with real-time tracking and interactive visualisations. This system helps users understand how animal attributes and interactions contribute to broader ecosystem patterns.

Animal Data Collection

The foundation of our analytics system is the `AnimalAnalytics` script, which records animal data at fixed intervals. Every 5 seconds, it logs the position and count of all animals in the scene. Key preprocessing steps include:

- **Grouping similar species:** Animals with variants (e.g., `goat_white` and `goat_black`) are grouped together for aggregated analysis.
- **Cleaning object names:** Unity's instantiation adds suffixes such as "(Clone)" to object names. These are removed to standardize naming (e.g., `(Clone) goat_white` becomes `goat_white`).

Data Visualization with XCharts

For visualizing data, we used **XCharts** [19], an open-source Unity charting library. This was integrated into the ANALYTICS scene with multiple chart types to convey different aspects of the ecosystem:

- **Animal Count Snapshot – Pie Chart:** Displays the proportion of each species at the most recent data point (Figure 5.3).

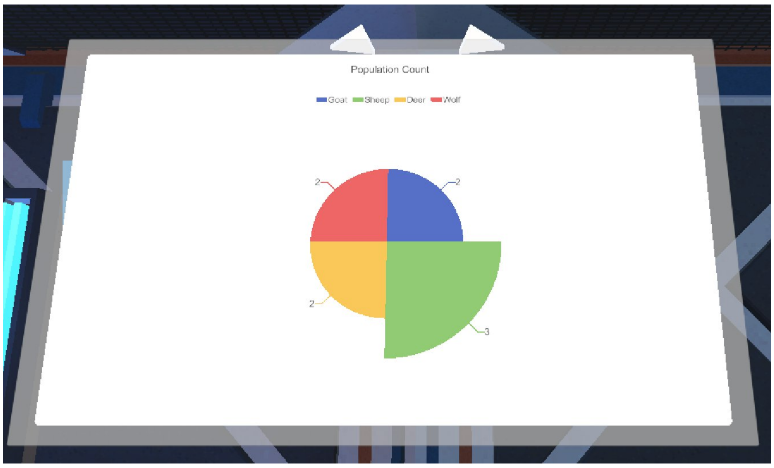

Figure 5.3: Pie chart of population count.

- **Animal Population Over Time – Line/Area Chart:** This visualisation (Figure 5.4) illustrates the ever-changing population dynamics of both prey and predators over time. Prey populations are depicted through solid line graphs with circular markers, while predator populations are represented with shaded area charts featuring triangle markers.

- **Animal Movement – Scatter Plot Heatmap:** A scatter plot visualises the spatial distribution of animals over time. Prey are displayed as circles and predators as triangles (Figure 5.5). A slider below the chart allows users to toggle and observe the changes in distribution over different timestamps (Figure 5.6). The axes are fixed using the maximum x and z positions recorded, ensuring consistent spatial reference.

- **Population Attributes – Radar Chart:** To explore how different animal statistics affect behavior, we plotted key attributes leveraging a radar chart (Figure 5.7):

Figure 5.4: Line chart of population over time.

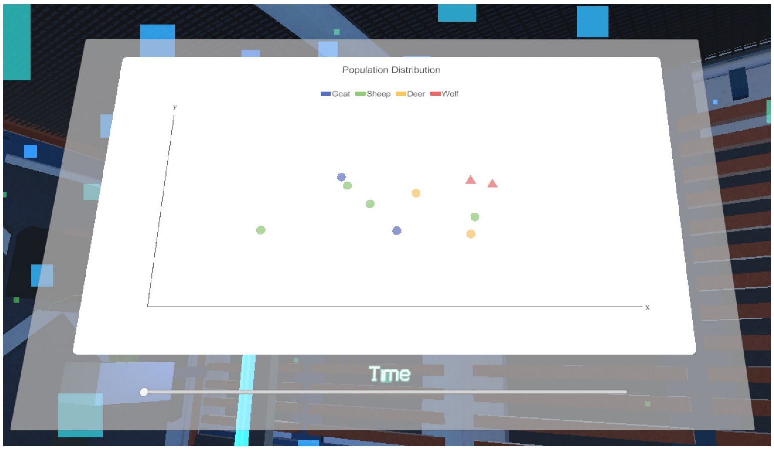

Figure 5.5: Population distribution heatmap.

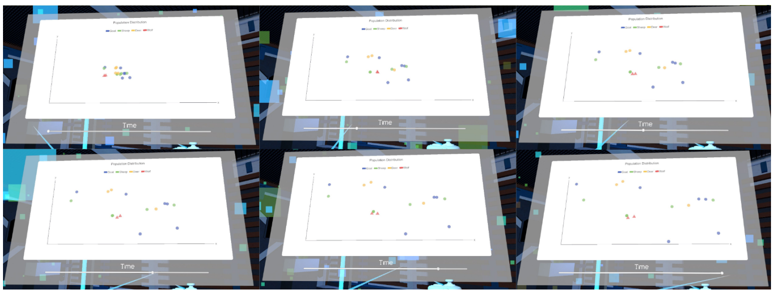

Figure 5.6: Population distribution heatmap change over time from left to right, top to bottom.

- **Predators:** Detection Range, Walking Speed, Hunting Speed, Attack Range, Attack Cooldown
- **Prey:** Move Speed, Rotation Speed, Detection Range, Flee Distance

These charts enable users to compare behavioral traits across species.

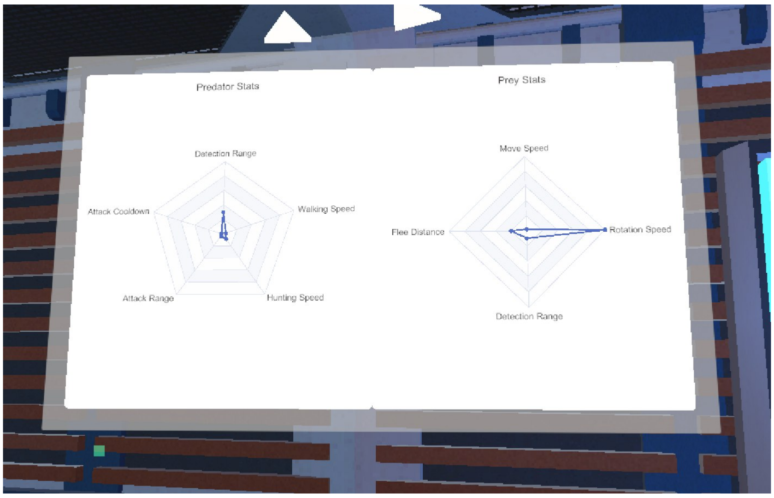

Figure 5.7: Radar chart of animal attributes.

- **Predicted Population – Forecast Line Chart:** We implemented an AutoRegressive Integrated Moving Average model to forecast future animal populations (Figure 5.8). The model takes in historical population data as inputs and generates future values displayed as dotted lines. This prediction helps users anticipate population trends and potential ecological imbalances.

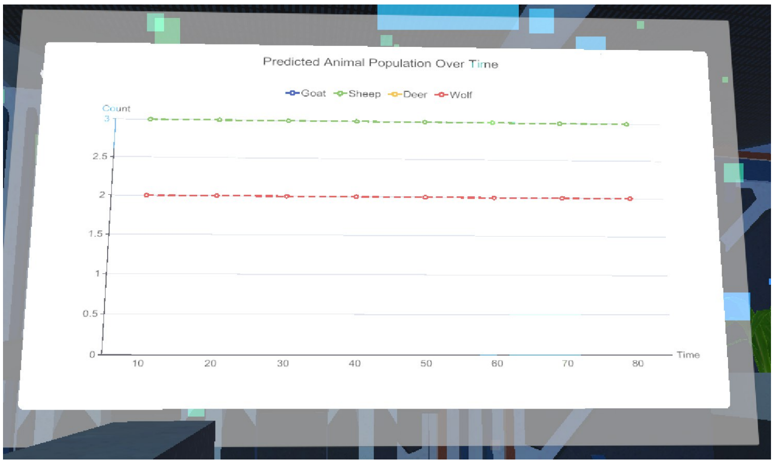

Figure 5.8: Line chart of predicted population.

Chapter 6

Survey Results

In order to evaluate the educational effectiveness and user experience of EcoVR, we conducted a survey targeting students and individuals with varying levels of familiarity with ecological concepts. The survey aimed to assess prior understanding, perceived learning outcomes, usability, and the overall value of the simulation as an educational tool.

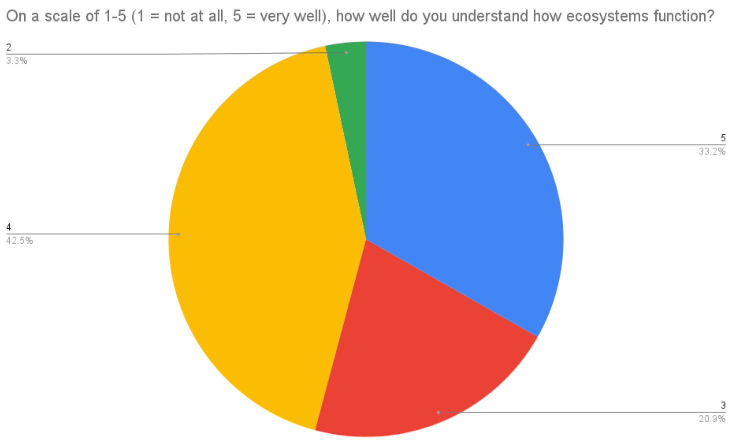

Figure 6.1: Users' understanding of ecosystem functions prior to the simulation

Participant Overview

A majority of participants reported moderate to high familiarity with ecosystem functionalities and predator–prey relationships prior to using the simulation. As highlighted in Figure 6.1, although most of the respondents indicated that they had some basic knowledge, the simulation ultimately deepened their understanding of species interactions and environmental dependencies, as seen in Figure 6.2.

Learning Impact

Participants overwhelmingly agreed that EcoVR helped them better understand:

- The effect of predator and prey populations on ecosystem stability.
- The influence of environmental factors such as weather and habitat on animal survival.
- The importance of balance in food chains and natural population control mechanisms.

As depicted in Figure 6.2, over 85% of users indicated an improved understanding of ecosystems after using EcoVR, with several open-ended responses mentioning how EcoVR made abstract ecological concepts “easier to grasp” and “more memorable through interaction.”

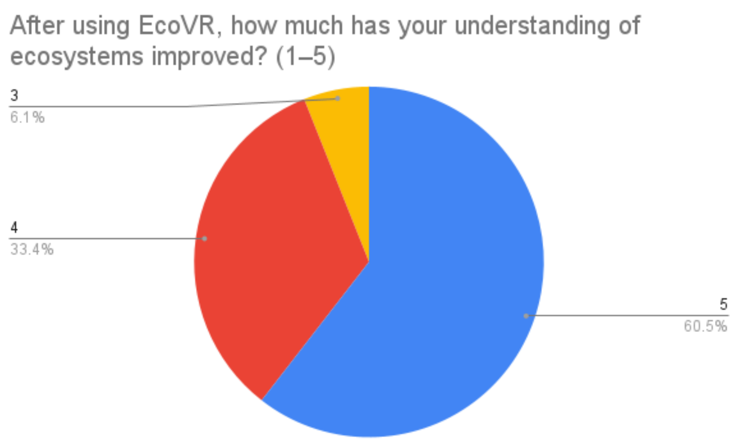

Figure 6.2: Users’ understanding of ecosystem functions after the simulation.

User Experience and Engagement

The majority of participants described the simulation as ”engaging” and ”easy to navigate”. Features that were highlighted to be particularly enjoyable included:

- Observing animal behaviors and interactions in real-time.
- Interpreting data through immersive analytics panels.
- Finding and feeding animals in real-time.

Educational Value and Recommendation

Nearly all respondents agreed that EcoVR could be a useful learning tool for students due to its ability to effectively visualise complex ecological relationships. One participant noted, "It felt like conducting a real experiment in a living ecosystem."

When asked how likely they were to recommend EcoVR as an educational tool, more than 90% gave a score of 4 or 5 out of a possible 5 as shown in Figure 6.3.

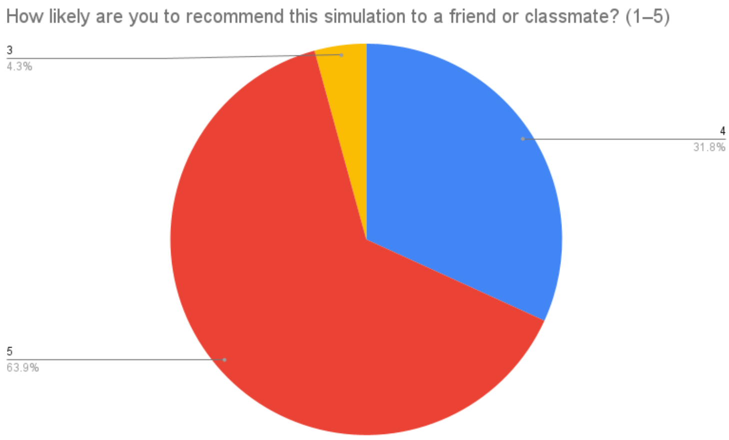

Figure 6.3: If users want to recommend the simulation to others.

Chapter 7

Conclusion

7.1 Challenges and Solutions

During development, we encountered several technical and design challenges which we addressed through the following targeted solutions:

- **High GPU Utilisation:** One major challenge faced was high GPU utilisation, primarily caused by excessive draw calls and overdraw resulting from our complex environments, dynamic lighting setups, and particle-heavy weather effects. On standalone VR platforms such as the Meta Quest 3, where resources are limited, this severely impacted user experience through laggy performance, stuttering, and inconsistent frame drops during gameplay. *Solution:* A mixture of VR optimisation measures, such as GPU instancing, baked lighting and modification of rendering settings, were leveraged. These changes significantly reduced GPU utilisation by minimizing redundant draw calls and eliminating real-time lighting overhead.
- **Lack of 3D Art Support:** Our team had no experience in 3D asset modelling through common softwares such as Blender. *Solution:* We implemented pre-made assets from the Unity Asset Store, allowing us to focus on gameplay and simulation features, without compromising on visual aesthetics.
- **Performance Strain from Animal AI:** Simulating behavior for many animals

at runtime caused performance issues. *Solution:* We modularised the AI into lightweight state machines and used time-staggered logic updates to reduce processing load.

7.2 Limitations and Future Work

While the current implementation successfully establishes the foundation of an interactive and educational ecosystem simulation, several limitations were encountered during development. These challenges also highlight promising directions for future enhancements.

Limitations

- **Time Constraints:** Due to the limited project duration, some features—such as procedural terrain generation, animal evolution, and long-term ecosystem persistence—were not fully implemented or explored.
- **Lack of 3D Art Support:** Without a dedicated 3D artist, we relied on existing assets from the Unity Asset Store. This limited the visual consistency and optimization of the environment and animals.
- **Limited Species and AI Complexity:** The simulation includes only a small set of prey and predator species with rule-based behaviors. AI logic is handled via finite state machines, which limits adaptability and complexity.
- **Static Simulation Parameters:** Users currently cannot modify animal attributes or environmental parameters during runtime, reducing opportunities for hands-on experimentation and scenario testing.
- **Lack of Long-Term State Saving:** The simulation does not support persistent sessions, making it difficult to analyze ecosystem changes across multiple runs.
- **Lack of Full Multiplayer Functionalities:** Full real-time multiplayer functionalities such as live co-op gameplay, synchronous interactions, and shared virtual spaces were all considered initially, but ultimately set aside due to the challenges

involved in networking, synchronization, latency management, as well as a lack of time. Hence, EcoVR is a single-player only experience.

Future Work

Based on the above limitations, we propose the following areas for future development:

- **Support for More Animals and Environments:** Expand the ecosystem to include a greater diversity of species and biomes (e.g., tundra, savannah, wetlands) to simulate broader ecological dynamics.
- **In-Game Modification of Attributes and Terrain:** Provide interactive tools for users to adjust animal attributes (e.g., speed, detection range) and reshape terrain in real-time to test different hypotheses.
- **Animal Evolution System:** Introduce long-term adaptation mechanics, where animal traits evolve based on environmental pressures and survival success.
- **Multiplayer Exploration and Collaboration:** Enable multiple users to explore and analyse the ecosystem together, fostering collaborative learning and shared discoveries.
- **Scenario-Based Challenges and Missions:** Add structured educational missions (e.g., rebalancing a collapsing ecosystem) to guide user interaction and learning outcomes.
- **Integration with Real Ecological Data:** Connect the simulation with real-world datasets to study actual ecosystems and test conservation strategies in a virtual environment.
- **Advanced Animal Intelligence:** Replace FSMs with more advanced decision-making systems (e.g., behavior trees or reinforcement learning agents) to simulate more complex interactions.
- **Procedural World Generation:** Currently, the explorable area in EcoVR is but a fraction of the entire world. As such, future iterations will see the current explorable area be expanded. Additionally, PCG can be leveraged to create new

terrain features such as rock formations, lakes or mangroves, and modeled similar to the existing environments in the game. This will allow for a more varied and dynamic world, offering players fresh areas to explore while maintaining the same aesthetic, as well as smooth and controlled navigation of the original simulation.

- **Data Export Tools:** Allow users to export population logs and movement data through tools such as Excel or Jupyter Notebook for external analysis and educational use.
- **Animal Shelter-Seeking Behavior:** Animals could detect adverse weather and move toward sheltered areas (e.g., trees, caves) to preserve hydration or avoid cold.
- **Weather Seasonal Cycles:** Introduce longer-term changes such as dry vs. wet seasons that affect food availability, animal behavior, and birth rates.

This roadmap highlights the potential of EcoVR as a flexible, extensible platform for ecological simulation, education, and exploration.

7.3 Conclusion

This project presents EcoVR, a forest-based VR ecosystem simulation designed to explore and visualise predator–prey dynamics, population trends, and species attributes all in an immersive, interactive environment. By combining real-time simulation with intuitive data analytics, we are confident that EcoVR will be the bridge between experiential learning and scientific observation, while being suitable for all ages.

Bibliography

[1] OpenAI. *GPT-4o*. https://openai.com/index/introducing-4o-image-generation/. GPT 4o Image Generation. 2025.

[2] Robert Costanza et al. "Simulation games that integrate research, entertainment, and learning around ecosystem services". In: *Ecosystem Services* 10 (2014), pp. 195–201.

[3] SERDP and ESTCP. *Ecosystem Model Comparisons and Applications on DoD Lands*. Youtube. 2023. URL: https://www.youtube.com/watch?v=fQSHtR0cKV0.

[4] Robert E Keane and Eva Karau. "Evaluating the ecological benefits of wildfire by integrating fire and ecosystem simulation models". In: *Ecological Modelling* 221.8 (2010), pp. 1162–1172.

[5] Tony J Pitcher, Eny A Buchary, and Trevor Hutton. "Forecasting the benefits of no-take human-made reefs using spatial ecosystem simulation". In: *ICES Journal of Marine Science* 59.suppl (2002), S17–S26.

[6] Cayelan C Carey et al. "Macrosystems EDDIE teaching modules significantly increase ecology students' proficiency and confidence working with ecosystem models and use of systems thinking". In: *Ecology and Evolution* 10.22 (2020), pp. 12515–12527.

[7] Talespin. "7 Benefits of Virtual Reality Soft Skills Training Simulations". In: *Virtual Reality Training* (2022).

[8] HigherEchelon. "What Are the Benefits of Virtual Reality Simulation?" In: *Cyber Gaming Simulation General High Performance Technology* (2024).

[9] Yuehong Zhu and Ying Han. “Marine Environment Monitoring Based on Virtual Reality and Fuzzy C-Means Clustering Algorithm”. In: *Mobile Information Systems* 2021.1 (2021), p. 2576919.

[10] John HM Thornley. *Grassland dynamics: an ecosystem simulation model.* 1998.

[11] Tom Chandler et al. “Immersive landscapes: modelling ecosystem reference conditions in virtual reality”. In: *Landscape Ecology* (2022), pp. 1–17.

[12] Synty Studios. *Sci-Fi Styled Modular Pack*. https://assetstore.unity.com/packages/3d/environments/sci-fi/sci-fi-styled-modular-pack-82913. https://assetstore.unity.com/packages/3d/environments/sci-fi/sci-fi-styled-modular-pack-82913. 2017.

[13] Innerverse Games. *Stylized Nature Kit Lite*. https://assetstore.unity.com/packages/3d/environments/stylized-nature-kit-lite-176906. https://assetstore.unity.com/packages/3d/environments/stylized-nature-kit-lite-176906. 2020.

[14] Polyart Studio. *Dreamscape Nature Mountains - Stylized Open World Environment*. https://assetstore.unity.com/packages/3d/environments/fantasy/dreamscape-nature-mountains-stylized-open-world-environment-264352. https://assetstore.unity.com/packages/3d/environments/fantasy/dreamscape-nature-mountains-stylized-open-world-environment-264352.

[15] Tanuki Digital. *Tenkoku Dynamic Sky*. https://assetstore.unity.com/packages/tools/particles-effects/tenkoku-dynamic-sky-34435.https://assetstore.unity.com/packages/tools/particles-effects/tenkoku-dynamic-sky-34435.

[16] UrsaAnimations. *Animated Goat and Sheep- 3D low poly-FREE*. https://assetstore.unity.com/packages/3d/characters/animals/animated-goat-and-sheep-3d-low-poly-free-251910. https://assetstore.unity.com/packages/3d/characters/animals/animated-goat-and-sheep-3d-low-poly-free-251910.

[17] Backrock Studios. *Low Poly Animals 3D assets*. https://assetstore.unity.com/packages/3d/characters/animals/low-poly-animals-3d-assets-258476. https://assetstore.unity.com/packages/3d/characters/animals/low-poly-animals-3d-assets-258476.

[18] JustCreate. *Low Poly Food Lite*. https://assetstore.unity.com/packages/3d/props/food/low-poly-food-lite-258693. https://assetstore.unity.com/packages/3d/props/food/low-poly-food-lite-258693.

[19] XCharts Team. *XCharts, a charting and data visualization library for Unity.* https://github.com/XCharts-Team/XCharts. https://github.com/XCharts-Team/XCharts. 2023.