

Defense Report 2

Tribz



Thomas Chen
Martin Doillon
Marie-Jasmine Andres-Yosrungruang
Emile Lassalle
Adrien Le Berre

March 2025

Contents

1	Introduction	4
1.1	Context and Objectives	4
1.2	Recap of Defense 1 Achievements	4
1.3	Report Structure	5
2	Project Design and Updates	6
2.1	Martin - AI Development and Game Balancing	6
2.1.1	Refining Villager Role Management	6
2.1.2	A* Pathfinding Integration	6
2.1.3	Balancing Resources and Game Difficulty	6
2.1.4	Planned Future AI Expansions	7
2.2	Emile - UI/UX Designer and Network Engineer	8
2.2.1	Enhanced Building Interface	8
2.2.2	Shop Categories and Navigation	8
2.2.3	Multiplayer Prototype	8
2.2.4	Challenges and Optimization	9
2.2.5	Future UI and Network Plans	9
2.3	Adrien - Game Designer	10
2.3.1	Multi-Stage Construction and Builder Role	10
2.3.2	Resource Zones and Territorial Dynamics	10
2.3.3	Advanced Creature Interactions	10
2.3.4	Emergent Gameplay and Testing	10
2.3.5	Roadmap for New Game Mechanics	11
2.4	Thomas - Sound Designer	12
2.4.1	Integrating Sound Effects in Unity	12
2.4.2	Triggering Sounds in C# Code	12
2.4.3	Managing Background Music	13
2.4.4	Difficulties Encountered	13
2.4.5	Experience Gained	13
2.4.6	Next Steps in Audio	13
2.5	Marie-Jasmine - 2D Artist	15
2.5.1	Creating a Consistent Visual Style	15
2.5.2	Character and Building Design Updates	15
2.5.3	Environment Creation and Biomes	15
2.5.4	Animations and Visual Effects	16
2.5.5	Designing Fantastic Creatures	16
2.5.6	Challenges and Future Development	16
2.5.7	Planned Artwork	16
3	Project Management and Challenges	18
3.1	Team Coordination and Task Distribution	18
3.2	Progress Overview	18
3.3	Ongoing Challenges	18
3.3.1	AI Complexity vs. Performance	18
3.3.2	Network Synchronization	18
3.3.3	Art Consistency	19

3.3.4	Audio Overlap and Volume Levels	19
3.3.5	User Onboarding	19
4	Next Steps and Conclusion	20
4.1	Future Development Goals	20
4.2	Longer-Term Vision	20
4.3	Conclusion	21

1. Introduction

1.1. Context and Objectives

In this second defense report for our video game project **Tribz**, we present the new milestones and improvements achieved since our first defense. During that earlier stage, we introduced the basic foundations of our prehistoric city-building experience, including rudimentary resource management, partial AI for villagers, an initial user interface, and a basic construction mechanic.

Since then, our team has been hard at work expanding these core features into a more cohesive and immersive framework. In particular, we have developed:

- A more nuanced **AI system** that includes an A* pathfinding approach and better resource balancing.
- Additional **UI/UX** components, such as improved building placement and category-based shops for easier navigation.
- Foundations for **multiplayer support**, enabling potential co-op or competitive interactions between players within the same game world.
- Further **sound design** integration, including dynamic ambient tracks, varied construction sounds, and initial voice triggers.
- More elaborate **2D assets** and digital art, creating a richer environment consistent with the game's prehistoric-fantasy theme.

This report retains the same structure as our first defense but provides a deeper view of each aspect of development. We begin with a recap of the initial accomplishments, then move into a detailed presentation of the current progress by each team member. Finally, we highlight the challenges encountered, lessons learned, and future objectives leading up to our next major milestone.

1.2. Recap of Defense 1 Achievements

In our first defense, we demonstrated:

- A playable proof-of-concept showing fundamental resource management (wood, stone, food).
- A rudimentary AI that allowed NPC villagers to gather resources, albeit with limited roles and no robust pathfinding.
- An early user interface featuring a main menu, a minimal shop interface, and a basic HUD for resource counts.
- Initial sound design concepts, such as background ambient music and a few construction or UI click effects.
- Preliminary concept art for buildings, terrain, and characters to guide the game's aesthetic direction.

We received valuable feedback concerning AI complexity, user feedback mechanisms (tooltips, error messages), environmental immersion, and potential multiplayer features. Over the last few weeks, each of these areas has been further refined, as described in the following sections.

1.3. Report Structure

The remainder of this report is divided into three major sections:

1. **Project Design and Updates:** Each team member details their recent contributions, from AI improvements to art assets, UI/UX developments, sound integration, and new game mechanics.
2. **Project Management:** A look at how we have organized tasks, collaborated via sprints, and tracked feature completion, along with ongoing hurdles.
3. **Conclusion and Next Steps:** We summarize our progress, discuss upcoming milestones, and outline future plans to polish and expand the *Tribz* experience.

By carefully addressing each domain—AI, UI/UX, game design, sound, and art—we aim to demonstrate our shared commitment to building a robust and engaging management game that captures the spirit of tribal survival in a fantastical prehistoric world.

2. Project Design and Updates

2.1. Martin - AI Development and Game Balancing

2.1.1. Refining Villager Role Management

During our first defense, the NPC AI featured relatively basic logic—villagers would gather resources or idle until reassigned. Based on feedback and our own observations, I restructured the AI to allow dynamic role switching and more nuanced behaviors:

- **Priority-Based Task Allocation:** At regular intervals, the AI checks which resources are critically low (food, wood, stone) or if the settlement faces imminent danger. Villagers can switch from gathering to defensive roles automatically.
- **Stateful Villagers:** Each villager tracks states like hunger, fatigue, and morale, influencing how quickly they work. If food supplies run low, gatherer roles are prioritized, and morale might drop, slowing building tasks.

This approach ensures that as the player's settlement expands, villagers remain responsive and relatively self-directed, reducing micromanagement overhead.

2.1.2. A* Pathfinding Integration

One of the largest technical changes since Defense 1 is the incorporation of **A* pathfinding** for more intelligent movement:

- **Grid Layout and Collision:** The map is subdivided into cells indicating whether an area is passable (grass, dirt) or blocked (building footprint, large rock). When a villager moves from point A to point B, the system rapidly computes an optimal route.
- **Updating Obstacles:** If a new building is placed or existing terrain is modified, the AI updates grid data so path calculations remain accurate. **Performance Considerations:** For large villages with many NPCs, pathfinding can be a performance bottleneck. We introduced small optimizations (e.g., caching partial routes, limiting path recalculations) to maintain acceptable frame rates.

With A* in place, villagers no longer wander aimlessly or get stuck around newly placed objects, leading to a more polished simulation overall.

2.1.3. Balancing Resources and Game Difficulty

Alongside pathfinding, the overall pacing of resource gathering and consumption was adjusted:

- **Gradual Difficulty Curve:** Early on, resource acquisition is fairly simple, allowing new players to learn core mechanics. As the tribe population grows, consumption rates scale upward, forcing expansions into new resource zones or advanced structures.
- **Random Events:** To keep the game dynamic, occasional events like a mild drought or a small wave of raiders disrupt the routine. The AI tries to adapt, shifting roles to gather food quickly or station villagers at watchtowers.

```

2 references
private bool Search(Node currentNode) {
    currentNode.State = NodeState.Closed;
    List<Node> nextNodes = GetAdjacentWalkableNodes(currentNode);

    nextNodes.Sort((node1, node2) => node1.F.CompareTo(node2.F));
    foreach (var nextNode in nextNodes)
    {
        if (nextNode.location == this.endNode.location)
        {
            return true;
        }
        else
        {
            if (Search(nextNode))
                return true;
        }
    }

    return false;
}

```

Figure 1: Part of the A* C# implementation

By merging pathfinding, role management, and balanced resource systems, the AI portion now provides a richer, more challenging environment. Players must consider how to fortify their base, manage expansions, and keep villagers well-fed, or face potential collapses.

2.1.4. Planned Future AI Expansions

Looking ahead, I intend to:

- Develop specialized behaviors for **hostile dinosaurs or raiders**, using the same pathfinding approach to threaten player territory in more advanced ways.
- Add simple **behavior trees** or hierarchical logic for villagers, allowing them to chain tasks (e.g., gather wood, deliver it, then assist in a building).
- Continue **performance optimizations** for pathfinding, especially crucial if we introduce larger maps or more numerous NPCs.

Overall, the AI developments since the first defense have made the game world feel more alive and reactive, serving as a strong foundation for the next layers of complexity we plan to introduce.

2.2. Emile - UI/UX Designer and Network Engineer

2.2.1. Enhanced Building Interface

A common request from testers after Defense 1 was an easier mechanism to move or rotate existing buildings. I therefore expanded the **building interaction UI**:

- **Grab-and-Drag Arrows:** Selecting a building now reveals directional arrows. Players can drag the structure around the grid, instantly receiving feedback if the chosen cell is blocked.
- **Upgrade and Info Panels:** Buildings above level 1 show an “Upgrade” button with cost details. Hovering over an info icon displays building stats (like storage capacity or attack range for a watchtower).

These refinements reduce confusion and empower players to plan base layouts more organically without repeatedly demolishing and reconstructing.

2.2.2. Shop Categories and Navigation

With an increasing variety of buildings and decorations, the shop needed better organization. I introduced:

- **Tab-Based Categories:** For instance, **Decorations**, **Defense**, **Resource**, and **Other**. Each category has a colored icon for quick recognition.
- **Scrolling Lists:** A horizontal scroll within each category organizes related items. For smaller screens or higher item counts, the user can flick left or right to reveal more structures without cluttering a single page.
- **Cost Highlighting:** If the player doesn’t have enough resources for an item, the cost label appears in red, while affordable items are shown in white or green text.

These UI changes have been well-received, significantly cutting down on accidental purchases or confusion about item costs and placement rules.

2.2.3. Multiplayer Prototype

The *early* network features revolve around:

- **Local Hosting and Lobby:** A user can start a session on their machine, while others on the same local network can join. We plan to expand this to a more general server browser eventually.
- **Shared State Updates:** When a building is placed or a resource is gathered, the server relays these changes to all connected players. This requires a carefully synchronized approach—particularly for building movement—to prevent “ghost buildings.”
- **Simple Chat Window:** Text-based chat allows players to coordinate or roleplay. We tested sending resource requests or discussing which zone to expand into next.

While currently limited, these features lay the groundwork for more robust co-op or competitive experiences. Once server authoritative code is fully established, we aim to introduce trade, alliances, and potential sabotage mechanics.

2.2.4. Challenges and Optimization

Balancing the UI for both single-player and (eventually) multiple players remains a challenge:

- **Simultaneous Actions:** If two players attempt to modify the same building at once, the server picks one action to ensure consistency. We'll refine locks or "ownership" rules to keep the user experience smooth.
- **Resolution Variety:** High-DPI monitors can show more shop items at once, while smaller screens require more scrolling or condensed layouts. We rely heavily on Unity's *Canvas Scaler* plus advanced anchoring to adapt the UI proportionally.
- **Network Latency:** Suboptimal connections can lead to delayed building placements or "rubber-banding." A partial client-side prediction system reduces the visual jitter, but further testing is needed for large-scale sessions.

2.2.5. Future UI and Network Plans

- **Resource Trading Panels:** If alliances or teams are introduced, players should easily exchange wood, stone, or specialized items via a drag-and-drop or slider interface.
- **Improved Lobby System:** A more user-friendly approach with server lists, game passwords, or direct IP connections for remote co-op.
- **Event Notifications:** Pop-up or side panel alerts for major events, like raider attacks on an allied base or a new building completed by a teammate.

Overall, the additions to UI/UX and the initial multiplayer prototype move *Tribz* closer to providing a cooperative or competitive tribal management experience. Through incremental refinement and user feedback, we aim to make these features both fluid and intuitive.



Figure 2: In game shop

2.3. Adrien - Game Designer

2.3.1. Multi-Stage Construction and Builder Role

To give more weight to the act of constructing buildings, I introduced the concept of **multi-stage building**:

- **Foundation:** Basic resource investment to define the building site on the map.
- **Framework Erection:** Workers place the scaffolding, requiring additional wood or stone.
- **Finalization:** Exterior details, roofing, or functional components (e.g., a watchtower's platform).

Only a few villagers can work on the same building at once, preventing instant completion. This system emphasizes resource planning and ensures the construction process is visually and mechanically layered.

2.3.2. Resource Zones and Territorial Dynamics

As *Tribz* grows in scope, I wanted players to strategically decide where to gather resources:

- **Forest, Mining, and Farming Zones:** Marking these areas helps the AI direct gatherers, preventing them from wandering to suboptimal spots.
- **Biome-Specific Challenges:** In future expansions, certain zones (like a swamp) could hinder movement or require specialized buildings to exploit.
- **Conflict or Cooperation:** In a multiplayer environment, overlapping zones can create friction or alliances, especially if a resource is scarce.

2.3.3. Advanced Creature Interactions

Although not fully realized, we have concept sketches for new dinosaur or mythical species:

- **Taming and Training:** Some creatures can be domesticated, unlocking benefits like faster transport or boosted gather rates.
- **Predatory Threats:** Large carnivores or winged beasts might raid outlying farms, prompting players to construct watchtowers or advanced defenses.

This expands the gameplay loop beyond pure city-building and fosters a survivalist vibe, tying in well with Martin's evolving AI that recognizes and responds to threats.

2.3.4. Emergent Gameplay and Testing

During internal tests, interesting dynamics surfaced:

- Some testers used multi-stage building as a “check” to ensure they never queued too many structures at once, balancing the workforce effectively.
- Others specialized heavily in certain resources, hoarding them for possible trade in future multiplayer sessions.

- Minor “crises”—like a shortfall of stone or a predator attack—quickly tested how well the AI and resource systems adapt, surfacing areas for further refinement.

Regularly playing these test scenarios helps refine the interplay among building expansions, resource zones, and the AI’s capacity to meet emergent needs.

2.3.5. Roadmap for New Game Mechanics

- **Raider System:** AI-driven raids from hostile tribes or bandit groups, forcing players to allocate manpower to defense or form alliances in multiplayer.
- **Advanced Farming:** Crop rotations or specialized plants that require different terrain conditions, pushing the player to explore or trade for seeds.
- **Wonders or Monuments:** Late-game structures offering global bonuses but requiring large resource investments, giving players an end-goal beyond pure survival.

The result is a **layered approach** to design, where each new mechanic—multi-stage builds, resource zoning, advanced creatures—reinforces the core city-building experience while adding breadth to the prehistoric fantasy setting.



Figure 3: Main look

2.4. Thomas - Sound Designer

2.4.1. Integrating Sound Effects in Unity

A significant focus in this second development phase has been to refine the process of **implementing sound effects** into the game engine. After creating a wide range of sounds with FL Studio and Audacity, the next step was integrating them in *Tribz* via Unity's **AudioSource** and **AudioClip** components.

Preparing and Exporting Sound Files

- **Optimization and Compression:** Each sound was exported as an MP3 to balance file size with audio quality. This is particularly important for background music loops, which can be lengthy.
- **Organized Folder Hierarchy:** To keep track of the wide array of effects (building, UI, creature roars, background loops, etc.), I subdivided them into categories inside the Unity project. This ensures each clip is easy to locate and reference in scripts.

Steps Taken to Reach This Point

1. **Sound Needs Assessment:** I collaborated with Adrien and Emile to list out every action in the game that requires an auditory cue—construction, attacks, resource gathering, UI clicks.
2. **Sound Effect Creation:** Using FL Studio's library of instruments and plugins, I designed short but distinct effects (e.g., hammer hits with slight reverberation for building).
3. **Editing and Cleanup:** In Audacity, I removed background noise or unwanted frequencies, normalized volumes, and trimmed silence to ensure each clip starts crisply.
4. **File Integration in Unity:** I placed each MP3 into an appropriate folder (e.g., `Assets/Audio/ConstructionSFX`) and assigned them to relevant GameObjects (like a `BuildingAudioHandler`) or a general SFX manager script.

2.4.2. Triggering Sounds in C# Code

In-game events often originate from *UI* or gameplay scripts. For instance, when the player selects a building, a short selection sound plays:

```
public class BuildingSelector : MonoBehaviour {
    public AudioSource selectionSource;
    public AudioClip selectionSFX;

    void OnBuildingSelected() {
        if (selectionSource != null && selectionSFX != null) {
            selectionSource.clip = selectionSFX;
            selectionSource.Play();
        }
    }
}
```

This ensures immediate feedback, enhancing immersion. Similar logic applies to construction events, resource gathering, or AI triggers (like a dinosaur roar when an aggressive creature appears).

2.4.3. Managing Background Music

For background music, I implemented a layered approach:

- **Looping Ambient Tracks:** A default “tribal day” track runs continuously until the in-game time changes to night or a special event triggers.
- **Event-Based Overlays:** Combat or raider events layer additional percussion or tension strings on top. If the event ends, the overlay fades out smoothly.
- **Biome Variation (Planned):** As the game introduces multiple biomes (desert, swamp, volcanic), we can fade into a different ambient loop or add localized environment sounds (wind, bubbling lava, chirping insects).

2.4.4. Difficulties Encountered

- **Audio File Compatibility:** Some FL Studio exports needed re-encoding because Unity’s default MP3 importer occasionally produced minor artifacts or pitch shifts.
- **Balancing Volumes:** Because *Tribz* can feature many events in a short span, stacking sound effects risked cacophony. A basic audio manager monitors channels, reducing volume if too many concurrent effects are triggered.
- **Scripting Timing:** A few events triggered slightly out of sync, especially if the AI and UI events overlapped. We refined the code to unify triggers at the time of an animation start or a building state change.

2.4.5. Experience Gained

Throughout this integration process, I gained:

- A better understanding of **Unity’s audio pipeline**, including Audio Mixer groups, audio source pooling, and runtime effect parameters.
- Insight into **resource optimization**—ensuring the project doesn’t bloat with large uncompressed files or an overload of near-duplicate SFX.
- Practice in **collaborative iteration**, adjusting volumes or effect triggers based on feedback from the rest of the team and from player testers.

2.4.6. Next Steps in Audio

Upcoming improvements include:

- **Refined Spatial Audio:** Attenuation or directional cues for large creatures, or for waterfalls in a future biome.
- **Action Variations:** Swapping between different hammer or saw sounds so that repeated construction doesn’t become repetitive for the player’s ear.

- **Narrative Cues:** Adding short voiced prompts for tutorial steps or tribal chants that signal major events (like discovering a new resource zone).

By building on this foundation, **sound design** in *Tribz* will continue to deepen the sense of immersion and provide players with clear, engaging feedback at every turn.

```
public class Building : MonoBehaviour
{
    public AudioClip selectionSound;
    private AudioSource audioSource;

    void Start()
    {
        audioSource = GetComponent<AudioSource>();
    }

    void OnMouseDown()
    {
        audioSource.PlayOneShot(selectionSound);
    }
}
```

Figure 4: Code snippet, when a player clicks on a building, the selection sound is played instantly.

2.5. Marie-Jasmine - 2D Artist

2.5.1. Creating a Consistent Visual Style

From the beginning, our goal was a unified, **prehistoric** style with a slightly **cartoony flair**. My role is to ensure that each 2D asset—whether a building sprite, a UI icon, or an environment texture—fits cohesively within this artistic direction.

2.5.2. Character and Building Design Updates

Main Characters

- **Silhouette Clarity:** Especially in an isometric or top-down view, I refined the shape of tribe members so that a gatherer's silhouette is different from that of a warrior. For instance, the gatherer might carry a small basket or net, whereas the warrior could be depicted with a spear or shield.
- **Color Palette:** I introduced subtle differences in clothing color to show faction roles, e.g., a hint of green for gatherers (symbolizing synergy with nature), and a more neutral brown for builders.

Buildings and Structures

- **Multi-Tier Buildings:** For each structure, I designed multiple tiers. A “Hut Level 1” is very rustic with rough straw roofs, while “Hut Level 2” has a more solid frame, decorative details, or improved textures. This tier progression visually communicates building upgrades to players.
- **Advanced Constructions:** In collaboration with Adrien, I created concept art for specialized buildings like a *Barrack* or *Gatherer Depot*, each featuring unique silhouettes and color accents (e.g., red roofs for the Barrack).

2.5.3. Environment Creation and Biomes

Current Biomes

- **Grassland (Default):** Rolling green plains, basic trees, small flowers. It's intended as the starting zone, relatively safe and resource-rich.
- **Marsh/Swamp Prototype:** Darker greens and browns, watery patches, bigger emphasis on hazards like mud or logs. Potentially slower movement for villagers, requiring special boots or bridging.

Adding Variety and Depth

- **Texturing Variation:** To avoid a repetitive “tiled” look, I created multiple variants for each tile type (some with small stones, others with weeds). The game randomly picks these variants when generating the map.
- **Layered Foliage:** Trees come in different shapes—some are short and bushy, others tall and slender. For additional detail, smaller shrubs or decorative rocks appear in clusters, giving the world a more organic feel.

2.5.4. Animations and Visual Effects

Character Animations

- **Walking and Idle Loops:** A gatherer leans forward slightly, as if carrying a load; a warrior stands upright and scans the horizon.
- **Simple Attack Animations:** For upcoming raider or wildlife conflicts, I created frames for villagers brandishing spears or bows.

Environmental Animations

- **Water Flow / Waterfalls:** In some concept scenes, waterfalls or river flows are gently animated by shifting textures, giving a sense of movement.
- **Subtle Weather Effects (Planned):** A future addition is light rain or dust storms for desert or swamp biomes, realized through overlay animations or particle systems.

2.5.5. Designing Fantastic Creatures

An essential element of *Tribz* is introducing a **fantastical prehistoric** ecosystem:

- **Docile Creatures:** Smaller dinosaurs or fantasy mammals with bright, friendly colors. They can coexist near the village or be tamed for resource transport.
- **Hostile Beasts:** Larger predators with spiky backs, vibrant patterns, or fearsome horns. They represent mid- or late-game threats that push players to build defenses.
- **Rare Legendary Creatures (Future):** Possibly large flying beasts or half-mythical reptiles with unique resources. Their design might incorporate a more dramatic silhouette, swirling patterns, or unusual coloration to signal their rarity.

2.5.6. Challenges and Future Development

Keeping Performance High While I aim for visually appealing, semi-detailed sprites, I must also ensure the game remains performant. This involves:

- **Sprite Atlases:** Grouping related textures in a single atlas to reduce draw calls.
- **Limited Frame Counts:** My character animations are kept short (4-8 frames) to achieve a smooth but memory-efficient look.

Expanding Visual Cohesion New features—like zone overlays for farmland or specialized building states—necessitate additional icons or partial overlays. I coordinate with Emile to make sure these overlays mesh seamlessly with existing UI elements.

2.5.7. Planned Artwork

In the coming weeks, I plan to:

- Create **desert** and **volcanic** biome tilesets, each with distinct color palettes and resource doodads (cacti, lava cracks).
- Add **construction stage visuals** for more building types (like a more advanced resource silo or blacksmith workshop).

- Expand **villager diversity**, so there are small style variations (hair, clothing patterns) among tribe members, giving the settlement a more lived-in feel.

Overall, the artistic evolution of *Tribz* emphasizes both clarity and charm, ensuring players can readily identify vital elements while staying immersed in a stylized, vibrant prehistoric environment.

3. Project Management and Challenges

3.1. Team Coordination and Task Distribution

We rely on a combination of **Agile** practices, version control (Git), and communication tools (Discord or Slack) to keep development organized:

- **Bi-Weekly Sprints:** Each sprint includes a planning session, daily asynchronous check-ins, and an end-of-sprint review where we test new features together.
- **Kanban Board:** We maintain a board listing tasks by categories—AI, UI, art, sound, and game design. Tasks move from “To Do” to “In Progress” to “Done,” providing transparency on each person’s workload.
- **Regular Builds:** After each sprint, we produce an internal build for testing, enabling quick feedback loops and revealing integration issues early.

3.2. Progress Overview

Reflecting on Table ?? from the earlier section, we see that each major feature has advanced substantially:

- **AI Pathfinding (80%):** The core logic is in place, though we continue to fine-tune performance and plan expansions (like raider AI).
- **UI/UX (75%):** Many of the requested refinements—such as building movement and category-based shops—are live, but deeper tutorial or alert systems remain to be implemented.
- **Multiplayer Basics (30%):** The essential architecture is functioning, but robust netcode for advanced interactions is still in progress.
- **Sound Design (60%):** We have integrated an array of effects and loops, though future expansions aim to differentiate biomes and refine event triggers.
- **2D Art (65%):** The main environment tiles and building sets are more comprehensive, but new biomes and creature sprites are on the roadmap.

3.3. Ongoing Challenges

3.3.1. AI Complexity vs. Performance

As the simulation grows more intricate, we must keep CPU usage stable. Martin is exploring ways to **batch pathfinding requests** or adopt a hierarchical approach for large maps, reducing the overhead if many villagers move simultaneously.

3.3.2. Network Synchronization

Early multiplayer sessions revealed occasional **desync** issues—like a building existing in one player’s view but not another’s. Emile is refining the authoritative server model to queue changes and confirm them with all clients before finalizing.

3.3.3. Art Consistency

With a variety of new tiles, structures, and soon multiple biomes, ensuring a **consistent palette** and style is crucial to avoid visual dissonance. Marie-Jasmine maintains a color and style guide that addresses everything from building outlines to creature silhouettes.

3.3.4. Audio Overlap and Volume Levels

Now that we have numerous sound effects, events can layer quickly in dense scenes. Thomas is calibrating volume curves and employing culling rules so the resulting audio mix doesn't overwhelm players or strain performance.

3.3.5. User Onboarding

Although not a critical priority yet, we recognize the eventual need for a **tutorial** or guided introduction. With all these new mechanics (multi-stage building, resource zones, AI interactions), new players risk confusion without a structured learning path. We plan to tackle this in upcoming sprints by integrating tooltip-based guides or a short “first mission.”

4. Next Steps and Conclusion

4.1. Future Development Goals

As we move toward the third defense and beyond, our key focus areas include:

- **Advanced AI Enhancements:** Implementing raider logic, specialized wildlife behaviors, and ongoing performance improvements for pathfinding.
- **Fuller Multiplayer Features:** Introducing alliances, a formal trade system, and potential small-scale PvP conflicts or co-op building efforts.
- **Biome Diversity:** Expanding the map generation to incorporate desert, swamp, or volcanic biomes, each with unique resources, challenges, and aesthetic flair.
- **Expanded Sound Design:** Creating distinct music or ambient loops for different biomes, refining event-based triggers for mid/late-game content, and possibly implementing voice lines for important game milestones.
- **Animation and Effects:** Adding day-night lighting transitions, improved weather effects, and building animations that clearly represent upgrade stages.

4.2. Longer-Term Vision

Looking beyond the immediate schedule:

- **Endgame Objectives:** Possibly “wonder” structures or major legendary boss creatures that require collective effort (in multiplayer) or well-prepared defenses in single-player.
- **Modular Building Upgrades:** More sophisticated systems that let players customize buildings with add-ons (e.g., a workshop extension for extra crafting or an archer platform on top of a hut).
- **Robust Player Onboarding:** A multi-step tutorial guiding players from basic resource gathering to advanced territory management, bridging the complexity gap for new arrivals to *Tribz*.

4.3. Conclusion

Since our first defense, *Tribz* has evolved substantially. Each aspect of the project—AI, UI/UX, game design mechanics, sound integration, and art—has received focused attention from our team members:

- **Martin** built on the foundation of villager AI, making them more adaptive and autonomous with role switching and A* pathfinding.
- **Emile** refined the user interface for building movement and shop interaction, while experimenting with network code for future co-op and competitive modes.
- **Adrien** introduced multi-stage construction, resource zoning, and deeper synergy between builder roles, setting the stage for strategic expansions.
- **Thomas** extensively integrated new sound effects and music loops in Unity, ensuring the audio environment adds immersion to every action or event.
- **Marie-Jasmine** expanded building sprites, character designs, environment tiles, and concept art for future biomes and fantastic creatures.

These improvements collectively shape *Tribz* into a more coherent and engaging experience, maintaining the balance between **city-building strategy**, **survival mechanics**, and a **prehistoric-fantasy aesthetic**. The next phase of development will center on polishing these systems, enabling more robust multiplayer interaction, and continuing to enrich the audio-visual layers.

We appreciate the ongoing feedback from advisors and testers, as each iteration brings us closer to a robust, fun, and visually appealing game. We look forward to showcasing further progress in our future defenses, demonstrating how *Tribz* continues to evolve into a fully-realized tribal management experience that blends imagination, strategy, and community interactions.

End of Defense Report 2: *Tribz*

