MATT'S LAIR

Professional Game Design Document

Project: Matt's Lair - VRChat World Experience **Version:** 1.0 **Date:** January 2026 **Platform:** VRChat **Target Filesize:** Under 200 MB

---

1. EXECUTIVE SUMMARY

Matt's Lair is a hub-and-spoke narrative world design for VRChat that evolves a proven 30-year-old design pattern from 1995 QBasic-based "Choose Your Own Adventure" interactive fiction into immersive 3D space. The experience centers on a central navigation hub (Matt's Lair) that provides access to four independent Empire worlds, each representing a distinct narrative branch of the larger worldbuilding structure.

This document details the technical architecture, narrative framework, systems design, and implementation strategy for delivering a cohesive, scalable VRChat experience while maintaining optimization constraints.

---

2. DESIGN PHILOSOPHY

2.1 Core Design Principle: Portals as Primitive Language

The foundational design treats portals and teleportation as low-level language primitives—the bytecode upon which all higher-level gameplay and narrative systems are built. This layered approach mirrors the IF statement logic used in the original 1995 QBasic implementation.

Primitive Layer (Bytecode):

- Portal mechanics (world transitions)
- Teleportation (spatial navigation within/between worlds)
- Spawn point management

Mid-Level Layer (Game Logic):

- Conditional access systems
- Narrative branching based on player choice

- State persistence across empire worlds
- Quest progression and checkpoints

High-Level Layer (Player Experience):

- Emergent storytelling
- Environmental narrative
- Consequence-driven gameplay
- Narrative closure and player agency

2.2 Design Lineage: 1995 to 2026

The original 1995 QBasic implementation used a simple IF-statement structure:

```
Matt's Lair (entry point)
  IF choice = 1 THEN GOTO Empire1_Story
  IF choice = 2 THEN GOTO Empire2_Story
  IF choice = 3 THEN GOTO Empire3_Story
  IF choice = 4 THEN GOTO Empire4_Story
```

This document translates that proven logic into 3D spatial architecture while preserving the fundamental design patterns that made the original effective: clear branching, player agency, narrative encapsulation, and scalability.

---

3. TECHNICAL ARCHITECTURE

3.1 World Structure Overview

Total Target Filesize: 200 MB maximum (smaller preferred)

Allocation Strategy:

- Matt's Lair (Central Hub): 20-30 MB
- Empire World 1: 40-50 MB
- Empire World 2: 40-50 MB
- Empire World 3: 40-50 MB
- Empire World 4: 40-50 MB

This allocation provides ~40 MB per empire world, sufficient for meaningful environmental storytelling with aggressive optimization.

3.2 Hub-and-Spoke Architecture

Matt's Lair (Central Hub)

- Function: Primary navigation node and portal index
- Visual Language: Minimal, functional architecture emphasizing navigation clarity
- Complexity: Low polygon count, single optimized skybox, simple materials
- Purpose: Gateway to four empire worlds; serves as narrative framing device
- Player Interaction: Walk to portals, read/absorb contextual information about each empire, make conscious choice

Four Empire Worlds (Independent Instances)

- Each functions as a separate, distinct loaded world
- Players load only the empire they choose to enter
- Each can be updated independently without affecting others
- Narrative, environmental design, and gameplay mechanics fully encapsulated within each world
- Allow for thematic differentiation and specialized storytelling

3.3 Navigation System: Portals Over Teleportation

Primary Navigation Method: Portals

- Use VRChat's native portal system
- Provides smooth world transitions
- Built-in player spawning at destination
- No physics glitches or positioning issues
- Clear visual feedback through portal presence

Rationale: Portals serve as the optimal primitive for this architecture. They are intuitive, require no scripting overhead, and provide visual clarity to player intent (walking through a portal is a deliberate, conscious choice—functionally equivalent to selecting an IF statement).

Secondary Navigation Method: Teleportation

- Reserved for intra-world navigation or advanced conditional systems
- Treated as a programmable primitive for building higher-level systems
- Enables sophisticated interactions layered atop the portal foundation

---

4. NARRATIVE ARCHITECTURE

4.1 Structure

The experience employs branching narrative structure modeled on interactive fiction principles:

- **Entry Point:** Matt's Lair (narrative framing, world context, choice presentation)
- **Branch Points:** Four portals representing four distinct narrative paths
- **Story Worlds:** Each empire world contains self-contained narrative progression, internal branching, and closure
- **Agency Model:** Player choice drives which narrative branch is experienced; consequences remain encapsulated within chosen world

4.2 Four Empires Framework

Empire 1-4 Specifications:

- Each represents a distinct political, environmental, cultural, or thematic context
- Each contains standalone narrative that functions independently of others
- Thematic differentiation allows for varied gameplay, visual language, and storytelling approaches
- Designed to support player replay: choosing different empires reveals different story perspectives on shared worldbuilding

4.3 Narrative Scaling and Encapsulation

Each empire world can contain internal narrative branching without bloating the overall project:

- Multiple choice points within each empire
- Sub-quests or narrative branches contained within world boundaries
- Closure mechanics that provide sense of completion regardless of path taken

This mirrors the original QBasic structure but distributes complexity across separate world files rather than monolithic code.

---

5. SYSTEMS DESIGN

5.1 Primitive Layer Systems

Portal System

- Function: World transition triggers
- Implementation: Native VRChat portal mechanics
- Behavior: Detects player proximity, initiates world load, spawns player in destination world
- Integration: Placed within Matt's Lair; linked to specific empire world IDs

Teleportation System

- Function: Spatial navigation mechanism
- Implementation: Scripted trigger zones and spawn points
- Behavior: Enables conditional movement within or between worlds
- Integration: Used for complex scenario-based movement; layered atop portal primitives

5.2 Mid-Level Layer Systems

Conditional Access Logic

- Restrictions on which portals/teleportation points are available
- Based on player progression, choices made, or state flags
- Built atop primitive portal/teleportation layer

State Persistence

- Tracking of player choices across world transitions
- Implementation via networked data or local tracking
- Enables branching logic dependent on previous decisions

Waypoint and Checkpoint System

- Predefined spawn points for different narrative scenarios
- Chained transitions enabling multi-step narrative sequences
- Built atop teleportation and portal primitives

5.3 High-Level Layer Systems

Environmental Narrative

- Visual storytelling through world design, asset placement, atmospheric elements
- Supports passive narrative understanding independent of active gameplay
- Reduces exposition burden on interactive systems

Quest and Progression Tracking

- Monitors player advancement through narrative
- Triggers conditional world state changes
- Enables consequence manifestation

Player Agency Mechanics

- Systems that ensure player choices carry weight
- Different outcomes/endings based on choice history
- Narrative variations based on player path through empire world

---

6. OPTIMIZATION STRATEGY

6.1 Asset Management

Shared Resources:

- Texture atlasing across all worlds
- Material library with thematically appropriate variations
- Reusable architectural elements with visual differentiation

Per-World Budget Discipline:

- Each empire strictly limited to ~40 MB allocation
- Prioritize high-impact visual elements
- Reserve detail work for player-frequent zones

Compression and Format:

- Aggressive texture compression (DXT5/BC7 formats)
- Shader-based effects prioritized over baked elements
- LOD systems for distant geometry

6.2 Performance Optimization

Level of Detail (LOD) Systems

- Reduced polygon counts at distance
- Culling of off-screen geometry
- Progressive asset loading

Efficient Skyboxes

- Single optimized skybox in Matt's Lair
- Shader-based dynamic effects where thematically appropriate
- Reuse across empire worlds where narratively sensible

Texture Strategy

- Texture atlases for batching efficiency
- Compressed formats for storage
- Mipmap chains for distance rendering

6.3 Filesize Discipline

Target Tracking:

- Monitor cumulative filesize throughout development
- Regular optimization passes before build
- Prioritize narrative impact over asset redundancy

What To Avoid:

- Duplicate assets across world files
- High-resolution textures for non-focal areas
- Excessive particle systems or dynamic lighting
- Unoptimized 3D models

---

7. IMPLEMENTATION ROADMAP

Phase 1: Foundation (Weeks 1-2)

- Create Matt's Lair core structure (portal placement, basic navigation)
- Define spawn points and portal destinations
- Establish filesize tracking systems
- Create asset library with shared resources

Phase 2: Empire Architecture (Weeks 3-6)

- Build foundational structure for Empire World 1
- Implement portal/teleportation primitive systems
- Establish narrative branching logic
- Test state persistence between worlds

Phase 3: Empire Worlds (Weeks 7-12)

- Complete Empires 2-4 with thematic variation
- Implement conditional access and waypoint systems
- Layer environmental narrative elements
- Populate quest and progression systems

Phase 4: Integration & Polish (Weeks 13-16)

- End-to-end testing of all narrative branches
- Optimization pass across all worlds
- Player testing and iteration
- Final filesize reduction and deployment preparation

Phase 5: Deployment & Support

- Launch to VRChat
- Monitor performance and user feedback
- Iterative updates to individual empire worlds
- Potential expansion with additional content

---

8. TECHNICAL SPECIFICATIONS

Engine: VRChat (Unity-based) **Target Filesize:** 200 MB maximum **Platform:** PC VRChat, potential console support pending filesize **Networking:** Leverage VRChat's built-in networking for state persistence **Save System:** Local progression tracking per player (VRChat user ID)

8.1 World Specifications

Matt's Lair Central Hub

- Polygon Budget: 50,000-75,000 triangles
- Texture Budget: 4-6 atlases at 2048x2048
- Filesize Target: 25-30 MB
- Key Assets: 4 Portal frames, navigation markers, environmental framing

Each Empire World

- Polygon Budget: 200,000-250,000 triangles
- Texture Budget: 8-12 atlases at 2048x2048
- Filesize Target: 40-50 MB
- Key Assets: Environmental storytelling, NPC areas, narrative hot-spots, teleportation waypoints

8.2 API and Systems Integration

Portal System Integration:

- Use VRChat's WorldDescriptor and SpawnPoint systems
- Portal ownership and linking within editor
- Automatic player spawning at destination

Scripting Framework:

- U#/C# for advanced conditional logic
- VRChat networking for shared state
- Local player tracking for single-player progression

---

9. DESIGN PRINCIPLES & CONSTRAINTS

9.1 Core Constraints

1. **Filesize Ceiling:** 200 MB total (smaller preferred)
2. **Navigation Clarity:** Portals prioritized over complex scripting
3. **Narrative Encapsulation:** Each empire world functions independently
4. **Player Agency:** Choice architecture ensures meaningful branching
5. **Optimization First:** Visual quality subordinate to performance requirements

9.2 Design Commitments

- **Respect the Original:** Preserve the proven IF-statement branching logic from 1995
- **Intuitive Navigation:** Spatial design should communicate choices clearly
- **Meaningful Scoping:** Each empire world delivers narrative closure despite filesize constraints
- **Sustainable Development:** Architecture supports iterative updates and expansions

---

10. FUTURE EXTENSIBILITY

10.1 Expansion Possibilities

Additional Empire Worlds:

- Architecture supports adding Empire 5, 6, etc.
- Each loads independently; hub can accommodate additional portals
- Minimal impact on existing filesize (new world file only)

Persistent World Features:

- Cross-empire progression tracking
- Unlockable content based on empire completion
- Shared narrative elements that reference other empire experiences

Multiplayer Elements:

- Cooperative instances within empire worlds
- Social hubs for player interaction
- Leaderboards or shared progression metrics

10.2 Technical Debt Management

- Regular optimization audits
- Asset library maintenance and deprecation
- Filesize tracking and budget adherence
- Performance profiling at deployment milestones

---

11. SUCCESS METRICS

Technical Success:

- Filesize consistently under 200 MB
- Load times under 30 seconds between worlds
- Zero critical performance issues in player testing
- Stable networked state persistence

Design Success:

- Clear communication of narrative choices
- Player engagement with 3+ different empire worlds per session
- Positive feedback on branching narrative impact
- Meaningful sense of player agency

Content Success:

- Each empire world delivers ~30-45 minutes of narrative content
- Multiple playthroughs reveal different story perspectives
- Environmental storytelling supports passive narrative engagement
- Narrative closure achieved regardless of choice path

APPENDIX A: GLOSSARY

Hub-and-Spoke Architecture: Centralized navigation node (hub) connecting to multiple independent branch worlds (spokes)

Primitive Layer: Foundation systems (portals, teleportation) upon which higher-level gameplay is constructed

Narrative Branching: Player choices that create divergent story paths with distinct outcomes

Environmental Narrative: Storytelling through world design, asset placement, and visual communication rather than explicit exposition

State Persistence: Retention of player progression and choice history across world transitions

Filesize Budget: Predetermined allocation of storage space for each world component

---

APPENDIX B: DESIGN DECISIONS LOG

Decision	Rationale	Alternative Considered
Portals as primary navigation	Native VRChat support, intuitive UI, no scripting overhead	Scripted teleportation (more complex, less reliable)
Four independent empire worlds	Enables parallel development, independent updates, clear narrative branching	Single monolithic world (filesize constraints, harder to manage)
40 MB per empire allocation	Supports meaningful environmental storytelling while respecting 200 MB ceiling	Higher allocation (exceeds budget), Lower allocation (insufficient for narrative)
Hub-and-spoke over interconnected world	Simplifies navigation, reduces complexity, mirrors original QBasic structure	Interconnected worlds (harder to navigate, larger filesize requirement)
Teleportation as secondary system	More flexible than portals for complex scenarios; can be layered atop portal primitives	Primary system (overcomplicates basic navigation)

Document Status: Complete and Ready for Development **Next Steps:** Architecture review, asset library creation, Empire 1 structural development **Contact/Updates:** [Development Team Information]