# Project Chimera: The Ultimate Comprehensive Development Plan

**Version:** 1.0 **Date:** May 27, 2025 **Project Lead:** [Developer Name/Studio]

## Table of Contents

1. **Grand Vision & Strategic Imperatives** 1.1. Project Chimera: Defining the Dream 1.2. Core Pillars: The Foundation of Gameplay 1.3. Unique Selling Propositions (USPs): Standing Apart 1.4. Target Audience: Engaging the Connoisseur & Strategist 1.5. Monetization Strategy: Ethical & Sustainable Growth
2. **Foundational Framework: Architecture & Technology** 2.1. Engine Selection & Configuration: Unity 6.2 Beta 2.2. Scalable Software Architecture: Designing for Complexity & Evolution 2.2.1. Modular Design Philosophy: Assemblies, APIs, and Single Responsibility 2.2.2. Data-Driven Design: The Power of ScriptableObjects 2.2.3. Event-Driven Architecture: Decoupling for Robustness 2.2.4. Core Game Loop & Advanced State Management 2.3. Technology Stack: Tools of the Trade 2.3.1. C# Best Practices: Crafting Clean, Performant Code 2.3.2. VS Code & Unity Integration: Optimizing the IDE Workflow 2.4. Data Management & Persistence: Safeguarding Progress & Content 2.4.1. Serialization Strategy for Complex Runtime Game State 2.4.2. Addressable Asset System: Efficient Asset Management
3. **AI-Assisted Development & Content Generation Workflow** 3.1. Philosophy: AI as a Force Multiplier, Human as Creative Director 3.2. AI for Code Generation & Assistance 3.2.1. Cursor AI: Primary C# Coding Assistant 3.2.2. Unity AI Assistant (Unity 6.2): In-Editor Support 3.3. AI for 3D Asset Generation 3.3.1. Rodin by Hyper3D: High-Fidelity Equipment & Base Meshes 3.3.2. Other 3D Tools (Meshy AI, Sloyd AI): Prototyping & Specialized Assets 3.4. AI for 2D Asset Generation 3.4.1. Texture Synthesis (Stable Diffusion, Substance 3D Sampler AI) 3.4.2. Icons, UI Elements, Concept Art (Leonardo.Ai, Google Gemini/Imagen) 3.5. AI for Potential In-Engine Runtime Systems (Research & Post-MVP) 3.5.1. Unity Sentis: Neural Network Inference 3.5.2. Unity Behavior (Behavior Trees/Graphs): Advanced NPC Logic 3.6. Mandatory Human Oversight, Optimization & Provenance Tracking 3.6.1. Rigorous Review & Refinement Pipeline 3.6.2. Provenance Metadata: Ensuring Legal & Technical Integrity
4. **Minimum Viable Product (MVP): Scope, Core Loops & Essential Features** 4.1. MVP Definition & Guiding Philosophy 4.2. Core Gameplay Loops for MVP Validation 4.2.1. Manual Cultivation & Observation 4.2.2. Basic Genotype x Environment (GxE) Interaction 4.2.3. Experimentation & Iterative Learning 4.3. Essential MVP Systems & Features (Detailed Breakdown) 4.3.1. Foundational Cultivation Mechanics (Manual Focus) 4.3.2. Foundational Genetics & Breeding (5 Strains, Basic F1 Crosses) 4.3.3. Rudimentary Environmental Control (Manual Adjustments, Basic Sensors) 4.3.4. Basic Nutrient Management (Manual Mixing & Monitoring) 4.3.5. Elementary Plant Health & Pest/Disease Management 4.3.6. Core Time Mechanics (Active Scales, Offline Pause, Transition Inertia) 4.3.7. Initial Sandbox Environment: The Residential House 4.3.8. NPC-Driven Contract Economy (Introduction) 4.3.9. Initial Player Progression (Simplified Skill Tree) 4.3.10. Core UI/UX Elements for Data & Interaction 4.4. MVP Asset Prioritization & Visual Quality Mandate 4.4.1. Master MVP Asset List (Table) 4.4.2. Commitment to High Visual Fidelity for MVP Assets 4.5. Explicit MVP Exclusions: Deferred Features for Future Phases
5. **Core Gameplay Systems: Deep Dive (MVP & Full Vision)** 5.1. **Cultivation Mechanics: From Seed to Harvest** 5.1.1. Plant Lifecycle Simulation (States: Seed, Germination, Seedling, Vegetative, Flowering, Harvest) 5.1.2. Genotype x Environment (GxE) Interaction: The Heart of Simulation 5.1.2.1. Conceptual Model: Genetic Potential x Environmental Modifiers 5.1.2.2. Key Environmental Factors & Their Impact 5.1.2.3. Environmental "Recipes" & Strain Profiling 5.1.3. Procedural Plant Morphology & Visual Realism (Details in Sec 6.3) 5.1.4. Player Cultivation Techniques (Manual & Automated) 5.1.4.1. Planting, Watering, Pruning, Training (LST/HST) 5.1.4.2. MVP: Manual Focus, Post-MVP: Advanced Automation 5.2. **Genetics & Breeding: The Pursuit of Perfection** 5.2.1. Genetic Representation: Polygenic Traits, Genes, Alleles 5.2.2. Inheritance Models: From Simple Mendelian to Complex Interactions 5.2.2.1. MVP: Basic F1 Crosses, Observable Visual Traits 5.2.2.2. Post-MVP: Polygenic Traits, Allelic Interactions (Dominance, Epistasis, Pleiotropy), Mutation 5.2.3. Foundational Strains (Initial 5 for MVP, Expanding Library) 5.2.4. Advanced Breeding Techniques (Post-MVP) 5.2.4.1. Backcrossing (BX), Inbred Lines (IBLs), Selfing (S1), Feminization 5.2.4.2. Pheno-Hunting: Observation & Selection 5.2.4.3. Genetic Marker Analysis (Late-Game Assist) 5.2.5. The "AI Research Lab": Predictive Breeding (Post-MVP) 5.2.6. Trait Library: Player-Populated Genetic Discoveries 5.2.7. Tissue Culture & Micropropagation (Advanced Post-MVP) 5.2.8. Genetic Acquisition: Landraces, Expeditions, NPC Contacts 5.3. **Environmental Control Systems: Mastering the Microclimate** 5.3.1. MVP: Manual Adjustments, Basic Standalone Sensors & Controllers 5.3.2. Post-MVP: Advanced Utility Networks 5.3.2.1. Detailed 3D Routing: Plumbing (Water, Nutrients), Electrical, HVAC 5.3.2.2. Component Sizing & Material Properties (Flow, Capacity, Efficiency) 5.3.2.3. Logical Source-to-Endpoint Connections & Visual Feedback 5.3.3. The "X-Ray" Utility View Toggle: Visualizing Hidden Infrastructure 5.3.4. Microclimate Modeling: Localized Environmental Variations 5.3.4.1. Grid-Based or Zone-Based Simulation 5.3.4.2. Propagation & Diffusion of Heat, Humidity, Light, CO2 5.3.5. Advanced Automation: Sensors, Controllers, PLCs (Post-MVP) 5.3.5.1. Tiered Sensor Systems (Basic, Intermediate, Advanced) 5.3.5.2. Programmable Logic Controllers (PLCs) & Central Management 5.4. **Nutrient Management: Feeding for Excellence** 5.4.1. MVP: Manual Mixing of Base Nutrients & Additives, Manual Monitoring (EC/PPM, pH) 5.4.2. Post-MVP: Automated Dosing Systems, Advanced Nutrient Recipes, Hydroponic Management 5.4.3. Nutrient Solution Tracking & Plant Uptake Simulation 5.5. **Plant Health & Integrated Pest/Disease Management (IPM)** 5.5.1. MVP: Visual Observation, Basic Scouting Tools (Loupes, Sticky Traps), Manual Organic Treatments, "Plant Problems Guide" 5.5.2. Post-MVP: Advanced IPM Strategies 5.5.2.1. Biological Controls (Beneficial Insects) 5.5.2.2. Preventative Treatments & Environmental Management for Pest Control 5.5.2.3. Advanced Diagnostic Tools & Disease Modeling 5.6. **Time Mechanics: Controlling the Flow of Growth** 5.6.1. Active Time Acceleration Scales (0.5x, 1x, 2x, 4x, 8x, Real-Time) 5.6.2. "Transition Inertia": Strategic Commitment to Time Scales 5.6.3. Offline Progression: Player-Controlled Speed or Full Pause 5.6.4. "Catch-Up Visualization" & "Facility Status Report" on Login 5.6.5. Subtle Risk/Reward Variables Tied to Time Scale 5.7. **Facility Construction & Management: Building the Ideal Grow Op** 5.7.1. Sandbox Environments: Residential House (MVP), Warehouse (Expansion) 5.7.1.1. Residential House: Predefined Layout, Room Unlocks, Surface-Mounted Utilities 5.7.1.2. Warehouse: Vast Empty Shell, Custom Room Construction, Hidden Utilities 5.7.2. Grid-Based Construction System (1-foot Grid Unit, Snap-to-Grid) 5.7.3. Manual Utility Routing (MVP in Warehouse, Full 3D) 5.7.4. Zoning & Layout Optimization: Functional Area Designation 5.7.4.1. MVP: Basic Room Zoning 5.7.4.2. Post-MVP: Granular Sub-Zones, Workflow Optimization, Contamination Control 5.8. **Post-Harvest Processing: From Flower to Finished Product** 5.8.1. MVP: Manual Drying, Curing (Jar Tech, Burping), Hand-Trimming 5.8.2. Post-MVP: Advanced Extraction Techniques 5.8.2.1. Solventless Extraction (Rosin Presses) 5.8.2.2. Solvent-Based Extraction (Abstracted: Vessels, Winterization, Purging) 5.8.2.3. Quality Factors: Yield, Purity, Potency, Terpene Retention 5.8.3. Post-MVP: Edibles & Topicals Manufacturing 5.8.3.1. Recipe-Based Formulation & Infusion 5.8.3.2. Dosage Consistency & Quality Control 5.8.3.3. Specialized Processing Equipment
6. **Art Direction & Asset Production Pipeline** 6.1. Master Art Style & Aesthetic: "Modern, High-Tech, Clinical/Scientific, Aspirational/Professional" 6.1.1. Visual Priorities: Cleanliness, Detail Focus, Rewarding Feedback, Engaging Data Presentation 6.1.2. Color Palette: Primary, Secondary, Accent, Functional 6.1.3. Lighting & Material Definition (PBR) 6.1.4. Typography: Clarity & Modernity (Poppins initial) 6.2. Master Asset List & Prioritization Strategy 6.2.1. Comprehensive Asset Inventory (Reference Table from Doc1, Expanded) 6.2.2. MVP Asset Focus: Essential for Core Loops & Residential House 6.2.3. Post-MVP Asset Expansion: Warehouse, Advanced Equipment, New Strains 6.3. Procedural Plant Generation Strategy: The Visual Heartbeat 6.3.1. Goal: Highest Possible Visual Realism & Biological Accuracy 6.3.2. Hybrid Approach: Human Artistry + Procedural Algorithms + AI Assist 6.3.2.1. Artist-Created Base Meshes & Textures for Plant Parts 6.3.2.2. Rule-Based Component System (High-Level Structure, Splines) 6.3.2.3. L-Systems for Fine Detail & Variation (Optional Enhancement) 6.3.2.4. Unity AI (6.2 Beta) for Base Asset Assistance & Procedural Tools 6.3.3. Dynamic Adaptation: Visuals Reflecting GxE, Growth Stage, Health 6.3.4. C# Implementation: Tight Integration with GxE Simulation 6.4. Environment & Equipment Asset Creation Workflow 6.4.1. Methodology: Human Modeling + AI-Assisted Generation (Concepts, Base Meshes, Textures) 6.4.2. Mandatory Human Review & Optimization Pipeline (Retopology, UVs, LODs, Texture Refinement) 6.4.3. Tools: Standard 3D/Texturing Software + AI Suite (Rodin, Imagen, Leonardo.Ai, etc.) 6.4.4. Specifics: Facility Components, Cultivation Equipment, Decorative Items 6.5. UI Asset Design & Production 6.5.1. Style Alignment: Dark Mode, Material Design Influence, Abstract Line Art Icons 6.5.2. Asset Types: Icons, Panel Backgrounds, Button Styles, Fonts 6.5.3. Implementation in Unity UI Toolkit 6.6. Provenance Tracking for All AI-Assisted Assets (Mandatory) 6.6.1. Metadata Requirements: Tool, Version, Prompts, Seeds, Dates, Human Artists, Modifications, Licensing 6.6.2. Importance: Legal Compliance, Quality Control, Reproducibility, Refining Prompts
7. **User Interface (UI) & User Experience (UX) Design: The Player's Window to Complexity** 7.1. UI/UX Philosophy: Critical Gameplay Asset - Clarity, Intuitiveness, Actionable Insights 7.2. Implementing the "Modern, Clean, Sophisticated" UI Aesthetic 7.2.1. Unity UI Toolkit: Recommended Framework 7.2.2. Structure (UXML), Styling (USS), Logic (C# with MVVM/MVC) 7.2.3. Responsiveness & Adaptability Considerations 7.3. Core UI Elements & Dashboards (MVP & Expansion) 7.3.1. Environmental Data Dashboards (Real-Time Readouts, Customizable) 7.3.2. Plant Health/Status Indicators & Detail Views (At-a-Glance & In-Depth) 7.3.3. Nutrient Management Interface (Mixing, Monitoring) 7.3.4. Genetic Data Display (Strain Info, Lineage, Traits, Pedigrees) 7.3.5. Facility Management Overlays (Resource Inventory, Zoning, Utility View) 7.3.6. Economic & Contract Management Panels 7.3.7. Skill Tree & Research Interfaces 7.4. Advanced Data Visualization: Transforming Data into Decisions (CRITICAL FEATURE) 7.4.1. Real-Time Environmental Readouts (Digital Style, Status Coding) 7.4.2. Historical Trend Graphs (Line Graphs for Environment, Nutrients, Growth) 7.4.3. Simulated Lab Analysis Reports (Cannabinoid/Terpene Profiles, NPK) 7.4.4. Grow Cycle Comparator & Advanced Analytical Tools 7.4.5. Heatmaps & Spatial Data Visualizations (Post-MVP) 7.4.6. Interactivity: Tooltips, Hover-Over Data, Filtering, Sorting 7.5. Player Feedback Systems: Keeping the Player Informed & Empowered 7.5.1. Tiered Alerts & Notifications (Informational, Warning, Critical - Visual & Auditory) 7.5.2. Comprehensive Logging Systems 7.5.2.1. Automated Event Logging (Planting, Harvest, Alerts, Research) 7.5.2.2. Player-Driven Logging ("Grower's Journal" - Notes, Observations, Tags) 7.5.3. Integrated Tutorials & Information Systems 7.5.3.1. Contextual Tutorial Overlays & Pop-Ups 7.5.3.2. In-Game "Plant Problems Guide" & Diagnostic Tools 7.5.3.3. Extensive Tooltips for UI Elements & Technical Terms 7.6. Navigation & Interaction Models: Ensuring Smooth Engagement 7.6.1. Hierarchical View System (Exterior Shell -> Blueprint -> Room Interior -> Detail Views) 7.6.2. Contextual "Action Modes" for Specific Tasks (e.g., Manual Data Acquisition) 7.6.3. Intuitive Controls for Construction, Placement, and Interaction 7.7. UI/UX Refinement & Iteration Process 7.7.1. Prototyping & Frequent Usability Testing 7.7.2. Iterative Design Based on Feedback (Information Density, Layouts, Ease of Use) 7.8. Accessibility Considerations (Future Phase) 7.8.1. Text Scalability, Colorblind Modes, Keyboard Navigation, Control Remapping
8. **Player Progression, Economy & Narrative Systems** 8.1. **Player Progression Systems: The Journey to Mastery** 8.1.1. The Skill Tree: "The Tree of Knowledge" 8.1.1.1. Thematic Visualization: Growing Cannabis Plant with 7 Main "Leaves" (Genetics, Cultivation, Environment, Construction, Harvest, Science, Business) 8.1.1.2. Nodes: Unlocking Concepts, Mechanics, or Simulations 8.1.1.3. Philosophy: Unlocking Concept vs. Achieving Mastery via Equipment/Research 8.1.1.4. Skill Point Acquisition: Objectives, Goals, Harvest Quality 8.1.1.5. Interconnectivity & Prerequisites between Nodes/Leaves 8.1.1.6. Illustrative MVP Skill Tree Node Examples (Table) 8.1.2. Research System (Post-MVP Focus, Foundational Elements in MVP) 8.1.2.1. Specialized Unlocks: Advanced Tech, Unique Genetic Lines, High-Tier Equipment 8.1.2.2. Resource Investment: Currency, Time, Research Points, Rare Materials 8.2. **Economic Systems: From Humble Beginnings to Cannabis Conglomerate** 8.2.1. MVP: NPC-Driven Contract Economy 8.2.1.1. Cultivation Contracts: Strain, Quantity, Quality, Deadlines, Rewards 8.2.1.2. Direct Sales to Limited NPC Buyers 8.2.1.3. Basic Operational Costs: Utilities, Consumables, Initial Equipment 8.2.1.4. Role as Organic Guidance & Tutorial Mechanism 8.2.2. Post-MVP: Dynamic Player-Driven Marketplace 8.2.2.1. Supply & Demand Mechanics for Genetics & Equipment 8.2.2.2. Player-to-Player Trading (Simulated or Asynchronous) 8.2.2.3. Order Management Dashboards, Transaction History, Market Wallet 8.2.2.4. Advanced Market Data Analysis Tools (Price Charts, Volume Info) 8.2.3. Player/Company Reputation System 8.2.3.1. Impact on NPC Interactions, Contract Availability, Market Trust 8.2.3.2. Influenced by Product Quality, Contract Fulfillment, Ethical Choices 8.2.4. Robust Resource & Currency Sinks 8.2.4.1. Marketplace Taxes & Listing Fees 8.2.4.2. High-Priced Essential/Rare NPC Vendor Items 8.2.4.3. Ongoing Operational & Maintenance Costs, Research Investments 8.3. **Narrative Systems & World-Building: Context & Immersion** 8.3.1. ADA (AI Digital Assistant): The Player's Guide 8.3.1.1. Role & Persona: Helpful, Functional, Non-Anthropomorphic 8.3.1.2. Communication: Text-Based (Inbox, Notifications, Pop-Ups), Sparse Synthesized Voice (Critical Alerts, Milestones) 8.3.1.3. Functions: Delivers Contracts, Research Directives, Milestone Notifications, Contextual Hints, World News 8.3.1.4. Trigger System: Event-Driven, Progression-Based, Contextual 8.3.2. Diegetic Lore Delivery Mechanisms: Organic World Discovery 8.3.2.1. In-Game Communications (Industry News Digests, Scientific Journal Excerpts, NPC Messages) 8.3.2.2. Item Descriptions (Equipment, Seeds, Nutrients - Manufacturer Lore, Tech History, Strain Origins) 8.3.2.3. Research & Skill Tree Unlock Blurbs (In-World History, Significance) 8.3.2.4. Ambient Environmental Cues (In-Game Radio, News Tickers, Branding) 8.3.3. Lore Themes: State of Cannabis Industry, Scientific Landscape, Cultural Context, Historical Milestones
9. **Advanced Technical Implementation & Optimization** 9.1. Advanced C# Programming Patterns for Unity 9.1.1. State Pattern: Managing Complex Entity Behaviors (Plants, Equipment, UI) 9.1.2. Observer Pattern / Event-Driven Architecture (ScriptableObject Event Channels) 9.1.3. Service Locator or Dependency Injection (DI) for Core Systems 9.1.4. Object Pooling: Efficient Management of Frequent Instantiations 9.2. Performance Optimization Strategies: Ensuring Smooth Simulation 9.2.1. Optimizing MonoBehaviours for Many Simulated Agents 9.2.1.1. Custom Update Managers for Batched Updates 9.2.1.2. Data Locality & Struct-Based Data for Core Simulations 9.2.1.3. Leveraging Unity's Job System & Burst Compiler for Parallel Processing 9.2.1.4. Hybrid Approach: Data-Oriented Principles within MonoBehaviour Architecture 9.2.2. Profiling Tools & Techniques (Unity Profiler, Frame Debugger, Memory Profiler) 9.2.3. GPU Optimization: Draw Calls, LODs, Occlusion Culling, Shader Efficiency, Texture Optimization 9.2.4. CPU Optimization: Algorithm Efficiency, Minimizing Update Loop Costs, GC Management 9.2.5. Memory Management: Efficient Data Structures, Asset Unloading 9.2.6. Optimizing Data Structures & Algorithms (Big O, Caching) 9.3. Load Time Optimization Strategies 9.3.1. Addressable Asset System for On-Demand Loading 9.3.2. Asynchronous Loading APIs (Scenes, Assets) 9.3.3. Minimal Bootstrap Scene 9.3.4. Efficient Serialization Formats for Save/Load
10. **Quality Assurance, Testing & Iteration Strategy** 10.1. Testing Philosophy: Integrated, Iterative, and Comprehensive 10.2. Phased Testing Approach (Aligned with Development Phases) 10.3. Unit Testing for C# Systems 10.3.1. Framework: Unity Test Framework (NUnit) 10.3.2. Scope: Core Algorithms, Data Validation, Critical Logic, AI-Generated Code Segments 10.3.3. AI Assistance (Cursor) for Test Boilerplate & Case Suggestion 10.4. Integration Testing: Verifying System Interconnectivity 10.5. Gameplay Testing & Feedback Loops 10.5.1. Internal Testing: Continuous Developer & Team Playtesting 10.5.2. External Alpha Testing: Core Mechanics, Stability, Major Bugs (Small, Trusted Group) 10.5.3. External Beta Testing: Balance, Progression, UI/UX, Fun Factor (Larger, Broader Group) 10.5.4. Feedback Collection Mechanisms (In-Game Reporting, Discord, Forums, Surveys) 10.6. Performance Profiling and Optimization (Continuous Cycle) 10.6.1. Regular Profiling Sessions Under Expected Player Loads 10.6.2. Targeting Frame Rates & Budgeting Frame Times 10.7. AI-Specific Testing Considerations 10.7.1. Cursor-Generated Code: Correctness, Performance, Standards Adherence, Edge Cases 10.7.2. Unity AI Runtime Components: Predictability, Stability, Performance, Desired Outcomes 10.7.3. AI-Assisted Assets: Technical Compliance, Visual Style Adherence (Post-Optimization) 10.8. Simulation Balance & Emergent Behavior Testing 10.8.1. Identifying & Addressing Unintended Exploits or Dead-Ends 10.8.2. Targeted Testing for Subtle Risk/Reward Mechanics 10.8.3. Telemetry Data Collection (Anonymized, During Beta) for Balance Insights 10.9. Iteration & Balancing Tools for Development 10.9.1. In-Game Data Visualization & Debug Menus (Runtime Inspection & Modification) 10.9.2. Advanced Time Control Utilities (Pause, Fast-Forward, Step) 10.9.3. Scenario Management & Testing Tools (Save/Load Presets, Scriptable Scenarios)
11. **Build Management, Deployment & Post-Launch Strategy** 11.1. Build Pipeline & Automation 11.1.1. Automated Build Process (Unity Cloud Build, Jenkins, GitHub Actions) 11.1.2. Build Configurations (Development, QA, Release) 11.1.3. Consistent Versioning System (Semantic Versioning) 11.2. Platform Considerations 11.2.1. Primary Target: PC Standalone (Windows, macOS, Linux) 11.2.2. Hardware Compatibility & Performance Targets (Min/Rec Specs) 11.3. Distribution Platform Strategy 11.3.1. Platform Selection (Steam, Epic Games Store, GOG.com) 11.3.2. Store Asset Preparation & SDK Integration 11.4. Post-Launch Support & Evolution 11.4.1. Bug Fixing & Patch Release Strategy 11.4.2. Phased Content Expansion Plan (Deep Genetics, Player Marketplace, etc.) 11.4.3. Alignment with Monetization Strategy (Buy-to-Play Base, Paid Expansions, Ethical Cosmetics) 11.4.4. Community Engagement & Feedback Channels
12. **Project Management & DevOps** 12.1. Version Control Strategy: Git + Git LFS 12.1.1. Repository Setup & Configuration (.gitignore, .gitattributes, Unity Settings) 12.1.2. Branching Strategy (GitHub Flow Recommended for Indie) 12.1.3. LFS Hosting & Management Considerations 12.2. Project Management Methodology: Agile Principles with Kanban 12.2.1. Iterative Development, Flexibility, Regular Reviews 12.2.2. Visual Task Management with Kanban Boards 12.3. Task Tracking & Roadmapping Tools (HacknPlan, Jira, Notion) 12.3.1. Master Feature List & Detailed Task Breakdown 12.3.2. High-Level Roadmap with Milestones (MVP, Alpha, Beta, Expansions) 12.3.3. Prioritization Framework (MoSCoW, RICE) 12.4. Documentation Strategy: Living Design Documents 12.4.1. Game Design Document (GDD) - Continuously Updated 12.4.2. Technical Design Documents (TDDs) for Complex Systems 12.4.3. AI Prompting Guides & Asset Provenance Records 12.5. Continuous Integration/Continuous Deployment (CI/CD) Pipeline 12.5.1. Automated Builds Triggered by Commits/PRs 12.5.2. Automated Unit Test Execution 12.5.3. Automated Deployment to Staging/QA Environments
13. **Risk Assessment & Mitigation Strategies** 13.1. Technical Risks (Complexity, Performance, AI Tool Instability, Beta Engine) 13.2. Scope Creep & Feature Bloat 13.3. Balancing Intricate Simulation Systems 13.4. Asset Production Bottlenecks (Especially for Diverse Plant Visuals) 13.5. Solo/Small Team Limitations & Burnout 13.6. Evolving Legal Landscape for AI-Generated Content 13.7. Market Reception & Player Expectations
14. **Future Vision & Long-Term Evolution** 14.1. Beyond Planned Expansions: Potential Future Directions 14.1.1. Multiplayer Functionality (Cooperative Cultivation, Competitive Breeding) 14.1.2. Deeper Narrative Storylines & Character Development 14.1.3. Expanded Facility Types (Outdoor Grows, Advanced Research Labs) 14.1.4. Modding Support & Community Content Creation 14.2. Project Chimera as an Evolving Platform
15. **Glossary of Key Terms & Acronyms**
16. **Appendix** 16.1. Detailed MVP Asset List (Expanded Table) 16.2. Phased Development Roadmap (Expanded Table from Doc1) 16.3. Illustrative Skill Tree Node Breakdown (Expanded Table) 16.4. AI Tool Integration Summary Table (Expanded from Doc2) 16.5. UI Data Display & Interaction Mapping Table (Expanded from Doc2) 16.6. Core Simulation Systems Overview Table (Expanded from Doc3)

This outline provides a comprehensive framework. Each section and subsection is intended to be fleshed out with exhaustive detail, examples, specific algorithms where appropriate, and further refined tables and diagrams as development progresses.