## Part 6: World & Facility Systems Design

This part of the development plan focuses on the design and implementation of the game's primary environments and the systems that allow players to construct, customize, and optimize their cultivation facilities. These systems are crucial for providing a tangible space for the core simulation mechanics to unfold and for offering players a canvas for strategic expression and expansion. From the initial, constrained Residential House to the vast potential of the Warehouse and beyond, the world and facility systems are designed to evolve with the player's progression and mastery.

### 6.A. Sandbox Environments: Residential House (MVP Detailed Design), Warehouse (MVP Basic Build-out & Expansion Path)

Project Chimera's gameplay unfolds within distinct sandbox environments, each offering different scales, challenges, and opportunities. The MVP establishes the Residential House as the starting point and introduces the Warehouse as a significant progression goal. (Doc1, Sec VI.A).

* **MVP Recap (as per Part 4, Sec 4.3.7):**
  + **Residential House:** Predefined layout, small starting space (e.g., closet), unlockable rooms for gradual expansion. Serves as an extended tutorial area. Utilities are abstracted (wall outlets, sink taps). No X-Ray view.
  + **Warehouse:** Conceptually unlocked as a future goal. MVP access might be limited to its basic empty shell to signify progression, but not for detailed build-out or complex operations.
  + **Exterior:** Both facilities exist in an abstract "endless white abyss" with basic ground/sky visuals and an ambient day/night cycle, having no mechanical impact in MVP.
* **Full Vision: Evolving Sandbox Environments & Specialized Facilities:6.A.1. The Residential House: Mastering the Fundamentals (Full Detail)** The Residential House remains the player's initial foray into cultivation, but its full vision sees it as a more polished and nuanced starting experience.
  + **Detailed Interior Design & Atmosphere:**
    - While the layout is fixed, the interior will be rendered with high-quality assets, reflecting a clean, slightly lived-in, but ultimately professional and aspirational feel. Subtle environmental storytelling cues (e.g., unpacked boxes hinting at a recent move-in, a few non-interactive hobbyist items) will add character without clutter.
    - **Lighting:** Realistic interior lighting (lamps, ceiling fixtures, window light) that contributes to the ambiance. The day/night cycle visible through windows will affect ambient light levels, potentially having a minor (abstracted) impact on plants near windows if not supplemented by grow lights.
    - **Sound Design:** Ambient sounds appropriate for a suburban house (distant traffic, birdsong, house creaks) to enhance immersion.
  + **Progressive Room Unlocking & Purpose:**
    - Unlocking rooms (closets, bedrooms, bathrooms, basement, garage) via the early Skill Tree or introductory objectives remains.
    - Each room type might offer slight inherent environmental characteristics or constraints:
      * **Closets/Small Bedrooms:** Good for very early grows, easy to control temperature/humidity due to small volume.
      * **Basement:** Naturally cooler, potentially higher humidity. Might require more effort for ventilation. Good for drying/curing or specific strain needs.
      * **Garage:** Larger, potentially less insulated, offering more space but harder to control environment initially. Could be a stepping stone to warehouse-scale thinking.
      * **Bathrooms/Kitchens:** Access to water sources (sinks) is more explicit.
  + **Utility Access & Limitations (Refined for Full Vision within House Context):**
    - **Power:** Still primarily from wall outlets. The concept of "household circuits" could be introduced abstractly. Plugging too many high-draw items (multiple powerful lights, heaters) into outlets conceptually on the same "circuit" (e.g., within one room or a section of the house) could lead to a (simulated, non-destructive) "tripped breaker" event for that area, requiring the player to redistribute load. This teaches basic power management before the detailed wiring of the Warehouse.
    - **Water:** Manual filling of reservoirs from sinks. Drainage for runoff might be a simple "floor drain" interaction point in some rooms (garage, basement).
    - **Ventilation:** Opening/closing windows (if rooms have them) provides basic, uncontrolled air exchange with the (still abstract) outside, affecting temperature/humidity/CO2 slightly. Bathroom/kitchen extractor fans (if present) could offer minor active ventilation.
  + **Transition to Warehouse:** Unlocking all rooms in the House and achieving specific milestones (e.g., mastering basic genetics, fulfilling a set of challenging contracts, accumulating a certain amount of capital) will trigger the full unlock and operational access to the Warehouse. ADA might present this as a major "graduation" or "business expansion opportunity."
  + **C# Implementation (Residential House - Full Vision):**
    - ResidentialHouseManager.cs: Manages the state of the house (unlocked rooms, circuit loads).
    - RoomDataSO.cs (for each house room): Defines its dimensions, inherent environmental modifiers (e.g., "basement\_cooler\_mod"), available utility points (conceptual outlets/taps), and unlock conditions.
    - HouseholdCircuit.cs: Abstractly simulates electrical load for sections of the house.

The Residential House, in its full vision, becomes a richer, more reactive tutorial environment that subtly introduces concepts (like circuit loads or basic ventilation) that will become critical in more advanced facilities.**6.A.2. The Warehouse: The Canvas for Grand Ambitions (Full Detail)** The Warehouse is where Project Chimera's "Satisfactory-like" infrastructure management and large-scale cultivation ambitions truly come to life. (Doc1, Sec VI.A).

* + **Vast, Customizable Space:**
    - A truly massive, open-plan interior (e.g., 100ft x 200ft x 30ft high, or larger) providing ample room for extensive, multi-room, and potentially multi-story custom facility designs.
    - Structural elements like support pillars are present and must be built around or incorporated into designs.
    - Loading docks / large access doors might be present as non-functional (initially) visual elements, hinting at future logistics mechanics.
  + **Full Construction Freedom (See Section 6.B):** Players use the advanced grid-based and free-form construction system to partition the Warehouse into numerous specialized rooms and zones, tailored to their operational strategy.
  + **Detailed Utility Engineering (See Section 5.3 & 6.B):** This is where the full 3D routing of plumbing, electrical, and HVAC networks becomes essential. The X-Ray Utility View is a primary tool here.
  + **Scaling Operations:** The Warehouse enables:
    - **Significantly Larger Grow Rooms:** Accommodating hundreds or thousands of plants.
    - **Dedicated Rooms for All Processes:** Separate, optimized spaces for each stage of cultivation, breeding, research, and processing.
    - **Implementation of Advanced Automation:** The space and utility infrastructure support the installation and operation of complex automation systems (PLCs, robotic systems).
    - **Large-Scale Resource Management:** Managing large reservoirs, high-capacity power generation (potentially requiring dedicated generator rooms), and extensive HVAC systems.
  + **Expansion Phases within the Warehouse:**
    - Players might initially only afford to build out a small section of the Warehouse.
    - Further expansion within the Warehouse itself becomes a mid-to-late game progression goal, requiring significant capital investment for new construction, equipment, and utility upgrades.
  + **Aesthetic Transformation:** As players build out the Warehouse, it transforms from an empty industrial shell into a gleaming, high-tech, personalized cultivation and research hub, reflecting their success and design choices.
  + **C# Implementation (Warehouse - Full Vision):**
    - WarehouseFacilityController.cs: Manages the overall state of the Warehouse, including its vast grid, constructed elements, and potentially large-scale environmental factors (e.g., ambient temperature of unconditioned space).
    - Dynamic loading/unloading of sections or "chunks" of the Warehouse if it becomes extremely large and detailed, to manage performance.
    - Systems for managing structural integrity for multi-story constructions (abstracted).

The Warehouse is the player's primary sandbox for the majority of the game, offering near-limitless potential for design, optimization, and scaling.**6.A.3. Future Specialized Facility Types (Long-Term Vision - Examples):** To provide continued late-game challenges and diverse gameplay experiences, future expansions could introduce entirely new types of facilities beyond the Warehouse.

* + **Automated Greenhouses:**
    - **Concept:** Large-scale, climate-controlled structures with transparent (e.g., polycarbonate, glass) roofing and walls.
    - **Unique Mechanics:**
      * **Sunlight Simulation:** Natural sunlight becomes a primary light source. Its intensity and duration vary with the in-game time of day and season (if seasons are implemented). Cloud cover (random or weather-event driven) can reduce light.
      * **Glazing Properties:** Different GlazingMaterialSOs (glass, double-pane, UV-blocking polycarbonate) affect light transmission, insulation (R-value), and heat gain (greenhouse effect).
      * **Thermal Management:** Significant heat buildup from sunlight ("greenhouse effect") requires robust ventilation (ridge vents, side vents - automated) and potentially shading systems (ShadeClothSO - deployable equipment).
      * **Supplemental Lighting:** Still required for precise photoperiod control or to boost DLI on cloudy days or during off-seasons.
      * **Different Pest/Disease Profile:** More susceptible to airborne pests/spores from the outside environment if not well-sealed or filtered.
    - **Player Challenge:** Balancing natural light with supplemental lighting, managing extreme temperature fluctuations, optimizing ventilation.
  + **Outdoor Cultivation Plots (Very Advanced Expansion):**
    - **Concept:** Growing cannabis directly in outdoor soil plots in a specific geographic region (player chooses or is assigned).
    - **Unique Mechanics:**
      * **Dynamic Weather System:** Simulates rainfall, temperature variations (diurnal, seasonal), wind, frost risk, storms. Weather directly impacts plant growth, stress, and risk of physical damage or disease.
      * **Soil Science (Detailed):** Players amend and manage outdoor soil plots. Soil type, pH, organic matter content, drainage, and nutrient retention become critical. Living soil/organic methods are highly relevant here.
      * **Seasonal Cycles:** Planting and harvesting are tied to in-game seasons. Requires long-term planning.
      * **Natural Pest & Disease Pressure:** Exposure to a wider range of local pests, beneficial insects, and plant diseases from the surrounding ecosystem. IPM becomes even more crucial.
      * **Security (Abstracted):** Potential for theft or damage from wildlife/NPCs if plots are not secured (fencing, basic deterrents).
      * **Strain Selection:** Genetics for outdoor hardiness, pest/disease resistance, and appropriate flowering times for the region become paramount.
    - **Player Challenge:** Adapting to unpredictable natural conditions, long-term crop planning, robust IPM. Highest risk, but potentially unique quality from "sun-grown" product or ability to cultivate landraces in their (simulated) native conditions.
  + **Underground Bunkers / Vaults:**
    - **Concept:** Highly secure, fully artificial environments, perhaps for ultra-sensitive genetic preservation, black-market operations (if narrative allows), or research into extreme environmental conditions.
    - **Unique Mechanics:**
      * **Limited Space & Expansion:** Construction is very expensive and constrained.
      * **Total Environmental Reliance:** No natural light or air. All life support (light, air, CO2, temp, humidity) must be artificially generated and meticulously controlled.
      * **High Power Demand:** Significant energy costs.
      * **Stealth/Security (If Thematic):** Mechanics related to avoiding detection or securing valuable genetics.
  + **Specialized Research & Development Labs (Standalone):**
    - **Concept:** Facilities focused purely on advanced genetic engineering, tissue culture, and new product development, rather than bulk cultivation.
    - **Unique Mechanics:** Expanded lab equipment, unique research projects, potential for breakthrough discoveries not possible in a standard grow op. Might have very strict cleanliness and contamination control requirements.
  + **C# Implementation (Future Facilities):**
    - Each new facility type would require a dedicated FacilityController (e.g., GreenhouseController.cs, OutdoorPlotManager.cs).
    - New simulation modules for sunlight, weather, advanced soil science.
    - New sets of StructuralElementSOs and EquipmentDataSOs specific to these environments.

These future facility types offer pathways for significant post-launch expansions, each introducing unique gameplay mechanics, challenges, and strategic considerations, ensuring Project Chimera remains an evolving and engaging experience for years.

### 6.B. Construction System: Grid-based, Manual Utility Routing (MVP & Full Vision)

The construction system is the player's primary toolset for shaping their operational spaces, particularly within the Warehouse and future customizable facilities. It needs to be intuitive for basic use yet offer depth for complex designs. (Doc1, Sec VI.B).

* **MVP Recap (as per Part 4, Sec 4.3.7 & Doc1, Sec VI.B):**
  + **Focus:** Primarily equipment placement within predefined rooms of the Residential House.
  + **Grid:** Basic grid-snapping for equipment.
  + **Utilities:** Abstracted in the House. Manual utility routing is a Warehouse feature, so largely conceptual for MVP (though the *code foundations* for 3D utility placement might be started if the Warehouse shell is accessible).
  + **X-Ray View:** Not available in Residential House MVP.
* **Full Vision: Flexible Construction & Detailed Engineering:6.B.1. Advanced Grid & Snapping System (Full Detail):** The foundation of precise construction is a robust and flexible grid system.
  + **Core Grid Unit:** A consistent unit (e.g., 1 foot / 0.3 meters) for all dimensional calculations, structural elements, and equipment footprints. This ensures modularity and predictable alignment.
  + **Multi-Axis Snapping:**
    - **Grid Lines & Intersections:** Standard snapping of objects to the primary X, Y, Z grid lines and their intersections.
    - **Vertex Snapping:** Snap to vertices of already placed objects (walls, equipment).
    - **Edge/Midpoint Snapping:** Snap to the midpoints or edges of existing objects.
    - **Surface Snapping:** Align objects to the surface normal of walls, floors, ceilings (e.g., wall-mounted fans, ceiling lights).
    - **Angular Snapping:** Snap object rotation to predefined increments (e.g., 15°, 30°, 45°, 90°) during placement. Hotkey for free rotation.
  + **Toggleable Snapping:** Players can temporarily disable snapping (e.g., by holding a modifier key) for fine-grained freeform placement of certain objects (like decorative items or some utility segments if needed).
  + **Visual Grid Display:**
    - Clear visual representation of the active construction grid in blueprint and detail views.
    - Grid lines might become more or less dense based on zoom level or current tool.
    - Contextual highlighting of valid snap points as the player moves an object before placement.
  + **Vertical Grid & Floor Levels:** For multi-story construction, the grid system must extend vertically, with clear definition of floor levels and ceiling heights.
  + **C# Implementation:**
    - GridSystem.cs: Manages grid settings (unit size, origin), provides methods to convert world coordinates to grid coordinates and vice-versa, and calculates snap points.
    - PlacementController.cs: Uses GridSystem.cs to determine valid placement locations and orientations for objects being constructed.
    - Visual rendering of the grid lines and snap indicators (potentially using GL class for dynamic drawing or transparent textured planes).

**6.B.2. Manual Utility Routing in Full 3D (Reiteration & Expansion - see 5.3.1):** This is a core engineering gameplay mechanic, allowing players to design and implement the circulatory systems of their facility.

* + **Tool-Based Routing:**
    - Player selects a utility type (Plumbing, Electrical, HVAC) and then a specific component (e.g., "1-inch PVC Pipe," "12-gauge Wire," "6-inch Insulated Duct").
    - **"Click and Drag" or "Point-to-Point" Placement:**
      * Click to start a segment from a valid connection point (on equipment, a junction box, or an existing utility segment).
      * Move cursor to route the segment in 3D space (snapping to grid/axis where appropriate).
      * Click again to place a node/vertex (e.g., for a bend) or connect to another valid endpoint.
      * The system automatically places appropriate fittings (elbows, tees) at bends and junctions if possible, or prompts player to select/place them.
  + **Visual Feedback During Routing:**
    - Preview of the segment being routed.
    - Highlighting of valid connection points.
    - Color-coding or visual cues for invalid placements (e.g., trying to connect incompatible utility types, exceeding max segment length without support, clipping through critical structures).
    - Indication of required fittings.
  + **Layering & Clipping Avoidance:**
    - While some controlled clipping of utilities within walls/floors/ceilings is expected (and visualized with X-Ray view), the system should try to prevent egregious clipping through other utility lines or critical equipment components.
    - Players will need to plan routing in 3D space to avoid clashes, e.g., running plumbing below electrical, or ensuring ducts have clearance. This is part of the puzzle.
  + **Support Structures for Utilities (Advanced):** For long spans of pipes or ducts, players might need to place PipeHangerSO or DuctSupportSO items to prevent (simulated) sagging or stress, or to meet a "build code" requirement.
  + **C# Implementation:**
    - UtilityRoutingTool.cs: Manages the state of the currently selected utility component and player input for routing. Uses raycasting and grid snapping.
    - PlaceableUtilitySegment.cs (for pipes, wires, ducts): Handles its own mesh generation/scaling between connection points, material application, and updates its status in the relevant UtilityNetworkManager.
    - Logic for automatically suggesting or placing fittings, or for highlighting where manual fitting placement is needed.

**6.B.3. The "X-Ray" Utility View Toggle (Reiteration - see 5.3.2):** This view mode is absolutely indispensable for managing and troubleshooting the manually routed 3D utility networks. Its full functionality (translucent structures, color-coded and status-highlighted utilities, interaction in X-Ray mode, information overlays, filtering) as detailed in 5.3.2 is a core part of the construction system.**6.B.4. Structural & Equipment Placement Rules & Validation:** To ensure functional and believable facilities, the construction system incorporates rules and validation.

* + **Prerequisites & Unlocks:** Many advanced structural components (StructuralElementSO) and most equipment (EquipmentDataSO) will be locked behind Skill Tree nodes or Research Projects. Players cannot build what they haven't learned or researched.
  + **Resource Costs:** Placing any structural element or piece of equipment consumes in-game currency and potentially specific material resources (e.g., "Steel Beams," "Insulation Panels" - if a resource crafting/acquisition system is added beyond just currency).
  + **Placement Validation Logic:**
    - **Collision Detection:** Cannot place objects interpenetrating other solid objects (unless specifically designed to, like a wall-mounted sensor). Uses Unity's physics collision system during placement preview.
    - **Support Requirements:** Large floor/ceiling spans might require support pillars. Heavy equipment might need reinforced flooring. (Abstracted structural integrity checks).
    - **Clearance Requirements:** Some equipment needs specific free space around it for operation or maintenance access (e.g., a large generator needing ventilation clearance).
    - **Utility Connection Availability:** Equipment requiring power/water/etc. can only be placed if valid connection points to the respective utility networks are within reach (or if the player plans to route new utilities to it).
    - **Zone Restrictions:** Some specialized equipment may only be placeable in appropriately designated zones (see 6.C).
  + **Visual Feedback for Validation:**
    - Placement "ghost" or preview model changes color (e.g., green for valid, red for invalid).
    - UI messages explaining why a placement is invalid (e.g., "Obstructed," "Requires Power Connection," "Needs Support Pillar").
  + **C# Implementation:**
    - PlacementValidator.cs: A module used by ConstructionController.cs that checks a list of placement rules for the selected object at the target location/rotation.
    - EquipmentDataSO and StructuralElementSO will contain data fields defining their placement rules (footprint, clearance needs, support requirements, utility connection points).

**6.B.5. Modularity & Prefab-Based Design:** The system relies heavily on modularity for both structures and equipment.

* + **Reusable Prefabs:** All placeable items (walls, doors, pipes, lights, fans, benches, etc.) are Unity prefabs. This allows for efficient instantiation and easy updates/variations.
  + **Custom Assemblies (Advanced):** Players might be able to create and save "Blueprints" or "Assemblies" of commonly used room layouts or equipment configurations (e.g., a fully kitted-out seedling station with lights, trays, and humidity dome) for quick re-deployment. This is a late-game QoL feature.
  + **C# Implementation:**
    - PrefabManager.cs: Handles loading and instantiating Addressable prefabs for construction.
    - BlueprintSystem.cs (for custom assemblies): Manages saving/loading player-created multi-object templates.

**6.B.6. C# Implementation (Overall for Construction System):**

* + **Key Managers:** ConstructionController.cs, GridSystem.cs, UtilityRoutingTool.cs, PlacementValidator.cs, PrefabManager.cs.
  + **Data Objects:** StructuralElementSO.cs, PlaceableEquipmentSO.cs, UtilityComponentSO.cs (for pipes, wires, etc.), BlueprintSO.cs.
  + **MonoBehaviours:** PlacedObject.cs (base class for all placed structures/equipment), UtilityConnectionPoint.cs.
  + Extensive UI for selecting items from build menus, displaying placement previews and validation feedback, managing layers/filters in X-Ray view.

**6.AI Tooling (Full Vision for Construction System):**

* + Cursor AI: Logic for grid calculations, snapping algorithms, placement validation rules, utility routing state machines, C# backend for the build menu UI.
  + AI for 3D Assets: Generating the vast library of modular structural prefabs (walls, floors, doors with variations) and utility components (pipes, wires, ducts, fittings with different materials/sizes). All require human optimization for snapping points, colliders, and performance.
  + AI for Concept Art: Visualizing different architectural styles for facilities, innovative room layouts, or the appearance of complex utility junctions.

The full vision for the Construction System in Project Chimera provides players with powerful, flexible tools to design and engineer highly customized and optimized cultivation facilities, making the facility itself a core expression of their strategic approach and progression.

### 6.C. Zoning and Layout Optimization

Strategic zoning and thoughtful facility layout are crucial for operational efficiency, effective environmental control, risk mitigation (contamination, pest spread), and maximizing the productivity of a large-scale cultivation enterprise. The MVP introduces basic room designation, while the full vision expands this into a deep strategic layer. (Doc1, Sec VI.C).

* **MVP Recap (as per Part 4, Sec 4.3.7 & Doc1, Sec VI.C):**
  + Basic designation of entire rooms in the Residential House for specific functions (e.g., "Veg Room," "Flower Room") via a simple UI.
  + Gameplay effects are straightforward (e.g., certain equipment might only be placeable in an appropriately designated zone).
* **Full Vision: Granular Zoning, Workflow Engineering, & Environmental/Risk Management:6.C.1. Advanced Zoning Tool & Interface:** Players gain much finer control over defining and managing zones within their facilities.
  + **Drawing & Editing Zones:**
    - In blueprint mode, a dedicated "Zoning Tool" allows players to:
      * Draw custom polygonal zone boundaries that can span parts of rooms, entire rooms, or multiple rooms. Zones are not necessarily constrained by physical walls, though walls help isolate them.
      * Assign a "Zone Type" to each drawn area from an extensive list.
      * Name custom zones (e.g., "Mother Plant Sanctuary A," "Flower Bay 3 - Sativa Dominant").
      * Color-code zones on the blueprint view for easy visual identification.
      * Resize, reshape, merge, or delete existing zones.
  + **Zone Type Library (ZoneTypeSO):**
    - Predefined zone types, each with associated properties and gameplay implications:
      * **Cultivation Zones:** Seedling/Clone Room, Vegetative Bay (Early/Mid/Late), Flowering Bay (Early/Mid/Late), Mother Plant Room.
        + *Properties:* Default target environmental recipes, specific equipment allowed/disallowed, potential impact on plant growth if mismatched (e.g., flowering plants in a veg-zoned lighting schedule will struggle).
      * **Post-Harvest Zones:** Drying Room, Curing Room/Area, Trimming/Packaging Room.
        + *Properties:* Critical environmental targets (especially for drying/curing), cleanliness requirements.
      * **Lab & Research Zones:** Genetics Lab, AI Research Lab, Tissue Culture Lab.
        + *Properties:* High cleanliness requirements (affects contamination risk for sensitive processes), specialized equipment restrictions, potentially positive/negative impact on research speed/success if not properly zoned.
      * **Utility & Storage Zones:** Nutrient Mixing Station, Water Treatment Area, Electrical Room, HVAC Utility Core, General Storage (for consumables/equipment).
        + *Properties:* May have specific safety considerations (e.g., ventilation for chemical storage) or impact efficiency of resource distribution.
      * **Specialized Zones:** Quarantine Room (for new plant intake), Waste Disposal/Composting Area, Staff Breakroom/Office (if staff are implemented).
        + *Properties:* Quarantine effectiveness, sanitation impact.
  + **Zone Overlays & Information:**
    - Toggleable overlay in blueprint view showing all defined zones and their types.
    - Selecting a zone displays its properties, current environmental summary (if applicable), assigned automation controller (if any), and any active alerts related to that zone.

**6.C.2. Gameplay Impact of Strategic Zoning & Layout:** Zoning and layout are not merely cosmetic; they have profound and interconnected effects on gameplay.

* + **Targeted Environmental Control & Microclimate Management:**
    - PLCs and Central Facility Computers (see 5.3.4) can be assigned to manage specific zones, applying tailored environmental recipes (temperature, humidity, CO2, light schedules) optimized for the plants or processes within that zone.
    - Physical separation via walls and airtight doors between zones with vastly different environmental needs (e.g., high-humidity Clone Room vs. low-humidity Drying Room) becomes critical for efficient and stable control. Poorly sealed zones lead to "environmental bleed," making it harder for HVAC systems to maintain targets and increasing energy costs.
  + **Workflow Optimization & Efficiency:**
    - **Logical Flow:** Players must design layouts that facilitate an efficient flow of plants and materials:
      * Clones/Seedlings -> Vegetative -> Flowering -> Harvest -> Drying -> Curing -> Trimming -> Packaging -> Storage/Sale.
      * Nutrients/Supplies -> Storage -> Mixing Station -> Reservoirs -> Grow Zones.
      * Waste -> Disposal/Composting.
    - **Minimizing Travel Time (Abstracted or Staff-Based):**
      * Inefficient layouts with long travel paths between dependent zones can lead to (abstracted) operational inefficiencies, increased labor time (if staff are simulated), or slower throughput.
      * The game might provide feedback on layout efficiency through ADA hints or a "Facility Efficiency Report."
    - **Accessibility for Maintenance:** Ensuring equipment in utility zones or within grow rooms is accessible for (simulated) maintenance or repair.
  + **Contamination & Pest/Disease Spread Management:**
    - **Zone Isolation:** Well-defined zones with controlled access (e.g., requiring "decontamination" actions like changing conceptual coveralls or walking through a foot bath when moving between a "dirty" outdoor access zone and a "clean" indoor grow zone) are crucial for preventing the spread of pests and diseases.
    - **Quarantine Zone:** All new plants (from NPC vendors, trades, or expeditions) *must* pass through a quarantine zone for a period of observation before being introduced to main production areas. Failure to do so carries a high risk of introducing new pests/diseases.
    - **Airflow Control (HVAC Design):** Strategic HVAC design can create positive/negative pressure differentials between zones to further control airborne contaminant flow (e.g., positive pressure in clone rooms to keep contaminants out, negative pressure in rooms with potential odor or spore release to exhaust air safely).
    - **Traffic Flow:** Designing layouts to minimize unnecessary foot traffic through sensitive areas (like flowering rooms or labs).
  + **Safety & Hazard Management (Especially for Advanced Processing):**
    - Solvent-based extraction labs must be zoned with specific safety features (explosion-proof fans, gas detectors, potentially fire suppression systems - all placeable equipment).
    - Chemical storage areas need proper ventilation and separation.
    - Failure to adhere to safety zoning for hazardous processes can lead to simulated accidents (fires, explosions) with severe consequences (facility damage, crop loss, financial penalties, research setbacks).
  + **Resource Distribution Efficiency:**
    - Centralizing utility hubs (electrical rooms, water treatment/nutrient mixing stations) and planning efficient distribution networks (pipes, wires, ducts) to various zones minimizes resource loss (e.g., pressure drop in long pipes, voltage drop in long wires) and construction costs.

**6.C.3. Visual Analysis Tools for Layout Optimization (Post-MVP - Full Detail):** To help players understand and optimize their complex facility layouts, a suite of visual analysis overlay tools will be available (unlocked via research or skill tree).

* + **Environmental Heatmaps/Overlays (Reiteration - see 5.7.4):**
    - Toggleable overlays in blueprint or X-Ray view showing spatial distribution of:
      * Temperature (identifying hot/cold spots).
      * Humidity / VPD (identifying overly damp or dry areas).
      * Light Intensity (PAR/PPFD at canopy level, showing uniformity or dark spots).
      * CO2 Concentration.
      * Airflow Quality/Velocity (showing areas of good circulation vs. stagnant air).
    - Data is derived from the microclimate simulation grid (5.3.3).
  + **Workflow Pathing Visualization (Abstracted):**
    - Player can select a process (e.g., "Seedling to Veg Transplant," "Nutrient Solution Delivery").
    - The tool highlights the most common or optimal paths for plants/materials/staff (if simulated) on the facility blueprint.
    - Helps identify inefficient routes, long travel distances, or bottlenecks (e.g., too few doors, narrow corridors).
  + **Contamination Spread Risk Overlay:**
    - Based on current pest/disease presence, zone adjacencies, airflow connections (HVAC), and defined traffic patterns, this overlay visualizes areas at high, medium, or low risk of cross-contamination.
    - Helps players identify critical control points for improving biosecurity.
  + **Utility Network Efficiency Overlay:**
    - **Electrical:** Highlights circuits near capacity, areas with significant voltage drop (if simulated), or inefficiently long wire runs.
    - **Plumbing:** Shows areas with low pressure, potential flow restrictions, or inefficiently long pipe runs.
    - **HVAC:** Visualizes airflow distribution efficiency, areas not adequately served by vents, or zones where HVAC is struggling to maintain targets due to poor insulation or sealing.

**6.C.4. C# Implementation (Full Vision for Zoning & Layout):**

* + Zone.cs (Data Class/SO): Stores zone type (enum ZoneType), custom name, list of encompassed grid cells or geometric boundary definition, assigned environmental recipe EnvironmentalRecipeSO, link to assigned PLC\_Controller.cs (if any), current contamination status/risk.
  + ZoningManager.cs:
    - Manages all player-defined Zone objects for a facility.
    - Provides methods for other systems to query zone information (e.g., GetZoneType(Vector3 worldPosition), GetEnvironmentalRecipeForZone(Zone zone)).
    - Handles UI interaction for drawing/editing zones.
    - Provides data to the visual analysis overlay tools.
  + FacilityEfficiencyCalculator.cs: Abstractly calculates workflow efficiency scores based on zone adjacencies, travel distances between key processing areas, and layout of resource distribution.
  + ContaminationSpreadSimulator.cs:
    - Models the probability of pests/diseases spreading between adjacent or connected zones.
    - Factors in zone type (e.g., higher risk spreading from a "Waste Disposal" zone), airlock effectiveness, HVAC connections (shared air), player/staff movement patterns (abstracted), and current sanitation levels.
    - Updates Zone.contaminationRisk values.
  + SafetyManager.cs: Monitors hazardous zones (e.g., solvent extraction labs) for compliance with safety equipment requirements and triggers accident events if safety protocols are breached.
  + UI C# scripts for the advanced zoning tools and the various analysis overlay renderers.

**6.AI Tooling (Full Vision for Zoning & Layout):**

* + Cursor AI: Logic for ZoningManager, FacilityEfficiencyCalculator, ContaminationSpreadSimulator, SafetyManager. C# backend for the zoning UI and analysis overlay data processing.
  + AI for Procedural Layout Suggestions (Highly Experimental Future R&D): Potentially, an AI could analyze a player's production goals and available space/equipment and suggest a few *template* optimal room layouts or zone configurations as starting points, which the player can then customize. This would be a very advanced feature requiring significant R&D into procedural generation and constraint satisfaction.
  + AI for Concept Art: Visualizing efficient vs. inefficient facility layouts, examples of good zoning for different operational scales.

Strategic zoning and layout optimization elevate facility design from mere construction to a deep, systems-thinking challenge. Players who master these concepts will create highly efficient, resilient, and productive cultivation empires, while those who neglect them will face inefficiencies, increased risks, and difficulties in scaling their operations. This system is key to the long-term strategic depth of Project Chimera.