**Spectra Game Core Systems Blueprint (Developer Design Document)**

**Version:** Draft 0.1  
**Design Lead:** [Brett Whitson]  
**Theme:** Optical Physics and Spectroscopy translated into interactive systems  
**Game Engine Target:** TBD (ASCII/WebGL prototype → later port to 2D engine e.g., Godot or Unity)

**1. Core Concept**

**1.1 Vision**

“*A self-aware spectrophotometer repairing itself from the inside.*”  
You control a **photon emission core** navigating the instrument’s optical modules, restoring functionality and unlocking new spectral ranges.  
Gameplay combines **precision puzzles, reaction chaining, and system management**, grounded in real optical principles.

**1.2 Genre**

Hybrid physics puzzle + simulation roguelike.  
Inspirations: *Portal*, *Opus Magnum*, *Factorio*, *Antichamber*, *Nova Drift* (for upgrade system).

**1.3 Player Goal**

Deliver photons of specific **wavelengths, polarization, and energy** to trigger, excite, or measure optical components.  
Rebuild modules, restore spectral range, and achieve full spectrum coherence.

**2. System Overview**

|  |  |  |
| --- | --- | --- |
| **System** | **Purpose** | **Core Loop Interaction** |
| **Photon Control** | Player movement, aiming, firing photons | Defines feel and precision |
| **Optical Environment** | Mirrors, prisms, polarizers, filters, absorbers, samples, detectors | Generates puzzles |
| **Spectroscopy Core** | Translates physical phenomena into game mechanics | Core educational/gameplay hook |
| **Power & Energy Management** | Battery, beam intensity, heat | Risk/reward system |
| **Progression Layer** | Unlocks new wavelength ranges, optics, and photochemical tools | Long-term motivation |
| **Data & Analysis UI** | Display results and mission success | Reinforces learning and feedback |

**3. Physics → Gameplay Translation**

**3.1 Photon Dynamics**

* **Base properties:**
  + Wavelength (λ, nm) → Energy (E = 1240 / λ eV)
  + Polarization (θ) → Determines pass/fail on polarizer tiles
  + Coherence (phase φ) → Used for interference-based puzzles
* **Behavior:**
  + Reflect, refract, absorb, scatter, or split on interaction.
  + Intensity decays over distance per Beer–Lambert law:  
    I = I₀ \* e−αx
  + Attenuation coefficient (α) varies by medium (air, glass, solution).

**3.2 Optical Interactions**

|  |  |  |
| --- | --- | --- |
| **Component** | **Behavior** | **Implementation Notes** |
| **Mirror** | Reflects beam; θout = θin | Axis-aligned or angled. Handle recursion safely. |
| **Prism** | Splits beam into component λ bands | Spawn new photons at adjusted λ offsets. |
| **Polarizer** | Transmits beam if polarization matches | T = cos² (θin − θfilter) intensity multiplier. |
| **Sample** | Absorbs selectively; may fluoresce | Gaussian absorbance curve; optional delayed emission. |
| **Detector** | Counts photons meeting target λ range | Triggers mission objectives; logs λ histogram. |
| **Filter / Grating** | Passes or deflects wavelengths | Useful for tuning and gating puzzles. |
| **Photochemical Target** | Requires threshold energy to react | If E ≥ , trigger event (bond break, emission, unlock). |

**3.3 Reaction Logic Flow**

Photon fired →

intersects optical object →

evaluate interaction type →

update beam (λ, E, I, θ, φ) →

spawn emission / trigger reaction →

update mission state (detector count, calibration progress)

**4. Player Control & Input Model**

|  |  |  |
| --- | --- | --- |
| **Action** | **Input** | **Output/Feedback** |
| Move | WASD (acceleration + friction) | Smooth inertial motion |
| Aim | Mouse (free) / Lock (RMB/L) | Rotates emission vector |
| Fire | Space / LMB | Emits photon packet(s) |
| Tune λ | Scroll | Real-time beam color shift |
| Adjust Polarization | Q / E | Rotates polarization axis |
| Overcharge Pulse | Hold Space | High-intensity emission, heat cost |
| Analyze / Pause | Tab | Opens spectrograph view (data overlay) |

**Design Notes:**

* Control latency target <50 ms input-to-display.
* Response curve: acceleration uses exponential ease-in/out to feel weightless but responsive.
* Visual feedback: wavelength hue + audio pitch = intuitive light feedback loop.

**5. UI / UX Framework**

**5.1 Layout**

**HUD Zones:**

* Top-Left: Photon status (λ, E, polarization, battery)
* Bottom-Left: Power & heat meters
* Top-Right: Mission panel (objectives, detector readings)
* Center-Right: Mini-spectrograph (histogram of detected wavelengths)
* Bottom-Right: Tooltip zone (context-sensitive optical data)

**5.2 UI Feedback**

* Real-time color shift as λ changes (hue = wavelength)
* Beam intensity falloff visualization (opacity fade)
* Subtle pulse audio at firing rate (frequency = λ)
* Detector success → click + flash
* Sample excitation → glow + afterimage decay

**5.3 Interface Design Principles**

* **Minimalism:** data-dense but visually calm.
* **Color = Data:** hue encodes λ, brightness encodes intensity.
* **Continuous Feedback Loop:** every interaction changes visuals immediately.

**6. Progression Systems**

**6.1 Unlock Tree**

|  |  |  |
| --- | --- | --- |
| **Tier** | **Unlock** | **Gameplay Impact** |
| **Tier 1** | Visible Range (600–700 nm) | Basic reflection and detection puzzles |
| **Tier 2** | Full Visible (400–700 nm) | Enables color-coded reactions |
| **Tier 3** | UV (200–400 nm) | Allows photochemical bond breaking |
| **Tier 4** | IR (700–1000 nm) | Heat manipulation and thermal gates |
| **Tier 5** | Coherence Control | Enables interference puzzles |
| **Tier 6** | Polarization Layer | Adds alignment and rotation mechanics |

**6.2 Upgrade Categories**

* **Source Power:** increase emission intensity (battery drain trade-off).
* **Optical Precision:** tighter beam divergence (improves accuracy).
* **Detector Sensitivity:** higher count accuracy, lower noise floor.
* **Cooling Systems:** raise thermal tolerance before overheating.
* **Spectral Expansion:** unlock new λ bands.

**6.3 Meta Progression Loop**

Each “run” = one calibration cycle.  
Player restores a module (e.g., monochromator).  
Completion yields **spectral tokens** used to unlock new ranges/modules in future runs.

**7. Environmental Systems**

|  |  |  |
| --- | --- | --- |
| **Type** | **Purpose** | **Behavior** |
| **Optical Chambers** | Puzzle rooms | Closed circuits of mirrors/prisms |
| **Reactive Zones** | Photochemical arenas | Change layout when triggered |
| **Dark Matter Fields** | Absorb all light | Force detour / spectral adaptation |
| **Detector Banks** | Goal zones | Measure cumulative photon data |
| **Cooling Flow Paths** | Manage heat through routing | Thermodynamic sub-puzzle |

**8. Future Systems & Scalability**

**8.1 Procedural Samples**

* Randomly generated Gaussian absorbance spectra.
* Hidden “unknowns” players identify by probing wavelengths.
* Dynamic difficulty via sample complexity.

**8.2 Beer–Lambert Integration**

Intensityout = Intensityin \* exp(−α(λ) \* thickness)

* α(λ) derived from sample spectra.
* Visual feedback: beam brightness dims proportionally.

**8.3 Reaction Trees**

|  |  |  |
| --- | --- | --- |
| **Event** | **Trigger Condition** | **Result** |
| Photo-excitation | E ≥ threshold | Emission (fluorescence) |
| Ionization | E >> threshold | Permanent change to material |
| Bond formation | Two beams overlap in phase | Create new optical path |
| Overheat | Sustained high energy | Local blackout or meltdown |

**9. Technical Architecture Overview**

**Subsystems:**

* PhotonEngine: manages emission, propagation, attenuation.
* OpticsSystem: handles collisions & interactions with tiles.
* SpectroscopySystem: tracks counts, absorption, emission.
* UIManager: updates displays and feedback.
* ProgressionManager: handles upgrades and unlocks.

**Entity Definition Example:**

{

"entity": "Sample",

"absorbanceCurve": [ {"λ": 520, "σ": 15, "strength": 0.9} ],

"fluorescence": { "shift": 60, "prob": 0.25 },

"state": "stable"

}

**10. Key Development Priorities**

1. Finalize photon physics module (reflection, absorption, emission).
2. Implement sample & detector object classes with event hooks.
3. Build scalable UI overlay (λ slider, battery meter, histogram).
4. Prototype one full mission loop: **Source → Sample → Detector → Analysis → Unlock.**
5. Integrate save system for meta progression.
6. Establish test suite for wavelength/energy consistency.