ChatGPT

DonutOS Optics & QFT Lab – Feature Specification

Key Metaphors and Features

Figure: Example of diffraction vs. interference. Left: a single-slit diffraction pattern (broad central maximum) vs. Right: a double-slit interference pattern with alternating bright and dark fringes, as originally observed by Young [1]. The Optics & QFT Lab will let users overlay and explore such interference patterns interactively.

- **Interference Overlays:** At the core is the ability to overlay multiple wave sources and visualize their **interference pattern** in real-time. When two or more coherent waves overlap, they produce alternating regions of high and low intensity (bright and dark fringes) due to constructive and destructive interference [2] [3]. The Lab will simulate classic setups (e.g. double-slit experiments) where adding a second source creates an interference pattern [4]. Users can toggle single vs. multiple sources to see how a stable fringe pattern emerges only when waves are phase-coherent. (Two independent light sources with random phase would produce no stable pattern, whereas splitting one source into two beams yields stable fringes [5].) These overlays convey wave **superposition** intuitively, helping users "see" interference by projecting fringe patterns onto a virtual screen or environment.

- **Phase-Lock & Holonomy Controls:** To explore coherence, the Lab provides controls to **phase-lock** sources or introduce controlled phase shifts. A "phase knob" slider can continuously adjust the relative phase between beams from 0 to 2π, causing the interference fringes to **shift position** or oscillate [6]. This simulates moving a mirror in a Michelson interferometer or changing path length – cycling the phase swaps bright and dark fringe locations [6]. In *unlocked* mode, each source's phase drifts independently (emulating incoherence, yielding washed-out or zero visibility pattern). In *locked* mode, a global phase can be tuned uniformly. Advanced **holonomy** mode (opt-in) will let users introduce a phase loop around a closed path – a conceptual nod to geometric phase. For example, users could simulate the **Aharonov–Bohm effect** by adding a phase shift when waves travel around a region with magnetic flux: even if no field acts locally on the beams, a phase holonomy (global loop phase) would shift the interference pattern [7]. This connects to QFT concepts (phase factors from gauge fields) in an accessible visual way. In short, phase-lock controls clarify that interference depends on maintaining a stable phase relationship, and holonomy controls illustrate how accumulated phase around loops can affect observable fringes.

- **Quasicrystal & Multi-Beam Patterns:** Moving beyond two-source interference, the Lab enables complex **multi-beam** superposition to produce intricate stationary patterns. By aiming several coherent beams at a screen (as in holographic interference lithography), users can generate both periodic and *aperiodic* lattices. For example, overlapping **5 laser beams** at equal angles produces a **five-fold symmetric** intensity pattern – a 2D *Penrose quasicrystal* interference pattern [8]. This pattern lacks translational symmetry but has long-range order (like a quasicrystal), demonstrating how interference can create structures with exotic symmetry [8]. Similarly, 6 beams can yield a periodic hexagonal lattice pattern. The UI will offer presets (e.g. "Penrose 5-beam" or "Hexagonal 6-beam") and allow tuning beam angles or phases. Users can observe how adjusting relative phases of the beams changes the pattern (e.g. shifting or blurring it) without altering its symmetry. This feature showcases interference as a **tool for pattern generation**, from the familiar Young's fringes to complex moiré and quasicrystal patterns. It highlights that by **phase-aligning multiple waves**, one can "paint" intricate intensity landscapes – a direct analogy to how crystallography or photonic lattices are made with laser interference [8]. Visualization of these patterns will be overlaid on a canvas with options to zoom or measure distances, reinforcing concepts like symmetry and periodicity in wave physics.

- **Safety & Comfort Guardrails:** Because the Lab deals with intense visual patterns and possibly animations, built-in guardrails will ensure a comfortable user experience. **No rapid flashing** or high-frequency strobe effects will be used that could trigger photosensitive epileptic responses. We will cap any brightness oscillation or blinking of patterns to **3 flashes per second or less**, since flashing above ~3 Hz (especially in the 5–30 Hz range) is known to induce seizures in susceptible individuals [9]. For example, if a user sets the phase knob to auto-oscillate, the system will limit the rate of fringe movement cycles to stay below this threshold [9]. Additionally, the Lab avoids stark red/blue flickering contrasts and high-intensity strobing (users are more sensitive to red flashing than other colors [9]). A **comfort mode** can further reduce motion: e.g. pausing animations by default and using softer color schemes for interference fringes. The UI will also provide warnings or require confirmation if a user tries to enable any dynamic effect that might be intense. These guardrails ensure the Lab's visualizations remain accessible and do not cause nausea, dizziness, or discomfort – aligning with web accessibility guidelines for animations and flashing content [10] [11].

Product Specification

User Stories and Use Cases

- *Student Physicist:* A high-school student exploring wave physics can use the Optics & QFT Lab to simulate classic experiments. For example, they can recreate **Young's double-slit**: selecting a laser source, enabling a barrier with two slits, and observing the interference pattern on a screen. The student adjusts slit separation and sees fringes spacing change, helping them learn that smaller slit spacing increases fringe spacing (and verifying $d \sin\theta = m\lambda$ predictions in an opt-in calculation mode). They can toggle a "Phase Randomizer" to see that if the two slits were fed by incoherent sources, the pattern disappears – reinforcing the need for coherent phase. They leave the session with a concrete visual of wave interference, satisfying a learning goal of identifying constructive vs. destructive interference patterns.

- *Advanced Learner (Optics/QFT enthusiast):* A college student or hobbyist can experiment with more complex setups. For instance, they choose a **multi-beam interference** scenario with 5 beams arranged in a circle to generate a quasicrystal pattern. Using the phase-lock controls, they vary the common phase and watch the entire stationary pattern simply fade in/out (constructive vs. destructive alignment) without losing its 5-fold symmetry. Then they deliberately *unlock* one beam's phase to see the pattern decohere into a blur, illustrating sensitivity to phase noise. In another use case, they enable *Holonomy Mode* and simulate an Aharonov–Bohm-like scenario: two paths around a region with a "magnetic flux" phase difference. By tuning that phase difference, they observe shifts in the interference fringes position, akin to the electron interference shift in the AB effect. This satisfies their curiosity about how optical phenomena can analogize quantum field effects.

- *Educator/Demo Presenter:* A physics teacher uses the Lab live in class to demonstrate interference phenomena safely, instead of a fragile physical setup. They quickly pull up preset scenes: a Michelson interferometer demo (showing circular fringes), a double-slit demo, etc. They utilize the **Educational Mode** (see below) so that key concepts and equations are annotated on-screen (e.g. a label showing “Path difference = $m\lambda$ leads to bright fringe” at a highlighted fringe). The educator also appreciates the **safety features** – e.g. when demonstrating fringe motion by moving the virtual mirror, the Lab automatically slows down the animation to a gentle oscillation, preventing any sudden flashes that could discomfort students. After the demo, students can individually explore the same simulations on their devices (the teacher provides the DonutOS Lab as it’s integrated into their DonutOS environment or accessible on the web), facilitating interactive learning and experimentation beyond the lecture.

UI/UX Design and Interaction

The Optics & QFT Lab’s interface is designed for clarity and interactivity, following familiar **lab bench metaphors**. The main area is a **2D canvas** or virtual optical table where wave sources, obstacles (slits, mirrors), and detectors/screens are depicted. Users can drag and position components (for example, place a screen farther away to see fringe spacing change, or add a second source by dragging it out of a toolbar). Key UI elements and patterns include:

- **Toolbar/Palette:** A side panel lists available components: point source (laser or point wave emitter), slit barrier, mirror, beam splitter, detectors, etc. Users drag components into the canvas to build a setup, or load a preset configuration with one click (presets for common experiments are provided to reduce setup time).

- **Overlay Controls:** When an interference pattern is present, a semi-transparent overlay shows the fringe pattern on the screen or across the field. This overlay can be toggled between a continuous intensity map (smooth color gradients for intensity) and discrete fringes (contour lines or bands). For instance, in a water wave mode, the peaks and troughs might be shown as blue and white patterns, whereas in light mode, bright fringes could be shown in yellow/white on a black background. Controls for overlay brightness/contrast are available so the user can make the pattern more or less pronounced against the background.

- **Sliders and Knobs:** Key experimental parameters are accessible via sliders:

- A *wavelength slider* to adjust the wavelength of the light (or water wave frequency). Moving this will immediately update the fringe spacing (shorter wavelength -> fringes get closer).
- A *slit separation slider* (or geometry control) for double-slit or grating components, updating the pattern accordingly.

- The **Phase Knob** – a circular dial UI or slider that controls the relative phase between a reference beam and another beam. For example, in a Michelson interferometer preset, this knob effectively moves one mirror: turning it shifts the concentric fringes continuously. The knob has markings from 0 to 2π and can be rotated; the UI might also allow manual input of phase in radians. If multiple phase parameters exist (e.g. 5 beams each could have a phase), a simplified mode either links them (common phase) or offers multiple knobs. We ensure that adjusting phase yields a smooth animation of fringes rather than a jump, to clearly see the movement of interference bands [6] .

- **Toggles and Buttons:** There are toggle switches for turning on/off certain effects:

- *Coherence Toggle:* When ON, all sources derive from a single master source (fully coherent, stable interference). When OFF, sources are treated as independent (introducing random phase jitter so that long-term observation washes out the pattern). This toggle lets users qualitatively see the effect of coherence (it essentially animates rapid phase drift when off).
- *Holonomy Mode:* When enabled, the UI highlights closed paths in the setup (e.g. loops around which a phase could accrue). The user can then add a “loop phase” via a dial, and the system will apply that phase offset to one path of a two-path interferometer. The result is an interference shift even though other parameters unchanged, visualizing geometric phase addition.
- *Intensity/Log Scale:* A toggle to view intensity on a logarithmic scale or linear scale, useful for patterns with high dynamic range (diffraction patterns with faint higher-order fringes).

- *2D/3D View:* If applicable, a toggle to switch between a planar view and a 3D visual (e.g. showing a 3D wavefront or surface in a ripple tank mode, though core interactions are 2D).

- **Feedback Display:** The UI provides real-time numerical feedback. For example, as the user adjusts parameters, a small tooltip or readout might show “Fringe spacing = X mm” or “Phase difference = Y radians”. If the user moves a detector to measure intensity at a point, the readout could give the intensity value and even the theoretical value for fully constructive vs destructive interference at that point.

- **Visual Aids:** The interface uses color and graphics to aid understanding. Coherent beams might be represented with the same color hue to indicate they are phase-linked, whereas an incoherent source might be gray. When phase is adjusted, subtle visual cues (like a rotating phase wheel icon next to the source) indicate the phase value. During holonomy mode, an arrow around the loop could depict the phase angle currently applied.

Overall, the UI pattern follows an **interactive simulation** style (akin to PhET simulations [4] but integrated in DonutOS), where users can directly manipulate the scene and see immediate visual results. Common actions (adding sources, toggling coherence, adjusting phase) are all one or two clicks away, minimizing any form-based input. This encourages playful exploration – a user can, for instance, drag five sources into a pentagon arrangement and instantly see the resultant interference pattern emerge on the canvas, then use a single slider to synchronously rotate all five (changing the pattern orientation) or detune one (seeing the pattern distort).

Metrics and Visualization Outputs

To quantify what users see, the Lab will compute and display certain **metrics** related to the interference patterns:

- **Fringe Visibility:** For two-source interference, the lab calculates the **Michelson visibility** of the fringes, defined as $V = (I_{\max} - I_{\min})/(I_{\max} + I_{\min})$ [12]. This value (between 0 and 1) appears on a dashboard. A visibility of 1.0 means perfect contrast (bright maxima and dark minima), indicating perfectly coherent sources [13] [14]. A visibility of 0 means no interference contrast (e.g. incoherent sources) [15]. This metric updates in real time – if the user slowly de-phases one source, they can watch visibility drop from ~1 toward 0, quantitatively reinforcing the concept of coherence.

For multi-beam patterns, a generalized visibility or contrast measure may be shown, or separate visibilities for different pairs of beams if relevant.

- **Order Parameter (Symmetry Metric):** In multi-beam/quasicrystal mode, the lab can display a measure of the pattern's order or symmetry. For instance, for a N-beam setup, it might detect the dominant symmetry (e.g. five-fold for N=5 beams) and show an "order parameter" indicating how well-defined that symmetry is. This could be as simple as detecting the contrast of the corresponding diffraction peaks: e.g., in a 5-beam Penrose pattern, if all five-fold symmetric diffraction spots are sharp, the order parameter ~1 (strong quasicrystal order); if the pattern is perturbed (e.g. one beam intensity lowered or incoherent), the diffraction peaks weaken and the order parameter drops. While a rigorous definition might be complex, qualitatively it gives feedback on pattern formation. Alternatively, an **FFT analysis** could run in the background: the presence of clear discrete peaks in the spatial frequency spectrum could be quantified and shown as evidence of long-range order. This is an advanced metric mostly for enthusiasts, but it connects to how physicists identify crystal vs quasicrystal order in interference patterns [16] [17].

- **Phase Difference Indicator:** When the phase knob is used, an onscreen indicator (like a rotating arrow or a phase clock icon) illustrates the phase difference. If holonomy is active, a small loop graphic might show the total phase around the loop (like a circle with an arrow partway around to indicate, say, a $\pi/2$ total phase). This is more of a visual cue than a metric, but it helps users correlate the numeric phase value with a visual representation.

- **Intensity Profiles and Data:** Users can place virtual photodetectors on the screen to record intensity across a line. The UI will then plot an intensity graph (I vs position) so the user can see a cross-sectional plot of the interference fringes. This graph can be analyzed to extract fringe spacing, etc. The user might use a built-in fitting tool which, for example, fits the intensity pattern to $I(x) = I_0 [1 + V\cos(\delta x)]$ for double-slit, verifying that the measured visibility V matches the theoretical value. Such data-centric features turn the visual patterns into quantitative experiments, aligning with educational goals of connecting theory and observation.

The metrics and visual outputs update live but can be frozen (paused) when needed so users can take measurements or screenshots. All these provide feedback that deepens understanding: e.g., seeing visibility go to 1 or 0, seeing how a five-beam pattern maintains high order parameter only if all beams remain phase-locked, etc. The Lab essentially becomes not just a toy but a measurement tool, reinforcing learning by quantification.

Opt-in Educational Mode

The Optics & QFT Lab includes an **Educational Mode** that can be toggled on, primarily for classroom use, self-study, or beginners. In this mode, additional explanatory overlays, guidance, and restraints appear:

- **Concept Annotations:** Important regions of the simulation are labeled. For example, in a double-slit setup, an annotation might appear on the screen saying "Bright Fringe (Constructive Interference, path difference = $m\lambda$)" at the central maximum, and "Dark Fringe (Destructive Interference, path difference = $(m+0.5)\lambda$)" at the first minimum [18]. Little callout boxes might track the moving fringes as phase is adjusted, explaining in text what is happening ("Fringes shifting: path difference changing by fraction of wavelength"). These annotations turn the simulation into a guided lesson.

- **User Prompts and Tasks:** The mode can present small tasks or questions to the user. For example: "Increase the phase difference until the fringes shift one whole fringe spacing. How much phase change is that? (Answer: 2π radians)." The Lab can detect the user's action (e.g. they turn the phase knob until one fringe moves over by one period) and then display the answer or next hint. Another task: "Try adding a second source. What pattern do you observe?" – after the user does so, the on-screen text might explain "You created an interference pattern [19] ! The alternating bands are where waves from Source 1 and 2 arrive in phase or out of phase."

- **Equations and Calculations:** In educational mode, relevant equations can be superimposed and even calculated live. For instance, alongside the double-slit pattern, the formula $d \sin\theta = m\lambda$ could be shown. The simulation knows d and λ, so if the user clicks on the m=1 fringe, it could compute $\sin\theta$ and verify the equation for them. Similarly, the lab could display the formula for fringe intensity: $I(\phi) = I_{\max}\cos^2(\phi/2)$ or $I_{\min}$ etc., and as the user changes phase φ, a small overlay might show the expected intensity at a certain point and compare to simulation. This bridges theory and experiment.

- **Safety Reminders:** Educational mode can also double down on comfort by reminding the user why some things are limited. For example, if they attempt to set a high flashing rate, a note might pop up: "Flashing too fast can be dangerous for some viewers [9] . The simulation is limiting this for safety." This not only prevents an action but educates about real-world constraints and accessibility.

- **Guided Presets:** Instead of dropping users in a blank slate, educational mode may start with a guided sequence of scenarios. E.g., Step 1: Single slit (observe diffraction), Step 2: Double slit (observe interference), Step 3: Increase slits (diffraction grating, more complex pattern), Step 4: Michelson interferometer (rings), etc. Each step automatically configures the scene and provides explanatory text. Users can still tinker at each step, but the mode ensures they don't get lost. It's like a built-in tutorial.

Educational mode is *opt-in* to keep the main interface uncluttered for advanced users. When off, the UI is clean and sandbox-like. When on, it essentially turns the Lab into an interactive textbook, with the simulation providing the experience and the annotations providing the pedagogy. This dual design means the Lab serves both as an open exploratory tool and a structured learning tool, depending on user needs.

Implementation Notes

- **Lightweight WebGL Demos:** The Lab will leverage WebGL for real-time wave computations and rendering, ensuring a smooth interactive experience even for complex patterns. Each wave source can be represented as a simple harmonic field (e.g., using shader code to sum up contributions of each source at each pixel). For instance, the intensity on the screen from two slits can be computed by summing two sinusoidal waves and squaring the magnitude – a trivial operation for a GPU fragment shader. We can easily extend this to multiple beams: e.g., a shader summing 5 plane waves (with vectors corresponding to chosen beam angles) will produce the Penrose interference pattern when their phases align [20] [16] . These calculations are efficient and allow dynamic updates as users move sliders (the uniforms in the shader for phase or angle can be updated each frame). For 2D wave propagation (like ripples or near-field diffraction), we can use a simplified wave equation simulation or interference integral computed via WebGL as well. The goal is to keep simulations **lightweight** – focusing on key optics formulas (superposition, interference) rather than full heavy physics – to ensure it runs in real-time on typical hardware (including possibly in a browser). Existing projects like PhET's Wave Interference (HTML5/JavaScript) and Falstad's Ripple tank (WebGL) prove this is feasible [4] [21]. We can crib from such open-source efforts (Falstad's rippleGL code [21] is a great reference) for handling wave math and GPU acceleration. Each demo (double-slit, multi-beam, etc.) will essentially be a configuration feeding into the same underlying engine that computes interference patterns by summing waves.

- **Phase-Lock and Knob Interaction:** Implementing the phase knob involves synchronizing the simulation update loop with the UI input. As the user drags the phase knob, the relative phase parameter in the shader or calculation is updated continuously. To represent incoherence when phase-lock is off, we can simulate it by rapidly randomizing the relative phase (e.g., changing it by a random jump every few frames) or by time-averaging the intensity. For simplicity, a fast method is to set the wave from an unlocked source to have a time-dependent phase $\phi(t)$ that drifts quickly (faster than the eye can follow), leading to a time-averaged uniform intensity. The result is the pattern "washing out" when unlock is toggled – effectively showing $V\to0$. We must ensure that toggling coherence on->off is visually clear: perhaps a short transition where the fringes fade away. When the user toggles back on, we can even illustrate re-phasing: e.g., briefly show the two beams' phase as different colors converging back in step. The holonomy control can be implemented as an added phase term in one path's phase calculation. For example, in a Mach-Zehnder interferometer setup, normally intensity ∝ $\cos^2(\Delta\phi/2)$; if a holonomy phase Φ is inserted in one arm, we just add Φ to $\Delta\phi$. This will shift the cosine curve accordingly. Thus, the implementation is straightforward: it's literally an added constant to one beam's phase accumulator. The challenge is providing a UI affordance for it, which we solved via the loop toggle and separate dial. We will ensure that phase changes are applied smoothly over a few frames to avoid any discontinuous jumps (unless intentionally demonstrating a sudden phase flip). As a validation, we can compare our fringe shifting to known physics: e.g. a full 2π knob turn should shift a fringe by one spacing – we'll test that visually in the sim to ensure accuracy.

- **Safe Animation & Visual Comfort:** Implementing the safety guardrails requires monitoring any time-dependent changes in the simulation. For any automated oscillation (like if we let the phase knob auto-rotate for a "continuous phase sweep" demo), we will cap the angular frequency such that the resulting intensity flicker at any given point is ≤ 3 Hz [9]. Concretely, if rotating phase by 2π causes one full brightness oscillation at a point, doing that three times a second is the limit. We will set default animations to something like 1 Hz for extra safety. Additionally, we will avoid stark contrast flashing. If an animation does cause a significant portion of the screen to flash (e.g., switching between an all-bright and all-dark pattern), the Lab can automatically insert intermediate frames or reduce contrast to soften the flash. The implementation can use CSS or WebGL shaders to **dim** extreme changes – for instance, if the entire screen would invert brightness, we can fade it over 0.5s instead of an instant swap. Furthermore, user settings in DonutOS accessibility preferences will be respected: if the OS signals "reduce motion," the Lab will disable any non-essential animation (so phase changes only occur on user drag, not auto-oscillate, etc.). We'll also include a quick **"Panic" stop** – a global pause button readily visible – so if a user feels discomfort they can stop all motion immediately. All these measures align with W3C and industry guidelines for avoiding photosensitive triggers (no content that flashes more than 3 times per second over a significant portion of screen

[9]). We will test the Lab with high-contrast modes as well, ensuring the color choices for interference patterns can be adjusted (some users may prefer a monochrome fringe display to avoid any problematic colors). The safety features will be thoroughly documented, and whenever they dynamically adjust the output (like slowing an animation), a gentle notification can inform the user why ("Animation speed reduced for comfort – see settings to adjust"). This way, the implementation not only prevents harm but is transparent about it.

- **Performance and Compatibility:** Since this Lab is part of DonutOS, we assume a modern web runtime or native environment where WebGL or similar GPU acceleration is available. We will provide fallbacks for non-WebGL environments (e.g., a simplified canvas 2D rendering for basic double-slit interference if WebGL is unavailable, though with reduced features). The design will keep computations minimal (summing a handful of waves is trivial for a GPU; even CPU can handle it at moderate resolution). Only if we attempted full wavefield simulations (ripple tanks with many sources, etc.) do we need to watch performance, but those can be optimized by lowering resolution or using spectral methods. Memory footprint is small, mostly just framebuffers for rendering the patterns. We will include options to reduce detail (like a "low resolution mode") for weaker devices. Testing on various hardware (including integrated graphics laptops and tablets) will ensure the Lab runs smoothly at an interactive frame rate (target 60 FPS for smooth knob dragging feedback). Given the simple nature of the graphics (mostly some colored fringes and simple shapes for apparatus), we don't expect GPU overload. We will however be careful with any 3D view or fancy graphical effects – those will be optional eye-candy (like a wavy surface view) and will auto-disable if performance suffers.

In summary, the implementation focuses on harnessing GPU for the heavy lifting of interference calculations, using stable numerical methods (direct analytical formulas for patterns where possible), and wrapping it in a responsive UI. By referencing existing proven simulations (PhET, RippleGL, etc.), we minimize risk and concentrate on integration and user experience. Each feature – from the phase knob to safety throttling – will be built with simplicity and robustness in mind, to deliver a seamless "virtual optics lab" experience on DonutOS.

Research Appendix

- **Historical Experiments & Analogies:** The Lab is grounded in real optical physics experiments. Thomas Young's 1801 double-slit experiment demonstrated that light exhibits interference, producing a sequence of bright and dark fringes on a screen [3] – a phenomenon our Lab reproduces virtually. The **Michelson interferometer**, invented in the late 19th century, produces circular interference fringes by splitting a beam and recombining it after different path lengths; our phase knob essentially emulates moving Michelson's mirror to shift those fringes [6]. These classical setups are special cases within the Lab. Moving to quantum analogies, the **Aharonov–Bohm effect** in 1959 showed that electrons passing on either side of a shielded magnetic field exhibit shifts in their interference pattern due to the enclosed flux, even though they travel through field-free regions [7]. This purely quantum phenomenon (a manifestation of gauge field holonomy) has an optical parallel in our holonomy mode where a loop phase mimics that effect. Likewise, the concept of **geometric phase** (Pancharatnam/Berry phase) from optics and quantum theory is introduced via the holonomy control – e.g., rotating polarization or path can yield a phase that is observable as an interference shift. All these tie the Lab's features to real scientific principles, some of which have won Nobel prizes or are fundamental in modern physics. By experimenting in the Lab, users effectively reenact parts of these famous experiments (e.g., zero-path-difference white-light fringe in Michelson, or the need for coherent sources in Young's experiment [5]), deepening their appreciation of the science history and theory.

- **Existing Simulations and Tools:** We drew inspiration from existing educational simulations. The PhET **Wave Interference** sim by University of Colorado is a well-known tool where users can "make waves with a dripping faucet, audio speaker, or laser" and "add a second source to create an interference pattern" [4], as well as explore single-slit vs double-slit setups. Our Lab extends similar concepts into a more open-ended OS-integrated environment, adding features like multi-beam interference and phase holonomy that go beyond PhET's scope. Another reference is Paul Falstad's **Ripple Tank** simulator [21], originally a Java applet and now available in WebGL. It simulates 2D wave phenomena including interference and diffraction, and its codebase (RippleGL on GitHub) demonstrates efficient techniques for simulating wave superposition in a browser. We have consulted such implementations to ensure our Lab's engine is both accurate and performant. Additionally, resources like Walter Fendt's simulations and various university applets have shown how UI can be kept intuitive – for instance, many simulations use draggable slits and real-time graphs, which we incorporated into our design. On the research front, the idea of using multi-beam interference to create quasicrystal patterns is supported by optics literature: e.g., holographically constructing a Penrose photonic quasicrystal with five-beam interference has been demonstrated experimentally [8]. This assures us that our simulated pattern for 5 beams is not fanciful – it's what real lasers would produce on a photosensitive plate. Safety guidelines are informed by **W3C and Epilepsy Foundation** recommendations on flashing content [9]; we looked at WCAG 2.3 success criteria and ensured to exceed the minimum requirements by design. In summary, the Lab stands on the shoulders of prior educational tools and physics experiments: it is an integration and expansion that benefits from prior art. All the components – from physical equations to UI widgets – are chosen based on proven success either in real labs or virtual ones. By combining these with the unique twist of QFT analogies, the DonutOS Optics & QFT Lab becomes a comprehensive, research-backed learning playground for waves and interference.

Sources: The development and design of this Lab referenced multiple sources for accuracy and best practices, including physics textbooks and open educational resources for interference theory [3] [12], research articles on multi-beam interference for quasicrystals [8], accessibility standards for visual media [9], and existing simulation descriptions and codebases [4] [21]. These sources ensured that the Lab's functionality is scientifically correct and pedagogically effective.

---

[1] [2] [3] [5] [18] Interference
http://physics.bu.edu/py106/notes/Interference.html

[4] [19] Wave Interference - Interference | Double Slit | Diffraction - PhET Interactive Simulations
https://excelschools.net/en/simulation/wave-interference.html

[6] web.physics.ucsb.edu
https://web.physics.ucsb.edu/~phys128/experiments/interferometry/michelson_interferometer.pdf

[7] Geometric and holonomic quantum computation - ScienceDirect
https://www.sciencedirect.com/science/article/abs/pii/S0370157323002065

[8] [16] [17] [20] Strongly linearly polarized low threshold lasing of all organic photonic quasicrystals | Scientific Reports
https://www.nature.com/articles/srep00627?error=cookies_not_supported&code=6194eceb-f1ad-4201-88d4-7750f95d6b95

[9] [10] [11] Web accessibility for seizures and physical reactions - Accessibility | MDN
https://developer.mozilla.org/en-US/docs/Web/Accessibility/Guides/Seizure_disorders

[12] [14] [15] 5.10: Fringe Visibility - Physics LibreTexts
https://phys.libretexts.org/Bookshelves/Optics/BSc_Optics_(Konijnenberg_Adam_and_Urbach)/05%3A_Interference_and_coherence/5.10%3A_Fringe_Visibility

[13] Interferometric visibility
https://grokipedia.com/page/Interferometric_visibility

[21] GitHub - pfalstad/ripplegl: Ripple Tank, webgl version
https://github.com/pfalstad/ripplegl