ChatGPT

Fractal Bullseye Interface for the Donut of Attention

Design Principles and Requirements

- **Fractal & Holographic Structure:** The interface should mirror the Donut of Attention's concept of attention as a *"toroidal, fractal–holographic field across scales"* [1]. In practice, this means each part of the bullseye menu can reflect the structure of the whole in a self-similar way. Much like how *"each part of the Flower [of Life] contains the pattern to generate the whole"* [2], the bullseye should nest controls and data in repeating patterns across concentric layers. This fractal organization ensures **scale-invariance** – zooming into one cluster reveals a layout similar to the higher-level view, preserving a sense of harmony and continuity. It resonates with how nature and even the brain organize information (scale-free networks) via *"the same shapes and patterns at all levels… like a series of Russian dolls"* [3]. In short, the bullseye's nested rings or segments should be self-similar, creating a holographic feel where any piece can hint at the structure of the whole field of data.

- **Intuitive & Resonant Mapping:** The layout must feel spatially natural and *"resonant"*, meaning users intuitively sense relationships between controls by their placement. Designs based on sacred geometry and human cognitive patterns can achieve this. For example, the **Golden Ratio** is a classic fractal proportion that yields aesthetic coherence – it's the *"only division in which the parts are still in resonance to the whole"* [4]. Using proportions and symmetries that echo natural patterns (circles, spirals, etc.) will make the interface instantly more comfortable and beautiful to the user. In fact, psychologists note that humans perceive golden-ratio fractal patterns as especially "harmonious" or beautiful because *"our biology literally loves"* these universally embedded patterns [5]. By aligning the bullseye's geometry with such ratios or symmetry (e.g. dividing rings or segments in golden-angle or Fibonacci counts), the UI will likely *"optimize the flow"* of attention and feel *coherent rather than arbitrary* [5]. The spatial grouping should also follow *"natural mappings"* – e.g. color-related controls could form a rainbow ring, frequency-related metrics could arrange from low to high in a circle – so that the position itself carries meaning.

- **Extensible, Layered Hierarchy:** The bullseye menu must accommodate **more membranes and clusters** over time without breaking the design. A fractal radial layout inherently supports this, since you can add additional concentric rings or additional spokes as needed while preserving overall structure. This approach is similar to a "radial tree" layout, where you have *"a central CORE node — the heart of your interface,"* and *"surrounding levels — each forming a circle of clickable nodes"* [6]. New functions or data types can simply become another ring outwards or another segment, maintaining balance. This design provides a clear *visual hierarchy* (center to outer layers) and *semantic structure* (grouping by rings or sectors), which users can naturally explore [7]. Crucially, it's *"expandable by layers"* [8] – you can always drill in or add more concentric circles. For example, today's five data clusters might become seven tomorrow; a well-structured bullseye could accommodate that by adding two sectors or an extra outer ring without redesigning the whole UI. This flexibility future-proofs the interface.

- **Multimodal Interaction Support:** The organizational scheme should work seamlessly with **gaze, pointer, rotation, and symbolic input** modalities. Radial layouts are known to be very efficient for quick selection and even muscle memory: *"Radial menus are faster to access than list-based menus"* because every option is equidistant from the center [9]. This means a user can flick a cursor or gesture in a direction to trigger a menu item rapidly (leveraging Fitts' Law) and, with practice, perform it by habit (as seen in "marking menus" where users swipe at, say, 2 o'clock or 6 o'clock without looking [10]). The bullseye's self-similar rings allow such gestures at multiple scales – e.g. a quick outward flick might drill into a sub-ring. For gaze/BCI input, a radial layout is advantageous because it avoids edge-of-screen issues: the user can simply dwell their gaze at a certain angular sector or ring, and since items are spatially distributed in a circle, it reduces clutter and ambiguity. The design can incorporate gaze dwell triggers at each segment (with clear highlighting when focused). **Rotation and tilt** of the torus come into play as additional degrees of input: by rotating the physical (or virtual) donut, the user might scroll through ring items or reveal different layers. The bullseye should treat rotation as a context shift – for example, turning the torus might spin the bullseye menu like a safe dial, or switch which ring is active. Symbolic tagging could be integrated by allowing users to "mark" certain segments with symbols (like pinning an icon or dropping a token on a bullseye sector) for quick reference. The design should allow these multimodal interactions by being spacious (to accommodate gaze targets and finger-sized hit areas) and by possibly encoding symbolic cues (icons, glyphs) on the radial segments that correspond to voice commands or mental commands in a BCI context. In summary, the bullseye's geometry should be usable with eyes, hands, or even head movement, making it a true **radial hub** for all input modes.

- **3D Aesthetic Integration:** Finally, the bullseye menu must feel like an organic part of the **toroidal 3D environment**. It's effectively the *"radial gateway"* at the core of the torus, so its visuals and behavior should mesh with the donut's look and physics. This can be achieved by using the same design language found in the Donut's existing UI: for example, glowing rings, translucent overlays, and subtle animations. The use of sacred geometry motifs can double as both functional layout and visual theme – for instance, a **Flower of Life** pattern could be lightly inscribed in the bullseye, providing both an aesthetic texture and a guide for placing interactive elements. The torus is already described as having "levogyre shells" and creative overlays (Nimbus, glimmer, etc.), so the bullseye might adopt complementary effects: a slight nimbus glow on the active ring, or a *"bindu"* (central point) indicator when phase-lock or focus is achieved [11]. Aligning with known harmonious proportions (as mentioned, Golden Ratio divisions, or using the **Flower of Life's** circle arrangement) will not only look beautiful but also *implicitly echo the torus's geometry*. (Notably, the torus itself is a form often associated with sacred geometry and phase-coherent fields [12] [13], and the Flower of Life pattern contains a "Fruit of Life" subset of 13 circles – 12 around 1 – which is essentially a top-down view of a torus-like symmetry [14] [15].) By designing the bullseye as a kind of **mandala** at the donut's core, with fractal details and smooth transitions, it will feel like an integral "eye" of the donut. Colors can be chosen to match the project's palette (perhaps each data type has a signature color that also appears on the torus overlays), and when the donut tilts or spins, the bullseye might gently rotate or tilt in perspective as well, reinforcing that it's anchored in the 3D space. In essence, the bullseye should appear as a natural *"field of resonance"* itself – a mini-torus or concentric field at the center of the larger torus – so that interacting with it feels like tuning the core of the system, not opening an unrelated flat menu.

Geometric Layout Options (Radial Patterns)

To fulfill those principles, we propose several **geometric layouts** for structuring the bullseye interface. Each pattern provides a fractal, symmetric way to arrange heterogeneous data in nested clusters, but with different aesthetic flavors:

1. **Flower of Life Mandala:** This layout draws directly on the *Flower of Life* sacred geometry pattern – essentially, a central circle surrounded by others in hexagonal symmetry. The basic "Seed of Life" has one center with 6 circles around it, and the full Flower can expand to multiple rings (with 12 around 1 in the next layer, and so on) [14] [15] . Using this as the bullseye design, the **central circle** could be an overview or "home" node, and around it could sit 6 or 12 circular icons corresponding to the main data clusters (color, geometry, EEG, etc.). The circles might slightly overlap or interlock, mimicking the Flower of Life look, which inherently creates a *nested Venn-diagram* effect (shared boundaries could indicate interactions between clusters). The benefit of this approach is deep **fractal coherence** and holography: the Flower of Life is literally known as a holographic matrix where *"each part… contains the pattern to generate the whole"* [2] . If the bullseye is structured this way, clicking any one cluster-circle could zoom into a *mini flower* of sub-options inside it, repeating the geometry at a smaller scale (like a hologram). Aesthetically, this pattern is highly harmonious and recognizable across cultures (it appears in many mandalas and knowledge systems). It naturally produces a **hexagonal/hexagram lattice** arrangement [16] , meaning the clusters will be evenly spaced in a ring and can tessellate if more are added. This feels "organic" and **resonant** to users, as it's akin to looking at a lotus or a snowflake – patterns we intuitively find meaningful. The Flower of Life also embeds **Platonic solids and golden ratio** relations [17] [18] , so using it could allow clever alignments (e.g. placing certain controls at golden-angle offsets, or grouping metrics in a hexagon corresponding to a Platonic solid face). Overall, the Flower of Life option offers a truly fractal, nested menu: A **central hub** with rings of "petals" that can themselves bloom into smaller petals, theoretically ad infinitum, all while maintaining a unified visual theme of overlapping circles. This option might especially suit the project's emphasis on *"scale-invariant, self-organizing fields"*, since the Flower is basically a field pattern of circles, self-similar at every scale.

2. **Hexagonal Honeycomb Grid:** A closely related but distinct option is to use a **hex-grid tiling** as the underlying structure for the bullseye. Instead of overlapping circles, think of the bullseye as concentric **hexagonal rings** or a target subdivided into hexagonal cells. Hexagons tile perfectly around a center (like a beehive honeycomb), which makes an efficient use of space and a clear adjacency structure. For instance, one could place a hexagon at the center and ring it with 6 hexagons (one on each side) for the first layer of options – exactly the "six-around-one" geometry that is also present in the Flower pattern. Additional membranes could form a second ring of 12 hexagons, then 18, and so on, expanding like a ripple. The advantage of hexagons is that they give a bit more **surface area and legible shape** to each menu item compared to circular petals, and their edges naturally line up, which could simplify drawing connector lines or boundaries between clusters. A hexagonal bullseye might resemble a *radial honeycomb* or a segmented dartboard. This can be very useful for grouping and tessellation: e.g., all hex cells in a certain ring could represent one category (by color coding the entire ring), or a wedge of contiguous hex cells from center outward could represent a category's sub-items. The hex pattern is inherently **modular** and can extend infinitely while preserving symmetry, making it highly extensible. It's also a pattern that appears in data visualization (e.g. hex-bin charts) and even cognitive models (some concept maps use hex grids for relationships). The hex layout is essentially a more angular, grid-like manifestation

of the Flower-of-Life's geometry (the Flower's circles' centers actually form a hexagonal lattice [16]). In terms of aesthetics, a hexagonal bullseye might evoke a *technological* or *geodesic* feel (like a soccer ball's pattern or a molecular diagram) which could match the "quantum metrics" aspect of the project. It can still be made to look beautiful – for example, each hex cell could have an icon or an abstract symbol, and the entire bullseye might glow with a subtle grid. If needed, one can combine this with the Flower approach: use circular graphics within a hex tiling, achieving both sacred geometry vibe and the UI clarity of distinct cells. The **Flower vs. Hex** decision largely comes down to visual style: soft overlapping circles vs. crisp geometric cells – both provide fractal nesting and 6-/12-fold symmetry. Either way, the hex-grid approach ensures that adding more elements results in a **predictable expansion** (another ring of hexagons) without breaking the pattern, keeping the interface scalable and **self-organizing**.

3. **Concentric Rings with Polar Segments:** Another approach is to structure the bullseye as classic **concentric rings**, each divided into **angular sectors** (pie slices). This is akin to a multi-level pie menu or "bullseye target" with slices – an approach that is straightforward and highly intuitive for users to navigate. Each ring can represent a depth of menu or a category layer, and each angular slice within a ring represents an item or sub-cluster. For example, the innermost ring could be split into (say) 5 segments for the 5 main clusters (color, geometry, quantum, EEG, cognitive). Selecting one of those segments (via click or dwell) could then highlight the next ring out, which is subdivided into options related to that cluster. This approach is effectively a nested **radial menu**, which leverages many known benefits of pie menus: quick selection and strong muscle-memory potential (since each option has a unique direction) [9]. It is also inherently *fractal* if designed consistently – e.g. every ring might be divided into the same number of segments, so the pattern of slices repeats outward (this repetition helps in learning and gives a pleasing self-similarity). Alternatively, the segment counts could grow with each ring (like 5 segments on inner ring, 10 on next, 20 on outer) to accommodate more detailed options while maintaining proportional geometry. The concentric segmented rings are highly **extensible**: one can always add another outer ring for a new cluster of controls, or subdivide a ring further if a category grows, and it remains visually coherent as a bullseye. One challenge with pure polar slicing is that too many slices in a ring can become small and hard to target – research on radial menus suggests *"eight items seems to be the reasonable maximum"* for a single ring before performance degrades [19]. To address this, we can combine rings and segments hierarchically (so no single ring has an excessive number of slices). The design could also use **color-coding or iconography per segment** to aid recognition (since reading text along a curve is harder). For instance, the geometry tools slice might always appear at 12 o'clock with a little cube icon, the EEG slice at 2 o'clock with a brainwave icon, etc., making it easy to snap the pointer or gaze to the right "pie slice." Concentric rings have the benefit of *clarity*: it's immediately obvious which layer you are in (by radius) and which category (by angle). This layout can be made fractal by ensuring that *sub-menus also appear as radial arrangements*: e.g. choosing one segment could spawn an inner mini-ring of its own sub-options (perhaps using a **spiral** transition or an expanding circle animation to reinforce depth). In essence, the polar ring layout offers an **"inside-out" exploration** of the interface – as one paper put it, *"the future of interfaces isn't linear. It's radial… Instead of flat lists, you build living structures of meaning."* [20] The user can navigate the field of controls by orbiting through rings rather than diving through flat menus, which fits the torus metaphor well. This pattern also maps neatly onto the torus: imagine each ring of the bullseye corresponding to a toroidal "latitude," and each segment corresponding to a "longitude" or angle around it – it's conceptually like unwrapping a torus surface into concentric circles. Therefore, polar rings are a safe, intuitive choice

that can still satisfy the fractal, nested requirement if implemented with repeating segmentation across levels.

4. **Nested "Flower of Life" vs.** Polar Hybrid: **We can also envision** hybrid patterns **that combine the above. For example, a *Flower of Life* arrangement for the high-level clusters (giving an initial 7-circle pattern: one center + six around) could then transition into a more** pie-slice detailed view **when one cluster is focused. In that case, the bullseye might normally show a mandala-like cluster, but upon selection it "unfolds" that circle into a concentric ring of options (perhaps the circle morphs into a ring). This would capitalize on the *intuitive symbolism* of sacred geometry for overview, and the *practical clarity* of concentric menus for drill-down. Another hybrid idea: use** spiraling concentric circles **– a pattern of circles that spiral inward (like a logarithmic spiral) to indicate a continuous zoom. This is inspired by the concept of a** fractal spiral UI **where each nested level rotates and shrinks inward** [21] **. It's a more experimental design, but it could allow a user to "fall into" deeper data by a spiral gesture, literally visualizing depth as going down a vortex. Such a spiral could be laid out radially so that each 360° turn corresponds to one level deeper (much like turning a combination lock multiple revolutions). The spiral idea aligns with the toroidal vortex notion (a torus can be seen as a spiral rotated 360°). However, this might be harder to execute cleanly and could confuse users if overused, so it may remain an optional aesthetic flourish (e.g., using a subtle spiral animation when transitioning states). In summary, a hybrid approach can leverage the** Flower/hex **pattern for global coherence and the** concentric/pie *pattern for local clarity. Both patterns are compatible since the Flower-of-Life's second ring is essentially 12 nodes around – which could correspond to 12 pie slices of a ring, etc. The key is to maintain self-similarity*: even if the shapes change (circle vs slice), the positions and counts should feel like they correspond across scales.

Mapping Strategies for Diverse Data Types

Regardless of the geometric pattern chosen, we need a clear **mapping of the heterogeneous data** into the bullseye's sectors, rings, and angles. Here are a few strategies to organize the content intuitively:

- **By Concentric Layer (Thematic Rings):** In this strategy, each **ring of the bullseye corresponds to one broad category** of data. For example, the **innermost ring** (or center) might contain the core **Geometry & Visualization tools** (since those define the fundamental shape and look of the torus). The next ring out could host **Color Controls & Overlays** (things like color filters, brightness, "glimmer" or "nimbus" toggles) so that all visual styling adjustments live in that layer. Beyond that, a further ring could represent **Analytic Metrics & Fields** (quantum metrics, complexity measures, mean fields, etc.), essentially a ring of gauges or toggles for the scientific overlays. One more outward ring might be dedicated to **EEG/BCI and Neurofeedback inputs** – for instance, indicators for delta/theta/alpha/beta/gamma brainwave levels, or controls for BCI mode, could be arranged here. Finally, the **outermost ring** (or a floating set of nodes on the periphery) could correspond to **Cognitive & Intention Mapping panels** – possibly shortcuts to open intention presets, cognitive state maps, or an "intention field" membrane. This layering from inside-out can follow a logical order: the most "core" settings (geometry) at center and more interpretative or higher-level data (cognitive maps) toward the outside. It also resonates with metaphors like chakras or concentric mandalas – e.g., inner rings for physical parameters, outer rings for mental/spiritual state (similar to how some mandalas put earth at center and heavens at the edge). A concentric mapping has the benefit that turning off an entire layer of functionality is as simple as collapsing or deactivating a

ring. It is also easy to extend (add a new ring for a new category). The user learns "if I want to adjust visuals, look at ring 2; if I want brain data, it's ring 4," etc. One could even allow the rings to be toggleable UIs: for example, if the user isn't using BCI, they could collapse the EEG ring to reduce clutter. The downside is that each ring might contain a mix of sub-options that need differentiation (we can solve that with inner labeling or color coding per ring). Overall, this **thematic layering** emphasizes the *scale aspect*: each ring is a membrane of a different "scale" of attention data (from raw geometry up to cognitive intent), fitting the idea of attention as scale-spanning.

- **By Radial Sector (Clustered Slices):** Another mapping scheme is to give each major data type its **own sector (wedge) of the bullseye**, which spans all or multiple rings. In this model, imagine dividing the circle into, say, 5 primary wedges (perhaps 72° each if equal). Each wedge is devoted to one content cluster and can be thought of as a "pie-slice menu" for that domain. For instance, one wedge could be **Color & Overlay** controls: within that slice, concentric positions might adjust different aspects (the inner portion of the wedge might let you pick base color themes or toggle color filters, the middle might adjust intensity or transparency, and the outer part might turn on artistic overlays like Nimbus or creative filters). The next wedge might be **Geometry & View**: its inner cells could switch torus parameters (inner radius, tube size, symmetry flips), its middle could handle rotation modes or camera locks, and its outer might toggle grid overlays or Platonic solids [22]. Another wedge for **Quantum/Analytics** could have sub-rings for different metrics (one ring of that wedge for mean-field vs. chaos metrics, another ring for RL or predictive coding toggles, etc. [23]). The **EEG/BCI** wedge could dedicate inner rings to device connection/status and outer rings to brainwave monitors or neurofeedback controls (for example, a small ring graph for each band power, arranged radially). Finally, the **Cognitive/Intention** wedge might start with inner controls for setting intentions or presets, and outer displays showing current mental state mapping or suggestions. Using radial sectors in this way clusters all related controls in one angular zone, which can be very intuitive: a user can mentally associate "the top-right quadrant of the bullseye is all my brain-tech stuff" and "the left side is all visuals," etc. This reduces context switching because one doesn't have to hunt around the full circle for related items – they're bundled. It also allows **simultaneous display** of different data types in different wedges. For example, while the color wedge might show a static palette, the EEG wedge could concurrently show live pulsing indicators (each wedge can have its own dynamic behavior). One design consideration is how to delineate the wedges clearly: visual separators (radial lines) or color-coded backgrounds can help. Also, if one category has *way* more sub-options than another, its wedge could theoretically span a larger angle than the others (a flexible sector size). The bullseye could start with equal sectors but allow a wedge to "grow" when activated (much like a pie menu that magnifies the chosen slice). This strategy aligns well with a **mental model of facets**: since attention is considered a field, each wedge could be seen as a *facet of attention* (like color facet, geometry facet, etc.) arranged around the center. It's also similar to the Taoist *Bagua* metaphor – the Bagua has eight sectors around a center, each representing a life aspect, providing an easy map of "areas" [24]. Here we'd have five (or more) sectors representing facets of the attention field. In summary, radial sectors emphasize *simultaneous clusters* and can make it very intuitive to find "all things related to X" by looking in one direction.

- **Hybrid / Matrix Mapping:** We could also combine the above two methods to create a **grid of rings vs. sectors**, essentially forming a polar matrix. In this hybrid, each **sector** is a major category and each **ring** is a sub-category or functional layer. For example, sectors might be as in the previous scheme (Color, Geometry, Analytics, EEG, Cognitive), and rings might represent a mode or type of interaction (perhaps inner ring = toggles on/off, next ring = continuous sliders or intensity, outer ring

= data displays). This way, to find a particular control, one would identify its category by sector and its nature by ring position. The intersection of that sector and ring is the control. For instance, "Geometry + slider ring" could correspond to the innerRadius and tubeRadius adjustments (continuous values), whereas "Geometry + toggle ring" might correspond to switching on symmetry or golden-ratio mode. Likewise, "EEG + display ring" could show live values (like an outer ring EEG histogram in that sector), and "EEG + toggle ring" could allow enabling/disabling BCI input or switching training modes. The benefit of this matrix approach is clarity in dense information: it's like a polar coordinate system for the UI. It also fully realizes the **fractal grid** idea: you have a repeating structure of cells (imagine a bullseye divided both radially and concentricly into a grid of patches). Each cell can be uniquely identified by (sector, ring) and could even be tagged with a symbol for cross-reference. This is perhaps the most extensible and regular approach, but care must be taken to not overwhelm the user visually. We would rely on subtle grouping cues – perhaps coloring the rings or using ring-specific background shading, plus distinct icons in each sector. If done right, it provides both the **focus of sectors** and the **depth of rings**. It's akin to having multiple rings of "chip menus" anchored around the donut. The existing design already has *"radial chip menus"* and *"donut anchors"* in use (per the prompt), so this scheme could build on that by formalizing a full ring for each chip menu. For example, if currently there's a radial menu for scene presets, that can become one sector in a unified bullseye; another existing donut anchor for phase locking can become another sector, etc., uniting all these radial menus into one nested bullseye system. In effect, this hybrid mapping is like a **Rosette** or a *lotus with multiple rings of petals*, where each ring of petals corresponds to a type of action and each petal around corresponds to a domain.

In choosing a mapping strategy, the key is to maintain **natural associations**. Elements that are frequently used together or conceptually linked should either be in the same sector (angular cluster) or same ring (radial layer). For instance, if adjusting a color filter and seeing its effect on the torus is a tight loop, those controls and displays should be adjacent (maybe one on an inner ring, one on outer ring of the same wedge). Similarly, if certain BCI inputs directly drive geometry changes (just hypothetically), linking those sectors visually (maybe even overlapping their boundaries or using a shared color accent) will help the user see the connection. The bullseye's advantage is that proximity can imply relationship (much more so than a linear menu). So we will leverage that: arrange the data types in a circle such that **adjacent sectors have some thematic resonance** if possible. For example, Cognitive/Intention mapping could be placed next to EEG, since those are related (mind state and brainwaves) – their boundary might blur, or perhaps a graphic arc straddles them, indicating mind<->brain link. Geometry might be next to Color/Overlays, since those both deal with visual form. Quantum metrics might sit between Analytics and Cognitive, bridging objective and subjective data. In other words, the ordering of sectors around the bullseye should itself tell a story (much like the sequence of trigrams around a Bagua or the progression of chakras along the spine). We can draw from metaphors: a **cognitive continuum** from physical (geometry) to physiological (EEG) to mental (intention) could wrap around logically. This way, the bullseye not only organizes data but educates the user about how these facets of attention relate in a circle.

Dynamic Behavior and Toroidal Interaction

One of the most exciting aspects of this bullseye design is how it can **dynamically respond** to the torus's movement and the user's perspective. The interface shouldn't be static; it can leverage the 3D nature of the Donut:

- **Perspective Shifts (Top vs Side Views):** Depending on how the user orients the torus (and their camera view), the bullseye can reveal different information. For example, in a **top-down view** (looking directly at the donut's core), the bullseye might emphasize a 2D overview – perhaps showing high-level attention maps or the overall configuration of active panels. This could manifest as a color-coded heatmap on the bullseye indicating which sectors are most "active" (like attention hotspots). However, when the user **tilts the donut to a side view**, the bullseye element could transform. One idea is that the bullseye flips into a **vertical mode** that acts like a depth gauge or timeline. The prompt suggests, for instance, *"top view shows attention maps, side tilt reveals quantum metrics"*. Concretely, if the donut is rotated 90° (so you're seeing the torus edge-on), the UI might automatically switch the bullseye from "menu mode" into a sort of **dashboard mode** focusing on metrics that benefit from a side perspective. Quantum or physics-inspired metrics might be better represented as vertical bars or radial oscillations – these could be shown when the bullseye is viewed edge-on, almost like a speedometer or oscilloscope look. Technically, the bullseye could be a 3D object (a disc) that rotates with the donut, so when it's face-on you see the menu, and when it's edge-on you see a thin line or a different UI. We could use that transition to display alternate data: e.g., the **thickness** or **glow** of the bullseye's edge could encode a value (like overall coherence or phase synchrony). Another approach is to have a mode such that by tilting the donut past a certain angle, the bullseye fades out the interactive controls and fades in a **global visualization** (like an attention hologram or quantum field lines emanating). This would be a dynamic, context-sensitive UI: the *orientation becomes an input* that chooses what the bullseye shows. It aligns with the idea of a *"self-organizing field"* – the view organizes the info shown.

- **Donut Rotation and Active Sector:** The torus also rotates around its axis (yaw rotation). We could tie this to the bullseye's sectors. One idea is to have the **currently top-most sector** of the bullseye be the active one, linked to the torus orientation. For instance, if the user rotates the donut such that the "EEG" sector of the bullseye comes to the 12 o'clock position, the system could automatically open or highlight the EEG panel/membrane in the HUD. In essence, rotating the donut could cycle through focus on different data clusters – the bullseye acting like a combination lock where each position resonates with a particular "membrane". This could be reinforced visually: as the user rotates to a sector, that sector's icon might enlarge or that wedge might glow, indicating it's aligned. This is a very spatially coherent way to navigate: spin the donut to choose a toolset. Because the bullseye is at the core, this feels like tuning the core of a safe or a puzzle box. We must ensure the bullseye markings are fixed to the torus rotation (not screen-fixed), so that they spin with it; indeed the **Sagittarius mode** mentioned in dev notes likely already couples rotation with bullseye appearance [25]. We will ensure *bullseye is only visible/enabled in the appropriate mode* (to avoid interference with free rotation when not desired) [25], but when in use, rotation and bullseye selection can go hand-in-hand. This dynamic is especially useful in VR/AR scenarios: the user could physically rotate a hand controller or their head, and see the menu scroll accordingly.

- **Contextual Morphing of Layout:** The bullseye might not always display the full complexity of all layers – it can **morph** based on context. For instance, when the user is in a high-level overview state,

we might show only the major clusters (maybe just icons in a ring). When they focus or select one cluster, the bullseye could smoothly animate to bring that cluster's details to the forefront. Perhaps the selected sector expands (its wedge grows bigger, others shrink or fade), or the selected ring could ripple outward to fill more space, presenting its controls prominently. This way, the user isn't overwhelmed by seeing *everything* at once unless they want to. It's akin to a flower that blossoms one part at a time under inspection, then returns to the whole form. Technically, we can employ animations like rings that rotate into view or segments that slide outward on selection. The multimodal triggers like gaze or dwell can initiate these changes: looking at a particular ring might cause it to highlight or even raise up (z-depth) to indicate focus. Audio or haptic feedback can reinforce which layer or sector is active.

- **Resonant Feedback and Phase Response:** Given the project's interest in resonance and neurofeedback, the bullseye could dynamically pulse or oscillate in response to data. For example, **EEG rhythms** can be mapped to a visual pulse on the corresponding ring or segment (this was suggested as *"nested pulses for EEG field zones"* in the prompt). Concretely, if there is a ring or sector for brainwaves, each band (delta, theta, etc.) might be a small arc that expands/contracts or glows in sync with the amplitude of that frequency. This would create a living interface element that literally *beats* with the user's brain – a powerful biofeedback cue. Similarly, if the user's attention level (perhaps measured by some internal metric) fluctuates, the bullseye center might brighten or dim like a heartbeat. The term "fractal-holographic field of resonance" implies that the UI could mirror the user's internal state in a fractal way; thus the dynamic behavior should include **visualizing temporal rhythms**. A user in a flow state might see the bullseye adopt a stable, symmetric pattern, whereas in chaotic state it might jitter or scatter (these are speculative ideas, but could be very engaging). Of course, all dynamic feedback would be **non-intrusive and optional** – we'd keep them subtle by default (coherence over chaos in UI).

- **Symbolic Tagging & Filters:** If the user employs **symbolic tagging** (e.g. marking certain data streams with symbols, or using symbolic gestures), the bullseye can react by highlighting relevant segments. For instance, suppose the user marks an "intention" with a certain sigil icon – that sigil could appear in the cognitive mapping sector of the bullseye, perhaps at the periphery, to remind them it's active. Or if the user draws a shape (say a triangle in mid-air) as a gesture to open geometry controls, the bullseye might respond by flashing the Geometry sector (maybe a triangle icon on that sector pulses). Symbolic tagging could also refer to filtering: maybe each sector has a unique symbol, and the user can tag a piece of content with that symbol to route it to that sector's controls. The bullseye would then route or display accordingly (imagine dragging and dropping an icon into the bullseye's sector to assign it). These interactions are somewhat speculative, but the architecture should be **prepared for symbolic interplay**, given the emphasis on cognitive and intention mapping (which often involves symbolic representation).

In all these behaviors, the guiding idea is that the bullseye is not a static HUD element – it's part of a *"self-organizing field"*. It reconfigures with the system's state, user's focus, and orientation, maintaining *spatial coherence* (so changes are spatially logical, not random). For example, a ring might literally rotate or scale to morph into a different function, rather than disappearing and showing a totally unrelated interface. This continuity helps the user trust the system; they can *see* how one state transforms to another.

Finally, since the bullseye is at the core of a 3D torus, we can explore slight 3D effects: perhaps the rings are slightly extruded or stacked in depth. Tilting the donut could show that the bullseye's rings are like **disks at**

different heights – maybe the EEG ring hovers a bit above the geometry ring, etc., like a stack of translucent plates. When viewed from angle, you'd literally see a tiny tower of rings – which might convey hierarchy (higher = more abstract data?). When viewed head-on, they collapse into the flat bullseye. This kind of 3D layering could subtly reinforce the fractal nature (each layer floats in its own "orbit"). If done delicately (with transparency and small offsets), it could enrich the aesthetic without confusing the 2D selection mechanics.

Examples and Visual Ideas

To ground these abstract ideas, here are a few concrete visualization proposals for how different data types and controls could manifest in the bullseye menu. These are meant to illustrate how an **aesthetically integrated, intuitive clustering** might look:

- ***Color & Overlay Controls:*** Imagine an **inner ring** of the bullseye dedicated to color and visual style. This ring could literally be a **color wheel** – a 360° hue ring that the user can rotate to shift the color scheme of the torus overlay. Small segments on this ring might toggle specific overlay effects: e.g. one segment with a sun icon for the *Solar Gate* or "Nimbus" cloud overlay (turning it on/off), another with a grid icon for grid/Platonic overlays [22]. Each segment could double as a button (tap to toggle overlay) and as a slider (drag around the ring to adjust intensity or opacity of that overlay). For example, a *"glimmer"* overlay control could be a segment that, when active, pulses with a faint glow on the ring itself. The entire ring might be color-coded (perhaps using the currently selected palette) to remind users that this is the visual settings layer. If using the Flower-of-Life pattern, the color controls might occupy one petal in a flower – maybe represented by a rainbow-colored petal. Users would intuitively recognize this as the "color wheel" section. As they adjust these, the torus's real-time visual feedback (hue shifting, brightness changes) will reinforce the connection. By placing these controls in a ring, we allow quick cyclic adjustments (color is inherently circular in hue space). Also, a color ring is a familiar UI component in creative software, so it provides comfort in an otherwise novel interface.

- ***Geometry & View Tools:*** The core shape controls could be represented by **geometric icons arranged radially**. For example, at the very center of the bullseye (the "bullseye" dot) there might be a draggable point that controls the torus's inner/outer radius (pulling it outward might increase the torus hole size, etc.). Around the center, maybe in the first concentric circle, we can have icons like a torus, a cube (for symmetry lock), a spiral (for toroidal vs polar spin toggles), etc. These could follow a **polar arc arrangement** – say 6 icons spaced evenly in a small circle. Each icon, when gazed at or clicked, could expand a mini-menu on the spot (for fine adjustments). For instance, selecting the torus radius icon might pop up a small radial slider for that value around the icon itself. This is like radial micro-interactions embedded in the bullseye. Using sacred geometry motifs: one could surround the center with a **Flower-of-Life "seed" of six small circles**, each representing a geometric transformation (scale, rotation, golden ratio toggle, etc.). The Flower's symmetry suggests a natural placement: e.g., opposite points could be pairs like increase/decrease or on/off for complementary settings. Aesthetically, these geometry controls can be drawn with thin line art that matches the wireframe of the torus (since geometry is structural). They should feel like an extension of the torus model – perhaps even anchored to it. For example, if the torus has an axis indicator, the bullseye's geometry icons might align to that axis. Additionally, a **levogyre (gyroscope) indicator** might ring the center – showing orientation lock or free-spin mode (this could be a small ring that looks like a gyroscope, turning when the user moves the mouse if *mouse orbit* is toggled [26]). All of this sits in

the bullseye without overwhelming because geometry tools are fundamental and likely fewer in number. By keeping them central, we reinforce that these are the primary "matter-shaping" controls of the donut.

- ***Quantum Metrics & Fields:*** Representing complex metrics (like quantum overlays, predictive models, manifold confidence, etc. [23]) calls for a design that can display numerical or graphical info in a compact form. A good approach is to use the **outer circumference** of the bullseye as a kind of **polar chart** or **radial infographic**. For instance, we could devote the outermost ring (or an outer border of the bullseye) to a set of arc indicators – each arc corresponding to a metric (e.g., one for "torus confidence level", one for "phase coherence", one for "edge-of-chaos measure"). These could appear as colored arcs that partially fill based on value, essentially radial progress bars encircling the bullseye. The arcs' positions or colors would tie them to their meaning (perhaps a legend is provided in a tooltip or learned by the user). When the donut is face-on, these arcs might just show as subtle rings; when tilted, they could extrude into bars (for easier reading of values). Another visualization: **field lines or wave patterns** drawn on the bullseye disc itself. For example, a low-opacity interference pattern or a mandala-like graphic that morphs with certain data – if the system has a "quantum field" value, it could generate a pattern of concentric ripples or a flower-like interference pattern on the bullseye background (constructive interference might form a Flower-of-Life-like image, interestingly). This would literally use the bullseye as a canvas for data visualization. Symbolic representations might include small **Platonic solid icons** (since those are tied to fields in the design [22]) placed at certain angles, or even animating particles orbiting the bullseye to indicate dynamic values (like a number of particles or their speed corresponding to a metric). Inspiration can be drawn from sci-fi HUDs that show circular graphs and rotating indicators for energy fields. Importantly, these metrics should remain *readable yet unobtrusive* – perhaps normally semi-transparent, but when the user tilts to "metrics view" or focuses on Analytics, they brighten up for inspection. By mapping abstract numbers to spatial, circular visuals, we keep the *resonance* principle: e.g., a **phase lock ratio** could be shown as two overlapping circles on the bullseye edge – if they align perfectly, you have phase lock (this visual directly uses geometry to convey state). The bullseye's ability to shift modes could allow these quantum/analytic displays to sometimes hide or shrink when not needed, and come alive on command. Tying a bit to sacred geometry: one might incorporate an **inscribed Metatron's Cube or Sri Yantra** pattern that highlights certain nodes when particular metric thresholds are crossed, giving a mystical but information-rich feedback (for example, if a complexity metric is high, a star tetrahedron shape on the bullseye might glow, hinting "high complexity"). These subtle cues can make the interface feel *alive with meaning* even before the user fully learns each symbol.

- ***EEG/Neurofeedback Inputs:*** For brainwave and BCI data, a **nested pulse** visualization fits perfectly. We can allocate one ring (perhaps the second outermost ring) exclusively for EEG bands. Picture a ring divided into, say, 5 segments – labeled (via color or tiny text) Δ, Θ, α, β, γ for the brainwave bands. Each segment acts like a tiny radial bar that expands/contracts in proportion to real-time power in that band. Visually, it could look like five radial **pips** around a circle, each pulsing to the rhythm of the user's brain. In fact, the *frequency* could also be shown by a subtle oscillation or glow. The dev code hints at specific colors for these bands (delta blue, theta purple, etc.) [27] – we can adopt those so that, for example, the delta segment is sky-blue and might throb slowly (delta = slow wave), while beta segment is yellow and flickers faster. This creates a direct sensory coupling: the user can glance at the bullseye and *feel* their mental state (calm vs aroused) from the pattern of pulses and colors. Additionally, if there are BCI "commands" or gaze focus indicators, these can

appear on the EEG ring as markers. For instance, if the system detects a user's focus (via EEG or eye-tracking) is above a threshold, a small *"attention dot"* could light up at the top of the EEG ring (maybe this is the *"bindu"* indicator [28]). Another idea: use the **center of the bullseye as a neurofeedback focal point** – e.g., a dot that expands when focus is high and contracts when low, guiding the user in meditation or training. The EEG ring can work in tandem with that, like ripples emanating from the center dot representing brain activity. In terms of interaction, the EEG sector/ring might also contain controls: e.g., a segment to calibrate or zero the EEG, or an icon to switch which user's EEG (if multi-user group sync) is shown. If a **group phase sync** feature exists [29], multiple user's brain rhythms could be layered on this ring (maybe multiple concentric sub-rings, one per user, or multiple dots per segment). However, that could get busy, so perhaps leave group visuals to another panel, but the bullseye could indicate group lock by an overall ring glow or a linking pattern between segments (like lines connecting two peoples' brainwave segments if they sync up). The key is that the EEG info is *visibly alive* in the bullseye, reinforcing the notion of the attention field. By literally embedding brainwave graphics into the center UI, we underline the project's theme of internal state (attention/neuro) and external visualization merging.

- ***Cognitive & Intention Mapping:*** Representing cognitive states or intentions might be more abstract, but we can use **symbolic icons and spatial zones**. One possibility is to include a **"cognitive wheel" or chakra-like display**. In many contemplative traditions, cognitive/energetic states are mapped in a circular schematic (for instance, the *chakra system* maps different psycho-spiritual states to colored lotus symbols arranged vertically – we could adapt that into a circle). We could assign a set of symbols for different mental modes (e.g., focus, creativity, stress, calm, etc.) and arrange them around a ring or as petals around the center. When the system detects or the user selects a particular intention/preset, the corresponding symbol on the bullseye could illuminate. For example, if the user chooses an "Intention preset" for creative brainstorming, an **icon of a lotus or a lightbulb** on the cognitive ring might glow, and maybe the ring itself takes on a color (like violet for imagination, akin to crown chakra color). If the user is in an analytical/problem-solving mode, a different icon (maybe a geometric symbol) at another angle lights up. This effectively makes the bullseye a **compass for the mind**, showing where on the map of mental states one is. We can take inspiration from the Taoist Bagua or Jungian/archetypal circles where each direction corresponds to an aspect (the interface could similarly define, say, North = focused attention, East = open awareness, South = relaxation, West = stimulation, etc., purely conceptually). The intention mapping panel might also allow input: the user could "nudge" their state by interacting with the bullseye. For instance, dragging a marker on the cognitive ring from one symbol toward another could mean blending those states or setting a target state (almost like moving a token on a board that represents where you want your attention to go). This is speculative but intriguing – it would turn the bullseye into a sort of **intention compass**. Visually, to integrate this, we could use the very center of the bullseye (or an inner ring) as a **symbolic mandala**: perhaps a faint overlay of a mandala that has segments corresponding to mind, body, spirit, etc., and as those aspects activate, that part of the mandala brightens. Since the project documentation references *"intention field"* and related concepts, we can assume there are discrete modes or continuous spectra for intentions. The bullseye is an ideal place to show that because it's front-and-center and metaphorically the "mind's eye" of the donut. If using the Flower-of-Life motif, note that one of its interpretations is indeed a map of consciousness – *"each circle represents ψ (consciousness) observing itself"* in a recursive way [30]. We could play off that: each circle/petal of the bullseye might be one aspect of consciousness, and their overlaps (vesica piscis regions) could be where combined states occur. This is a deeper

theoretical mapping, but if done visually (even as simple as overlapping translucent icons when two intentions are active), it could impart a holographic sense of cognitive state.

- ***Dynamic Shift with Donut Tilt:*** As a concrete example of dynamic view: when the donut is flat (top view), the bullseye might show the full multi-layer menu (all rings and sectors visible as described). But if the user tilts the donut 90°, the bullseye's appearance could simplify into a **single vertical stack** – perhaps all the rings collapse into a line, or the sectors fold behind one another. In this state, rather than interactive controls, the bullseye might show aggregate info: e.g., an **"attention meter"** – a vertical bar or concentric circle indicating overall attention level or coherence. The edges of the bullseye (which are now what we see) could have tick marks or a scale. For instance, a side view could reveal a **stack of rings each emitting a glow** – from top ring (outermost originally) to bottom (innermost). The intensity of each glow could correspond to how active that layer is. If quantum metrics are high, maybe the outer ring (now at top of stack) glows bright, if cognitive focus is low, the inner ring (bottom of stack) is dim, etc. This gives a quick at-a-glance sense of the state of each layer without the detail of individual controls. It's almost like seeing the cross-section of the "attention torus". And upon tilting back to flat, those rings spread out again into the full menu. This behavior would certainly impress the user with a feeling that the interface is truly **3D and alive**, not just a flat overlay. It also cleverly uses the *fractal self-similarity*: the UI re-represents itself in a different projection (flat vs side) but conveys related information.

All these examples aim to **inspire intuitive clustering** by using spatial, color, and symbolic cues that the user can *feel* as much as learn. By referencing geometric knowledge systems (Flower of Life, mandalas, chakras, Bagua, etc.) we tap into patterns humans have used for centuries to organize complex ideas – now applied to organizing complex interface data. The bullseye, as the "gateway" at the donut's core, will thus become a kind of *universal remote* for the user's attention: fractal (repeating structures at every zoom), holographic (each piece reflects the whole), and resonant (every control mapped in a natural, meaningful spot). With these structural models and configuration options, a developer could begin prototyping a bullseye menu that is not only highly functional across many data types, but also **mathematically beautiful** – a UI that itself feels like a little cosmos of attention, living at the heart of the Donut interface.

Sources: The design concepts are informed by a variety of spatial interface metaphors and references. Radial, layered menus (as in the Radial Tree UI) highlight how concentric circles around a core aid natural exploration [6] [8]. Sacred geometry patterns like the Flower of Life demonstrate fractal self-similarity and holographic grouping (each part reflecting the whole) [2], which inspire the nested circular layouts. Research into fractal design in interfaces shows that repeating patterns across scales create cohesive and scalable UIs [31] [3]. Moreover, leveraging known symmetrical arrangements (e.g. "12-around-1" as in the Fruit of Life, or the eight sectors of the Taoist Bagua) provides intuitive clustering of concepts [14] [24]. The efficiency of radial menus for multimodal input is supported by HCI studies (faster selection and muscle memory) [9] [10], which underpins the bullseye's suitability for gaze or gesture control. Finally, the importance of using resonant proportions and patterns (like the golden ratio) to enhance user comfort and perception is noted in design literature [18] [5] – guiding the aesthetic integration of the bullseye into the toroidal environment. All these sources collectively shape a proposal where the bullseye interface is a **fractal, holographic menu system** that is intuitive to use, extensible in design, and artfully united with the Donut of Attention's 3D world. [1] [20]

---

1 11 22 23 26 28 29 DONUT_SPEC.md
file://file_00000000767472438d5150a0756c4cd9

2 14 15 16 17 30 Chapter 018: The Flower of Life - Creation Matrix | AllTheory
https://spirit.dw.cash/docs/psi-codex-spiritual-totality/appendix/volume-02-geometry-of-awakening/chapter-018-flower-life-creation-matrix

3 4 5 12 13 18 Sacred geometry and the fractal universe – ArgosVu
https://argos.vu/sacred-geometry-and-the-fractal-universe/

6 7 8 20 Radial Tree UI — A Core-Centric Interface for Modular Thinking - DEV Community
https://dev.to/andysay/radial-tree-ui-a-core-centric-interface-for-modular-thinking-5am2

9 10 19 Touch Means a New Chance for Radial Menus | Big Medium
https://bigmedium.com/ideas/radial-menus-for-touch-ui.html

21 Fractal Spiral UI: A New Way to Represent Depth in Interfaces - DEV Community
https://dev.to/andysay/fractal-spiral-ui-a-new-way-to-represent-depth-in-interfaces-147f

24 Bagua - Wikipedia
https://en.wikipedia.org/wiki/Bagua

25 DEV_NOTES.md
file://file_00000000c2c87243922c2ea3ad76eecf

27 torus.js
file://file_00000000a0b07243a65b5ef6a63943a1

31 Exploring the Fractal Design Framework in Modern Design
https://think.design/blog/fractal-design-framework/