```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>The Octagonal Labyrinth - eTwinning Project</title>
    <style>
        * { margin: 0; padding: 0; box-sizing: border-box; }
        body {
            font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Arial,
sans-serif;
            background: linear-gradient(135deg, #1e1b4b 0%, #581c87 50%, #1e3a8a 100%);
            min-height: 100vh;
            color: #fff;
            padding: 20px;
        }
        .container { max-width: 1200px; margin: 0 auto; }
        .screen { display: none; }
        .screen.active { display: block; animation: fadeIn 0.5s; }
        @keyframes fadeIn { from { opacity: 0; } to { opacity: 1; } }
        @keyframes pulse { 0%, 100% { opacity: 1; } 50% { opacity: 0.5; } }
        @keyframes spin { from { transform: rotate(0deg); } to { transform: rotate(360deg); } }
        .animate-pulse { animation: pulse 2s infinite; }
        .animate-spin-slow { animation: spin 8s linear infinite; }
        h1 { font-size: 2.5rem; text-align: center; color: #fef3c7; margin-bottom: 1rem; }
        h2 { font-size: 1.8rem; color: #fcd34d; margin-bottom: 1rem; }
        .subtitle { text-align: center; font-size: 1.3rem; font-style: italic; color: #e9d5ff;
margin-bottom: 2rem; }
        .card { background: rgba(15, 23, 42, 0.6); border: 3px solid #d97706; border-radius:
1rem; padding: 2rem; margin-bottom: 2rem; backdrop-filter: blur(10px); }
        .grid-3 { display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap:
2rem; margin: 2rem 0; }
        .path-card { background: linear-gradient(135deg, var(--color-from), var(--color-to));
border: 3px solid rgba(255, 255, 255, 0.3); border-radius: 1rem; padding: 2rem; text-align:
center; cursor: pointer; transition: transform 0.3s, box-shadow 0.3s; }
        .path-card:hover { transform: scale(1.05); box-shadow: 0 20px 50px rgba(0, 0, 0, 0.5); }
        .path-card.blue { --color-from: #2563eb; --color-to: #0891b2; }
        .path-card.purple { --color-from: #9333ea; --color-to: #db2777; }
        .path-card.rose { --color-from: #e11d48; --color-to: #f97316; }
        .icon { font-size: 3rem; margin-bottom: 1rem; }
        .btn { background: linear-gradient(135deg, #d97706, #ea580c); color: white; border: 3px
solid #fbbf24; padding: 1rem 2rem; font-size: 1.1rem; font-weight: bold; border-radius:
0.5rem; cursor: pointer; transition: transform 0.2s; display: block; text-align: center; width:
100%; margin-top: 1rem; }
        .btn:hover:not(:disabled) { transform: scale(1.05); }
        .btn:disabled { opacity: 0.5; cursor: not-allowed; }
        .btn-secondary { background: linear-gradient(135deg, #7c3aed, #db2777); border-color:
#a855f7; }
```

```
        .question-card { background: rgba(255, 255, 255, 0.1); border: 3px solid rgba(255, 255, 255, 0.3); border-radius: 1rem; padding: 2rem; }
        .question { font-size: 1.4rem; font-weight: bold; margin-bottom: 2rem; text-align: center; line-height: 1.6; }
        .options { display: flex; flex-direction: column; gap: 1rem; }
        .option { background: rgba(255, 255, 255, 0.2); border: 2px solid rgba(255, 255, 255, 0.4); padding: 1rem; border-radius: 0.75rem; cursor: pointer; transition: all 0.3s; text-align: left; }
        .option:hover { background: rgba(255, 255, 255, 0.3); border-color: white; transform: scale(1.02); }
        .maze-container { background: rgba(15, 23, 42, 0.6); border: 4px solid #d97706; border-radius: 1.5rem; padding: 2rem; height: 600px; position: relative; margin-bottom: 2rem; }
        .maze-svg { width: 100%; height: 100%; }
        .legend { background: rgba(15, 23, 42, 0.6); border: 2px solid rgba(217, 119, 6, 0.5); border-radius: 0.75rem; padding: 1.5rem; margin-bottom: 1rem; }
        .legend-grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(120px, 1fr)); gap: 1rem; text-align: center; }
        .legend-item { display: flex; flex-direction: column; align-items: center; gap: 0.5rem; }
        .legend-dot { width: 24px; height: 24px; border-radius: 50%; }
        .score-display { text-align: center; font-size: 1.3rem; margin-bottom: 1rem; }
        .center-message { background: linear-gradient(135deg, #667eea 0%, #764ba2 100%); border: 4px solid #fbbf24; border-radius: 1rem; padding: 2.5rem; text-align: center; margin: 2rem 0; }
        .center-title { font-size: 2rem; color: #fcd34d; margin-bottom: 1rem; }
        .center-text { font-size: 1.2rem; line-height: 1.8; margin-bottom: 1.5rem; }
        .dots { display: flex; justify-content: center; gap: 0.5rem; margin: 1rem 0; }
        .dot { width: 12px; height: 12px; border-radius: 50%; background: #fbbf24; }
        .info-grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); gap: 2rem; text-align: center; margin: 2rem 0; }
        .info-item h3 { font-size: 1.2rem; margin-bottom: 0.5rem; }
        @media (max-width: 768px) {
            h1 { font-size: 1.8rem; }
            h2 { font-size: 1.4rem; }
            .subtitle { font-size: 1.1rem; }
            .question { font-size: 1.1rem; }
            .maze-container { height: 400px; }
            .icon { font-size: 2.5rem; }
        }
    </style>
</head>
<body>
    <div class="container">
        <div id="startScreen" class="screen active">
            <div style="text-align: center; margin-bottom: 2rem;">
                <div class="icon animate-spin-slow">🧭</div>
                <h1>The Octagonal Labyrinth</h1>
                <p class="subtitle">Three Paths Through Awareness</p>
```

```
      </div>
      <div class="card">
        <h2 style="text-align: center;">How to Play</h2>
        <div class="info-grid">
          <div class="info-item">
            <div class="icon">📍</div>
            <h3>1. Choose Your Path</h3>
            <p>Select one of three ways</p>
          </div>
          <div class="info-item">
            <div class="icon">👣</div>
            <h3>2. Cross the Labyrinth</h3>
            <p>Answer question-obstacles</p>
          </div>
          <div class="info-item">
            <div class="icon">👁️</div>
            <h3>3. Reach the Center</h3>
            <p>Pause in awareness</p>
          </div>
        </div>
      </div>
      <div class="grid-3">
        <div class="path-card blue" onclick="startPath('truth')">
          <div class="icon">👁️</div>
          <h2>Path of Truth</h2>
          <p>4 questions</p>
        </div>
        <div class="path-card purple" onclick="startPath('justice')">
          <div class="icon">⚖️</div>
          <h2>Path of Justice</h2>
          <p>4 questions</p>
        </div>
        <div class="path-card rose" onclick="startPath('love')">
          <div class="icon">❤️</div>
          <h2>Path of Love</h2>
          <p>4 questions</p>
        </div>
      </div>
    </div>
    <div id="playingScreen" class="screen">
      <div style="text-align: center; margin-bottom: 2rem;">
        <div class="icon" id="pathIcon">👁️</div>
        <h1 id="pathName">Path of Truth</h1>
        <div class="score-display">Score: <span id="score">0</span> | Checkpoint:
<span id="checkpoint">1/5</span></div>
      </div>
      <div class="maze-container">
```

```
                <svg class="maze-svg" viewBox="0 0 100 100" preserveAspectRatio="xMidYMid meet" id="mazeSvg"></svg>
            </div>
            <div class="legend">
                <div class="legend-grid">
                    <div class="legend-item"><div class="legend-dot" style="background: #10b981;"></div><div>Entrance</div></div>
                    <div class="legend-item"><div class="legend-dot" style="background: #fbbf24;"></div><div>Center</div></div>
                    <div class="legend-item"><div class="legend-dot" style="background: #ef4444;"></div><div>Exit</div></div>
                    <div class="legend-item"><div class="icon" style="font-size: 1.5rem;">🚶</div><div>You are here</div></div>
                </div>
            </div>
            <button class="btn" onclick="moveNext()" id="moveBtn">Advance to Next Checkpoint →</button>
        </div>
        <div id="questionScreen" class="screen">
            <div style="text-align: center; margin-bottom: 2rem;">
                <div class="icon" id="qPathIcon">👁️</div>
                <h2 id="qPathName">Path of Truth</h2>
                <div class="score-display">Score: <span id="qScore">0</span></div>
            </div>
            <div class="question-card">
                <div class="icon">🚧</div>
                <div class="question" id="questionText"></div>
                <div class="options" id="optionsContainer"></div>
            </div>
        </div>
        <div id="centerScreen" class="screen">
            <div style="text-align: center;">
                <div class="icon animate-pulse" style="font-size: 5rem;">⭐</div>
                <h1 style="margin-top: 2rem;">The Center of Awareness</h1>
                <div class="center-message">
                    <p class="center-text">You have reached the heart of the sacred octagon. Here, the three paths meet. Breathe. Reflect. This is not an arrival, but a transformation. What you have learned so far will illuminate the rest of the journey.</p>
                    <div class="dots">
                        <div class="dot animate-pulse"></div>
                        <div class="dot animate-pulse" style="animation-delay: 0.2s;"></div>
                        <div class="dot animate-pulse" style="animation-delay: 0.4s;"></div>
                    </div>
                    <p style="font-size: 1.2rem; font-style: italic; color: #e9d5ff; margin-top: 1rem;">This is a place of pause, not of arrival.</p>
                </div>
                <button class="btn" onclick="leaveCenter()">Continue the Journey →</button>
            </div>
        </div>
        <div id="completeScreen" class="screen">
            <div style="text-align: center;">
                <div class="icon" style="font-size: 5rem;">⛰️</div>
                <h1>You Completed the Journey!</h1>
                <p class="subtitle" id="completePath">Path of Truth</p>
                <div class="card">
                    <div class="icon">🏆</div>
                    <div style="font-size: 1.8rem; font-weight: bold; color: #fcd34d; margin-bottom: 1rem;">Final Score: <span id="finalScore">0</span></div>
                    <p style="font-size: 1.1rem; line-height: 1.8;">You have crossed the octagon, paused at the center of awareness, and completed your journey. But remember: every end is a new beginning.</p>
                </div>
                <button class="btn btn-secondary" onclick="reset()">Explore Another Path</button>
            </div>
        </div>
    </div>
    <script>
        const paths = {
            truth: { name: "Path of Truth", icon: "👁️", color: "#3b82f6", entrance: {x: 50, y: 5}, exit: {x: 50, y: 95}, checkpoints: [{x: 50, y: 20, id: 0}, {x: 35, y: 35, id: 1}, {x: 50, y: 50, id: 2, isCenter: true}, {x: 65, y: 65, id: 3}, {x: 50, y: 80, id: 4}], questions: [{q: "You find an error in a colleague's work that could cost them their job. What do you do?", options: ["Tell them privately", "Report it to the boss", "Ignore it", "Correct it myself"], correct: 0}, {q: "A small lie could spare a loved one's feelings. Is truth always absolute?", options: ["Yes, always", "It depends on context", "No, protecting loved ones is more important", "Truth is relative"], correct: 1}, {q: "After the center: How would you define honesty?", options: ["Always saying what I think", "Aligning words, thoughts, and actions", "Never lying", "Being true to myself"], correct: 1}, {q: "Are you ready to see yourself as you truly are?", options: ["Yes, without compromise", "Yes, but with self-compassion", "I prefer to improve gradually", "Some veils are necessary"], correct: 1}]},
            justice: { name: "Path of Justice", icon: "⚖️", color: "#a855f7", entrance: {x: 85, y: 30}, exit: {x: 15, y: 70}, checkpoints: [{x: 75, y: 35, id: 0}, {x: 65, y: 42, id: 1}, {x: 50, y: 50, id: 2, isCenter: true}, {x: 35, y: 58, id: 3}, {x: 25, y: 65, id: 4}], questions: [{q: "You see someone desperately stealing food. What do you do?", options: ["Alert security", "Offer to pay for it", "Pretend not to see", "Talk to them to understand"], correct: 3}, {q: "An unjust system oppresses a minority. Is it right to break the law?", options: ["No, change it legally", "Yes, moral justice supersedes legal justice", "It depends on the means", "We must obey"], correct: 2}, {q: "After the center: How would you define justice?", options: ["Giving according to merit", "Guaranteeing dignity for all", "Balance between rights and duties", "Harmony between individual and community"], correct: 3}, {q: "What truth about justice do you carry with you?", options: ["Justice without mercy is cruelty", "Every rule serves human dignity", "True justice transforms", "Justice is seeing humanity in everyone"], correct: 3}]},
            love: { name: "Path of Love", icon: "❤️", color: "#f43f5e", entrance: {x: 15, y: 30}, exit: {x: 85, y: 70}, checkpoints: [{x: 25, y: 35, id: 0}, {x: 35, y: 42, id: 1}, {x: 50, y: 50, id: 2, isCenter: true}, {x: 65, y: 58, id: 3}, {x: 75, y: 65, id: 4}], questions: [{q: "Someone you love
```

```
asks you to change a fundamental aspect of yourself. How do you react?", options: ["I
change for love", "I refuse: I must remain myself", "I dialogue to find a compromise", "I
wonder if it's really love"], correct: 2}, {q: "You must choose between someone you love and
your life path. What prevails?", options: ["The one I love, always", "My path", "I seek a
synthesis", "True love doesn't make you choose"], correct: 3}, {q: "After the center: How do
you understand love now?", options: ["Wanting the other's good", "Being yourself beside the
other", "Balance between giving and receiving", "Seeing the infinite in the other"], correct: 3},
{q: "What wisdom about love do you carry with you?", options: ["Love is the courage to be
vulnerable", "To love is to see the other as they are", "The greatest love leaves others free",
"To love is to choose every day"], correct: 2}]}
        };
        let currentPath = null, currentPosition = 0, score = 0, visitedCenter = false,
currentQuestionIndex = 0;
        function showScreen(screenId) { document.querySelectorAll('.screen').forEach(s =>
s.classList.remove('active')); document.getElementById(screenId).classList.add('active'); }
        function startPath(pathKey) { currentPath = pathKey; currentPosition = 0; score = 0;
visitedCenter = false; currentQuestionIndex = 0; updatePlayingScreen();
showScreen('playingScreen'); }
        function updatePlayingScreen() { const path = paths[currentPath];
document.getElementById('pathIcon').textContent = path.icon;
document.getElementById('pathName').textContent = path.name;
document.getElementById('score').textContent = score;
document.getElementById('checkpoint').textContent = `${currentPosition +
1}/${path.checkpoints.length}`; drawMaze(); const btn =
document.getElementById('moveBtn'); btn.textContent = currentPosition >=
path.checkpoints.length - 1 ? 'Complete the Journey' : 'Advance to Next Checkpoint →'; }
        function drawMaze() { const svg = document.getElementById('mazeSvg');
svg.innerHTML = ''; const octagon =
document.createElementNS('http://www.w3.org/2000/svg', 'polygon');
octagon.setAttribute('points', '30,10 70,10 90,30 90,70 70,90 30,90 10,70 10,30');
octagon.setAttribute('fill', 'none'); octagon.setAttribute('stroke', '#d4af37');
octagon.setAttribute('stroke-width', '0.5'); octagon.setAttribute('opacity', '0.3');
svg.appendChild(octagon); Object.keys(paths).forEach(key => { const p = paths[key]; const
isCurrentPath = key === currentPath; const opacity = isCurrentPath ? 1 : 0.2; const line =
document.createElementNS('http://www.w3.org/2000/svg', 'polyline');
line.setAttribute('points', p.checkpoints.map(cp => `${cp.x},${cp.y}`).join(' '));
line.setAttribute('fill', 'none'); line.setAttribute('stroke', p.color); line.setAttribute('stroke-width',
'0.8'); line.setAttribute('opacity', opacity); if (!isCurrentPath)
line.setAttribute('stroke-dasharray', '2,2'); svg.appendChild(line); const entrance =
document.createElementNS('http://www.w3.org/2000/svg', 'circle'); entrance.setAttribute('cx',
p.entrance.x); entrance.setAttribute('cy', p.entrance.y); entrance.setAttribute('r', '2');
entrance.setAttribute('fill', '#10b981'); entrance.setAttribute('opacity', opacity);
svg.appendChild(entrance); p.checkpoints.forEach((cp, idx) => { if (cp.isCenter) { const
center = document.createElementNS('http://www.w3.org/2000/svg', 'circle');
center.setAttribute('cx', cp.x); center.setAttribute('cy', cp.y); center.setAttribute('r', '4');
center.setAttribute('fill', '#fbbf24'); center.setAttribute('opacity', opacity);
svg.appendChild(center); } else { const isPassed = isCurrentPath && idx < currentPosition;
const isCurrent = isCurrentPath && idx === currentPosition; const point =
document.createElementNS('http://www.w3.org/2000/svg', 'circle'); point.setAttribute('cx',
cp.x); point.setAttribute('cy', cp.y); point.setAttribute('r', isCurrent ? '3' : '2');
point.setAttribute('fill', isPassed ? '#10b981' : isCurrent ? '#fbbf24' : p.color);
point.setAttribute('opacity', opacity); if (isCurrent) point.classList.add('animate-pulse');
svg.appendChild(point); } }); const exit =
document.createElementNS('http://www.w3.org/2000/svg', 'circle'); exit.setAttribute('cx',
p.exit.x); exit.setAttribute('cy', p.exit.y); exit.setAttribute('r', '2'); exit.setAttribute('fill',
'#ef4444'); exit.setAttribute('opacity', opacity); svg.appendChild(exit); }); if (currentPath) {
const path = paths[currentPath]; const pos = path.checkpoints[currentPosition]; const player
= document.createElementNS('http://www.w3.org/2000/svg', 'text'); player.setAttribute('x',
pos.x); player.setAttribute('y', pos.y + 1); player.setAttribute('text-anchor', 'middle');
player.setAttribute('fill', 'white'); player.setAttribute('font-size', '5'); player.textContent = '🚶';
svg.appendChild(player); } }
        function moveNext() { const path = paths[currentPath]; const nextPos = currentPosition
+ 1; if (nextPos < path.checkpoints.length) { const checkpoint = path.checkpoints[nextPos]; if
(checkpoint.isCenter && !visitedCenter) { currentPosition = nextPos;
showScreen('centerScreen'); } else { currentPosition = nextPos; showQuestion(); } } else {
showComplete(); } }
        function showQuestion() { const path = paths[currentPath]; const question =
path.questions[currentQuestionIndex]; document.getElementById('qPathIcon').textContent =
path.icon; document.getElementById('qPathName').textContent = path.name;
document.getElementById('qScore').textContent = score;
document.getElementById('questionText').textContent = question.q; const container =
document.getElementById('optionsContainer'); container.innerHTML = '';
question.options.forEach((opt, idx) => { const div = document.createElement('div');
div.className = 'option'; div.innerHTML = `<strong>${String.fromCharCode(65 +
idx)}.</strong> ${opt}`; div.onclick = () => answerQuestion(idx === question.correct);
container.appendChild(div); }); showScreen('questionScreen'); }
        function answerQuestion(isCorrect) { if (isCorrect) score += 10;
currentQuestionIndex++; setTimeout(() => { if (currentPosition >=
paths[currentPath].checkpoints.length - 1) showComplete(); else { updatePlayingScreen();
showScreen('playingScreen'); } }, 800); }
        function leaveCenter() { visitedCenter = true; updatePlayingScreen();
showScreen('playingScreen'); }
        function showComplete() { const path = paths[currentPath];
document.getElementById('completePath').textContent = path.name;
document.getElementById('finalScore').textContent = score; showScreen('completeScreen');
}
        function reset() { currentPath = null; currentPosition = 0; score = 0; visitedCenter =
false; currentQuestionIndex = 0; showScreen('startScreen'); }
    </script>
</body>
</html>
```