

The Fighting Lambs of Threadland

Phaser Development Plan – 9:16 Portrait, No Physics, Swipe Transitions

1. Core Targets

- Orientation locked to 9:16 portrait.
 - No physics engine; all movement and collision handled manually.
 - Portrait HUD for mobile-first flow.
 - Dynamic world transition swipe system between waves that also powers desktop side panels.
-

2. Technical Setup

- Engine: Phaser 3 (ES6 or TypeScript).
- Storage: IndexedDB for runs, localStorage for settings.
- Deployment: GitHub → Cloudflare Pages (PWA enabled).

File Structure

```
/public
index.html
manifest.webmanifest
sw.js
/assets/
  /sprites
  /maps
  /audio
  /interstitials
/src
main.js
```

scenes/{Boot,Menu,Game,UI,Transition}.js
systems/{path,towers,enemies,combos,save}.js
data/{waves.json,towers.json,combos.json}

3. Display Rules

- Logical canvas size: 540×960 px.
 - Mobile: fills screen.
 - Desktop: centered portrait with optional side art panels (uses transition images).
 - Grid: 10×18 tiles, 54 px each.
 - Camera fixed; enemies move bottom → top.
-

4. Core Systems

A. Path + Movement

- Waypoint-based linear interpolation (no physics).
- Air units follow higher offset path.

```
function lerpPath(points, t) {  
  const i = Math.min(points.length - 2, Math.floor(t * (points.length - 1)));  
  const a = points[i], b = points[i + 1];  
  const lt = t * (points.length - 1) - i;  
  return { x: a.x + (b.x - a.x) * lt, y: a.y + (b.y - a.y) * lt };  
}
```

B. Collision

- Circle vs. circle for projectiles and enemies.
- Range radius for towers, damage on overlap.

C. Combos

- Adjacent towers (≤ 60 px) trigger combo bonuses automatically.
- Combos stored in combos.json.

5. Towers and Combos

Tower	Function	Combo Synergy
Pin Launcher	Basic fast shooter.	Pairs with Button → Needleburst.
Button Mortar	Splash AoE.	Pairs with Spindle → Sky Stitcher.
Wool Cannon	Heavy beam.	Pairs with Patch → Guardian Loom.
Patch Totem	Healer/support.	Boosts reload + durability nearby.
Spindle Ward	Anti-air darts.	Pairs with Button → Sky Stitcher.

6. Enemies

[illegible]

Wolves	Ground	Fast rushers.
Threadlings	Ground	Swarm units.
Nightmoths	Air	Bypass ground towers.
Tatterbeasts	Ground	Slow tanks.
Spindlers	Air	Buff nearby allies.

Max: 30 enemies, 15 towers, 60 active projectiles.

7. Wave Transitions (New)

Concept

When the player presses Ready, before the next wave spawns, a thread-swipe transition runs across the screen:

- A stitched image unique to the world slides across (right→left or bottom→top).
- It creates a visual pause between waves, resetting tempo.
- On desktop, this same image serves as the side panel background, giving a unified look.

Assets

```
/assets/interstitials/
prairie.webp
desert.webp
forest.webp
```

Phaser Implementation

```

// GameScene.js
function startNextWave() {
  this.scene.launch('TransitionScene', { world: this.currentWorld });
}
// TransitionScene.js
export default class TransitionScene extends Phaser.Scene {
  init(data) { this.world = data.world; }

  preload() {
    this.load.image(this.world, `assets/interstitials/${this.world}.webp`);
  }

  create() {
    const img = this.add.image(this.scale.width, this.scale.height / 2, this.world)
      .setOrigin(1, 0.5)
      .setAlpha(0);

    this.tweens.timeline({
      targets: img,
      tweens: [
        { alpha: 1, x: this.scale.width / 2, duration: 300 },
        { x: -this.scale.width / 2, duration: 600, delay: 300 },
      ],
      onComplete: () => {
        this.scene.stop();
        this.scene.resume('GameScene');
      },
    });
  }
}

```

Desktop Integration

```

<aside class="world-art left"></aside>
<canvas id="game"></canvas>
<aside class="world-art right"></aside>

```

```

<style>
.world-art {
  flex: 1;
  background: url('/assets/interstitials/prairie.webp') center/cover no-repeat;
  opacity: 0.25;
}
</style>

```

Swap side art when the world changes:

```
document.querySelectorAll('.world-art').forEach(e =>
  e.style.backgroundImage = `url('/assets/interstitials/${world}.webp')`
);
```

8. HUD Design (Portrait)

- Top bar: fleece (gold), lives, wave counter, ×2 speed, pause.
 - Bottom tray: tower icons, cost, upgrade/sell buttons.
 - Right edge mini-panel: selected tower stats + combo badges.
 - Minimal tap distance, full touch area, no hover-only logic.
-

9. Audio + Feel

- Gentle woven ambient loop between waves.
 - “Thread swipe” transition triggers fabric-rustle SFX.
 - Mute, SFX, and music sliders in settings.
-

10. Performance Targets

- 60 FPS goal.
 - Object pools for enemies and projectiles.
 - WebP textures under 200 KB each.
 - Lazy-load transitions for next world only.
-

11. PWA + Offline

- manifest.webmanifest:

```
{ "orientation": "portrait", "display": "standalone" }
```

-

- sw.js: cache index.html, JS bundles, atlases, audio, interstitials.

12. Development Phases

Phase	Deliverable	Description
1	Core loop	Single world, path, enemies, towers.
2	Combos + UI	Adjacency bonuses, portrait HUD.
3	Transition System	Swipe art + desktop side panels.
4	Art & Sound	Embroidered assets, SFX, music.
5	Deploy	PWA build to PatternRipple Lab + Spoonflower pattern sync.

Result

The thread-swipe system gives every world a recognizable signature motion while doubling as desktop framing art.

This keeps load times low, gives visual continuity, and reinforces your textile brand identity across game and Spoonflower product lines.