

Kinoko Survivor プロジェクト俯瞰

今までの実績とこれからの作業整理

2025-12-16

プロジェクトの狙い

- 60FPS 固定の 2D サバイバルアクション（raylib ベース）
- ステート駆動でゲーム全体のライフサイクルを管理
- Entity + Component + StateMachine でキャラクタ挙動を分離
- Factory／Strategy／Observer を組み合わせた拡張性重視の設計

ここまでの実績

- **アプリ層:** `StateManager` と `Start/Play/ResultState` が稼働し、メインループに統合済み
- **エンティティ層:** `Player` / `Enemy` と `Position`・`Render`・`Health` 各コンポーネントを実装、`StateMachine` を介して挙動分離
- **AI/StateMachine:** `IState` , `StateMachine` , `PlayerMoveState` , `EnemyChaseState` 完了、`SimpleChaseAI` / `RangedAI` を導入
- **ファクトリ:** `EnemySpawner` が波ごとのスポン制御を提供し、ゲーム進行を担保
- **設計資産:** クラス構成資料と WBS を整備し、レイヤー毎の責務と依存が明文化

進行中・着手待ち領域

- StateMachine 拡張: `PlayerAttackState` / `PlayerDamageState` は骨組みのみ
- Factory/Item: `CharacterFactory` , `ItemFactory` , `TreasureChest` は API 宣言段階
- EventBus 連携: `EventBus` , `AudioManager` , UI 各種は Observer 実装待ち
- ステージ/背景: `StageBasic` , `BackgroundScroll` はプレースホルダ状態

TODO (docs/TODO.md 反映)

カテゴリ	優先タスク
アート/演出	プレイヤー/敵スプライト、BGM・SE、アニメーション
コアゲーム	スコア/レベル/ヒットボックス、プレイヤー攻撃・ジャンプ
コンテンツ	アイテムバッグ、各種アイテム、宝箱、エリート/ボス
システム	UI 改善、AI 強化、バランス調整、テスト・最適化
発表準備	ゲームオーバー演出、最終ビルド、ドキュメント更新

次の 4 スプリント指針

1. **戦闘の深度化:** プレイヤー攻撃/被弾ステート＋ヒットボックス整備
2. **成長サイクル:** スコア・経験値 UI、アイテム効果を EventBus で連動
3. **演出強化:** 新規アート、BGM/SE、背景スクロールを統合
4. **ボス解放:** エリート/ボス AI とステージ演出を一体テスト

リスクとフォロー策

- **アトリソース不足** → 仮素材テンプレを短期導入し並行開発
- **システム複雑化** → レイヤー毎のインターフェースを固定し結合テストを段階化
- **バランス調整の工数** → テレメトリ（基本ログ）を仕込み、テストデータを収集

まとめ

- コアループ・主要エンティティ・AI は既に動作し、骨格は完成済み
- 次フェーズは「戦闘の多層化」と「演出/UI」のテコ入れが鍵
- WBS/TODO をベースにスプリント計画へ直結できる状態