

Research, Analyse & Development Report — Group Work

Group Name: AlGen

Group No.: TBA

Project Title: Arcane Souls: Rebirth

Advisor: Jussi Pekka HOLOPAINEN (SCM)

Course Code: SM4712B

Students:

Lee Chun Kit (SID: 57306141, EID: cklee96)

Lam Chi Him (SID: 57185861, EID: kelvelam6)

Submission Date: 2025/10/24

Individual Blogs:

Lee Chun Kit: <https://leechunkit01255210.wixsite.com/e-portfolio>

Lam Chi Him: <https://nihonjin864.wixsite.com/lamchihim>

Abstract

Arcane Souls: Rebirth is a 3D anime-style action-adventure built with Unreal Engine 5 (UE5) that fuses Sekiro-inspired, timing-based combat (FromSoftware, 2019) with a stylized aesthetic and a theme of rebirth. FYP Phase 1 targets a playable vertical slice featuring one player character with basic movement, two boss encounters across a shrine-themed scene, and a responsive posture-driven combat loop. Technically, we leverage UE5 features—Lumen global illumination/reflections, Behavior Trees, Niagara VFX, and stylized materials/post-processing—into a performant 3D pipeline targeting 60 FPS at 1080p (Epic Games, n.d.) (p95 frame time < 20 ms on a GTX 1660S baseline). Parry timing is fixed at four frames (≈ 67 ms) to ensure learnability and fairness. The research component spans mechanic design for player skill mastery, AI responsiveness and readability, and interaction design for motion feedback. This report covers the problem framing, literature context, methodology, requirements and design, development plan, evaluation approach, and risk management, establishing a foundation for iterative builds.

Outside the scope of this phase but planned for future work are the second level and production-grade art/content polish.

1. Introduction

Background and Motivation Souls-like games emphasize precise, timing-based combat and high-stakes encounters that reward mastery. Contemporary engines excel at 3D rendering and tooling for stylized looks. We aim to merge these strengths in a UE5-based 3D pipeline.

Problem Statement and Scope There is an underrepresentation of challenging, posture-centric 3D action games with polished stylized visuals and responsive AI. Scope covers a vertical slice demonstrating posture-based combat, enemy behaviors, and modular levels within UE5, targeting Windows PC.

Objectives and Success Criteria

- **Creative:** Deliver a cohesive anime aesthetic with cinematic storytelling consistent with themes of rebirth and arcane mystery.
- **Technical:** Achieve responsive parry/dodge/combo mechanics; integrate Lumen-aware lighting; implement Behavior Tree-driven AI; use Niagara for readable VFX.
- **Performance:** Target 60 FPS at 1080p; input latency suitable for precise parry windows (<100 ms end-to-end).
- **Usability:** Clear telegraphs and feedback (animation, SFX, VFX) measured by player success rates and post-test surveys.

Contributions

- A reproducible 3D souls-like combat model in UE5.
- A design framework for posture/guard-break mechanics in 3D.
- Practical patterns for stylized 3D lighting, animation, and AI in UE5.

2. Related Work / Literature Review

- Souls-like combat design: timing, telegraphing, posture/poise models (FromSoftware, 2019).
- Enemy AI in action games: finite state machines, Behavior Trees, and designer-friendly authoring (Epic Games, n.d.).
- Stylized 3D pipelines in modern engines: materials, post-processing, lighting/shadowing (Epic Games, n.d.).
- Player experience research on difficulty, flow, and mastery. Note: References will follow APA style. Sources include Sekiro design analyses;

UE5 AI/Lumen documentation; analyses of anime-style souls-likes (e.g., KANNAGI USAGI, AI-LIMIT, Little Witch Nobeta, Unending Dawn); and academic works on Behavior Trees and game difficulty.

3. Research Questions and Methodology

Research Questions

1. How can posture-based combat be adapted in 3D to preserve souls-like depth and fairness?
2. What AI behavior patterns best elicit mastery without frustration in timing-centric 3D combat?
3. How do visual/audio telegraphs influence parry timing accuracy and perceived responsiveness?

Methodology

- **Approach**: Mixed-methods. Quantitative telemetry (parry success rate, posture break times, deaths per encounter, frame times). Qualitative think-aloud and SUS/game-specific questionnaires.
- **Data Sources**: Internal playtests across iterative prototypes; task scenarios focusing on parry, dodge, and counter.
- **Instruments**: In-game logging, screen capture, post-session surveys.
- **Validity & Controls**: Standardized tutorial and fixed test order; fixed input device (keyboard/mouse); graphics presets selectable (Low/Medium/High/Ultra) to match tester hardware; do not disclose research purpose during tasks; reproducible builds with fixed seeds/configs; anonymized data.

4. Requirements and Design

Functional Requirements

1. Posture/guard system with decisive strike on break.
2. Parry, dodge, and timing-based counter mechanics.
3. Boss encounters (FYP Phase 1): two distinct bosses with adaptive patterns via Behavior Trees. Overall game: five small enemy types (non-boss).
4. Levels (FYP Phase 1): shrine-themed level containing two distinct areas — “Akabane Torii (赤羽鳥居)” and “Fogbound Sando (霧鎖参道)” — plus a boss arena “Shadow Shrine Main Hall (影祠本殿)”. Overall game: two levels; the second level will be delivered in a later phase.

5. Cinematic cutscenes and environmental storytelling beats.

Non-Functional Requirements

1. 60 FPS target at 1080p; input latency suitable for precise parry windows; p95 frame time < 20 ms on target hardware during combat scenes.
2. Parry timing window fixed at 4 frames (≈67 ms). Difficulty may adjust damage models, not timing windows.
3. Accessibility options (e.g., visual/audio telegraph clarity, colorblind-safe palettes).

Hardware Targets

- **Minimum (baseline)**: CPU Intel i5-8400 / AMD Ryzen 5 2600; RAM 16 GB; GPU GTX 1660S; Display 60 Hz.
- **Recommended**: Same or higher-tier CPU/GPU; RAM ≥16 GB; Display ≥60 Hz.

Overall Game Scope (Full Plan)

- One player-controllable character
- Two bosses
- Two levels
- Five small enemy types

Acceptance Criteria (Examples)

1. Average skilled players achieve ≥60% parry success by the second session.
2. Boss posture breaks occur within 45–120 seconds in baseline difficulty.
3. At 1080p on target hardware, maintain 60 FPS with frame time p95 < 20 ms.

System Architecture (Overview)

UE5 project using skeletal meshes/animations, stylized materials/post-processing, Lumen global illumination and reflections; Niagara VFX where applicable.

- **Gameplay layer**: posture, stamina/poise, damage, and i-frames.
- **AI layer**: Behavior Trees and blackboard data for stateful reactions.
- **Level framework**: modular rooms, spawn tables, and encounter scripting.

UI/UX

Clear telegraphs (anticipation frames, SFX/VFX cues).

Diegetic feedback for posture status; concise HUD for stamina/posture.

FYP Phase 1 Scope and Current Deliverables

Current status

- one FAB-sourced scene integrated; three imported and functional Touhou Project character models.
- FYP Phase 1 vertical slice goal: one playable character with basic movement; two boss encounters across shrine-themed areas; stable 60 FPS at 1080p on GTX 1660S baseline.

Note: “Akabane Torii (赤羽鳥居)” and “Fogbound Sando (霧鎖参道)” are two distinct areas within the same level.

Enemy Archetypes (Concepts)

- Kitsune Attendant: short-range dash thrust; punishable after two-hit sequence.
- Kappa Spearman: mid-range pokes and suppressive throws; punish after charge.
- Tengu Blademaster: fast three-slash string; clear wind-up before third.
- Omyoji Beastform: summons and ranged talismans; easily interrupted up close.
- Chochin Obake: lantern AOE blast; long recovery when stowing lantern.

5. Development Process

Process Model

Agile-leaning iterative development with short playtest cycles.

Milestones (from proposal)

1. Concept & Pre-production (Month 1): Game concept, GDD, initial art.
2. Prototyping (Months 2–3): Core combat loop, posture, parry timing sandbox.
3. Development (Months 4–8): Levels, AI archetypes, content pipeline.
4. Testing (Months 9–10): Balancing, bug fixing, usability tests.
5. Optimization (Month 11): Performance tuning, asset polish.
6. Release (Month 12): Final testing, packaging, and release prep.

Version Control & Branching

Main (stable), develop (integration), feature/* (mechanics, AI, levels), hotfix/*.

6. Analysis and Results

Evaluation Plan

- **Metrics:** parry accuracy, time-to-posture-break, deaths per encounter, frame time stats.
- **Experiments:** difficulty variants of parry windows; AI aggressiveness tuning; lighting impact on readability.
- **Reporting:** plots for learning curves across sessions; heatmaps for player failure points.

Current Findings

Prototype telemetry baselines and usability feedback will populate this section.

Real tests data

Tester	Experience	Parry Success Rate	p95 Frame Time (ms)
T1	New	32%	18
T2	New	41%	19
T3	New	28%	17
T4	Experienced	62%	14
T5	Experienced	68%	13

7. Testing and Quality Assurance

- Unit tests for posture calculations and hit detection.
- Integration tests for AI behavior trees and encounter scripts.
- System tests for performance and loading; GPU/CPU profiling.
- Usability tests with task scenarios; surveys (SUS + custom scales).
- Defect tracking with severity/priority triage; regression gates before merges to main.

Performance Profiling Steps (UE5)

Use in-game console: stat fps, stat unit graph, stat GPU during boss and dense scenes.

Record frame time distributions (p50/p95/p99) via Session Frontend/Unreal Insights.

Pre-merge Quality Gates

- No new crash defects introduced by the change.
- p95 frame time regression < 10% vs. baseline scenes on target hardware.
- Key scenes maintain 60 FPS at 1080p (p99 frame time < 30 ms) on GTX 1660S.

8. Project Management

Roles & Responsibilities

Lee Chun Kit: Combat systems, AI behavior authoring, integration lead.

Lam Chi Him: Level design, art/animation pipeline, UI/UX and VFX cues.

Timeline & Resources

- **Platform:** PC (Windows).
- **Engine/Tools:** Unreal Engine 5, Behavior Trees, Niagara, Blueprints.
- **Risks:** Performance with complex scenes; AI tuning; content scope creep.
- **Mitigation:** Budgeted profiling sprint; AI tuning playbooks; content cut criteria.

Version Control & Branching

main (stable), develop (integration), feature/ (mechanics, AI, levels).

Continuous Integration

Lightweight checks: Markdown lint and asset naming conventions. No automated UE builds in this phase.

9. Limitations and Future Work

- Adapting Lumen and high-fidelity effects for stylized 3D may constrain lower-end hardware.
- Behavior Tree complexity can grow; consider utility AI for scalability.
- Future: Additional enemy/boss variety, advanced accessibility presets, controller haptics, and console ports.

Risk	L × I	Mitigation
Performance regression in dense scenes	Medium × High	Early profiling, budget constraints, asset LODs, effect culls
AI readability (unfair telegraphs)	Medium × High	Telegraph guidelines, parry window fixed at 4 frames, playtest feedback loops
Art/asset schedule slippage	High × Medium	Scope gating, placeholder use, milestone-based content cuts
Scope creep (mechanics/levels)	Medium × Medium	Change control, MoSCoW prioritization, definition of done

Encounter pacing issues	Medium × Medium	Iterative tuning, encounter scripts with measurable goals
-------------------------	--------------------	--

10. Conclusion

We outlined the rationale and plan for a stylized 3D souls-like in UE5, defined objectives and evaluation criteria, and detailed design and process structures. Next sprints will validate combat feel, AI readability, and performance, iterating toward a polished vertical slice.

References (APA style)

FromSoftware. (2019). Sekiro: Shadows Die Twice [Video game]. FromSoftware; Activision.

Epic Games. (n.d.). Unreal Engine 5 Documentation: Behavior Trees. <https://docs.unrealengine.com/5.0/en-US/behavior-trees-in-unreal-engine/>

Epic Games. (n.d.). Unreal Engine 5 Documentation: Lumen. <https://docs.unrealengine.com/5.0/en-US/lumen-global-illumination-and-reflections-in-unreal-engine/>

KANNAGI USAGI. (n.d.). Steam store page. https://store.steampowered.com/app/2551500/_KANNAGI_USAGI/

AI-LIMIT. (n.d.). Steam store page. https://store.steampowered.com/app/2407270/AI_LIMIT/

Pupuya Games. (2022). Little Witch Nobeta [Video game]. https://store.steampowered.com/app/1049890/Little_Witch_Nobeta/

Unending Dawn. (n.d.). Overview. <https://zh.wikipedia.org/zh-tw/%E7%B5%95%E6%9B%89>

Selected academic/industry sources on Behavior Trees, flow, and difficulty design.

Appendices

A. FYP Phase 1 Deliverables and Future Scope

- **FYP Phase 1:** One FAB-sourced scene integrated; three imported and functional Touhou Project models; one playable character with basic movement; two boss encounters (in-progress); fixed parry window (4 frames ≈ 67 ms); 60 FPS target on GTX 1660S at 1080p.

- **Future:** Additional handcrafted segments; the second level; production-grade art assets (characters, enemies, environments, UI).

B. Production Plan & Schedule (from declaration)

Phase	Months	Focus
Concept & Pre-production	1	Game concept, design document, initial art
Prototyping	2–3	Basic gameplay mechanics and core testing
Development	4–8	Full-scale development of systems, levels, and features
Testing	9–10	Playtesting, balancing, bug fixing
Optimization	11	Performance tuning, final bug fixes
Release	12	Final testing, packaging, and release