

Final Year Project Title:
Developing Multi-
Character Control System
for Switching in Blueprint

Post Project Title:
My Gameplay Sandbox

Game Title:
Sigil Commander's
Modular Chronicle

Personal Design Document

By Paul Wong

REMARKS:

This document is used for personal design process only, and will not be a formal Game Design Document that detail.

The first version is for my [Final Year Project\(FYP\)](#), mark in blue and using English for marking.

Second Version, which is the personal project using the [FYP](#) as the base to reform a new game, will be using my first language in [Traditional Chinese-繁體中文](#) for my idea organisation, with marked with Purple.

[A translated English version](#) will be added at the end and mark with Orange.

Contents

Contents	3
Game Overview	5
Inspiration	5
Game Concept	5
Genre	7
Game Flow Summary	8
Flow	8
Look and Feel	9
Core Pillars of Design	10
Pillar 01	10
Pillar 02	10
Pillar 03	10
Pillar 04	10
Pillar 05	11
Gameplay Systems & Mechanics	13
Mechanic 01	13
Mechanic 02	13
Mechanic 03	13
Mechanic 04	13
Mechanic 05	13
Mechanic 06	13
Mechanic 07	13
Mechanic 08	13
Narrative	20
Story	20
<i>Background</i>	20
<i>Introduction</i>	20
Game World	20
<i>Setting and Tone</i>	21
<i>Areas</i>	21
Characters	21
<i>Protagonist</i>	21
<i>Antagonist</i>	21
<i>Other Characters</i>	21

Levels & Maps	22
Level 01 –	22
Level 02 –	22
Level 03 –	22

Game Overview

Inspiration

FYP:

Pokémon, Genshin Impact, Baldur's Gate 3, Dragon Quest XI, Arknights (Reclamation Algorithm mode), and Escape Room games.

After FYP, Personal Project:

遊戲系統設計上概念部份參考中式傳統命理學中的「干支」與「紫微斗數」結構，但僅作為結構邏輯與詞組分類之基礎，實際遊戲內不會直接使用此類術語，亦不含玄學元素。

The concepts in the game system design refer to the " Sexagenary cycle " and " Chinese astrology / Ziwei doushu " structures in traditional Chinese numerology, but they are only used as the basis for structural logic and phrase classification. Such terms will not be used directly in the actual game, and it does not contain metaphysical elements.

Game Concept

Main idea:

Player character: role- commander, control/giving orders to other characters in party

Other characters: NPC, can invite to the team

Probably like Pokémon concept in general.

Story: not the main focus of the game, but still needed, make it simple

Gameplay related:

Simple Puzzles in escape room games, need to control different characters towards to exit

Turn base combat

Other things in game want to have:

Resource Management

Character Growth

Save Game option

Core:

Overworld/main world

Combat Level

Escape Room/Puzzle Level

The game focuses on the player assuming the role of a commander who gives orders to characters in a party. It integrates character switching as a core mechanic, combined with turn-based combat, puzzle-solving, and exploration elements.

- **Player Role:** Commander, tactically controlling characters.
- **Team Composition:** NPCs can be recruited into the player's party.

- **Story:** A simplified narrative acts as a backdrop, while gameplay remains the focus.

Core Gameplay

- **Combat Levels:** Turn-based battles using speed/dexterity to determine action order.
- **Puzzle Levels:** Simple character-based puzzles requiring players to solve challenges through character switching.
- **Overworld Exploration:** Players can interact with NPCs, accept quests, and manage resources.

MoSCoW:

Must:

Basic Character Control System
Character Switching System
Basic UI for Character Information
Components Necessary for Switching System

Should:

Add Mechanics Showcasing Character Switching
Develop Mechanics to Enhance Gameplay
Optimize the Switching System
Build a Playable Game

Could:

Simple Level Design
Add Sound and Visual Effects
Other Game Juice Elements (e.g. Particles Effects, Materials, Post Process on Camera, etc)

Won't:

Create Art-Related Assets
Create New Animations

遊戲核心概念（**Game Concept**）本作為一款中世紀幻想風格的模組化策略 **RPG**，融合回合制戰鬥、多角色編隊、模組技能建構、隨機角色組合與高重玩性機制。每一局為一次完整的遊戲週目，玩家需管理隊伍、配置模組、完成主要任務，逐步揭露世界系統規律與其背後力量

玩家角色定位（**Player Role**）

玩家為一名中世紀背景下的指揮官（具地位者），擁有一座城堡作為行動與戰鬥中樞，能夠招募人才、發展隊伍、處理事件與規劃任務。身份細節與敘事背景可依劇情展開進一步設定。

Game Core Concept (Game Concept)

This is a medieval fantasy style modular strategy RPG, integrating turn-based combat, multi-character formation, modular skill construction, random character combination and high replayability mechanism. Each game is a complete game cycle. Players need to manage teams, configure modules, complete major tasks, and gradually reveal the laws of the world system and the power behind it.

Player Role

The player is a commander (a person of status) in a medieval background. He has a castle as an action and combat centre. He can recruit talents, develop teams, handle events and plan tasks. Identity details and narrative background can be further set according to the plot.

Genre

3D Adventure RPG with Turn-Based Combat.

- 主 Genre：RPG（角色扮演）
- 子 Genre：創新型模組化 **Rogue-like Build RPG**
- 特點組合：
 - 隨機角色抽取（角色模組為單位）
 - 回合制戰鬥（原 FYP 架構）
 - 高重玩性（每局為一次世界命運週期）
 - 每局體驗差異化，類似 *Baldur's Gate 3* 式選擇導向
- Main Genre: RPG (Role Playing)
- Sub-Genre: Innovative Modular Rogue-like Build RPG
- Feature combination:
 - Random character extraction (character module as a unit)
 - Turn-based combat (original FYP architecture)
 - High replayability (each game is a world destiny cycle)
 - Differentiated experience in each game, similar to the choice-oriented Baldur's Gate 3

Game Flow Summary

Flow

Main Menu

For first time: Player Character customises

To the main world level

Until Pressing Quit Game

Things could do in world level:

conversation with NPC to battle(towards combat level), invite, get quest, buy or sell items?

Trigger puzzle levels (probably in dungeon)

Special events?

Combat Loop - Turn Base Combat:

Similar to Baldur's Gate, each character have few actions in each move

Since having dex/speed in each character(both player and enemies) that the next move base on who is faster, when form a looping in base on the speed of the characters(option 1)

(option 2 if having time for visual game feel: dynamic progress bar/timeline, using character run towards the end point as next move)

Calculation:

Speed = Base Speed + (Dexterity * Speed Multiplier)

Cooldown Delay = Base Cooldown + (Dexterity * Cooldown Multiplier)

Auto switching character: to next moved character

Non linear storyline and have multi-ending, need event tree to present

Using classic game loop as overall game loop, Trigger event->Reward->Strengthen

Other loops: Story Loop, Economy/Enhancement Loop

1. **Main Menu:** Player customises their character.
 2. **Overworld:** Exploration, NPC interaction, resource management, and quest initiation.
 3. **Combat Levels:** Turn-based battles where players make strategic decisions.
 4. **Puzzle Levels:** Environmental challenges that require players to switch between characters.
 5. **Game End:** Outcomes are determined by player actions, offering multiple endings.
-

開局：生成當局遊戲週目配置

教學：第一局強制進入 Tutorial（可選擇跳過）

村落行動階段：

- 消耗 AP 進行：招募角色、探索事件、收集模組…

• 每日結束前選擇任務 (Quest) : 所有 Quest 均為主線、進度關聯劇情

戰鬥場景階段 (Castle) : 進行回合制戰鬥

Quest 結算 : 獲得報酬、模組 / 插槽 / 詞組 / 金錢等

重覆第 3~5 步, 直至觸發 Final Boss 任務 → 戰鬥勝利

完成一局遊戲週目 → 進入繼承畫面, 選擇繼承元素 → 開始下一局

Start: Generate the game configuration

Tutorial: The first game is forced to enter the Tutorial (optional to skip)

Village Action Phase:

- Consume AP to: recruit characters, explore events, collect modules...
- Select tasks (Quest) before the end of each day: All Quests are main lines and progress-related plots

Battle Scene Phase (Castle): Conduct turn-based battles

Quest settlement: Get rewards, modules/slots/phrases/money, etc.

Repeat steps 3 to 5 until the Final Boss mission is triggered → Battle victory

Complete a game cycle → Enter the inheritance screen, select the inheritance element → Start the next game

Look and Feel

Fantasy, allow all kind of style character import to the game,
Dungeon for battle

Fantasy aesthetic with dungeon-like environments.

The world is immersive, with areas designed for exploration, puzzles, and battles.

Core Pillars of Design

Pillar 01

Characters and Switching system

In base Character:

Variables:

1. Health – integer
2. Name – Text
3. Icon – image/Texture
4. Speed/Dexterity – integer/float
5. Power – integer/float
6. Level – integer
7. EXP – float

Character Switching System Character switching forms the backbone of gameplay, enabling tactical decision-making.

- **Character Attributes:**
 - Health (HP)
 - Speed/Dexterity
 - Power
 - Level
 - Unique abilities or traits.

Pillar 02

Gameplay related/utilising character switching, e.g. Puzzles, Combat

Turn-Based Combat

- A speed-based system determines action order.
- Cooldowns and resource costs encourage strategic use of character actions.

Pillar 03

Puzzle-Solving and Exploration

- Environmental puzzles require players to switch characters to progress.
- Areas are designed with interactive elements that challenge the player.

Pillar 04

Resource Management

- Limited resources like health potions, keys, and special items must be managed carefully.
- Choices made early in the game impact difficulty later on.

Pillar 05

Party System

- Players can recruit and manage NPCs.
- Dynamic switching allows players to adapt to different challenges.

模組化構建與高度自由策略性

玩家所擁有的每一位角色、技能模組與技能插槽，皆以模組化方式隨機或條件生成，並可由玩家依情況進行配置。這種組合式建構機制，使得每一次編隊與出戰都能形成截然不同的 **Build** 策略，強調自由構建與高複雜度的系統相互作用。

週目式重玩設計（Run-based Play Loop）

遊戲以「一局一週目」的形式推進，每一局開局條件與世界規則皆會變化，包括角色生成、敵人風格、任務配置等，並在清局後允許玩家選擇繼承部分元素。這種架構可大幅提升重玩性與變化性，讓玩家每次進入新週目都有新的策略挑戰與長期構築目標。

系統驅動式世界互動與選擇意義

本作非線性劇情進程以系統互動為核心，玩家每次選擇角色組合、配置技能模組與選擇任務方向，都會改變當前世界中可觸發的事件、敵人行為與資源分佈。這不單強化了「**Build** 有結果」的感受，也使劇情與世界演化能回應玩家操作而非僅由預設主線推動。

Modular construction and high degree of free strategy

Each character, skill module and skill slot owned by the player are randomly or conditionally generated in a modular way, and can be configured by the player according to the situation. This combination construction mechanism allows each

formation and battle to form a completely different build strategy, emphasizing the interaction between free construction and highly complex systems.

Run-based Play Loop

The game is advanced in the form of "one round per round", and the starting conditions and world rules of each round will change, including character generation, enemy style, task configuration, etc., and players are allowed to choose to inherit some elements after clearing the game. This architecture can greatly improve replayability and variability, allowing players to have new strategic challenges and long-term construction goals every time they enter a new round.

System-driven world interaction and choice significance

The non-linear plot process of this game is centred on system interaction. Every time a player chooses a character combination, configures a skill module, and chooses a task direction, it will change the events that can be triggered in the current world, enemy behaviour, and resource distribution. This not only strengthens the feeling of "Building with Results", but also enables the plot and world evolution to respond to player operations rather than being driven solely by the preset main line.

Gameplay Systems & Mechanics

Mechanic 01

Character Switching System:

Core mechanic allowing players to switch characters during combat, exploration, and puzzles.

Mechanic 02

Puzzles, maybe having some board game or chess ideas

Puzzle Mechanics:

Environmental challenges requiring strategic character use.

Mechanic 03

Combat, turn base

Turn-Based Combat:

Players issue commands based on speed and character abilities.

Mechanic 04

Party Management:

Players recruit and organise a team of NPCs.

Mechanic 05

Player character customisation at the beginning

Mechanic 06

Quest

Mechanic 07

Skil Tree

Mechanic 08

Character Spawning:

Circular Layout Method for Character Spawning

Objective

To spawn multiple characters around a central base location with calculated offsets, ensuring that each character maintains a minimum distance from others to prevent collisions.

Key Concepts

1. **Base Location:** The central point around which characters will be spawned.
2. **Radius:** The distance from the base location to each character's spawn point.
3. **Angle Calculation:** Determining the position of each character around the circle based on evenly spaced angles.
4. **Radians vs. Degrees:** Mathematical functions like cos and sin in UE5 use radians, so angle conversion is essential.
5. **Random Angle Offset:** Introducing slight randomness to angles to avoid perfectly uniform placement, enhancing visual diversity.

Steps to Implement Circular Layout Spawning

1. Define Parameters

- **Base Location:** The central point (e.g., (1070, 2230, 60)).
- **Radius:** The distance from the base location to spawn each character (e.g., 100 units).
- **Number of Characters:** Total characters to spawn (e.g., 3).
- **Random Angle Offset Range:** The range of randomness to apply to each character's angle (e.g., -30 to +30 degrees).

2. Calculate Angle Interval

Formula:

$$\text{Angle Interval} = \frac{360^\circ}{\text{Number of Characters}}$$

Example: For 3 characters:

$$\text{Angle Interval} = \frac{360^\circ}{3} = 120^\circ$$

3. Loop Through Each Character

For each character in the Characters array, perform the following:

a. Calculate Base Angle

- Formula: $\text{Base Angle} = \text{Index} \times \text{Angle Interval}$
- Example:
 - Character 1: Index = 0, Base Angle = 0°
 - Character 2: Index = 1, Base Angle = 120°
 - Character 3: Index = 2, Base Angle = 240°

b. Apply Random Angle Offset

- Formula: $\text{Final Angle} = \text{Base Angle} + \text{Random Offset}$
- Random Offset: A random float value within the specified range (e.g., -30 to +30 degrees).
- Purpose: Adds variability to spawn positions, preventing uniform and predictable placement.

c. Convert Degrees to Radians

- Formula: $\text{Angle Radians} = \text{Final Angle} \times (180 \div \pi)$
- Reason: UE5's trigonometric functions (cos and sin) require angles in radians.

d. Calculate Spawn Location

- Formulas:
 $X = \text{Base X} + \text{Radius} \times \cos(\text{Angle Radians})$
 $Y = \text{Base Y} + \text{Radius} \times \sin(\text{Angle Radians})$
 $Z = \text{Base Z}$
- Result: A Vector (X, Y, Z) representing the spawn location for the character.

e. (Optional) Calculate Rotation

- Purpose: Make characters face towards the center.
- Method:
 - Use the Find Look at Rotation node.
 - Start: Character's spawn location.
 - Target: Base location.
 - Result: Rotation that makes the character face the center.

f. Spawn the Character

- Action:
 - Use Spawn Actor from Class node.
 - Class: Current character class from the Characters array.

- Spawn Transform: Location calculated above and optional rotation.

Blueprint Implementation Overview

1. Variables

- a. Characters (Array of Character Classes)
- b. BaseLocation (Vector, e.g., (1070, 2230, 60))
- c. Radius (Float, e.g., 100.0)
- d. RandomOffsetMin (Float, e.g., -30.0)
- e. RandomOffsetMax (Float, e.g., 30.0)

2. Event BeginPlay

- a. Loop: For Each Loop over Characters array.
 - i. Loop Body:
 1. Calculate Base Angle: $\text{Index} * \text{AngleInterval}$
 2. Generate Random Offset: Random Float in Range (RandomOffsetMin, RandomOffsetMax)
 3. Final Angle: Base Angle + Random Offset
 4. Convert to Radians: Degrees to Radians (Final Angle)
 5. Compute Spawn Location:
 - a. $X = \text{BaseLocation}.X + \text{Radius} * \cos(\text{AngleRadians})$
 - b. $Y = \text{BaseLocation}.Y + \text{Radius} * \sin(\text{AngleRadians})$
 - c. $Z = \text{BaseLocation}.Z$
 6. (Optional) Compute Rotation: Find Look at Rotation (Spawn Location -> BaseLocation)
 7. Spawn Actor: Spawn Actor from Class with calculated location and rotation.

Troubleshooting Common Issues

1. Characters Spawning Too Close

- a. Cause: Radius is too small relative to the characters' collision sizes.
- b. Solution: Increase the Radius value to ensure sufficient spacing between characters.

2. Incorrect Position Calculations

- a. Cause: Angles not correctly converted to radians.
- b. Solution: Verify that angles are properly converted using Degrees to Radians before applying cos and sin.

3. Overlap Despite Correct Calculations

- a. Cause: Random angle offsets too small or collision sizes larger than expected.
- b. Solution:
 - i. Increase the range of random angle offsets to enhance spacing variability.

- ii. Reassess character collision settings and ensure that the Radius accommodates them.
4. Characters Not Facing Center (if rotation is applied)
 - a. Cause: Rotation calculation or application issues.
 - b. Solution: Ensure Find Look at Rotation is correctly set up to face the center from the spawn location.

Best Practices

- **Dynamic Radius Adjustment:** If the number of characters varies, consider dynamically adjusting the Radius based on the number of characters to maintain spacing.
- **Collision Checks:** Implement additional collision checks (e.g., using Sphere Trace) before spawning to ensure the spawn location is free from obstacles.
- **Modular Design:** Encapsulate the spawning logic within a function or a dedicated Blueprint to promote reusability and maintainability.
- **Visual Debugging:** Use debug lines or markers to visualize spawn positions during development, ensuring characters are placed correctly.

遊戲時間系統 (Time System)

日數 (Day)：每當玩家消耗行動點 (AP) 進行任務、事件等，即視為「過去一日」。AP 每日補滿，不同行動會消耗不同數量。

任務與行動系統

玩家每日有固定行動點 (AP)，用於進行招募、探索、任務等操作。任務設有時間限制與不同報酬，引導玩家規劃資源分配與戰術節奏。

遊戲內時間進程：任務與事件會推進遊戲世界的月份與年份，影響劇情演變、敵人配置與任務時限。

遊戲週目：每一局遊戲即為一個世界模擬週期，控制整體 AI 行為傾向、資源分佈、Meta 結構與可選角色等。

角色系統 (Character System)

抽角機制：角色為模組化單位，抽取時會隨機生成一組詞組效果 (Effect Pair)，一組插槽配置 (Trigger Slot)，與若干技能模組 (Skill Module)。

詞組系統 (主屬性+效果方向)：每名角色隨機獲得一組主屬性與對應效果方向，用以修正其技能模組的觸發條件或表現，例如：Boosted Delay、Reduced Heal。

插槽（Slot）：角色最多擁有 3 個技能插槽，插槽類型決定可裝入模組類型與觸發方式。

技能模組（Skill Module）：具類型、描述與技能效果的主動技能，插入插槽方能使用，受詞組與觸發條件影響。

戰鬥系統（Battle System）

回合制系統：玩家可操作 1~3 名角色，以輪替回合方式出招。

技能觸發：技能需插入對應插槽才可觸發，其表現受角色詞組條件強化／削弱。

敵方 AI：所有非玩家角色共用統一 AI 行為邏輯模組，其技能選擇與戰術傾向將根據角色持有的詞組組合進行調節。實作層預留使用行為樹（Behavior Tree）以實現更多策略反應。

遊戲週目系統（Game Run System）

每局遊戲開局時會隨機（根據上一局權重加成）生成一個遊戲週目配置。

遊戲週目會影響：

- 敵人 AI 傾向（攻擊型、防守型、延遲型...）
- 抽角出現類型（某類詞組組合出現率上升）
- 任務風格與環境特性（控制系敵人出現率、持續傷害環境等）

每局戰鬥最終會有 **Final Boss** 作為該週期挑戰核心，勝出者可選擇部分繼承項目（角色／模組／插槽／世界條件）進入下一局。

Time System

Day: Every time a player consumes action points (AP) to perform tasks, events, etc., it is considered as "the past day". AP is replenished daily, and different actions consume different amounts.

Task and Action System

Players have a fixed number of action points (AP) every day, which are used for recruitment, exploration, tasks, etc. Tasks have time limits and different rewards to guide players to plan resource allocation and tactical rhythm.

In-game time process: Tasks and events will advance the month and year of the game world, affecting the evolution of the plot, enemy configuration and task time limit.

Game cycle: Each game is a world simulation cycle, which controls the overall AI behaviour tendency, resource distribution, Meta structure and optional characters.

Character System

Character extraction mechanism: Characters are modular units. When extracted, a set of phrase effects (Effect Pair), a set of slot configurations (Trigger Slot), and several skill modules (Skill Module) will be randomly generated.

Phrase system (main attribute + effect direction): Each character randomly obtains a set of main attributes and corresponding effect directions to modify the triggering conditions or performance of its skill module, such as Boosted Delay, Reduced Heal.

Slot: A character has up to 3 skill slots, and the slot type determines the module type and triggering method that can be installed.

Skill Module: An active skill with type, description and skill effect, which can only be used after being inserted into a slot, and is affected by phrases and triggering conditions.

Battle System

Turn-based system: Players can operate 1 to 3 characters and make moves in a rotating turn.

Skill trigger: Skills must be inserted into the corresponding slot to be triggered, and their performance is strengthened/weakened by the character's phrase conditions.

Enemy AI: All non-player characters share a unified AI behaviour logic module, and their skill selection and tactical tendencies will be adjusted according to the phrase combination held by the character. The implementation layer reserves the use of behaviour trees to achieve more strategic responses.

Game Run System

At the beginning of each game, a game run configuration will be randomly generated (based on the weight bonus of the previous game).

The game run will affect:

- Enemy AI tendency (offensive, defensive, delayed...)
- Draw type (increased appearance rate of certain phrase combinations)
- Mission style and environmental characteristics (control enemy appearance rate, continuous damage environment, etc.)

At the end of each battle, there will be a Final Boss as the core of the cycle challenge. The winner can choose to inherit some items (character/module/slot/world conditions) to enter the next round.

Narrative

Narrative Design 筆記（開發預備區）

玩家身份將設定於中世紀幻想背景，擁有城堡與地位，具管理／指揮功能。

→ 頭銜與階級尚未決定，將依故事主題與世界觀架構進一步確立。

遊戲世界尚未設定明確國家、宗教或地理範圍，

→ 可根據模組化系統特性，設計一個與符印／技能插槽系統對應的文化或技術體系（如：符印文化、命構理論）。

主線劇情將環繞「週目式演化世界」展開，玩家每次進入新局，會因選擇、角色、任務與技能構成，引發不同世界反應。

→ 每局結束後可強化命構記憶，逐步揭開整體世界之結構與重複輪迴原因。

任務設計方向初步：

所有 **Quest** 屬主線，但可藉每日行動點（AP）解鎖支線／人物事件；

任務將以「事件碎片」形式，拼湊每一週目的命構圖景；

部份任務帶有時間限制，會錯過而影響週目分歧。

Narrative Design Notes (Development Preparation Area)

The player's identity will be set in a medieval fantasy background, with a castle and status, and management/command functions.

→ Titles and classes have not yet been determined, and will be further established based on the story theme and worldview structure.

The game world has not yet set a clear country, religion or geographical scope.

→ According to the characteristics of the modular system, a cultural or technical system corresponding to the sigil/skill slot system (such as sigil culture, fate theory) can be designed.

The main storyline will revolve around the "round-based evolutionary world". Every time the player enters a new game, the world will react differently due to the choices, roles, tasks and skill composition.

→ After each game, the fate memory can be strengthened, and the structure of the entire world and the reasons for repeated reincarnation can be gradually revealed.

Preliminary direction of task design:

All Quests belong to the main line, but side quests/character events can be unlocked by daily action points (AP);

The tasks will be in the form of "event fragments" to piece together the fate picture of each round;

Some tasks have time limits, and missing them will affect the differences in the round.

Story

Background

Introduction

Game World

Setting and Tone

Areas

Characters

Protagonist

Antagonist

Other Characters

Levels & Maps

Level 01 –

World Level

Village（村落）：玩家主 HUB，進行招募、任務選擇、事件處理、AP 消耗等日常行動。

Village: The player's main HUB, where daily operations such as recruitment, mission selection, event processing, AP consumption, etc. are carried out.

Level 02 –

Combat Level

Castle（城堡）：戰鬥專用場景，所有主線戰鬥與 Final Boss 戰鬥皆於此展開。

Castle: A special battle scene where all main battles and Final Boss battles take place.

Level 03 –

Dungeon（地城）：暫未實裝，預留作探索類資源區域，或擴展特殊挑戰／模組收集用途。

Dungeon: Not yet implemented, reserved for exploration resource areas, or expansion of special challenges/module collection purposes.

