개발팀이 코드를 작성하고 시스템을 구현하는 데 바로 사용할 수 있도록, **직군 시스템 표를 한 줄로 깔끔하게 정돈**하여 통합한 **최종 게임 기획서**입니다.

# [Project: 도리천] 전투 시스템 명세서 (Combat System GDD)

* **문서 버전:** v1.0 (Final)
* **작성자:** Gemini (AI Thought Partner)
* **수신:** 클라이언트/서버 프로그래머, 시스템 기획자
* **핵심 로직:** 속성 화학 반응(Universal Physics) & 미션 게이지(Mission Gauge)

## 1. 시스템 개요 (System Overview)

본 프로젝트의 전투는 \*\*[관측자(플레이어)의 미션 선언] $\rightarrow$ [탐사대(캐릭터)의 수행] $\rightarrow$ [인과 확정]\*\*의 3단계 루프로 진행된다. 단순한 턴제 공격이 아니라, \*\*속성 조합(Combo)\*\*을 통해 목표한 \*\*에너지 게이지(Gauge)\*\*를 채우는 것이 승리 조건이다.

### 1.1 용어 정의 (Terminology)

| **용어** | **설명** | **비고** |
| --- | --- | --- |
| **관측자 (Observer)** | 플레이어. 인과를 조율하고 명령을 내리는 주체. | User ID |
| **미션 (Mission)** | 매 턴 달성해야 할 목표 에너지 (혼돈/정적). | Turn Goal |
| **현상 (Phenomenon)** | 두 가지 속성이 충돌하여 발생하는 특수 효과. | Combo Effect |
| **인과 부채 (Paradox Dmg)** | 미션 실패 시 아군이 입는 페널티 데미지. | Fail Penalty |

## 2. 전투 루프 로직 (Combat Loop Logic)

전투는 아래의 Phase 순서대로 처리한다.

### Phase 1. 관측 (Observation)

1. 시스템이 현재 전황을 분석하여 2종의 미션(혼돈/정적) 중 하나를 제시한다.
2. UI에 목표 게이지(0/100)와 아이콘이 표시된다.

### Phase 2. 연산 (Calculation) - ***Main Gameplay***

1. 플레이어는 캐릭터 스킬을 사용한다.
2. **[1타: 부착]** 적에게 속성을 묻힌다. (상태이상 Slot 1)
3. **[2타: 반응]** 다른 속성으로 타격 시 \*\*[현상]\*\*이 발생한다.
4. 발생한 현상의 종류와 스킬 속성에 따라 **미션 게이지가 충전**된다.

### Phase 3. 확정 (Realization)

1. **Check:** 턴 종료 시 게이지가 **100% 이상**인가?
   * **TRUE (성공):** [인과 확정] 연출 출력 + 선언된 보상 효과 발동 (예: 적 전체 300% 데미지).
   * **FALSE (실패):** [타임 패러독스] 경고 출력 + 아군 전체 체력 20% 감소.
2. 게이지와 적의 부착된 속성을 초기화하고 다음 턴으로 진행.

## 3. 속성 시스템: 절대 물리 법칙 (Universal Physics)

직군(Class)과 무관하게, \*\*[속성 A + 속성 B]\*\*는 항상 고정된 \*\*[결과 C]\*\*를 출력한다.

### 3.1 미션 타입별 조합 테이블 (Data Table)

#### Type A: 🔥 혼돈 에너지 (Chaos)

* **목표:** 적 파괴 / 데미지 누적
* **키 속성:** **🔴 엔트로피 (Entropy)** 필수 포함

| **Slot 1 (Target)** | **Attack (User)** | **결과 현상 (Result)** | **효과 (Effect)** |
| --- | --- | --- | --- |
| **🟣 중력** | **🔴 엔트로피** | **💥 핵융합** | [광역] 타겟 및 주변 확산 피해 |
| **🔵 스테이시스** | **🔴 엔트로피** | **💔 열충격** | [관통] 방어력/보호막 무시 피해 |
| **🟡 공명** | **🔴 엔트로피** | **🔥 플라즈마** | [확산] 상태이상(DoT) 주변 전이 |
| *(※ 반대 순서도 결과 동일)* |  |  |  |

#### Type B: ❄️ 정적 에너지 (Silence)

* **목표:** 적 제어 / 행동 방해
* **키 속성:** **🔵 스테이시스(Stasis)** 또는 **🟣 중력(Gravity)** 포함

| **Slot 1 (Target)** | **Attack (User)** | **결과 현상 (Result)** | **효과 (Effect)** |
| --- | --- | --- | --- |
| **🟣 중력** | **🔵 스테이시스** | **❄️ 절대영도** | [빙결] 행동 불가 (Turn Skip) |
| **🟡 공명** | **🔵 스테이시스** | **⚡ 과부하** | [마비] 스킬 쿨타임 증가 |
| **🟡 공명** | **🟣 중력** | **🌑 블랙홀** | [실명] 명중률 50% 감소 |

### 3.2 조커 속성 로직 (Special Logic)

일반 반응 테이블을 무시하고(Override) 우선 처리한다.

* **⚫ 패러독스 (Paradox):**
  + Condition: 적에게 Any Attribute 존재 시.
  + Effect: 미션 게이지 **100% 즉시 충전**.
  + Penalty: 시전자의 현재 체력 10% 감소.
* **⚪ 공리 (Axiom):**
  + Condition: 적에게 Any Attribute 존재 시.
  + Effect: 미션 게이지 **30% 충전**.
  + Bonus: 아군 전체 보호막(최대 체력 10%) 생성.

## 4. 직군 시스템 (Class System)

직군은 캐릭터의 스탯(Stat)과 스킬 타입을 결정할 뿐, **속성 반응 공식에는 관여하지 않는다.**

| **직군명 (Class)** | **역할 정의 (Role)** | **주요 스탯 및 특징** |
| --- | --- | --- |
| **🛡️ 인터셉터 (Interceptor)** | 전방에서 적의 공격을 받아내고 아군을 지키는 **[탱커]** | 높은 HP/DEF, 도발 및 피해 감소 스킬 보유 |
| **⚔️ 엑시큐터 (Executor)** | 강력한 화력으로 적을 섬멸하고 결과를 확정하는 **[메인 딜러]** | 높은 ATK/CRIT, 단일/광역 누킹 스킬 보유 |
| **⚕️ 스테빌라이저 (Stabilizer)** | 상처 입은 아군을 치유하고 상태를 정상화하는 **[힐러]** | 높은 회복력, 체력 회복 및 상태이상 해제 스킬 보유 |
| **🚩 패스파인더 (Pathfinder)** | 아군을 가속하거나 적을 약화(디버프)시키는 **[서포터]** | 높은 속도(AGI), 턴 게이지 주유 및 버프/디버프 스킬 보유 |

## 5. 미션 게이지 산정 로직 (Algorithm)

유저의 행동이 미션 게이지를 얼마나 채우는지 계산하는 로직이다. Add\_Gauge(Action\_Type) 함수로 구현한다.

C#

// Pseudo-code Example  
  
void OnAttackHit(User attacker, Target enemy) {  
   
 // 1. 속성 반응 체크  
 string reaction = CheckReaction(enemy.currentElement, attacker.element);  
 float score = 0;  
   
 // 2. 미션 타입에 따른 점수 계산  
 if (reaction != null) {  
 // 콤보(현상) 발생 시: 미션과 일치하면 50점, 아니면 20점  
 score = IsReactionMatchingMission(reaction, currentMission) ? 50.0f : 20.0f;  
 }   
 else if (attacker.element == Special.Paradox) {  
 score = 100.0f; // Cheat: 패러독스 (리스크 적용 별도)  
 }  
 else if (attacker.element == Special.Axiom) {  
 score = 30.0f; // Safe: 공리  
 }  
 else {  
 // 단일 속성 적중 시: 미션 속성과 일치하면 15점, 아니면 0점  
 score = IsElementMatchingMission(attacker.element, currentMission) ? 15.0f : 0.0f;  
 }  
  
 // 3. 게이지 적용 및 UI 갱신  
 CurrentMissionGauge += score;  
 UpdateUI();  
}

## 6. UI/UX 가이드라인 (Implementation Guide)

유저가 복잡한 표를 외우지 않도록 **시각적 힌트**를 제공해야 한다.

1. **스킬 버튼 하이라이트 (Traffic Light):**
   * 현재 미션이 \*\*[🔥혼돈]\*\*일 경우, 플레이어의 **🔴/🟣/⚫/⚪** 스킬 버튼 테두리가 Looping Glow 애니메이션 처리된다.
2. **타겟팅 프리뷰 (Preview):**
   * 스킬을 드래그하여 적에게 가져가면, 적 머리 위에 \*\*[예상 결과 아이콘]\*\*이 팝업된다.
     + *예: 💥 아이콘 + 게이지 +50 표시*
3. **공리(⚪) 아이콘 처리:**
   * 공리 속성 스킬 아이콘은 현재 미션 색상에 맞춰 **실시간으로 테두리 색이 변한다.** (카멜레온 효과)

### [개발팀 전달 사항]

* **우선순위 1:** 속성 간 반응 테이블(Reaction Logic)의 하드코딩 및 유닛 테스트.
* **우선순위 2:** 3단계 미션 루프(Observe -> Act -> Result)의 턴 구조 및 게이지 연동 구현.
* **우선순위 3:** 직군별 스탯 테이블 작성 및 밸런싱 작업.