

개인 프로젝트

로캣: 잔향의 던전
메인 기획서

핵심 메카닉스 — 완전 동기 턴 / 전투 / 포획 / 혼 / 성장

문서 최초 작성	문서 최종 수정
20260816	20260816
박도원	박도원

목차

개요 – 2P

1. 목차
2. 로그
3. 문서 계층

현 상황 – 5P

1. 컨셉 확정 시점

목표 재확인 – 6P

1. 목표와 기둥의 하위 목표

핵심 메카닉스 – 7P

1. 메카닉스와 목표 달성 방식
2. 월드 턴 1스텝
3. 스탯 변환과 데미지
4. 상성과 데미지 계열 분리
5. 포획과 파티
6. 혼과 전멸

구현 방식 – 14P

1. 모듈 구조
2. 기술 선택과 이유

구현 필요 요소 – 16P

1. 시스템, 데이터, 도구

구현 계획 – 17P

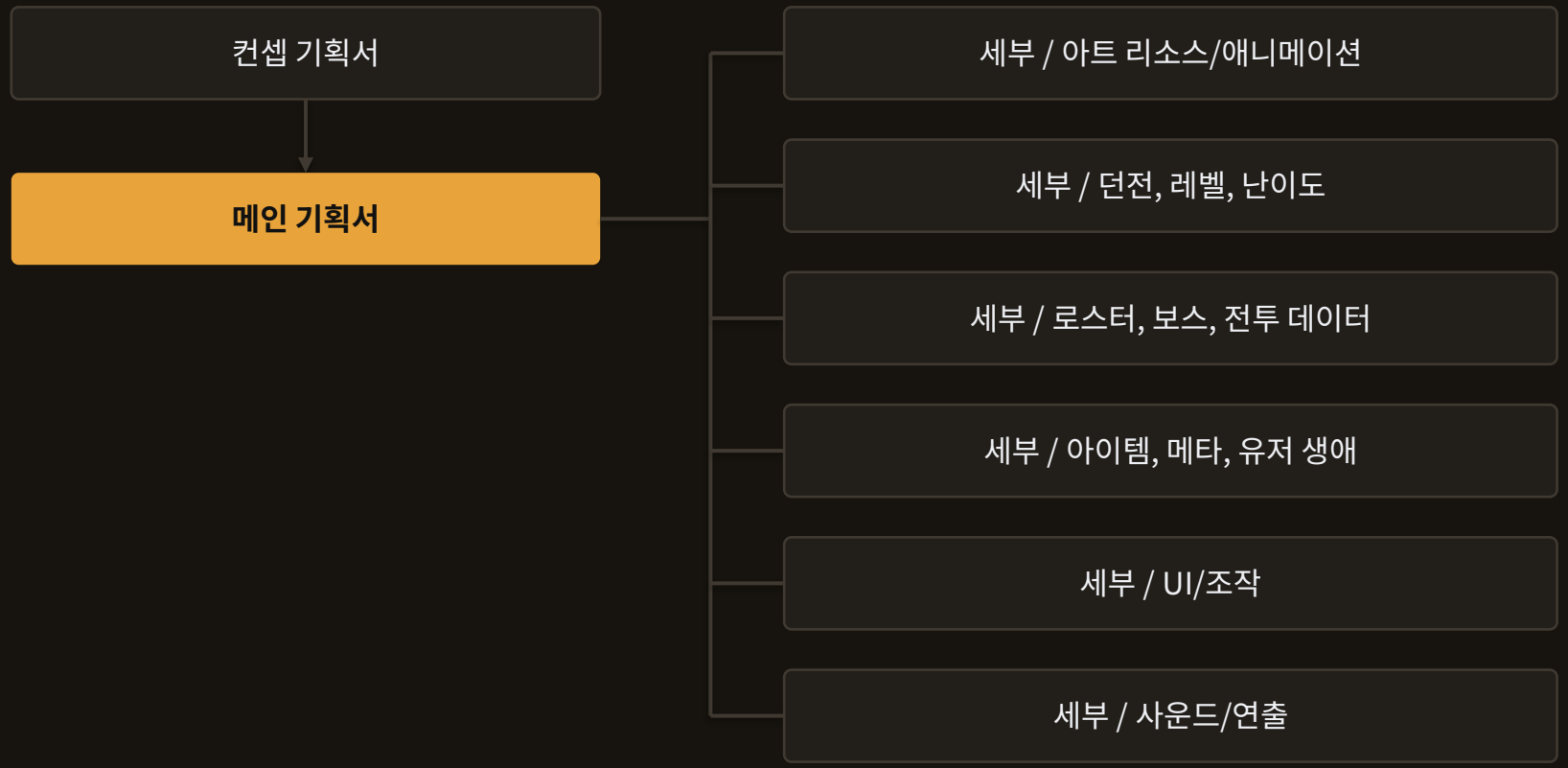
1. 구현 순서와 완료 기준

리소스 – 18P

개요 – 로그

순번	작성일	작성자	작성 내용	비고
1	2026-08-16	박도원	메인 기획서 Ver 0.2 작성	메카닉스/구현 구조 설계
2	2026-08-16	박도원	리소스 선행 제작 시작	정면 기준 이미지/규격 128px 전환
3	2026-08-16	박도원	엔진 구현 착수	M0 턴 엔진부터 순서대로 구현
4	2026-08-16	박도원	핵심 메카닉스 구현 완료	자체 검증 20/20 통과
5				
6				

개요 – 문서 계층



컨셉 → 메인 → 세부 순으로 상위 문서의 결정을 목표로 받아 구체화함.
강조 표시 = 이 문서

로캣: 잔향의 던전 / 메인 기획서

현 상황 – 컨셉 확정 시점

컨셉(목표/3대 기둥/루프)이 확정되고 규칙 수치는 아직 없는 상태임.

확정된 것

컨셉 기획서 Ver 0.1

목표 42체, 20층, 40~70분, 기둥 3개 확정

원본 엑셀

42체의 이름, 컨셉아트, 기술 이름/스코어 보유

근거 2026-08-13자 엑셀

리소스 기획서 v2.0

42체 지시표, 규격, 검수표를 담은 초안 보유

근거 32px 안은 형태가 뭉개져 128px 도트로 확정함

미정

턴/전투 수치

스탯 변환, 데미지, 포획 공식이 없음

엔진 구조

모듈 분할/데이터 형식/검증 방법이 없음

리소스 수량/규격

프레임 수/색 수/용량 한도가 미확정임

로캣: 잔향의 던전 / 메인 기획서

목표 재확인 – 목표와 기둥의 하위 목표

한 수씩 계산하는 수집 던전 (42체 / 20층 / 단일 HTML)

완전 동기 턴

목표: 입력 없으면 600회 해석에도 월드 정지

같은 시드/입력 = 같은 상태

적은 곧 아군

목표: 표시 확률과 실측 ±2%p

3인 파티/혼 15턴

데이터 주도

목표: 미정의 토큰 0/기획서/JSON 불일치 0

토큰 101종 사전

단일 HTML

목표: 외부 의존 0, 약 8MB 이하

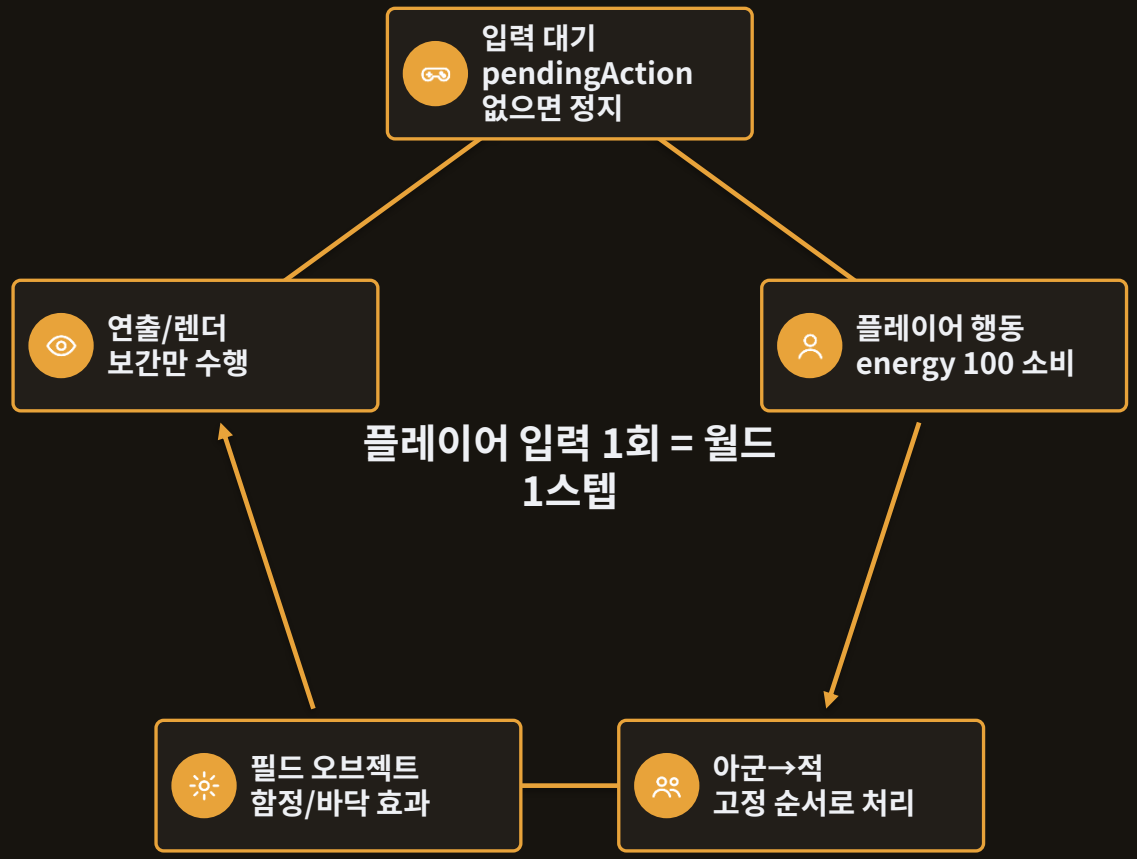
24색 감색

로캣: 잔향의 던전 / 메인 기획서

핵심 메카닉스 – 메카닉스와 목표 달성 방식

메카닉스	달성하는 기둥	방식	근거
에너지 스케줄러	완전 동기 턴	입력 소비 시점에만 월드 턴 +1	실시간/ATB는 '멈춘다'와 충돌
스탯, 데미지, 상성	데이터 주도	스코어 합 30을 공식으로 변환	42체를 동급으로 맞춰 개성만 남김
포획 확률	적은 곧 아군	확률 % 공개 + 고리/상태 보정	계산으로 읽히게 하려 함
혼(魂)/퍼머데스	적은 곧 아군	전투불능 15턴 유예 + 부활	긴장과 부활 수단을 함께 의미 있게 함
레벨, 각인, 메타	데이터 주도	레벨 하나로 난이도/성장 통일	티어 없이 강함을 일관되게 함

핵심 메카닉스 – 월드 턴 1스텝



3턴 주기

3턴마다 투지 +1/상태 감소/만복 −1

결정론

시드 xorshift RNG만 사용, 랜덤 정렬 금지

핵심 메카닉스 – 스탯 변환과 데미지

원본 스코어 s(3~9, 합 30)를 공식으로 변환하며 42체는 모두 동급임.

항목	공식	비고
HP / ATK	20+8s / 5+1.8s	레벨당 성장 base×0.09
DEF/MAG / SPD	2+1.6s / 40+10s	SPD는 성장 제외
데미지	ATK×POW/100×100/(100+DEF)×상성×VAR	VAR 0.90~1.10, 하한 1
보정	POW_SCALE 1.35	1:1 중앙값 8턴→약 6턴 목표
랭크	−6~+6	ATK/DEF/MAG ×(1+0.15r), SPD +10r

핵심 메카닉스 – 상성과 데미지 계열 분리

속성 상성 (캐릭터)

육 → 공 → 해 → 육

유효 ×1.5, 동속 ×1.0, 반감 ×0.7

의도 상성 격차를 약 2.2배로 둠

기호 병기 ▲●■

색맹 대응, 로그에 매 타격 표시

데미지 계열 (기술)

야성 / 신비 / 신성

참조 방어(DEF/MAG/평균)와 부가 규칙만 결정

근거 상성과 섞이면 가장 오해되기 쉬움

신성은 상성 ×1.0 고정

치명은 야성만 8% ×1.8

핵심 메카닉스 – 포획과 파티

1

대상 지정

확률 % 즉시 표시

P=0.32×고리×HP×상태×레벨

2

판정

흔들림 3회 후 확정

3

합류

HP 50%/투지 5로 파티 진입

4

만석 교체

확인 필수, 되돌리기 없음

핵심 메카닉스 – 혼과 전멸

1

전투불능

즉시 혼 상태 전환

2

15턴 유예

행동/피해 없이 남음

3

부활 수단

날숨/되살림 씨앗으로 HP 50%

4

전멸 판정

전원 혼/사망이면 회차 종료

핵심 메카닉스 – 레벨 곡선과 각인

항목	규칙	목표
필요 경험치	5×n^2.1 ×M(n), 16F↑ 가중	2F L5 / 10F L14 / 19F L20
적 레벨	1+층×0.95	도달 레벨이 적 레벨에 근접
심층 각인	8F~ 레벨 +2	HP ×1.19
엘리트 각인	12F~ 레벨 +4	HP ×1.52, 전 스탯 +1랭크
보스 각인	깊이 비례 +2~+6	HP ×3.54, 포획 ×0.25

구현 방식 – 모듈 구조

rocat.html (src 14파일 + 데이터 JSON)

시뮬레이션
역방향 의존 금지

- 10_core
- 20_world
- 50_action

전투/AI
토큰 인터프리터

- 40_combat
- 45_effects
- 60_ai

월드
BSP 던전/스폰

- 30_mapgen
- 55_spawn

표현/입력
Canvas2D + DOM

- 70_render
- 75_audio
- 80_ui
- 85_touch

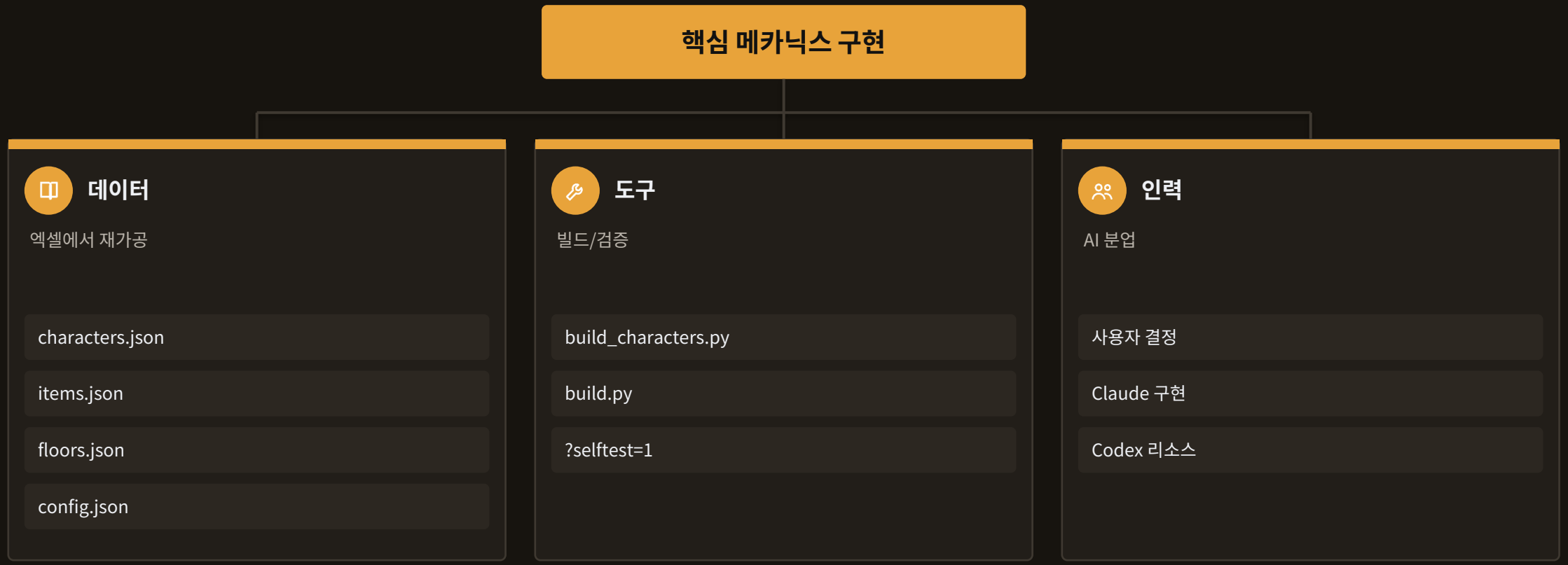
진입/검증
자체 검증 하네스

- 90_main
- 95_selftest

구현 방식 – 기술 선택과 이유

선택	대안	이유
자체 엔진(Canvas 2D+DOM)	외부 엔진	단일 HTML/외부 의존 0 목표
시드 RNG + lint	Math.random	재현 가능한 검증과 결정론
효과 토큰 문자열	기술별 코드	42체 기술을 JSON만으로 확장
BSP+MST+flood fill	수작업 맵	도달 불가 0을 자동 검증
WebAudio 절차 생성	오디오 파일	번들 용량을 늘리지 않으려 함

구현 필요 요소 – 시스템, 데이터, 도구



구현 계획 – 구현 순서와 완료 기준

1

M0 턴 엔진

도형 스프라이트로 30초 방치

적 0칸 이동

2

M1 던전

시드 100개 맵 생성

도달 불가 0

3

M2 전투

토큰 인터프리터 구현

1:1 4~6턴

4

M3 포획/AI

혼/아군 AI 6프로필

표시=실측 ±2%p

5

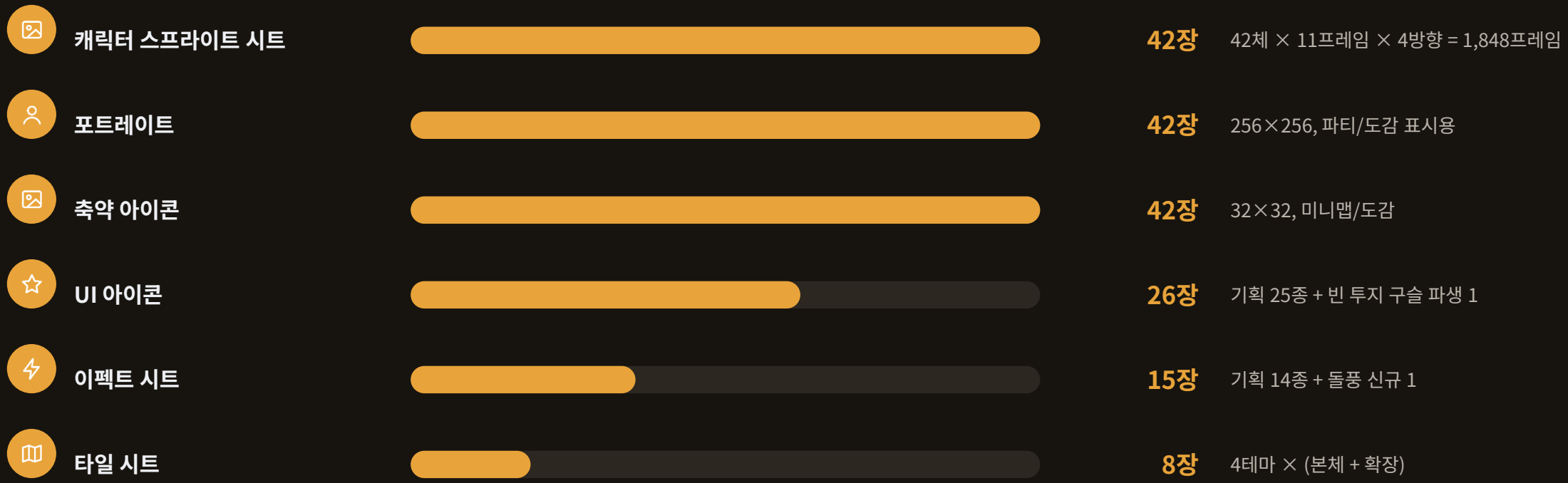
M4 성장

42체, 레벨, 각인, 보스

20F L20

리소스 – 수량 계획

수량은 규격 확정 후 실제 폴더에서 센 값이며 번들 용량 한도 안에서 정함.



리소스 – 규격

리소스	규격	파일명	근거
스프라이트	1408×512 / 11×4 / 24색+투명	spr_001~042.png	실측 1408×512 확인
포트레이트	256×256	por_001~042.png	고DPI 대비 축소 안 함
타일	256×256(64×64×16) + 확장 256×128	tile_<테마>[_ext].png	캐릭터 128px가 타일 64px 위로 솟는 구조
이펙트	192×192 프레임 가로 배열(4, 5, 6, 8)	fx_*.png	가산 합성, 소프트 알파 허용
공통	AA 금지 / 외곽선 2px / 접지선 y=116	asset_manifest.csv	24색은 용량 성립 조건

리소스 – 생산 파이프라인

1

생성

이미지 AI로 정면→측/후→시트

단계별 승인

2

정규화

배경 제거, 정렬, 접지선 고정

Python PIL

3

감색

24색/PNG-8, 원본은 별도 보존

quantize_assets.py

4

검수

기계 검수 후 눈 검수

verify_assets.py