GAME DESIGN PORTFOLIO

시스템을 숫자로 검증하는 기획자

김태준

시스템 · 밸런스 기획

ktj4060@gmail.com · hellobot99.github.io/game-design-portfolio

01 02 03

ABOUT

기획한 것을 직접 만들고, 숫자로 확인합니다

게임 서버를 직접 구현하다가 시스템·밸런스 기획으로 방향을 잡았습니다. 기획서로 끝내지 않고 플레이 가능한 프로토타입과 계산 시트까지 만들어서, 의도한 숫자가 실제로 나오는지 확인하는 방식으로 일합니다.

직접 플레이로 검증

영웅 7명의 클리어 시간을 직접 재면서 스킬 보정을 다섯 번 고쳐 목표 범위에 맞췄습니다.

프로그래머 관점

C++ IOCP 서버를 직접 구현해봐서, 데이터 구조와 예외 상황까지 생각하며 기획합니다.

AI로 빠르게 만들고 확인

프로토타입, 시뮬레이터, 시트를 AI(Claude Code)로 만들고 결과는 직접 확인합니다.

경북대학교 컴퓨터학부 (2027.07 졸업 예정) · 시바우라공업대학 교환학생 (2026.04 ~ 2027.01)

01

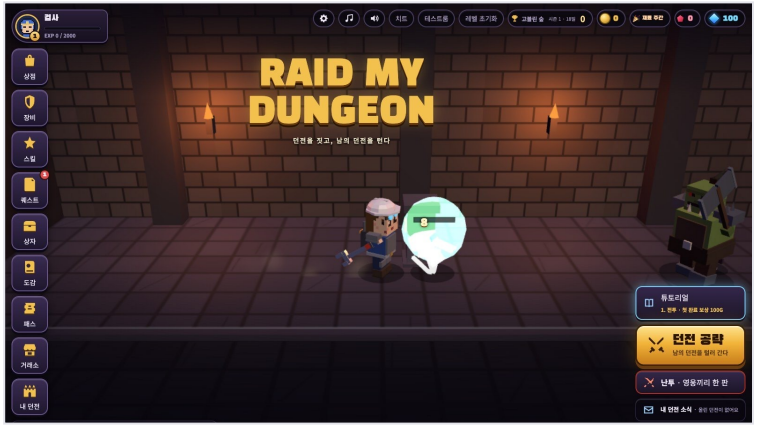
RAID MY DUNGEON

던전을 짓고,
남의 던전을 턴다

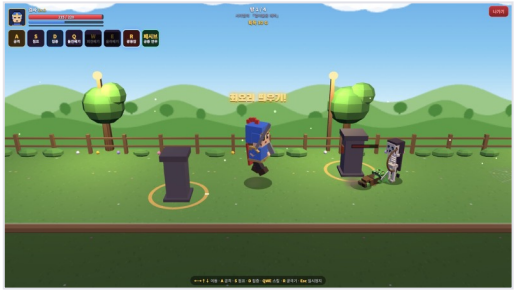
장르 액션 + 던전 빌딩
플랫폼 PC (웹 프로토타입)
기간 2026.07 ~
팀 5인
역할 기획 전담

던전을 짓고 남의 던전을 터는 비동기 PvP 게임

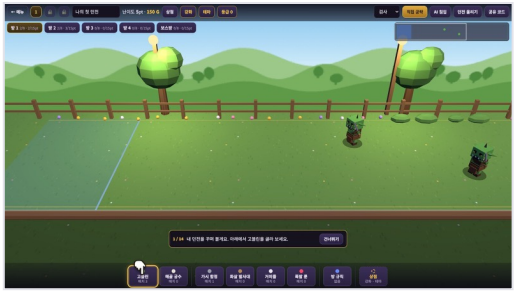
기획을 혼자 맡고 있습니다. 기획서를 시스템, 콘텐츠, 밸런스, 제작 네 갈래로 나눠 쓰고, 같은 수치로 돌아가는 웹 프로토타입과 밸런스 시트로 숫자를 확인했습니다. 클라이언트 2명, 모델링과 애니메이션 각 1명이 함께합니다.



웹 프로토타입 로비



남의 던전을 공략하는 화면



던전 편집기 (방마다 배치 예산)

19편
시스템 기획서

4편
밸런스 기획서와 엑셀

24건
이슈와 결정 기록

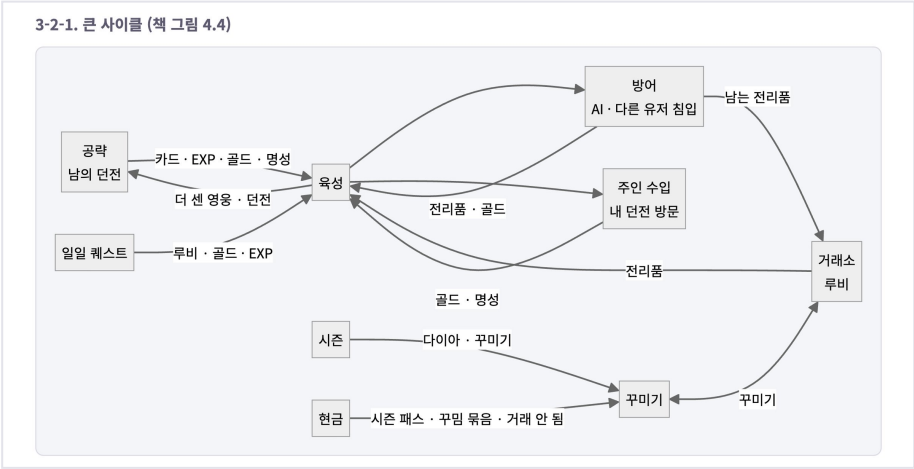
웹 기획서 웹 프로토타입

01

RAID MY DUNGEON

시스템 구조

플레이마다 얻는 자원을 하나씩 정해 두었습니다



경제 순환도 (책《게임 밸런스 수치 기획 바이블》 그림 4.4 형식)

거래는 거래소 한 곳에서만

무엇을 어떤 화폐로 얼마나 거래할지는 시스템이 정한다. 던전을 잘 지어 남은 전리품을 팔면 루비가 되니, 던전을 잘 지을수록 돈이 된다.

플레이별 중심 자원

공략	카드
방어	장비 재료
주인 수입	골드
일일 퀘스트	루비
시즌	꾸미기

횟수로 막지 않는다

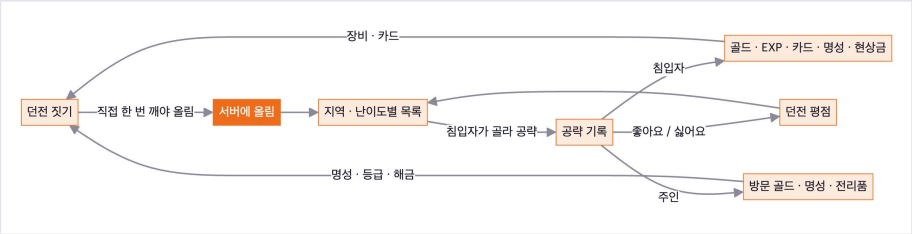
행동력이나 하루 횟수 대신, 한쪽만 하면 다른 쪽 자원을 못 얻게 했다

01

RAID MY DUNGEON

핵심 시스템
온라인 던전

남의 던전을 터는 규칙과 악용을 막는 장치



던전을 올리고, 털리고, 다시 짓는 순환

잘 꾸민 던전이 이긴다
수입은 방문 수보다 적정 난이도, 접전, 조합, 평가로

어려워도 사람이 온다
어려울수록 보상이 크고, 오래 안 깨진 던전엔 현상금

털려도 잃지 않는다
주인에겐 방문이 곧 이득이라 마음 편히 공개

막아야 할 것	규칙
아무도 못 깨는 던전	주인이 자기 영웅으로 **직접 한 번 깨야** 올릴 수 있다
너무 쉬운 던전으로 방문료 벌기	방문 골드는 적게, 수입의 중심은 **적정 난이도 명성** (클리어율 30~70%에서 가장 큼)
친구끼리 돌려 깨기	같은 침입자는 그 던전에서 **하루 한 번만** 보상
들어왔다 바로 나가서 싫어요	방 2개 이상 들어가야 평가할 수 있다
결과 위조, 자동 사냥	서버가 공략 기록을 검사한다 (시간과 처치 수가 던전 내용과 맞는지, 사람이 낼 수 있는 기록인지)
오프라인 기록을 몰아서 보상	서버가 끊기면 편집과 연습만, 보상은 연결됐을 때만

01

RAID MY DUNGEON

시스템
기획서

시스템 기획서는 개발자가 바로 옮길 수 있게 썼습니다

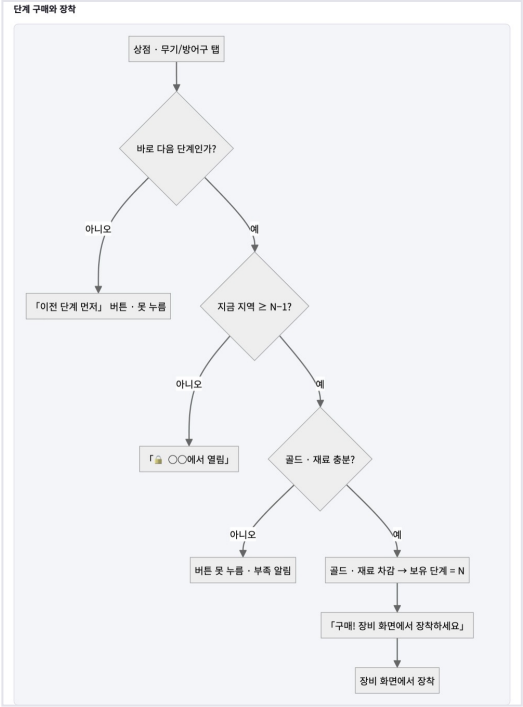
원칙 코드와 문서가 다르면 코드가 기준, 숫자의 원본은 CSV, 결정은 날짜와 이유를 함께 남긴다

절	장비 시스템 문서에서 쓴 것
1 목적	왜 있나, 어떤 시스템을 잇나
2 용어	단계 · 칸 강화 · 장비 재료
3 규칙	6단계 · 재료 6종 · 칸 강화 +0~10 · 능력치 공식
4 흐름	흐름도 3개 (구매 · 강화 · 재료 순환)
5 예외 처리	모자람 · 잠김 · 최대 · 지역 하락 · 예전 저장
6 UI	화면별 표시, 알림 문구
7 데이터	CSV 칼럼 · 저장 필드 · 코드 상수
8 연관 시스템	상점 · 재화 · 지역 · 온라인 던전
9 미정 사항	플레이 테스트로 볼 것

5. 예외 처리

상황	처리
골드가 모자람	구매 버튼을 못 누른다. 눌린 경우 「골드가 모자라요」 알림, 아무것도 차감하지 않는다
재료가 모자람	버튼을 못 누른다. 「○○이(가) 모자라요 (가진 수 / 필요 수)」
지역에 아직 못 닿음	카드에 🔒, 버튼에 「○○에서 열림」. 눌린 경우 「○○에 닿아야 살 수 있어요」
이전 단계를 안 삼	「이전 단계 먼저」 버튼, 못 누른다
칸 상한에 닿음	다음 칸 버튼이 「🔒 +N 은 「○○」에서」로 잠긴다
+10	「최대 강화」로 표시, 더 오르지 않는다
산 뒤 장착하지 않음	자동으로 갈아입히지 않는다. 알림으로 장비 화면을 안내한다 (§9 미정 1)
지역이 내려감	명성이 떨어져도 「가장 멀리 간 지역」은 그대로라서, 이미 열린 단계와 칸 상한은 닫히지 않는다
입어보기	미리보기에만 적용된다. 탭을 바꾸거나 무엇을 사면 원래 장착으로 돌아간다
예전 저장 (3단계 시절)	무기 강철 → 1, 룬 → 3, 전설 → 6 / 방어구 가죽 → 1, 강철 → 3, 미스릴 → 6. 예전 전리품 `loot` 는 녹슨 쇳조각으로 바꾼다

위는 시스템 문서 19편이 같이 쓰는 9개 절, 아래는 장비 문서의 예외 처리



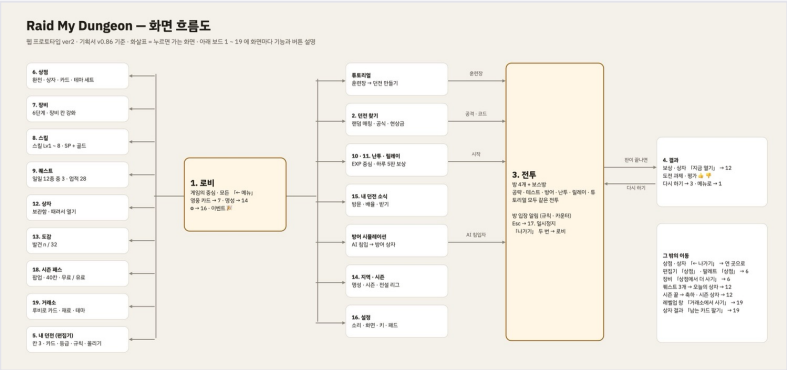
장비 문서의 흐름도 (단계 구매와 장착)

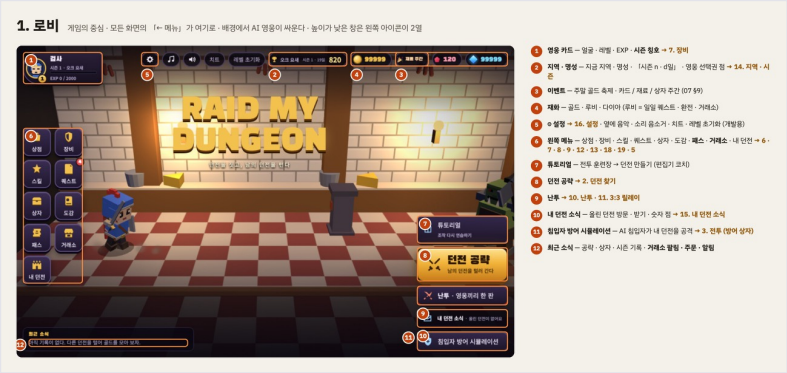
01

RAID MY DUNGEON

화면 흐름

화면마다 버튼의 기능과 이동할 곳을 적었습니다







웹 기획서 목차

화면 19개

실제 화면 위에 번호를 달고 기능과 이동할 화면을 적음

01

RAID MY DUNGEON

밸런스
방법

책의 다섯 단계를 시트와 측정으로 옮겼습니다

책의 단계	정한 것	엑셀 시트	확인 방법
1 준비	전투 리듬, 생존 주기, 허용 오차	기준	
2 전투 수치	속성과 공식, 직업 보정, 몬스터	속성·공식, 직업 밸런스, 몬스터 템플릿	**딜량 측정, AI 공략 측정**
3 육성	성장 분할, 육성 주기, 카드 레벨	성장, 육성 플랜	**스킬 딜 비중 측정 (24명)**
4 경제	화폐, 소비, 생산, 가치	소비 모델	**하루 플레이 경제 시뮬레이션**
5 검증	직접 플레이, 1대1, 실측	플레이 검증, 결정 기록	**테스트룸에서 직접 플레이**

원칙: **직접 플레이 기록이 먼저**, 평타는 손대지 않고 스킬 보정으로, 모자란 만큼의 **70%만** 고치고 다시 측정, **합계를 먼저 정하고 나눈다**, 힘은 팔지 않는다

플레이 검증 (실측) · AI가 추천 던전을 실제로 공략한 시간 (초)

클래스	던전	방 1	방 2	방 3	방 4	보스	전체	공략 성공
검사	고블린 소굴	5.9	5.6	7.7	9.7	39	77.7	3/3
궁수	고블린 소굴	10.6	9.2	8.9	9.9	41.8	90.5	3/3
마법사	고블린 소굴	9	8.5	8.1	8	33.2	80.2	3/3
도적	고블린 소굴	7.4	6.7	7.3	9.9	29.6	66.3	3/3
격투가	고블린 소굴	9.8	11.9	9.8	10.9	31.7	80.9	3/3
성기사	고블린 소굴	6.3	6.1	6.9	9.4	32.2	69.2	3/3
거너	고블린 소굴	8.1	9.4	7.1	8.1	39.6	83.9	3/3

출처: 프로토타입 자동 측정 (밸런스_공략측정.csv)

밸런스 시트의 플레이 검증 (봇이 추천 던전을 실제로 공략)

밸런스_설계.xlsx · 결정 기록 (일부)

날짜	결정	값	근거
2026-10-03	직업 설계 방향	제어 삼대장 — 모든 직업이 딜러, 가중 딜을 같게	3.3.1 직업 밸런스: 싱글 플레이 · 개인 경험 중…
2026-10-03	허용 오차 · 원거리 비율	±10% · 원거리도 근접과 같게 (1.0)	3.4.3 리듬 밸런스 — 추천값, 필요하면 조정
2026-10-03	마법사 평타 예외	평타 폭발 둘레 피해 70% → 40%	평타만으로 몰린 적을 다 때려 스킬을 낮춰도 다수 딜이…
2026-10-03	몬스터 수치	몬스터 HP ×2.5 · 보스 HP ×1.9 · 몬스터 · 보…	5.3.1 실측: 방 7초 · 보스 22초 → 13초 …
2026-10-03	성기사 2차	패시브 광휘의 불꽃 2 → 9 · 보스에게 2배 · Q·W·E…	플레이 피드백: 잡몹은 빠른데 보스전이 너무 깁 (AI…
2026-10-03	마법사 2차	스킬 보정 0.65 → 0.85 · 평타 12 · 13 · 1…	직접 플레이 클리어 시간: 다른 영웅 65~80초 · …
2026-10-03	마법사 3차	스킬 보정 0.85 → 1.0 · Q · W · E 마나 25…	2차 뒤 직접 플레이 87초 — 평타보다 스킬 위주(집…
2026-10-04	1차 7명 3차 (그로기 뒤)	스킬 보정 검사 1.85 → 1.6 · 궁수 1.0 → 1.2…	보스 그로기를 넣은 뒤 1레벨 직접 플레이 클리어 시간…

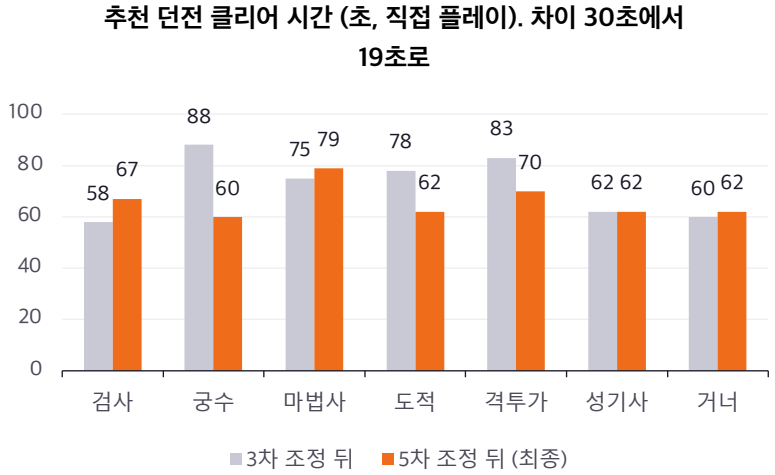
결정 기록 (보정값마다 날짜와 이유)

01

RAID MY DUNGEON

전투
밸런스

영웅 7명을 직접 플레이하며 다섯 번 맞췄습니다



최종 스킬 보정

검사	1.5
궁수	1.1
마법사	1.1
도적	1.3
격투가	1.0
성기사	0.45
거너	1.3

모든 직업이 딜러
혼자 깨는 게임이라 딜은 같게, 플레이 방식만 다르게

평타는 그대로
손맛이라 건드리지 않고 스킬 보정만 조정

70%만 반영
모자란 만큼의 70%만 고치고 다시 측정

사람 기록 우선
봇은 마법사 보호막을 못 살려서 직접 재며 103초에서 79초로

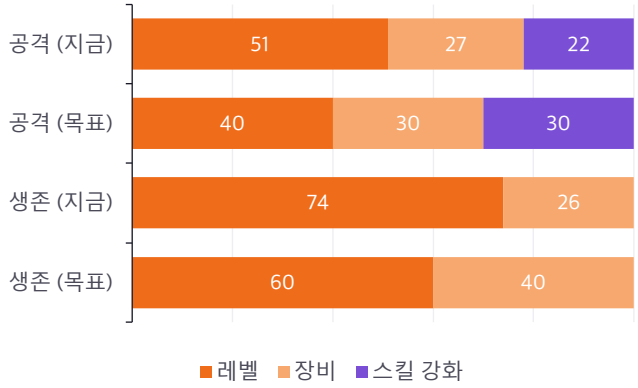
01

RAID MY DUNGEON

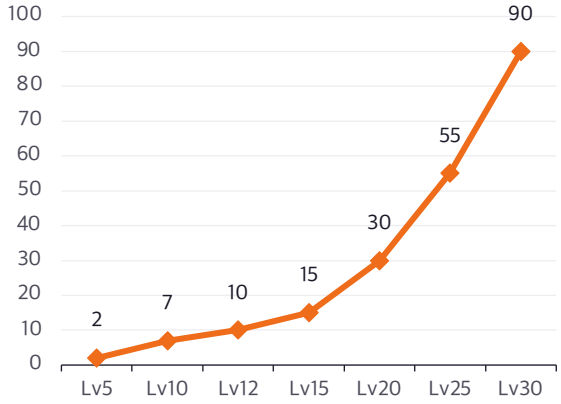
성장 밸런스

성장 총량을 먼저 정하고 경로마다 나눴습니다

경로별 성장 몫 (%)



레벨별 목표 도달 일수 (하루 30분~1시간)



왜 나눴나

레벨 몫이 크면 경험치가 비싼 후반에 성장이 멈춘다. 그래서 골드로 사는 장비와 스킬 몫을 늘렸다. 장비는 모든 영웅이 같이 써서 새 영웅도 빨리 큰다.

스킬 몫은 측정해서

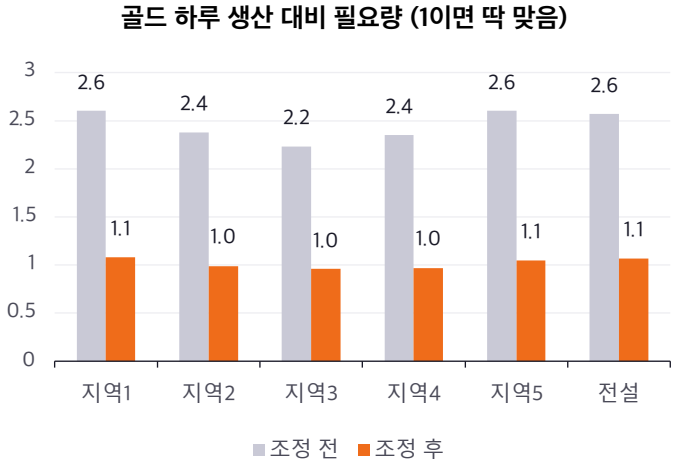
딜 중 스킬 강화가 붙는 몫을 영웅 24명 모두 재 보니 평균 57%. 그래서 스킬 칸마다 +14.3%로 정했다.

01

RAID MY DUNGEON

경제
밸런스

하루에 버는 양과 필요한 양을 맞췄습니다





다이아는 돈으로 팔지 않는다

유료 게임(Steam 4,900원) 안에 유료 화폐를 두지 않기로 했다. 돈으로는 시즌 패스와 꾸밈 묶음만 판다.

루비를 없애고 다이아로 거래하는 안도 봤지만, 값 단위가 너무 커져서 루비를 남겼다.

가격을 올림

보상 숫자는 그대로 두고 장비, 스킬, 카드 가격을 2.5배로

환전 한도

루비를 골드로 바꾸는 양은 하루 골드 생산만큼까지

가성비 톱니

한 단계 안에서는 떨어지고, 새 지역에서 다시 오르게

01

RAID MY DUNGEON

문제와 결정

찾은 문제와 정한 것을 문서로 남겼습니다

24건 찾은 문제

24건 정한 것

4갈래 시스템, 콘텐츠, 밸런스, 제작 문서

문제	정한 것
빈 던전 공유 코드로 보상을 무한히 받을 수 있음	같은 던전은 첫 클리어만 보상. 이후 **20pt 이상 + 보스**가 있어야 보상
연계 배율이 곱으로 쌓여 특정 순서만 피해가 폭증	연계 쪽 배율을 모두 곱한 값에 **상한 2.5배**
스킬 강화가 항목 고르기라 「피해부터」가 정답	**직선 강화**로 바꿈. 스킬마다 Lv1~8, 레벨마다 오르는 것 고정
레벨 몫이 커서 후반에 성장이 멈춘 느낌	성장 몫을 공격 레벨 40 · 무기 30 · 스킬 30, 생존 레벨 60 · 방어구 40으로 나눔
골드가 필요량의 2.5~3배 나옴	보상 숫자는 두고 **가격을 2.5배**로 (보상 받는 맛은 유지)
유료 게임 안에 유료 화폐를 돌지	**다이아를 돈으로 팔지 않음**. 루비를 없애는 안도 봤지만 값 단위가 너무 커져서 유지

보상을 무한히 받는 구멍은 공유 코드와 방어 시뮬레이션을 직접 돌려 보다 찾았습니다. 정할 때마다 날짜, 이유, 바뀐 문서를 적어 두어서 팀원이 문서만 보고도 왜 이렇게 됐는지 알 수 있습니다.

02

Netrunner's Playlist

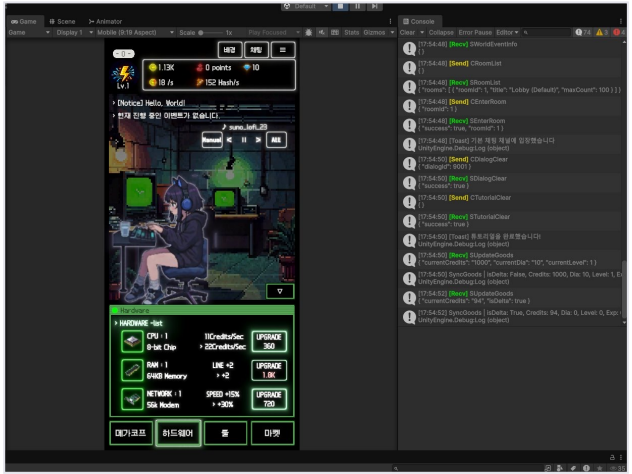
사이버펑크
방치형 게임

장르 방치형 성장
플랫폼 모바일 (Unity)
기간 2026.01 ~
팀 2인
역할 기획 + 서버(C++)

해커가 기업 서버를 뚫는 방치형 게임

기획과 서버를 맡았습니다. 기획서를 시스템과 밸런스 중심으로 다시 정리하고, 기존 데이터에서 문제를 찾아 공식과 수치를 새로 설계한 뒤 시뮬레이션으로 확인했습니다.





실제 게임 화면 (왼쪽은 클라이언트, 오른쪽은 직접 만든 서버의 패킷 로그)

서버 코드 (GitHub)

16건

코드와 데이터를 대조해 찾은 수치 문제

12시간

이전 지점 복귀 시간
(2회차는 48시간)

10회차

30일, 100티어까지 시뮬레이션으로 확인

02

Netrunner's Playlist

시스템 구조

성장 세 축이 돈(CPS)과 보스(DPS)로 나뉘어 들어갑니다

해킹 자동 + 클릭 → **크레딧 · 경험치** CPS · 오프라인 80% → **하드웨어 · 툴** CPU · RAM · NET → **메가코프 돌파** DPS 체크 20초 → **포맷(환생)** 악명 획득

성장 축	포맷 시	주는 재미	CPS(돈)	DPS(보스)
하드웨어 CPU · RAM · NET	1레벨로 초기화	숫자가 폭증하는 쾌감	○	○
해킹 툴 100종 · 슬롯 3~7	장착만 해제, 영구 소장	갈아 끼우는 전략	○	○
레벨 1 ~ 100	초기화 (시작 레벨 스킬)	단기 목표		○ 1.04^Lv
악명 + 악명 스킬 100	영구 누적	다음 회차를 빠르게		○ 회차마다 커지는 배율

보스 도전 → **20초 동안** 매초 HP −DPS · 클릭 −DPS 5% → **HP 0 → 다음 레이어** 최초 돌파 보상 · 점령 수입 교체 / **시간 초과 → 파밍 모드** 이전 단계에서 수입 쌓고 재도전

서버 검증	방식
보스 피해 조작	보고 피해 ≤ DPS × (경과 + 1) × 1.2
클릭 매크로	5초 배치 150회 초과 거절
오프라인 시간 조작	서버 시각 기준 · 12시간 상한

보스전 한 판의 흐름과 서버 검증

포맷하면
하드웨어와 레벨은 처음부터,
툴과 악명은 남는다
돈은 다시 벌지만 보스는 빨리 깬다

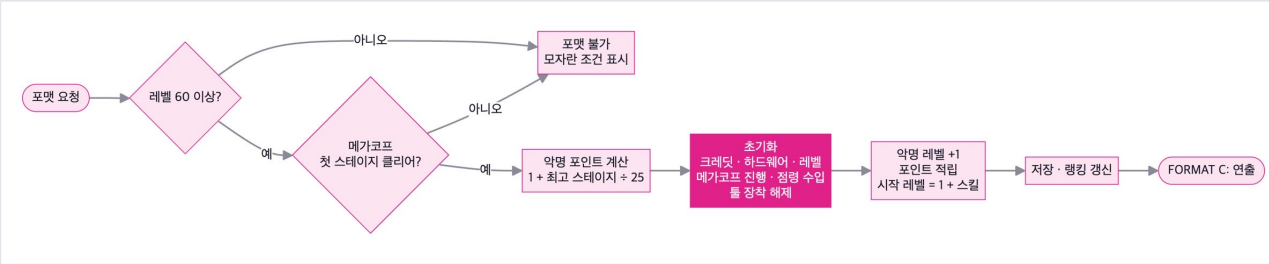
툴 슬롯은 레벨로 열림
포맷 직후엔 좋은 툴도 다 못 껴서
레벨을 다시 올릴 이유가 생긴다

02

Netrunner's Playlist

시스템 설계
포맷

포맷(환생) 때 지울 것과 남길 것을 나눴습니다



초기화	남는 것
크레딧 0	해킹 툴 (장착만 풀림)
하드웨어 1레벨	다이아
레벨 1 (+시작 레벨 스킬), 경험치 0	악명 레벨 · 포인트 · 스킬
메가코프 진행 · 점령 수입	업적

포맷 조건과 포인트 공식, 초기화 목록은 서버 코드 기준

나눈 이유

하드웨어는 다시 올리는 재미가 있어서 지운다. 툴은 모은 게 사라지면 허무해서 소장은 남기고 장착만 푼다.

예외 처리

조건이 모자라면 어떤 조건이 모자란지 보여 준다. 시작 레벨은 1에 악명 스킬만큼 더해, 툴 슬롯이 바로 열리게 한다.

02

Netrunner's Playlist

데이터 설계

데이터는 기획 표, 저장 데이터, 밸런스 로그로 나눴습니다

기획 데이터 (시트 → 서버 CSV)

- **megacorp** 7001~7500
  Tier · Layer · Theme · Maxhealth · Reward · CPS
- **tool** 3001~3100
  effect · value · Method · Price · Grade
- **skilltree** (악명 스킬) 5001~5100
  Effect · Value · Cost
- **achievement** 6001~6100
  EventTrigger · CheckKey · TargetValue · Reward
- **CPU · RAM · Network** 1~ · 1001~ · 2001~
  Level · Name · Visual State
- **exp · dailydungeon · worldevent · cps_boost**
  8001~

유저 저장 데이터 (DB)

- **users**
  credits · dia · level · exp · infamy_level · infamy_points · last_logout_time
- **user_hardware**
  cpu_level · ram_level · network_level
- **user_tools / user_equipped_tools**
  tool_id → tool.ID · slot_index
- **user_megacorp**
  current_stage_id · max_cleared_id → megacorp.ID
- **user_skills / user_achievements**
  skill_id → skilltree.ID

밸런스 로그 (분석용)

- **balance_cps_log**
  레벨별 최종 CPS · 배율
- **balance_warning_log**
  효율 상한 · 소프트캡 도달
- **player_levelup_log**
  레벨업 시점 · 보상
- **megacorp_clear_log**
  클리어 시간 · DPS · 보스 HP
- **tool_transaction_log · credit_tracking_log**
  툴 획득 · 크레딧 흐름

ID 앞자리로 종류를 나눠 **한 번호만 보고 어느 표인지** 알 수 있게 했다. 유저 데이터는 기획 데이터의 ID만 들고, 수치는 늘 표에서 읽는다 (표만 고치면 전체 반영)

서버 저장 구조와 로그 테이블은 직접 설계하고 구현했습니다. 로그는 레벨별 CPS와 상한 도달을 모아 숫자가 의도대로 가는지 보는 데 씁니다.

02

Netrunner's Playlist

공식 ·
서버 구현

같은 하드웨어를 돈과 보스에 다른 계수로 씁니다

CPS 평소 해킹 → 크레딧

```
11·CPU × (1 + RAM/2) × (1 + 0.15·NET)
× (1 + 크레딧 보너스, 상한 +300%)
× (1 + 0.5% × 점령 티어)
전체 배율 10배 넘으면 로그로 증가
오프라인: × 0.8 · 최대 12시간
```

DPS 보스전 20초 → 보스 HP

```
120·CPU × (1 + 0.1·RAM) × (1 + 0.05·NET)
× 1.04 ^ (레벨 − 1)
× 악명 배율 (×1.25, ×1.55, ×1.85 …)
× (1 + 섹터 피해) × (1 + 보스 피해)
클릭 1회 = DPS × 5%
```

부품마다 계수와 비용을 다르게 둔 이유: 셋 다 같은 때 같은 돈으로 올리면 고르는 재미가 없어서, 실제 장비 가격을 참고해 부품마다 느낌을 다르게 했다. 레벨과 악명은 **DPS에만** 곱해서, 포맷하면 돈은 처음부터지만 보스는 빨리 깬다

```cpp
// User::CalculateCPS() · 서버 (C++)
double toolCpuEff = toolBonus.hwCpuEff;
if (toolCpuEff > cpuCap) {            // 툴 효율 상한
    LogManager::Instance().LogBalanceWarning(playerId_, level_,
        "CPU_EFF_CAP", toolCpuEff, cpuCap, ...);
    toolCpuEff = cpuCap;
}
...
if (totalMultiplier > multiplierCap) {  // 10배 이후 로그 증가
    totalMultiplier = multiplierCap
        + std::log10(totalMultiplier − (multiplierCap − 1.0));
    LogManager::Instance().LogBalanceWarning(..., "LOG_SCALING", ...);
}
```

직접 구현한 서버의 CPS 계산 (상한에 닿으면 밸런스 로그를 남김)

기획자가 숫자를 확인하는 장치

툴 효율과 크레딧 보너스에 상한, 전체 배율은 10배부터 완만하게

상한에 닿을 때마다 **balance_warning_log**

레벨별 평균 CPS를 뽑아 보는 분석 쿼리

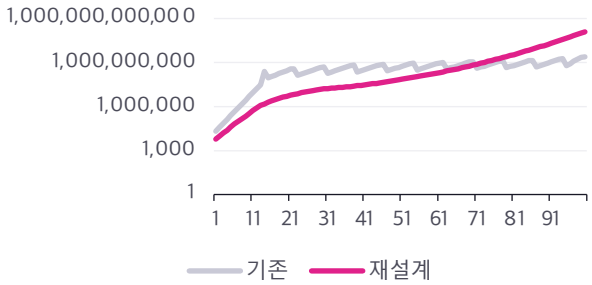
02

Netrunner's Playlist

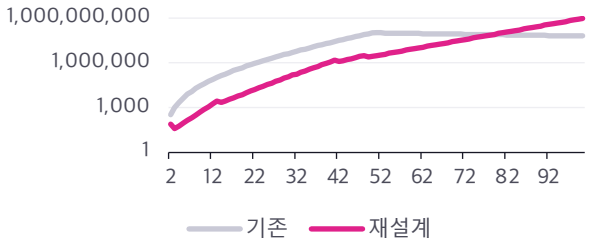
기존 데이터 진단

코드와 데이터를 대조해 수치 문제 16건을 찾았습니다

티어별 보스 HP (로그 축). 기존은 톱니 모양



레벨별 필요 경험치 (로그 축). 기존은 51부터 감소



기존 데이터의 문제	재설계
경험치: 첫 포맷까지 84시간, 51레벨부터 필요량이 **줄어듦**	목표 시간에서 역산, 계속 증가
레벨업 보상: 레벨마다 ×1.4라 Lv100에 **7.2경**	그 레벨 CPS 3분치
악명 스킬 RAM_ADD 3,000으로 DPS가 수천 배 (DPS엔 상한 없음)	ADD는 하드웨어 레벨 +1로
보스 HP: 8티어마다 떨어지는 톱니, 14티어만 튐	목표 DPS × 20초 × 레이어 계수
점령 수입: 100티어도 후반 CPS의 0.3%	티어당 CPS +0.5%
악명 스킬 총량: 다 찍으려면 포맷 약 400회	약 15회
툴 가격: 희귀 799억, 영웅 800조	해금 시점 CPS 1 · 3 · 8시간치

서버 코드와 데이터 테이블을 대조해 찾은 16건 중 일부

찾은 방법

서버 공식과 데이터 시트를 직접 읽고, 목표 CPS로 레벨 도달 시간을 계산해 봄

02

Netrunner's Playlist

수치 재설계

목표 플레이 시간에서 거꾸로 수치를 정했습니다

기준값 > **성장 구축** > **성장 분할** > **세분화** > **검증**

하루 2번 20분
1회차 80시간, 30티어

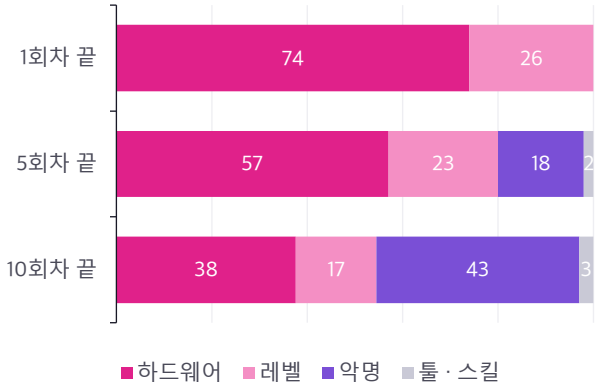
접속마다 목표 DPS

하드웨어, 레벨,
악명, 툴

데이터 표 8종

10회차
시뮬레이션

DPS 성장이 어디서 나오나 (%)



한 회차 안에서는 하드웨어가, 회차를 넘어서는 악명이 성장을 맡는다

바뀐 공식

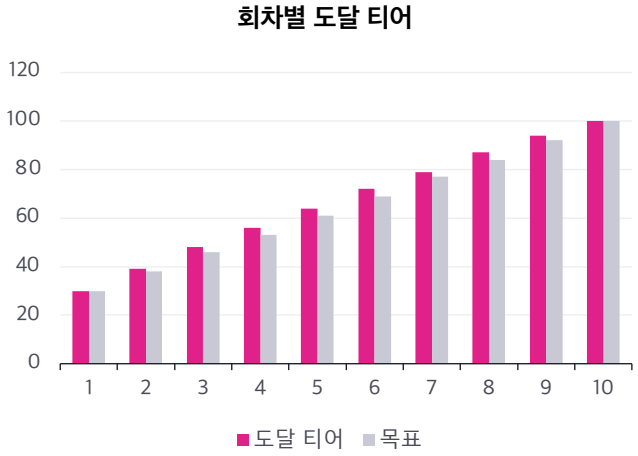
하드웨어 비용	1만, 3만, 2만 × 1.3^Lv
레벨 계수	1.04^(Lv-1)
악명 배율	×1.25, ×1.55, ×1.85 순으로 커짐
보스 HP	목표 DPS × 20초 × 레이어 계수
보상 · 가격	CPS 몇 분 · 몇 시간치
오프라인	80% · 12시간 (오프라인 위주)

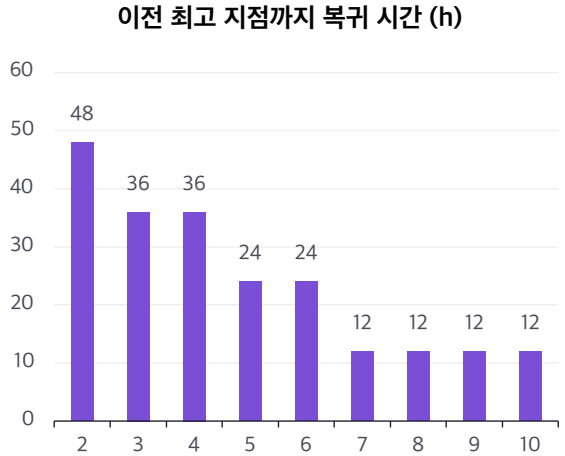
02

Netrunner's Playlist

시뮬레이션
검증

30일 동안 포맷 10번을 시뮬레이션으로 확인했습니다





1회차

72시간에 30티어, Lv60
첫 포맷 조건을 채움

복귀

48, 36, 24, 12시간으로
회차마다 짧아짐

10회차

100티어 도달
악명 누적 ×1,936

수입

오프라인이 86%
방치 중심 설계

가정: 하루 2번(오전 8시, 오후 8시) 20분 접속, 접속 중 초당 2회 클릭, 구매와 보스 도전은 접속 중에만. 툴과 스킬은 회차마다 평균 효과(CPS ×1.1, DPS ×1.06, 시작 레벨 +3)로 계산

02

Netrunner's Playlist

밸런스 시트

기준값 몇 개로 500개 스테이지가 계산되는 시트

Netrunner's Playlist 밸런스 데이터 (초안 v3)

시스템 구조는 그대로, 공식과 수치만 재설계 · 책 「게임 밸런스 수치 기획 바이블」 성장 구축 → 분할 → 세분화 → 검증 순서

1. 기준값 (미션 0)

항목	값
하루 접속	2번 × 20분
방향	오프라인 위주 (효율 80%, 상한 12시간)
1회차	80시간 안에 30티어 · Lv60 → 첫 포맷
2회차부터	이전 최고 지점 복귀 + 신규 진행, 복귀 시간은 점점 짧게
엔드게임	30일 · 포맷 10번 · 100티어

2. 바뀐 공식 · 수치

항목	기존	초안 v3	해결한 문제
하드웨어 비용	300 · 1,500 · 600 × 1.2^Lv	1만 · 3만 · 2만 × 1.3^Lv	1회차 끝 하드웨어 40레벨대
레벨 계수 (DPS)	1 + 0.1Lv + 0.2·ln(경험치)	1.04^(Lv-1)	레벨 · 경험치 이중 계산
악명 배율 (DPS)	1 + 0.5 × 악명	×1.25, ×1.55, ×1.85 … 누적	회차가 갈수록 복귀가 빨라짐
경험치 표	51레벨부터 필요량 급소	1회차 끝 Lv60, 계속 증가	경험치 과다 · 역전
보스 HP	손으로 입력 (8티어마다 튐니)	목표 DPS × 20초 × 레이어 계수	HP 튐니 · 섹터 무작위
레벨업 보상	×1.4씩 (Lv100 7.2경), 의지금	그 레벨 CPS 3분치	경제 붕괴 위험
툴 가격	최저 799억 · 영웅 800조	해금 시점 CPS 1 · 3 · 8시간치	가격 폭
점령 수입	고정 CPS (후반 0.3%)	티어당 CPS +0.5%	후반 체감 없음
악명 스킬	ADD 3,000 (DPS 폭주), 전부 400회	ADD = 하드웨어 레벨 ×1, 전부 약 15회	DPS 폭주 · 장기 목표 과다
오프라인	50% · 12시간	80% · 12시간	오프라인 위주 방향

3. 검증: 1회차 (세션 끝 기준)

경과 (시간)	티어	레벨	CPU	RAM	NET	CPS	DPS
0.3	5	12	9	5	5	622	3,117
12.3	12	30	25	21	22	14,415	60,907
24.3	18	40	34	30	30	35,874	188,348
36.3	22	47	38	34	35	52,198	335,184
48.3	25	53	41	36	37	63,143	495,794
60.3	28	57	42	38	39	72,155	641,742
72.3	30	60	44	39	40	79,872	785,088

요약 시트 (기준값, 바뀐 공식, 1회차 검증 표, 옆에 차트 5개)

기준값 (노란 칸을 바꾸면 다른 시트가 다시 계산됨)

공식과 수치 초안 v3 · 시뮬레이션: sim/np_sim.py

하드웨어		
CPU 기본가	10,000	RAM ×3 · NET ×2
RAM 기본가	30,000	
NET 기본가	20,000	
비용 배수 (레벨당)	1.30	Lv → Lv+1 비용 = 기본가 × 배수^Lv
전투		
레벨 계수 밑 (DPS × 밑^(Lv-1))	1.04	
보스 제한 시간 (초)	20	
레이어 1 계수	0.30	
레이어 2 계수	0.45	
레이어 3 계수	0.60	
레이어 4 계수	0.80	
레이어 5 계수 (보스)	1.00	
31티어 이후 티어당 성장 시작값	1.050	31티어 ×1.05 → 99티어 약 ×1.23
31티어 이후 성장 가속 (티어마다)	0.0023	
악명		
악명 1레벨 배율	1.25	악명 k레벨 배율 = 1.25 + 0.3 × (k-1)
레벨마다 배율 증가	0.30	
경제		
오프라인 효율	80.0%	오프라인 위주 방향
오프라인 상한 (시간)	12	
점령 수입 (티어당 CPS %)	0.5%	30티어 +15%, 100티어 +50%

기준값 시트 (노란 칸을 바꾸면 전부 다시 계산)

메가코프 500 스테이지

HP = 목표 DPS × 제한 시간 × 레이어 계수 · 점령 수입은 CPS %로 (교체 방식 유지)

ID	이름	티어	산업 분야	레이어	목표 DPS	HP	최초 돌파 보상	점령 수입 (CPS +%)
7001	NovaShop	1	Finance	1	300	1,800	240	0.1%
7002	NovaShop	1	Finance	2	300	2,700	240	0.2%
7003	NovaShop	1	Finance	3	300	3,600	240	0.3%
7004	NovaShop	1	Finance	4	300	4,800	240	0.4%
7005	NovaShop	1	Finance	5	300	6,000	1,440	0.5%
7006	TechSphere	2	Finance	1	479	2,874	520	0.6%
7007	TechSphere	2	Finance	2	479	4,311	520	0.7%
7008	TechSphere	2	Finance	3	479	5,748	520	0.8%
7009	TechSphere	2	Finance	4	479	7,664	520	0.9%

메가코프 시트 500행 (모두 수식)

시트 10개: 요약, 기준값, 하드웨어, 티어 곡선, 메가코프, 경험치, 악명, 악명 스킬, 툴, 일일 던전

03

메이플스토리 스타포스

강화 시스템
역기획

대상 상용 라이브 게임
범위 스타포스 강화
형식 개발 전달용 기획서

메이플스토리 스타포스 강화 역기획

작성 중

1 규칙: 강화 결과 · 결과를 바꾸는 요소 · 예외
2 플로우 차트: 강화 1회 흐름과 분기
3 데이터 테이블 설계: 컬럼 · 키 · 실제 값
4 수치 분석: 목표 성까지 기대 비용 · 불운 상위 10%
5 다른 게임 비교와 개선안

HOW I WORK

AI로 빠르게 만들고, 판단과 검증은 직접 했습니다

1 기획 결정 직접 → 2 구현 · 반복 작업 AI → 3 플레이 · 수치 확인 직접 → 4 기획서 · 기록 반영 직접



RaidMyDungeon

기획 수치로 바로 도는 프로토타입

AI 전투 · 던전 편집기 · 로비 코드, 점검 스크립트

직접 규칙과 수치 결정, 영웅 7명을 직접 플레이하며 다섯 번 보정



Netrunner's Playlist

10회차 진행 시뮬레이션과 시트

AI Python 시뮬레이터, 수식이 연결된 엑셀 10개 시트

직접 접속 패턴과 목표(첫 포맷 72시간, 10회차 티어 100) 설정, 결과를 보고 공식 수정



기존 데이터의 문제	재설계
경험치: 첫 포맷까지 84시간, 51레벨부터 필요량이 **줄어듦**	목표 시간에서 역산, 계속 증가
레벨업 보상: 레벨마다 ×1.4라 Lv100에 **7.2경**	그 레벨 CPS 3분치
악영 스킬 RAM_ADD 3,000으로 DPS가 수천 배 (DPS엔 상한 없음)	ADD는 하드웨어 레벨 +1로
보스 HP: 8티어마다 밀어지는 톰니, 14티어만 튐	목표 DPS × 20초 × 레이어 계
점령 수입: 100티어도 후반 CPS의 0.3%	티어당 CPS +0.5%
악영 스킬 총량: 다 찍으려면 포맷 약 400회	약 15회
톰 가격: 희귀 799억, 영웅 800조	해금 시점 CPS 1 · 3 · 8시간치

서버 코드와 데이터 테이블을 대조해 찾은 16건 중 일부

Netrunner's Playlist

코드와 기획서가 어긋난 곳 찾기

AI 서버 코드 · 데이터 테이블과 기존 기획서 대조

직접 찾은 16건의 심각도를 나누고 무엇을 어떻게 고칠지 결정

그 외

붕괴: 스타레일 전투 밸런스 역기획

대미지 공식 · 성장 비중 역산, 부옵션 가중치 개선안을 확률로 검증

The Last Harbor (개인 · 진행 중)

대항해시대 무역 시뮬레이션, 데이터 CSV 13종과 페이스 봇으로 검증

밸런스 수치 시각화 툴

기준 스탯부터 경험치 곡선까지 이어지는 웹 도구, 몬테카를로 · 가챠 천장 부록

온라인 경매장 게임 · 좀비 텍스트 어드벤처

DB · 시스템프로그래밍 팀 프로젝트



감사합니다

Email
ktj4060@gmail.com

Portfolio
hellobot99.github.io/game-design-portfolio

GitHub
github.com/Hellobot99