

게임 UI/UX, 유저 플로우에서 인엔진 출시 화면까지.

언리얼 엔진 AAA 라이브 서비스 10년+ 시니어 UI/UX 디자이너 — 블레이드&소울 2 · 리니지 W/2M · 불소 NEO · 배틀 크러쉬 (PC · 콘솔 · 모바일). 문제 정의와 UX 플로우에서 Figma 와이어프레임·비주얼 디자인(Photoshop · Illustrator)을 거쳐, 인엔진 프론트엔드 구현(UMG)과 HTML5/CSS까지 한 기능을 끝까지 책임집니다. 제 차별점은 폭(range)입니다 — VFX와 테크니컬 아트까지 갖춘 UI/UX 디자이너로, 플로우를 설계하고 모션을 입히고 셰이더·이펙트를 만들고 인엔진에서 직접 화면을 출시합니다. 한 사람이 비주얼 전체 스택을 책임집니다.

6

AAA 라이브 타이틀 · UI/UX 참여

10+

년 · 게임 UI/UX 경력

PC·콘·모바

멀티플랫폼 UI 경험

플로우 → 출시

기획 → 와이어프레임 → 인엔진, 끝까지

라이브 AAA 작업 보기

▶ 게임 라이브 데모

VFX · TA →

* 구현까지 하는 디자이너입니다. 이 페이지의 UI 모션은 외부 영상이 아니라 자체 제작 플립북 엔진이 실시간 재생하며, 플레이 가능한 게임들은 실제 프론트엔드 빌드입니다. 테크니컬 아트·VFX 심화는 맨 끝 VFX·TA 섹션에 — 설계한 것을 직접 만든다는 증거로 담았습니다.

유저 플로우에서 출시 화면까지 — 플레이어가 느끼는 품질

UI/UX 디자인은 결국 한 가지 질문에 답하는 일입니다. "이 순간이 플레이어에게 어떻게 느껴지는가?" 저는 그 답을 플로우·와이어프레임·비주얼로 설계하고, 인엔진에서 직접 구현합니다.

UI/UX 디자인

UX → SHIPPED UI

문제 정의에서 유저 플로우·와이어프레임·비주얼까지, 그리고 인엔진 구현까지 — 한 기능을 끝까지 책임집니다.

- ▶ 유저 플로우 · 정보구조 · 화면 상태 설계 (**Figma**)
- ▶ 라이브 대작 **UMG UI 디자인·구현** — 불소2 · 불소 NEO · 리니지 W/2M · 배틀 크러쉬
- ▶ 멀티플랫폼(PC·콘솔·모바일) — **게임패드 포커스 내비 · 타이틀 세이프 영역**
- ▶ 비주얼 — **Photoshop · Illustrator** 컴포넌트 · 디자인 토큰 시스템

UI 모션 디자인

UI MOTION DESIGN

버튼 하나의 눌림부터 보상 획득의 카타르시스까지, UI 위 모든 움직임을 연출합니다 — 이 페이지에서 실시간으로 확인됩니다.

- ▶ **언리얼 UMG** 라이브 UI 연출 — 강화 · 메뉴 · 씬 트랜지션
- ▶ **UIParticle(uGUI) · DOTween** 재화 비행 · 획득 연출
- ▶ UI 전용 커스텀 셰이더 (**Grayscale** , **RGBMask Dissolve**)
- ▶ CSS/웹 HUD 모션 — 체력바 트랜지션, 레벨업 카드 연출 (HTML5)

UI + VFX + 테크니컬 아트 — 한 사람이

THE FULL VISUAL STACK

진짜 차별점은 폭입니다 — 화면만 그리는 게 아니라 **모션을 입히고, 셰이더·VFX를 만들고, 인엔진에서 직접 구현합니다.** UI·모션·테크니컬 아트를 한 손으로 커버하는 드문 디자이너입니다.

- ▶ **인엔진 프론트엔드** — 언리얼 UMG & HTML5/CSS, 동작하는 화면으로 핸드오프
- ▶ **VFX · 셰이더 제작** — 나이가가라/캐스케이드 스킬 FX · UI 전용 셰이더 · 마스터 머티리얼 → 인스턴스
- ▶ **실시간 UI 모션** — 인터랙션의 손맛 (이 페이지에서 검증 가능)

라이브 AAA UI/UX — 출시 · 멀티플랫폼 · 인엔진

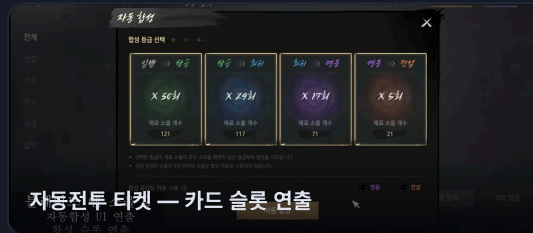
모두 언리얼 엔진 기반 AAA 라이브 서비스입니다. 중심은 **UMG UI 디자인·연출** — 블레이드&소울 2 · 블소 NEO · 리니지 W/2M · 배틀 크러쉬의 강화·메뉴·HUD를 직접 설계·구현했습니다. **세븐나이트 2는 VFX(스킬 연출)로 참여한 작업**으로, 전투 중 HUD 시인성과 직결되는 실시간 연출 역량을 보여주기 위해 함께 했습니다. 아래 캡처는 본인 작업 소릴에서 발췌했으며, 각 카드의 버튼으로 영상 전체를 볼 수 있습니다.

★ 대표 작업 — UI 연출

▶ 블소2 UI 연출 쇼릴

블레이드 & 소울 2

UNREAL ENGINE · UMG — 메뉴 · 강화 · 시즌 이벤트 · 씬 연출



● 라이브 UI 연출 — 실제 모션 루프 (PDF에서는 대표 컷)



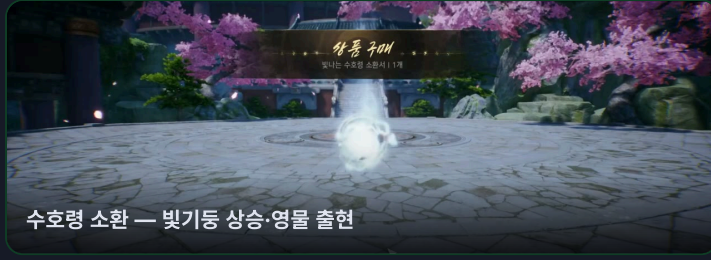
수호령 합성 성공 — 결과 공개 연출



수호령 컬렉션 달성 — 토스트 배너 연출



장비·스탯 팝업 — 슬롯 링 전개 연출



수호령 소환 — 빛기둥 상승·영물 출현



아루의 상자 재련 — 황금 링·보석 채도



강화 성공 — 수목 먹션·룬 링 연출



다중 강화 — 연속 임팩트 폭발



운명의 성도 — 별자리 점등 씬



시즌 이벤트 출석부 — 설산 전개 연출

- ◆ 수목의 모션 언어 — 강화 성공의 먹션 소용돌이처럼, UI 등장이 '배치'가 아니라 '획'으로 느껴지는 동양적 커브
- ◆ 확정 순간의 무게 — 일상 인터랙션은 절제하고, 강화 성공 같은 확정 순간에만 화면 전체 연출을 허용
- ◆ 씬 연출 — 운명의 성도 별자리, 아루의 상자 등 시스템 화면을 세계관의 한 장면으로

세븐나이츠 2 · VFX 참여작

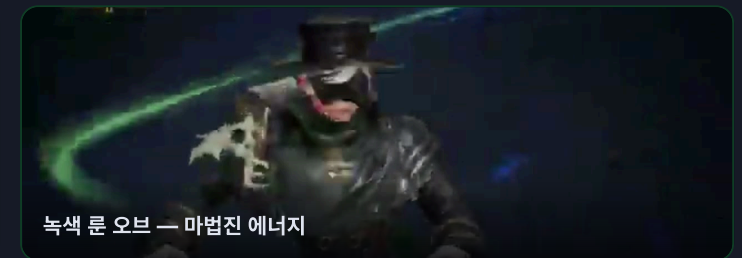
▶ 스킬 FX 쇼릴

▶ UI 연출 쇼릴

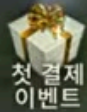
UNREAL ENGINE · 스킬 연출 · 시퀀서 시네마틱 — 전투 HUD 시인성과 직결되는 실시간 연출



● 라이브 스킬 FX — 실제 모션 루프 (PDF에서는 대표 컷)



02일23시남음



첫 결제 이벤트



오늘의 콘텐츠 현황판



점핑 이벤트

[메인] 빛의 신전을 향해 보통

파멸 군단 처치 6 / 10

단장D@3151

팀 전투력 : 52,435

세븐나이츠 2

인게임 — HUD UI 연출, 전투 중 스킬 FX (난전 시인성 관리)

- ◆ 궁극기 3막 설계 — 캐스팅(기대) → 임팩트(폭발) → 여운(잔광). 임팩트는 짧을수록 강하다는 원칙으로 타이밍 배분
- ◆ 시퀀서 컷인과의 협업 — 컷인 FOV 변화·블러가 이펙트 밀도를 보완, 파티클 수를 늘리지 않고 체감 화력을 올림
- ◆ 속성별 모션 언어 — 화염·신성·암속성 스킬마다 전개 곡선과 잔광 톤을 분리해 캐릭터 개성을 강화

리니지 W

UNREAL ENGINE · UMG — 장비창 · 이벤트 메시지 · 대화 UI



장비창 연출 — 원형 슬롯 힐 전개



이벤트 메시지 · 캐릭터 대화 UI 연출

- ◆ **질제가 만드는 무게감** — 다크 톤 위에 금속 질감의 미니멀 모션, 정보 밀도가 높아도 시선이 흐트러지지 않게

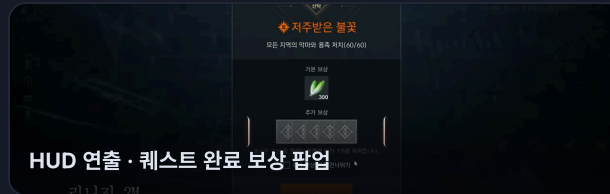
▶ 쇼릴에서 보기

리니지 2M

UNREAL ENGINE · UMG — 강화 · HUD · 뿔기 · 컬렉션



강화 · 다중강화 UI 연출



HUD 연출 · 퀘스트 완료 보상 팝업



뿔기 · 컬렉션 완성 연출

▶ 쇼릴에서 보기

블소 NEO · 배틀 크러쉬

UNREAL ENGINE · UMG — 인벤토리 · 스탯 슬롯 · 메인메뉴 · 튜토리얼



블소 NEO — 인벤토리 · 스탯 슬롯 연출



배틀 크러쉬 — 메인메뉴 · 영웅 아이들 연출



배틀 크러쉬 — 튜토리얼 가이드 UI 연출

- ◆ **톤의 스펙트럼** — MMO 다크 톤부터 밝한 캐주얼까지, IP에 맞춰 모션 언어를 전환

▶ 쇼릴에서 보기

Blood Invasion

담당 — VFX 시스템 · UI 파티클 연출 · 커스텀 셰이더 · TA 전반

Unity 6 · URP 17

2D 타워디펜스 · 모바일

뱀파이어 테마의 모바일 타워디펜스. "피"라는 단일 테마를 122개 VFX 프리팹과 60+ 셰이더그래프로 변주하면서도 시인성을 잃지 않는 것이 과제였습니다. 적 9종(암살자·사냥꾼·창병·사제·마녀 등) 각각의 피격 이펙트를 속성 시스템으로 체계화했고, 전투의 핵심 보상 루프(혈액 오브 획득 → 제단 강화)는 UIParticle로 월드→UI를 끊김 없이 연결해 연출했습니다.

🔥 속성 기반 VFX 아키텍처

VFX.Hit.* 50+ 변형을 적 타입별로 운영하되, 머티리얼·텍스처는 공유 풀에서 조합. 상태이상(화상·빙결·감전)은 **VFX.Status*** 계열로 분리해 중첩 표현이 가능하게 설계.

🌀 커스텀 URP 셰이더 8종

RadialShockwave (충격파 왜곡),
VerticalPannerWithSideFade (혈류 기동), **URPStylizedLava**,
UI용 **RGBMask Dissolve** 등 셰이더그래프로 안 되는 영역은 HLSL로 직접 작성.

🌟 UIParticle 보상 연출 — UI 프리팹 366종 운영

mob-sakai **ParticleEffectForUGUI** 기반.
Shared/Prefabs/UI 366종(버튼·게이지·배지·재화 연출)을 단일 체계로 운영하며, 재화 비행 연출의 Canvas 소팅·마스킹 충돌을 해결하고 획득 타이밍에 게이지 펀치 스케일을 동기화.

●●● RadialShockwave.shader — 극좌표 충격파 링 (URP HLSL, 직접 작성)

```
// 파티클 버텍스 컬러 알파를 '시간'으로 사용 → 머티리얼 복제 없이
// 파티클 시스템의 Color over Lifetime만으로 링 확산을 구동
float waveRadius = input.color.a;

// 극좌표 샘플링: 각도별로 반지름을 왜곡해 '유기적인' 링을 만든다
float2 polarUV = float2(dist, normalizedAngle) * _MainTex_ST.xy + _MainTex_ST.zw;
float radiusDistortion = (tex.r - 0.5) * _Distortion;
float distToWave = abs(dist - (waveRadius + radiusDistortion));

// 링 두께 + 전/후 에지 페이드 + 원형 마스크
float ringMask = 1.0 - smoothstep(0.0, _Width, distToWave);
float waveMask = ringMask * frontSoftness * backSoftness * circleMask;
```

인게임 캡처 — 전투 VFX 자체 배치 레코더로 캡처한 실제 인게임 이펙트



UI 연출 — 재화 획득 / 피드백 루프 uGUI 캔버스 위에서 재생되는 UIParticle 계열



플립북 텍스처 아틀라스 이펙트의 원재료 — 시트 설계도 함께 보여드립니다



K2 — 블록 퍼즐

담당 — 게임필 FX · UI 연출 · 에디터 툴링

Unity · 모바일 퍼즐

라이브 운영 빌드

캐주얼 퍼즐에서 이펙트는 곧 **게임필(game feel)**입니다. 매치 한 번의 짧은 임팩트부터 주문 완료·시즌 진행 같은 보상 연출의 여운까지 — 모두 "플레이어의 손가락이 뽐났다"고 말하는 **피드백**으로 설계했습니다. 연출 제작과 함께, 아트·기획의 반복 속도를 끌어올리는 에디터 툴링을 담당했습니다.

⚡ 타이밍이 곧 설계

임팩트(~0.1s) → 보상(0.3~0.4s) → 무드 루프(2s)의 3계층 템포. 빠른 피드백은 짧고 강하게, 보상은 여운 있게.

📁 에디터 툴링 환경

`FlyEffectTestMenu` (연출 즉시 테스트),
`TMPFontResamplerWindow` (폰트 리샘플), `StageEditor` (레벨 편집),
`AutoPlayEditor` (자동 플레이 검증) — 아트·기획 반복 속도를 끌어올리는 도구들.

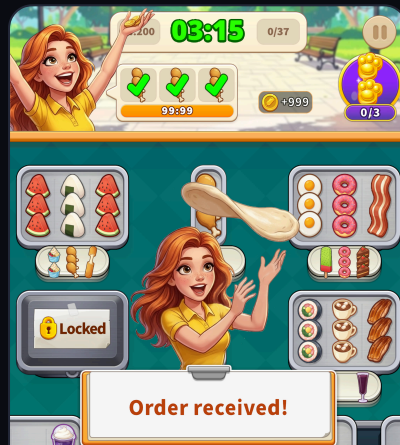
🚗 라이브 빌드에 실린 연출

푸드트럭 테마 로비의 시즌 진행 HUD, 주문 보드의 점수·클리어 피드백 — 연출은 시안이 아니라 **운영 빌드에서 검증**됩니다.

라이브 빌드 — 인게임 Unity 실빌드 스크린샷



로비 - 푸드트럭 테마 · 시즌 진행 HUD



인게임 - 주문 보드 · 클리어 피드백 연출

WoolSaga (울사가)

1인 제작 — 기획 · 아트 디렉팅 · 세이더 · VFX · UI 전부

Unity 6 + PixiJS 8 듀얼 빌드

털실 드래곤을 조작하는 캐주얼 퍼즐. 같은 게임을 **Unity(네이티브)**와 **PixiJS(웹)** 두 갈래로 구현하면서, 플랫폼이 달라도 동일한 비주얼 아이덴티티를 유지하는 것이 핵심 과제였습니다. Unity 쪽은 양털 질감을 위한 커스텀 셰이더를, 웹 쪽은 44프레임 플립북 VFX를 제작했습니다.

🐏 WoolFluff 셰이더 — "양털은 PBR로 안 됩니다"

표준 Lit은 양털을 광택 플라스틱처럼 만듭니다. **Wrapped Lambert**(부드러운 명암 경계) + **Fresnel Rim**(실루엣의 보풀 빛) + 무광 처리로 섬유 질감을 표현. ShadowCaster/DepthOnly 패스까지 URP 완전 대응.

🌟 런타임 파티클 — ExplosionFX

프리팸 없이 코드로 생성하는 2계층 파티클(보풀 퍼프 + 스파크). 색상 틸트를 게임 로직(실 색깔)과 연동해 9색 변형을 에셋 추가 없이 처리.

📄 코드 기반 HUD 빌더

HudController.cs — HP/실타래 게이지, 퍼즐 그리드, 승패 모달을 전부 런타임 생성. 와일드카드 펄스 링 등 상태 연출 포함. 프리팸 머지 충돌 없는 UI 워크플로.

● ● ● WoolFluff.shader - 핵심 라이팅 (URP HLSL, 직접 작성)

```
// Wrapped Lambert - 명암 경계를 부드럽게, 섬유 내부의 빛 번짐을 모사
float wrapped = saturate((NdotL + _Wrap) / (1.0 + _Wrap));

half3 diffuse = baseRGB * mainLight.color * (wrapped * shadow + _AmbientStrength);

// Fresnel rim - 실루엣에서 밝게, 안쪽으로 감쇠.
// 림을 베이스 컬러로도 틸트해 "털 자체가 빛을 머금은" 느낌으로.
float fresnel = pow(1.0 - saturate(dot(N, V)), _RimPower);
half3 rim = fresnel * _RimColor.rgb * _RimIntensity * baseRGB;

return half4(diffuse + rim, _BaseColor.a);
```

WS

Yarn Burst

실타래 파일 · 44f 플립북

WS

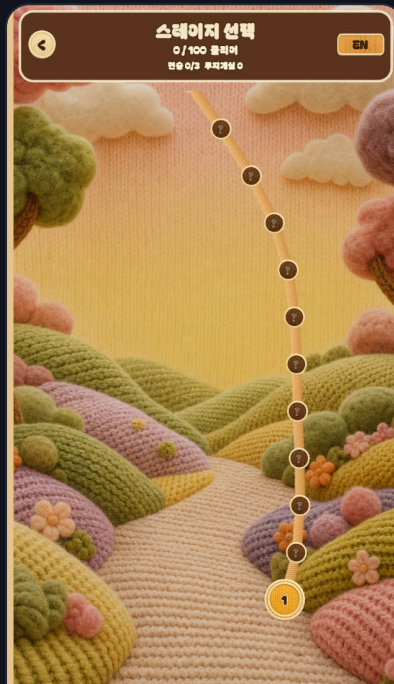
Yarn Poof

소멸 연기 · 28f 플립북

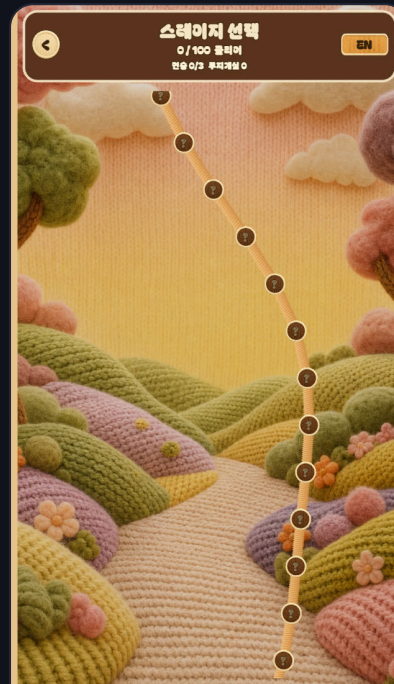
실제 빌드 — 인게임 PixiJS 웹 빌드 스크린샷



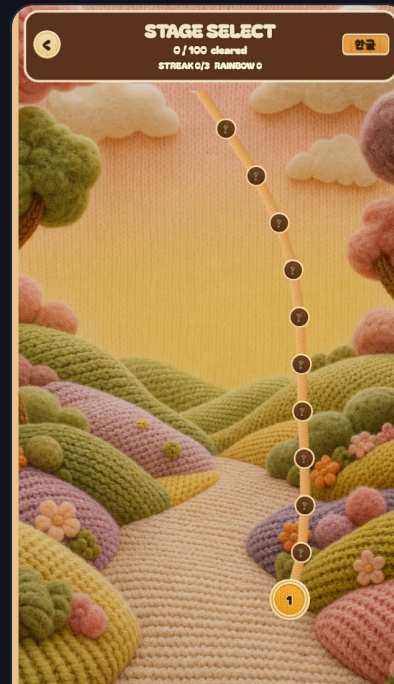
타이틀 화면 — 울 드래곤



스테이지 선택 — 니트 월드맵



월드맵 — 상위 스테이지 구간



EN 로컬라이제이션 — 언어 토글



인게임 배경 - 메인



테마 배경 - Forest



테마 배경 - Snow



스태이지 선택 맵



드래곤 헤드 - 메인 캐릭터



고양이 - 서브 캐릭터

던전 서바이버스 (Dungeon Survivors)

1인 제작 — HUD 모션 · 전투 연출 · 시스템 전부

Phaser 3 · HTML5

8캐릭터 · 5존 · 풀 메타 루프

범서라이크 장르의 HTML5 게임. 수백 마리 적이 쏟아지는 화면에서 **UI는 0.12초 트랜지션으로 즉각 반응**하되 전투를 가리지 않아야 합니다. 캔버스(전투)와 DOM(HUD)을 분리해 각자의 최적 기술로 연출했습니다 — 체력/경험치 바는 CSS 그라디언트+트랜지션, 레벨업 룰렛은 DOM 오버레이, 전투 이펙트는 Phaser 파티클.

하이브리드 렌더링 전략

전투(수백 엔티티)는 Phaser 캔버스, HUD는 DOM. HP바 `width .12s` 트랜지션, 장착 아티팩트의 `inset glow` 강조 등 CSS가 더 잘하는 일은 CSS에게.

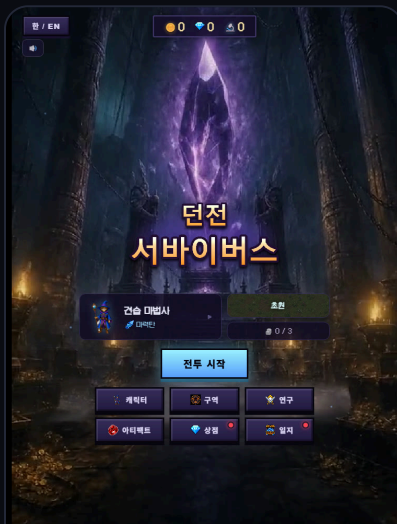
레벨업 룰렛 연출

전투 정지 → 룰렛 스피ن → 당첨 슬롯 강조로 이어지는 호흡. 카드 선택형에서 룰렛형으로 개편해 레벨업마다 기대감을 끌어올리고, 등급(Common~Legendary)별 컬러 시스템을 룰렛 슬롯·칩·아이콘 전체에 일관 적용.

씬 전환과 정보 위계

허브(캐릭터/구역/연구/상점/일지) 6개 화면을 단일 오버레이 시스템으로 — 진입·이탈 모션의 방향성을 통일해 공간감을 부여.

인게임 — 최신 빌드 던전 허브 · 실전투 · 레벨업 스피ن 룰렛 (실제 캡처)



던전 허브 — 포털 연출 · 메타 메뉴 6종



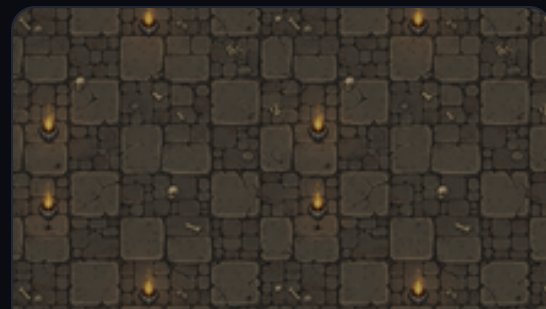
실전투 — 적 웨이브 · 피격 FX · HUD



레벨업 룰렛 — 스피ن → 당첨 카드 · 리볼(광고/보석)



허브(메인 메뉴) 배경



존 - Laboratory



존 - Arboretum



존 - Sewer



스킬 아이콘 시스템 - 메이지 계열 8종



캐릭터 4종 + 등급 컬러 시스템 4단계



보스 - Goliath

Grid Rush — 신작 블록 퍼즐

프로토타입도 기획서가 아니라 **실행되는 빌드**로 관리합니다. Grid Rush는 4개 모드와 기록·데일리 메타 루프를 갖춘 블록 퍼즐 — 아래 캡처는 모두 실제 인게임이며, **이 자리에서 바로 플레이**됩니다.

1인 제작 — 기획 · 아트 · 연출 · 코드 전부

HTML5 · 모바일 퍼즐

클래식 · 스테이지 48 · 타임어택 · 데일리

배치가 곧 쾌감

조각 배치 → 라인 클리어 → 콤보 이어지는 즉각 피드백. K2에서 다진 퍼즐 게임필 감각을 자체 타이틀로 확장.

4모드 + 데일리 루프

클래식(무한) · 스테이지 48 · 타임어택(60s) · 데일리 챌린지. BEST 기록과 일일 미션이 재방문 동기를 만드는 메타 설계.

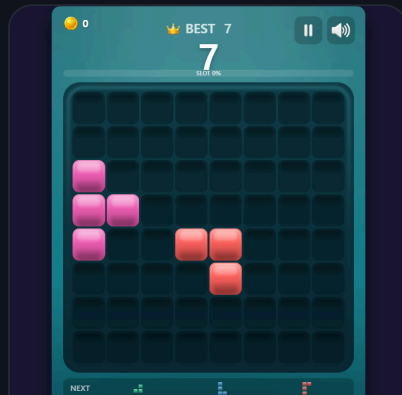
아이템 연출

LINE(가로 클리어) · BOMB(범위 폭파) 아이템 — 보드를 가르는 클리어 연출과 슬롯 게이지 피드백으로 역전의 손맛을 설계.

인게임 실제 빌드 캡처 — 메인 메뉴 · 클래식 모드



메인 메뉴 - 모드 선택 · 데일리
챌린지



클래식 - 배치 · NEXT 큐 ·
LINE/BOMB 아이템

▶ Grid Rush 플레이 (START_PORTFOLIO.bat 실행 후 접속)

PLAYABLE

말보다 플레이 — 라이브 데모

아래 게임들은 이 포트폴리오 폴더에 실제 빌드가 동봉되어 있습니다. **던전 서바이버스 · Grid Rush**는 index.html 더블클릭만으로 바로 플레이되고, **WoolSaga**는 에셋을 동적 로드해 START_PORTFOLIO.bat (로컬 서버)나 호스팅 주소에서 실행됩니다.



WoolSaga — 울사가

PixiJS 8 웹 빌드. 실타래 드래곤 퍼즐 — 100스테이지 맵·플립북 VFX·UI 연출. (서버/호스팅에서 플레이 — START_PORTFOLIO.bat)

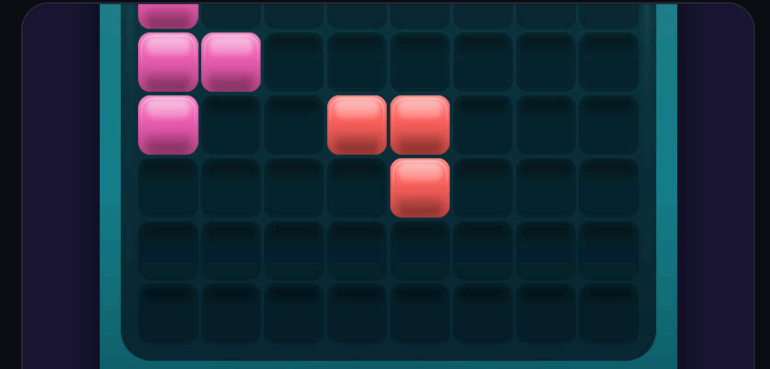
▶ **플레이** (START_PORTFOLIO.bat 실행 후 접속)



던전 서바이버스 — VSLike

Phaser 3 풀 게임 (8캐릭터 · 5존 · 메타 진행). 던전 허브 연출, 레벨업 스피너, 하이브리드 HUD를 플레이로 검증하세요.

▶ **플레이** (START_PORTFOLIO.bat 실행 후 접속)



Grid Rush — 블록 퍼즐

HTML5 웹 빌드. 4개 모드 · NEXT 큐 · LINE/BOMB 아이템 · 데일리 메타 루프를 갖춘 블록 퍼즐 — 이 자리에서 바로 플레이됩니다.

▶ **플레이** (START_PORTFOLIO.bat 실행 후 접속)

나머지 절반 — VFX · 테크니컬 아트 역량 (한 사람이)

라이브 대작들의 이펙트·UI 작업과 병행해, 게임 속 이펙트를 **자동으로 캡처·분류하는 파이프라인**을 직접 만들었습니다. 이펙트 **3,815개**를 자동 캡처하고 그중 **3,087개**를 **13개 카테고리**로 자동 분류해 검색·재생·추출이 되는 라이브러리(VFX Library v3)로 만들었습니다 — **아래에서 실제 화면**을 볼 수 있습니다. "이 이펙트 다른 게임에서 어땠더라?"라는 질문에 3초 안에 답하는 시스템입니다.

STEP 1 — CAPTURE

VfxBatchRecorder.cs

Unity Editor 상주 레코더.

`_VfxBatchTrigger.json` 감지 → VFX

프리팸을 스폰·재생·프레임 캡처. 해상도/FPS/길이 설정 가능.

STEP 2 — EXTRACT

vfx-sprite-extractor

TypeScript CLI. Unity YAML/.meta 파싱 → GUID

맵 → 텍스처 추적. UE5 프로젝트도 지원.

STEP 3 — PROCESS

Auto-Trim + Sheet Export

투명 여백 자동 트림 → N×N 그리드 자동 산출 → 시트

출력(원본+50%). Sharp 기반 이미지 처리.

STEP 4 — SERVE

VFX Library v1→v3

캡처 3,815개 → 분류 3,087개 검색·재생·추출 웹 뷰어. 3차에

걸쳐 자체 개량.

왜 만들었나

이펙트 레퍼런스 탐색이 작업 시간의 30%를 먹고 있었습니다. 게임별로 흩어진 이펙트를 한 번 캡처해서 연구 재사용 — 신규 프로젝트 무드보드 제작이 일 단위에서 분 단위로 줄었습니다.

v1 → v3, 96.5% 경량화

v1(5.3GB 정적 HTML)은 무거웠고, v2(메타데이터+서버)는 관리성을 얻은 대신 느렸습니다. v3는 썸네일 분리·지연 로딩으로 코어를 **184MB까지 경량화**. 도구도 제품처럼 인터렉션합니다.

이 포트폴리오도 산출물입니다

지금 보고 계신 페이지의 플립북 플레이어·시퀀스 재생·뷰포트 기반 재생 제어 모두 같은 파이프라인 철학으로 작성한 자체 제작 엔진입니다.

VFX Sprite Library v3 — 실제 화면 직접 설계·구현한 이펙트 레퍼런스 도구 · 라이브 데모 동봉

The interface shows a top navigation bar with six category filters: Nature (50 effects), Blood / Gore (16 effects), UI / Text (494 effects), Loop / Ambient (60 effects), Item / Object (301 effects), and Other (333 effects). Below the filters, a status bar indicates 13 categories, 3,087 effects, and options for search, playback, and download.

The interface displays a grid of explosion effects. Each effect is shown in a preview window with its name, file size, and format (ZIP/PNG). The effects include AppleBlast, AppleExplosion, BalloonPop, BananaBlast, BananaExplosion, BloodExplosion, BloodExplosion2, BloodExplosion3, BloodExplosion4, BloodExplosion5, BloodExplosion6, BloodExplosion7, BloodExplosion8, BloodExplosion9, BloodExplosion10, BloodExplosion11, BloodExplosion12, BloodExplosion13, BloodExplosion14, BloodExplosion15, BloodExplosion16, BloodExplosion17, BloodExplosion18, BloodExplosion19, BloodExplosion20, BloodExplosion21, BloodExplosion22, BloodExplosion23, BloodExplosion24, BloodExplosion25, BloodExplosion26, BloodExplosion27, BloodExplosion28, BloodExplosion29, BloodExplosion30, BloodExplosion31, BloodExplosion32, BloodExplosion33, BloodExplosion34, BloodExplosion35, BloodExplosion36, BloodExplosion37, BloodExplosion38, BloodExplosion39, BloodExplosion40, BloodExplosion41, BloodExplosion42, BloodExplosion43, BloodExplosion44, BloodExplosion45, BloodExplosion46, BloodExplosion47, BloodExplosion48, BloodExplosion49, BloodExplosion50, BloodExplosion51, BloodExplosion52, BloodExplosion53, BloodExplosion54, BloodExplosion55, BloodExplosion56, BloodExplosion57, BloodExplosion58, BloodExplosion59, BloodExplosion60, BloodExplosion61, BloodExplosion62, BloodExplosion63, BloodExplosion64, BloodExplosion65, BloodExplosion66, BloodExplosion67, BloodExplosion68, BloodExplosion69, BloodExplosion70, BloodExplosion71, BloodExplosion72, BloodExplosion73, BloodExplosion74, BloodExplosion75, BloodExplosion76, BloodExplosion77, BloodExplosion78, BloodExplosion79, BloodExplosion80, BloodExplosion81, BloodExplosion82, BloodExplosion83, BloodExplosion84, BloodExplosion85, BloodExplosion86, BloodExplosion87, BloodExplosion88, BloodExplosion89, BloodExplosion90, BloodExplosion91, BloodExplosion92, BloodExplosion93, BloodExplosion94, BloodExplosion95, BloodExplosion96, BloodExplosion97, BloodExplosion98, BloodExplosion99, BloodExplosion100.

Explosion 카테고리 608종 — "Play All"로 60개 동시 재생, 알파 배경 도구, ZIP/PNG 원클릭 추출

● ● ● _VfxBatchTrigger.json - 아티스트용 캡처 명세

```
// 아티스트는 JSON에 프리팹 경로만 적으면 됩니다.  
// 에디터 상주 레코더가 감지해서 알아서 캡처합니다.  
{  
  "prefabs": ["Assets/ ... /VFX.BloodSurge.prefab"],  
  "width": 512, "height": 512,  
  "fps": 30, "duration": 5.0,  
  "background": "transparent"  
}
```

● ● ● VfxBatchRecorder.cs - 프레임 캡처 루프 (발췌)

```
for (int f = 0; f < frameCount; f++)  
{  
    yield return new WaitForEndOfFrame();  
  
    RenderTexture.active = rt;  
    GL.Clear(true, true, new Color(0,0,0,0)); // 알파 보존  
    cam.Render();  
    readTex.ReadPixels(new Rect(0, 0, w, h), 0, 0);  
  
    File.WriteAllBytes(  
        Path.Combine(dir, $"frame_{f:D4}.png"),  
        readTex.EncodeToPNG());  
}
```

SANGYUL LEE — Senior Game UI/UX Designer · Multiplatform · In-Engine

kingbig1@gmail.com

Unreal Engine (나이야가라 · 캐스케이드 · 시퀀서 · UMG · 머티리얼) · Unity (URP · HLSL · VFX Graph · UIParticle · DOTween) · PixiJS · Phaser · TypeScript Pipeline

이 포트폴리오의 모든 이펙트는 자체 플립북 엔진으로 실시간 재생되며, 페이지 자체가 제 TA 파이프라인의 산출물입니다.

본 페이지는 인쇄(PDF) 최적화가 적용되어 있습니다 — Ctrl+P로 A4 가로 PDF를 바로 출력할 수 있습니다.

본 자료의 프로젝트 이미지·코드는 포트폴리오 열람 목적으로만 사용되었습니다. 무단 전재 및 재배포를 금합니다.

상용 게임 캡처는 본인 참여 작업의 쇼릴 영상에서 발췌했으며, 각 게임의 저작권은 해당 게임사에 있습니다.

레퍼런스 라이브러리(LIB 배지)에는 다양한 출처의 이펙트가 포함되며, 파이프라인 기술 증명 목적으로만 제시됩니다.