시티러쉬 — 컨셉기획서

> 재구성 기획서 (역설계) / 작성 기준 시점: 2026-07-08 (프로젝트 시작 시점) / 문서 버전: Ver 0.1 프로젝트 ID: P013 / 근거 수준: B / 작성: Claude (재구성) 이 문서는 결과물과 제작 기록으로부터 기획 의도를 역추정해 당시 작성한 것처럼 재구성한 것이다. 확인된 사실과 추정을 [근거]/[추정] 태그로 구분한다.

1. 한 줄 컨셉과 기획 의도

한 줄 컨셉: 경찰에게 쫓기며 도시 곳곳의 열쇠 아이템을 모두 모은 뒤 탈출구로 빠져나가는 3D 차량 추격/탈출 게임. [근거: ItemCollector.cs, EnemyChaser.cs, Goal.cs]

기획 의도. - 짧은 시간에 목표가 이해되는 캐주얼 게임으로 만든다. 목표는 "다 모으고 탈출"의 두 단계뿐이다. [근거: ItemCollector.cs] 이유는 3인 팀의 약 11일 개발 기간(2026-07-08~07-19)에 완성도 있는 한 판을 내기 위해서다. [추정: git 기간] - 쫓기는 긴장과 도망칠 수 있다는 구제 장치를 함께 둔다. 충돌 후 경찰이 3초 멈추고 플레이어가 1초 무적이 되는 설계가 근거다. [근거: vehicle_system_analysis.md] - 팀 구성(기획, 구현, 아트 역할 분담)에 따라 외부 에셋(차량, 로우폴리 시티)을 적극 활용해 빠르게 3D 도시 무대를 만든다. [근거: Assets/ExternalAssets] 역할 분담 자체는 기록 없음. - 과제/제출용 유니티 캐주얼 게임 프로젝트로 시작했다. [추정: 인벤토리의 "유니티캐주얼게임프로젝트" 제출 폴더명]

2. 타깃, 플랫폼, 엔진, 범위, 인원, 도구

- 플랫폼: PC(Windows) 빌드, 기본 해상도 1024×768(창 모드 기본값). [근거: ProjectSettings.asset, City_Rush.exe]
- 엔진: Unity 6000.3.19f1, URP, Cinemachine, AI Navigation, TextMeshPro, Input System 패키지 포함. 코드는 구 `Input` 클래스를 쓴다. [근거: ProjectVersion.txt, City_Rush_Data/Managed, CarController.cs]
- 범위(S): 한 스테이지 중심의 코어 루프 1개. 메타 진행/저장 없음. [근거: 저장 관련 스크립트 없음]
- 인원: 3인(윤희성, 박도원, 이제희). 도구: Unity, GitHub, Antigravity 정황, Unity MCP 설치 흔적. [근거: 인벤토리, com.IvanMurzak.Unity.MCP.Installer.csproj]
- 타깃: 별도 타깃 문서 미확인. [추정: 캐주얼 수업 과제 특성상 평가자/동료]

3. 핵심 재미와 차별점

- 플레이어 판타지: 경찰차에 쫓기는 도둑이 되어 도시를 질주한다. [추정: 경찰 AI/열쇠 수집 구조]
- 핵심 재미 1, 추격 압박: NavMeshAgent 경찰이 끝까지 쫓아온다. [근거: EnemyChaser.cs]
- 핵심 재미 2, 도주 쾌감: 피격 후 경찰 정지, R키 순간이동 리스폰과 고스트로 만회할 수 있다. [근거: CarController.cs]
- 핵심 재미 3, 수집 → 개방: 탈출구가 숨겨져 있다가 열린다. [근거: ItemCollector.cs]
- 차별점은 별도 기록 없음. 레퍼런스 게임도 기록 없음.

4. 세계관, 톤, 아트 방향

- 세계관: 도시. 서사 기록 없음. 열쇠 아이템의 이름/의미는 코드에 없고 문서에서 "키 아이템"으로만 지칭된다. [근거: analysis_results.md]
- 톤: 가볍고 밝은 로우폴리 도시. 건물(스카이/상점 다수)과 잔디, 도로 타일을 쓴다. [근거: MDMapCreator.cs]
- 아트: 외부 에셋 조합. SimplePoly City(도시), ARCADE FREE Racing Car(차량), Aletheia(용도 미확인). [근거: ExternalAssets 폴더]
- 포스트 프로세싱 기획서가 별도로 있다. 내용 미확인.

5. 게임 흐름 개요

1. 시작: 플레이어 차량이 맵에 배치, HP 100, 탈출구 비활성. [근거: PlayerHP.cs, ItemCollector.cs]
2. 진행: 도시를 돌며 아이템 수집. 경찰이 추격하고 충돌하면 HP 10 감소. [근거: EnemyChaser.cs]
3. 개방: 아이템을 모두 모으면 탈출구 활성화. [근거: ItemCollector.cs]
4. 종료: 탈출구 진입 = 클리어, HP 0 = 게임오버. 둘 다 `Time.timeScale = 0` 후 패널 표시. [근거: Goal.cs, GameManager.cs]
5. 재시작: 현재 씬 재로드. [근거: GameManager.cs]

6. 제작 방향 (파트별 설계 블록)

시스템 파트 — 차량 물리와 추격 AI

- 의도(Why): 이 게임의 재미는 차량 조작감과 추격 압박이다. 가장 먼저 완성해야 하는 핵심 파트다.
- 설계(How): 4륜 WheelCollider 차량(전륜 구동/전륜 조향/전륜 4륜 제동)과 NavMeshAgent 경찰. 상세는 02/03 참조. 물리 기반을 택한 이유는 충돌감과 속도감이 직접 전달되기 때문이고, 키네마틱 이동은 더 간단하지만 충돌 체감이 약하다. [추정]
- 필요(What): 차량 프리팹(바퀴 4개), NavMesh 베이크, 태그 `Player`.
- 작업 방법: 차량 → 경찰 AI → 충돌/HP 순서. 담당 분담은 기록 없음. 검증은 플레이 테스트(`TEST`, `Proj_TEST` 씬).
- 완료 기준: 차량이 입력대로 움직이고 경찰이 따라붙어 HP가 줄어든다.
- 리스크/대안: NavMeshAgent가 물리 차량과 충돌 시 튕김. 3초 정지로 완화.
- 근거: [근거: CarController.cs, EnemyChaser.cs]

콘텐츠 파트 — 도시 맵과 수집 목표

- 의도(Why): 아이템 배치가 플레이 동선을 만든다. 도시 맵이 곧 레벨이다.
- 설계(How): 로우폴리 타일 프리팹을 에디터 스크립트로 일괄 배치하고, 카테고리별 부모(`Roads`, `Natures`, `Buildings`, `Vehicles`, `Props`) 아래로 정렬한다. [근거: MDMapCreator.cs, NewMapSceneSorter.cs] 스크립트화한 이유는 반복 수정이 쉽기 때문이다. [추정]
- 필요(What): 에셋 프리팹 경로, 아이템 프리팹, 스폰 지점(`spawnPoints`).
- 작업 방법: 생성 → 정렬 → 아이템/탈출구 배치 → 개수 입력(`totalItemCount`).

- 완료 기준: 아이템 전부 도달 가능, 탈출구 도달 가능.
- 리스크/대안: 건물 콜라이더에 차량이 끼임 → R키 리스폰이 구제책.
- 근거: [근거: MDMapCreator.cs, NewMapSceneSorter.cs, CarController.cs]

아트 파트 — 외부 에셋 조합과 화면 연출

- 의도(Why): 제작 인원/기간이 작아 에셋으로 시각 품질을 확보한다.
- 설계(How): 로우폴리 도시 + 레이싱 카 에셋. 포스트 프로세싱은 별도 기획서로 정리했다. [근거: 기획서 파일명] 구체 설정은 미확인.
- 필요(What): 에셋 라이선스 확인, URP 볼륨.
- 작업 방법: 에셋 임포트 → 머티리얼 통일 → 볼륨 적용.
- 완료 기준: 도시 전역이 통일된 톤.
- 리스크/대안: 에셋 출처 서로 달라 톤 불일치 → 포스트 프로세싱으로 통일.
- 근거: [근거: ExternalAssets 폴더] [추정: 포스트 프로세싱 목적]

사운드 파트 — 엔진/브레이크 음

- 의도(Why): 속도감은 소리로 가장 쉽게 전달된다.
- 설계(How): 속도에 따라 엔진 피치를 0.8~2.0으로 보간(속도 30m/s에서 최대), 브레이크 입력+속도 1 초과 시 마찰음. [근거: CarController.cs] 별도로 시동, 루프, 종료 3클립 상태 머신 스크립트가 있다. [근거: vehicle_system_analysis.md]
- 필요(What): `Engine_Start.wav`, `Engine_Loop.wav`, `Engine_End.wav`, 브레이크음.
- 작업 방법: 클립 연결 → 피치 범위 튜닝.
- 완료 기준: 정지 시 무음, 가속 시 피치 상승.
- 리스크/대안: 두 시스템 중복 → 하나만 사용.
- 근거: [근거: CarController.cs, vehicle_system_analysis.md]

UI 파트 — HP, 아이템, 속도, 결과 패널

- 의도(Why): 위험도(HP), 목표(수집), 속도감을 한눈에 보여준다.
- 설계(How): HP 슬라이더+`HP : n` 텍스트, 현재 `/` 전체 아이템 텍스트, `n km/h` 속도계, 클리어/게임오버 패널. [근거: UIManager.cs, ItemCollector.cs, Speedometer.cs]
- 필요(What): Canvas, TMP 폰트, 패널.
- 작업 방법: 캔버스 구성 → 참조 연결.
- 완료 기준: 값 변화가 즉시 반영.
- 리스크/대안: 한글 폰트 TMP 미포함 시 깨짐 → 폰트 에셋 생성.
- 근거: [근거: UIManager.cs, Speedometer.cs]

데이터 파트

- 의도(Why): 수치를 에디터에서 조정해 밸런스를 맞춘다.
- 설계(How): 외부 데이터 테이블 없이 `[SerializeField]` 필드로 보관. [근거: 스크립트 전반]
- 필요(What): 03 데이터 명세서 참조.

- 작업 방법: 03 데이터 명세서 참조.
- 완료 기준: 주요 수치 7개 이상을 인스펙터에서 조정 가능.
- 리스크/대안: 씬마다 값이 달라짐 → 프리팹화.
- 근거: [근거: CarController.cs, PlayerHP.cs]

7. 제작 단계(마일스톤)

단계	목표	산출물	근거
M1 (07-08)	프로젝트, 씬, 렌더 에셋 세팅	첫 푸시	git
M2 (07-13)	차량, AI, 수집 프로토	`Proj_TEST` 씬	git, 씬
M3 (07-14~15)	맵, UI, 사운드 보강	`NewMap` 씬	git, 추정
M4 (07-19)	제출 빌드	`City_Rush.exe`	인벤토리
단계 내용은 커밋 날짜에 맞춘 추정이다. [추정]			

8. 리스크와 결정이 필요한 사항

- 기획서 본문과 구현의 차이 확인(본문 미열람).
- 빌드 씬 구성과 최종 레벨 수 확정.
- 엔진음 시스템 통일, 구 Input 대신 Input System 전환 여부.
- 2D 테스트 스크립트(`Scripts2`) 정리 여부.