# Middle (AR 마그리트) — 메인기획서

> 재구성 기획서 (역설계) / 작성 기준 시점: 2021-03-15 (프로젝트 시작 시점) / 문서 버전: Ver 0.1 프로젝트 ID: P037 / 근거 수준: C / 작성: Claude (재구성) 이 문서는 결과물과 제작 기록으로부터 기획 의도를 역추정해 당시 작성한 것처럼 재구성한 것이다. 확인된 사실과 추정을 [근거]/[추정] 태그로 구분한다.

## 1. 핵심 메카닉스

핵심 메카닉스는 하나다. **거리 스크러빙**: 카메라와 마커 사이 거리를 애니메이션 재생 위치로 직접 대응시킨다. 시간이 아니라 사용자의 몸 움직임이 재생을 이끈다. [추정: 거리를 normalizedTime에 대입하는 구조]

#### 거리 스크러빙 (입력 → 상태 → 피드백)

- 의도(Why): 버튼 없이 "다가가는 몸짓"만으로 연출을 조작하게 해 AR의 공간 감각을 체험시킨다.
- 설계(How): 입력은 `ImageTargetGetDistance`가 반환하는 카메라-마커 거리(`distance`). `FloatRemap`으로 MinIn 8, MaxIn 15를 MinOut 1, MaxOut 0에 대응시켜 `dist`(0~1)를 만든다. 즉 거리 8 이하는 1, 15 이상은 0이며 가까울수록 1에 가깝다. 결과를 글로벌 이벤트 `SetVidedTime`으로 보내 두 번째 FSM을 시작시킨다. 두 번째 FSM은 `Compare 0`(dist 대 0), `Compare 1`(dist 대 1)을 거쳐 `Effects`에서 `AnimatorPlay`(대상 Corona 오브젝트, 상태명 `corona`, normalizedTime = `dist`)를 실행한다. 비교 두 번은 범위 밖 값을 걸러 내는 용도로 추정하며 이벤트 연결의 세부는 미확인이다. [근거: Magritte.unity] [추정: 비교 상태의 용도]
- 필요(What): Vuforia ImageTarget, PlayMaker, Animator 컨트롤러 `Corona`, 클립 `corona`.
- 작업 방법: ① ImageTarget에 FSM `Checkdistance` 부착 ② 같은 오브젝트에 FSM `SetVidedTime` 부착 ③ Corona 스프라이트에 Animator 연결 ④ 실기기에서 8~15 범위를 확인한다.
- 완료 기준: 마커 앞에서 앞뒤로 움직이면 코로나 프레임이 앞뒤로 매끄럽게 이동한다.
- 리스크/대안: 거리 단위는 마커 물리 크기에 의존하므로 마커 크기가 바뀌면 8~15가 어긋난다. 범위를 변수로 노출한다. [제안값] 범위 외 입력은 `Clamp`로 처리한다.
- 근거: [근거: Magritte.unity]

#### 마커 인식과 장면 표시

- 의도(Why): AR 체험의 입구. 마커를 비추는 순간 그림이 나타난다.
- 설계(How): ImageTarget(스케일 0.8427069) 자식으로 하늘, 산, 벽, 양복, gurry, Corona를 둔다. 하늘만 별도 루트에 두고 z 30.91, 스케일 6.33으로 크게 멀리 배치해 배경 역할을 맡긴다. 인식 해제 시 표시 처리 방식은 Vuforia 기본 동작(`DefaultTrackableEventHandler`)이라 추정한다. [근거: Magritte.unity Transform] [추정: 기본 핸들러 사용]
- 필요(What): 마커 이미지, Vuforia DB.
- 작업 방법: 마커 등록 → 배치 → 인식 테스트.
- 완료 기준: 마커 인식 시 모든 레이어가 보이고 해제 시 사라진다.
- 리스크/대안: 하늘이 루트에 있어 마커를 놓쳐도 남을 수 있다. 하늘을 마커 자식으로 옮기거나 의도로 남긴다. [추정: 루트 배치의 의도 불명]

- 근거: [근거: Magritte.unity]

## 2. 구현 방법

- 구조: 씬 1개. 주요 오브젝트는 ARCamera, ImageTarget(FSM 2개, Animator), Directional Light, PlayMakerGUI, 스프라이트 6개(하늘, 산, 벽, 양복, gurry, Corona). [근거: Magritte.unity]
- 상태 머신: FSM `Checkdistance`(단일 상태 `Check Distance`에서 매 프레임 거리 측정, 환산, 이벤트 송신) → FSM `SetVidedTime`(State 1 → Compare 0 → Compare 1 → Effects). [근거: Magritte.unity]
- 데이터 파이프라인: AE 시퀀스(PNG 77장, 이름 컴포지션 `100~176`) → Unity 스프라이트 → `corona.anim`(샘플레이트 60, 길이 약 1.283초, 루프 켜짐). 77프레임 ÷ 60 ≈ 1.283초로 일치한다. [근거: corona.anim, coro 폴더]
- 도구: PlayMaker, ARExtensionVuforia v1.1 패키지, After Effects. InkPainter 샘플은 포함돼 있으나 씬 참조는 확인하지 못했다. [근거: raw/d2.md]
- 자작 C#: 없음 또는 미확인. 로직은 모두 FSM으로 구현했다.

## 3. 시스템 간 상호작용과 밸런스 기준

- 핵심 변수는 거리(`distance`)와 정규화값(`dist`) 두 개뿐이다.
- 밸런스 기준은 8~15라는 "체험 가능 거리 폭"이다. 폭 7 단위가 좁으면 조작이 예민하고 넓으면 반응이 둔하다. 실기기에서 조정 대상이다. [추정: 값의 의미]
- 레이어 간 상호작용은 없다. 시각 배치만 있다.

## 4. 게임의 나머지 부분의 방향

- 콘텐츠 구조: 단일 씬. 스테이지/챕터 없음.
- 메타 진행, 저장, 설정: 없음(해당 없음, 사유: 과제용 단일 체험).
- UI/UX: 화면 UI 없음. 마커 안내 문구는 추후 추가 후보.
- 아트/사운드: 세부기획서 03 참조.
- 미사용/미완성으로 보이는 부분: 씬 `Scratch`, `SampleScene` 및 완성도 약 60% 판정의 근거. 확장 방향은 [제안값]으로 03에 둔다.

## 5. 제작 과정 설계

- 순서: 환경 설정 → 리소스(AE 렌더, 이미지) → 씬 구성 → 애니메이션 → FSM → 기기 테스트.
- 병렬 가능: AE 렌더와 Vuforia 설정.
- AI/도구 분담: 이 시기 제작에 AI를 썼는지 기록 없음. 수작업으로 취급한다.
- 검증: Unity 에디터 Play(웹캠) 또는 기기에서 마커를 비추고 거리 변화에 따른 프레임 이동을 눈으로 확인한다.
- 인수인계: 문서 없음. 이 재구성 문서가 대체한다.

## 6. 세부기획서 목록

| 문서 | 정해야 할 것 |
|---|---|
| `03_세부기획서_ar-연출과-리소스.md` | 레이어/애니메이션 리소스 명세, FSM 변수/수치, 거리 연출 UX, 검수 기준 |