

■ 1) Core Layer (핵심 관리 계층)

- StateManager
- EventManager
- InputManager
- SaveLoadManager
- TimeManager (선택)

게임 핵심 제어 계층.

■ 2) System Layer (도시 시스템 계층)

- BuildSystem
 - GridSystem
 - GridManager
 - ResourceManager
 - CitizenSystem
 - UpgradeSystem
 - EconomySystem (경제/세금)
 - EnvironmentSystem (오염, 땅값 등)
 - AI/PathSystem (시민/차량 이동)
-

■ 3) UI Layer

- UIManager
- HUD/Panel UI들

✿ 각 시스템/매니저별 역할 & 상호작용 구조

■ CORE LAYER

■ StateManager (상태 관리자)

게임의 전체 상태를 관리

예:

- Normal
- Build
- Demolish
- UI_Menu
- Pause
- Inspect

역할

- 현재 모드/상태 저장
- 상태 변경 Event 발생

상호작용

- UIManager → StateManager (ESC, Build 버튼 등)
 - InputManager → StateManager 상태 읽기
 - EventManager → StateManager.ChangeState
-

■ EventManager (게임 전체 중계자 / 메시지 버스)

역할

- 시스템 간 직접 통신 차단
- 모든 이벤트 전달 (OnGridClicked, OnResourceChanged, OnBuildingPlaced 등)
- 한 시스템에서 발생한 결과를 다른 시스템에 중계

상호작용

- Input → EventManager
- GridSystem → EventManager
- BuildSystem → EventManager
- ResourceManager → EventManager
- UpgradeSystem → EventManager
→ EventManager가 이를 각각 필요한 시스템에 브로드캐스트

가장 많이 참조되는 시스템

■ InputManager

역할

- 마우스/키보드 입력을 감지
- 현재 StateManager 상태에 따라 입력 해석을 다르게 함
- “의미 있는 입력”을 EventManager에게 전달
 - 좌클릭(셀)
 - 우클릭
 - 드래그
 - 카메라 입력
 - UI 클릭 신호

상호작용

- StateManager (현재 모드 확인)
 - EventManager (입력 이벤트 전달)
-

■ SaveLoadManager

역할

- 게임 모든 시스템의 데이터를 저장/로드
- JSON / Binary 포맷
- GridManager에 저장된 건물 정보, ResourceManager의 값, Citizen 리스트 등 요청

상호작용

- GridManager → 저장 요청
 - ResourceManager → 자원 상태 제공
 - CitizenSystem → 시민 정보 제공
 - BuildSystem → 설치된 건물 목록 제공
-

■ SYSTEM LAYER

■ GridSystem (그리드 계산 및 기능 제공)

역할

- 클릭된 월드좌표 → 그리드 셀 변환
- 설치 가능 여부 판단
- 라인 디버그 표시

상호작용

- Input → EventManager → GridSystem (클릭된 지점)
- GridSystem → EventManager (설치 가능/불가 결과 전달)
- GridSystem → BuildSystem (이것은 EventManager를 통해 전달)

■ GridManager (그리드 데이터 저장소)

역할

- 전체 그리드의 "데이터 저장"
- 어떤 셀에 어떤 건물이 설치됐는지 기록
- 건물 ID, 점유 셀 목록 저장
- 저장/로드 시 SaveLoadManager가 데이터를 가져감

상호작용

- EventManager → GridManager (건물 설치 정보 저장)
 - SaveManager → GridManager (데이터 저장/로드)
-

■ BuildSystem (건물 설치 시스템)

역할

- 이벤트로 전달된 "건물 설치 요청"을 처리
- 건물 prefab Instantiate
- 배치된 건물의 BuildingComponent 활성화

상호작용

- EventManager에서 "설치 가능" 이벤트 받음
 - EventManager에 "건물 설치됨" 이벤트 전달
 - GridManager에게 "해당 셀 점유" 정보 전달
-

■ ResourceManager (자원/수치 시스템)

역할

- 돈, 전기, 물, 치안, 교육률, 실업률 등 모든 수치 관리
- 건물 설치/삭제 등으로 수치 변동
- 수치가 바뀔 때마다 EventManager로 이벤트 발행

상호작용

- BuildSystem → EventManager → ResourceManager (건물 설치 시 자원 소모)
 - ResourceManager → EventManager(OnResourceChanged)
 - UpgradeSystem이 수치를 읽어 조건 판단
-

■ CitizenSystem (시민 시뮬레이션)

역할

- 인구 증가/감소
- 시민 계층 관리
- 실업/범죄/교육 시스템과 연계

상호작용

- ResourceManager (실업률, 치안 수치 제공/변경)
 - UpgradeSystem (교육률 조건 제공)
 - EventManager로 시민 변화 이벤트 발행
-

■ UpgradeSystem (건물 업그레이드/다운그레이드)

역할

- ResourceManager와 시민 수치 변화를 받아
- 건물 업그레이드 조건 판단 및 수행

상호작용

- EventManager로부터 ResourceChanged, CitizenChanged 등 이벤트 수신
 - EventManager에 UpgradeOccurred 이벤트 발행
 - BuildSystem(혹은 건물 컴포넌트)에게 업그레이드 요청
-

■ UI LAYER

■ UIManager

역할

- 메뉴, HUD, 패널, 건설 패널 등 관리
- UI 버튼 → StateManager 상태 변경
- UI 이벤트는 대부분 EventManager로 전달

상호작용

- InputManager로부터 UI 클릭 여부 전달
- StateManager 변경
- EventManager를 통해 BuildSystem/ResourceManager에게 UI 명령 전달

🌸 전체 시스템 상호작용 요약

From	To	방식	설명
InputManager	EventManager	이벤트	클릭/드래그/줌 등 입력 전달
UIManager	StateManager	직접 호출	모드 변경
StateManager	InputManager	직접 읽기	입력 처리 방식 결정
GridSystem	EventManager	이벤트	클릭 지점 → 그리드 정보
EventManager	BuildSystem	이벤트	건물 설치 명령
BuildSystem	GridManager	직접 데이터 저장 점유 셀 저장	
BuildSystem	EventManager	이벤트	건물 설치됨
ResourceManager	EventManager	이벤트	자원/수치 변경
UpgradeSystem	ResourceManager	직접 읽기	업그레이드 조건 판단
UpgradeSystem	BuildSystem	이벤트	업그레이드 실행
SaveLoadManager	모든 매니저	직접 호출	현재 게임 데이터를 모아서 저장