**[설명]**

Unity 기반 간단한 어드벤처 게임 예제입니다.

VContainer를 활용한 의존성 주입 구조와 대화/선택지 시스템, 간단한 세이브/로드, 인게임 메뉴 등이 구현되어 있습니다.

**[실행 가이드]**

1. Unity 6000.0.26f1에서 동작을 확인했습니다. (다른 버전에서도 실행 가능하지만 일부 API 차이가 있을 수 있습니다.)

2. 게임 실행

Assets/Scene/0\_StartScene.unity씬을 실행하면 메뉴->게임 진행 순서가 이어집니다.

|  |  |
| --- | --- |
| 키 | 기능 |
| W,A,S,D | 플레이어 이동 |
| Q,E | 카메라 회전 |
| SPACE | 대화 진행/ NPC 상호작용/ 메뉴 선택 |
| ESC | 인게임 메뉴 열기/ 닫기 |

**[주요 클래스 & 기능 요약]**

1. 의존성 주입(DI)

|  |  |
| --- | --- |
| 클래스 | 기능 |
| RootLifetimeScope | 게임 전역에서 사용되는 매니저(IMainStateManager, ISaveLoadManager, GameSwitch)를 싱글턴으로 등록 |
| GameLifetimeScope | 인게임에 필요한 매니저, UI 컨트롤러, 플레이어/루트 객체 등록 |

2. 상태 관리

|  |  |
| --- | --- |
| 클래스 | 기능 |
| MainStateManager | 메뉴와 게임 등 상위 상태 전환 |
| GameStateManager | 인게임 내부 상태(Dialogue, Select, InGameMenu 등) 전환 |
| StateMenu, StateGame | 씬 로드/해제와 초기화 과정을 담당 |
| DefaultState | 플레이어 이동/조작이 가능한 기본 상태 |
| DialogueState | 대화 UI 표시, 대화 진행 |
| SelectState | 선택지 UI 표시, 결과에 따른 스위치 ON/OFF |
| InGameMenuState | 인게임 메뉴 UI 표시 |

3. 게임 엔진/플레이어

|  |  |
| --- | --- |
| 클래스 | 기능 |
| GameRoot | DontDestroyOnLoad로 유지되는 게임 Root오브젝트, IGameRoot인터페이스 제공 |
| PlayerController | 이동.카메라 조작, NPC 충돌 감지 후 대화 실행 |
| ZombieController | ScriptableObject 기반 이벤트 실행(대화/선택지) |
| PlayerProvider | DI로 플레이어 인스턴스를 제공 |
| SceneChageTrigger | 트리거 충돌 시 다른 게임씬으로 전환 |
| CheckOnOffManager,  EventCheckOnOff | 스위치 값에 따라 오브젝트 활성/비활성 제어 |
| GameSwitch | bool 스위치 배열을 관리, 이벤트 체크,초기화 |
| SaveLoadManager | JSON 기반 게임 상태 세이브/로드 및 스폰 위치 적용 |

4. UI

|  |  |
| --- | --- |
| 클래스 | 기능 |
| UIDialogue | 대화 UI 표시 및 페이징 처리 |
| UISelect | 선택지 UI 표시, SPACE로 확정, A/D/W/S로 선택 |
| UIInGameMenu | 인게임 메뉴 표시/숨김, 저장/로드/종료 선택 가능 |
| UINameMarker | NPC 이름을 화면에 표시 |
| ScreenMaskController | 화면 전환 시 페이드 인/아웃용 마스크 제어 |
| UIMediator 및  각 Controller | UI 간 중재자 패턴.  Dialogue/Select/Menu/NameMarker 컨트롤러들을 묶어 전달 |

5. 스크립터블 오브젝트

|  |  |
| --- | --- |
| 파일 | 목적 |
| Scripts | 대화 텍스트, 조건, 후속 이벤트 정보 |
| Selects | 두 개의 선택지와 해당 스위치 결과를 저장 |
| PlayerSpawnPos | 씬별 플레이어 스폰 위치. 회전 정보 |

**[참고사항]**

* 모든 UI와 스크립트는 VContainer를 통해 필요한 매니저/서비스를 주입받습니다.
* SaveLoadManager는 Application.persistentDataPath에 save.json 파일을 생성하여 게임 데이터를 저장합니다.
* GameSwitch 스위치 값을 이용해 이벤트 결과에 따른 오브젝트 활성화.비활성화를 처리하며, 세이브 파일에도 포함됩니다.