

UGESystem 커스텀 아키타입(Archetype) 추가 가이드

● [매우 중요] 패키지 업데이트 시 초기화 경고

본 가이드는 Assets/UGESystem/ 폴더 내의 핵심 스크립트(GameEventArchetype.cs, UEGameEventController.cs 등)를 직접 수정하는 방법을 설명합니다.

UGESystem 패키지를 업데이트할 경우, 이 파일들은 덮어쓰워지므로 사용자의 모든 수정 사항이 초기화(삭제)됩니다.

패키지 업데이트 후에는 이 가이드를 참고하여 반드시 변경 사항을 다시 적용(Re-apply)해야 함을 잊지 마시기 바랍니다.

이 문서는 UGESystem에 새로운 게임 플레이 모드(아키타입)를 추가하는 방법을 설명합니다.

🔊 웹 스토리 메이커 관련 안내

- 커스텀 아키타입은 웹 툴에서는 지원되지 않습니다.
- Unity 에디터에서만 생성 및 설정이 가능합니다.

목표: 무엇을 만드나요?

예제로 "탐험 모드 (Exploration)"를 만들어 보겠습니다. 이 모드에서는 복잡한 대화나 선택지 없이, 간단한 독백이나 사물 관찰 텍스트만 나오도록 제한된 환경을 구성해 봅니다.

1단계: 이름표 만들기 (Enum 등록)

시스템에 새로운 모드가 생겼음을 알려주기 위해 두 군데 파일에 이름을 등록해야 합니다.

1. 파일 1 열기: Assets/UGESystem/Core/Scripts/UGESystem/GameEvents/Enums/GameEventArchetype.cs
2. 코드 추가:

```
public enum GameEventArchetype
{
    Generic,
    Dialogue,
    CinematicText,

    // [추가] 새 아키타입 이름
    Exploration
}
```

3. 파일 2 열기: Assets/UGESystem/Core/Scripts/UGESystem/GameEvents/Enums/GameEventType.cs
4. 코드 추가: (위와 동일한 철자로 작성하세요)

```
public enum GameEventType
{
    Dialogue,
    CinematicText,

    // [추가]
    Exploration
}
```

2단계: 실행 전략 설정 (Controller 등록)

이 새로운 모드(Exploration)일 때, 각 명령어가 어떻게 작동할지 정의하는 가장 중요한 단계입니다.

1. 파일 열기: Assets/UGESystem/Core/Scripts/UGESystem/GameEvents/Managers/UGEGameEventController.cs
2. 코드 수정: InitializeCommandHandlers 함수 내 _commandHandlers 딕셔너리에 새로운 항목을 추가합니다.

```
private void InitializeCommandHandlers()
{
    // ... 기존 핸들러 생성 코드들 ...

    _commandHandlers = new Dictionary<GameEventType, Dictionary<Type, ICommandHandler>>
    {
        // ... 기존 Dialogue, CinematicText 설정들 ...

        // [추가] 탐험 모드 정의
        {
            GameEventType.Exploration, new Dictionary<Type, ICommandHandler>
            {
                // 이 모드에서 *허용할 커맨드*만 골라서 등록합니다.

                { typeof(DialogueCommand), dialogueHandler },
                { typeof(CharacterCommand), characterHandler },
                { typeof(PlaySoundCommand), playSoundHandler },
                { typeof(EndCommand), endHandler }
            }
            // 필요한 만큼 추가하세요.
        }
    };
}
```

3단계: 확인 및 활용

1. Unity 에디터로 돌아가 컴파일을 완료합니다.
2. 프로젝트 창에서 Create > UGESystem > New Game Event를 선택합니다.
3. 인스펙터 창의 **Archetype** 드롭다운을 눌러보세요. **Exploration**이 목록에 나타납니다.
4. 이것을 선택하면, 이제 해당 이벤트는 2단계에서 정의한 규칙(딕셔너리에 넣은 커맨드들)에 따라서만 동작하게 됩니다.

심화: 에디터 메뉴에서 특정 커맨드 숨기기 (선택 사항)

특정 커맨드가 이 아키타입을 선택했을 때 아예 **Add Command** 메뉴에 안 뜨게 하고 싶다면?

1. 파일 열기: 제한하고 싶은 커맨드 스크립트 (예: ChoiceCommand.cs)
2. 속성 추가: 클래스 위에 [AvailableIn] 속성을 사용하여 허용할 아키타입을 지정합니다.

```
// 예: 선택지는 Dialogue 모드에서만 쓰고, Exploration에서는 메뉴에 안 뜨게 하고 싶다.
[AvailableIn(GameEventArchetype.Dialogue)]
public class ChoiceCommand : EventCommand { ... }
```