UGESystem 커스텀 시작 조건(Start Condition) 추가 가이드

> 🔘 **[매우 중요] 패키지 업데이트 시 초기화 경고**
>
> 이 가이드는 편의를 위해 `Assets/UGESystem/` 폴더 내에 파일을 생성하도록 안내하고 있습니다.
> **UGESystem 패키지를 업데이트하면 이 폴더 내의 변경 사항(새로 만든 파일 포함)이 초기화(삭제)될 수 있습니다.**
>
> 안전을 위해 커스텀 코드는 가급적 **패키지 폴더 밖(예: `Assets/MyGame/Scripts/`)**에 작성하거나,
> 패키지 업데이트 전에 **반드시 별도로 백업**해 두시기 바랍니다.

이 문서는 사용자가 UGESystem의 스토리보드 실행 조건을 확장하여, 게임 내 다양한 상황에 맞춰 이벤트가 자동으로 시작되도록 만드는 방법을 설명합니다.

> 📢 **웹 스토리 메이커 관련 안내**
>
> - 커스텀 시작 조건은 **웹 툴에서는 지원되지 않습니다.**
> - 오직 **Unity 에디터**의 스토리보드 에디터(Storyboard Editor) 윈도우에서만 설정할 수 있습니다.

목표: 무엇을 만드나요?

예제로 **"플레이어의 체력이 30% 이하로 떨어지면 자동으로 시작되는 이벤트 조건 (`HealthLowCondition`)"**을 만들어 보겠습니다.

1단계: 저장용 데이터 상자 만들기 (DTO 클래스)

조건을 파일에 저장하고 불러올 때 사용할 데이터 구조(DTO)를 만듭니다.

1. **폴더 이동**: `Assets/UGESystem/Core/Scripts/UGESystem/GameEvents/Data/Storyboard/Conditions/` (또는 여러분의 커스텀 폴더)
2. **파일 생성**: `HealthLowConditionDto.cs` (C# 스크립트)
3. **코드 작성**: `BaseEventConditionDto`를 상속받아 작성합니다.

```
using Newtonsoft.Json;

namespace UGESystem
{
    // 파일 저장 및 로드를 담당하는 데이터 클래스입니다.
    public class HealthLowConditionDto : BaseEventConditionDto
    {
        // 체력이 몇 % 이하일 때 발동할지 저장하는 변수
        [JsonProperty] public float ThresholdPercentage { get; set; }

        public override AbstractEventCondition ToCondition()
        {
            // 이 DTO를 실제 로직 클래스(Condition)로 변환합니다.
            return new HealthLowCondition(this);
        }
    }
}
```

2단계: 조건 로직 만들기 (Condition 클래스)

실제로 게임 내에서 상황을 감시하고 조건을 충족시키는 핵심 로직을 만듭니다. **여러분의 게임 프로젝트에 있는 체력 시스템과 연결하는 작업이 필요합니다.**

1. **폴더 이동**: 위와 동일 (또는 여러분의 커스텀 폴더)
2. **파일 생성**: `HealthLowCondition.cs` (C# 스크립트)
3. **코드 작성**: `AbstractEventCondition`을 상속받아 작성합니다.

```
using UnityEngine;

namespace UGESystem
{
```

```
    [System.Serializable]
    public class HealthLowCondition : AbstractEventCondition
    {
        // 인스펙터에서 설정할 임계값 (0.0 ~ 1.0)
        [SerializeField] private float _thresholdPercentage;

        // 1. 기본 생성자 (에디터에서 새로 추가할 때 호출됨)
        public HealthLowCondition() : base("Player health is low")
        {
            _thresholdPercentage = 0.3f; // 기본값 30%
        }

        // 2. 로드 생성자 (파일에서 불러올 때 호출됨)
        public HealthLowCondition(HealthLowConditionDto dto) : base(dto)
        {
            _thresholdPercentage = dto.ThresholdPercentage;
        }

        // 3. 저장 메소드 (파일로 저장할 때 호출됨)
        public override BaseEventConditionDto ToDto()
        {
            return new HealthLowConditionDto
            {
                Description = Description,
                ThresholdPercentage = _thresholdPercentage
            };
        }

        // 4. [핵심] 이벤트 감지 시작 (Subscribe)
        public override void Subscribe(System.Action onStateChanged)
        {
            base.Subscribe(onStateChanged);

            // [사용자 구현 필요] 여러분의 게임에 있는 플레이어 체력 스크립트의 이벤트에 연결하세요.
            // 예시: PlayerHealth.Instance.OnHealthChanged += CheckHealth;
        }

        // 5. 이벤트 감지 중단 (Unsubscribe) - 메모리 누수 방지
        public override void Unsubscribe()
        {
            base.Unsubscribe();

            // [사용자 구현 필요] 연결했던 이벤트를 반드시 해제해야 합니다.
            // 예시: PlayerHealth.Instance.OnHealthChanged -= CheckHealth;
        }

        // 6. [사용자 구현 필요] 체력이 변할 때마다 호출되어 조건을 검사하는 함수
        private void CheckHealth(float currentHealth, float maxHealth)
        {
            // 이미 조건이 충족되었다면 중복 실행을 막기 위해 리턴
            if (IsMet) return;

            float percentage = currentHealth / maxHealth;

            // 현재 체력이 설정한 값(예: 30%) 이하인지 확인
            if (percentage <= _thresholdPercentage)
            {
                IsMet = true; // 조건 달성!

                // [선택 사항] 게임플레이 정지 처리
                // 이벤트가 시작되기 직전, 캐릭터가 죽거나 다른 행동을 하지 못하도록
                // 여기서 입력을 막거나 무적 상태로 만드는 처리를 할 수 있습니다.

                _onStateChanged?.Invoke(); // 시스템에 "이벤트 시작하세요!"라고 알림
            }
        }
    }
}
```

3단계: 확인하기

커맨드와 달리, 조건 클래스는 별도의 컨트롤러 등록 과정이 필요 없습니다. 클래스만 존재하면 시스템이 자동으로 인식합니다.

1. Unity 에디터로 돌아가 컴파일을 완료합니다.
2. 상단 메뉴에서 `Window` > `UGESystem` > `Storyboard Editor`를 엽니다.
3. 그래프 상의 아무 노드나 클릭하여 선택합니다.
4. 인스펙터 창의 `Start Conditions` 섹션에서 `+` 버튼을 누릅니다.
5. 드롭다운 목록에 **HealthLowCondition**이 나타나는지 확인하고 선택합니다.
6. `Threshold Percentage` 값을 설정합니다.

이제 플레이어의 체력이 설정한 값 이하로 떨어지면, 해당 노드의 이벤트가 자동으로 실행됩니다.