A cinematic scene from the video game The Last of Us. A female character, Ellie, is seen from behind, looking out over a desolate, rocky landscape. In the distance, the ruins of a city are visible, including a large, partially collapsed structure. The sky is filled with dramatic, dark clouds, and the sun is setting, casting a warm, orange glow over the scene.

게임 엔진

LEC 04 블루프린트 (2)



한국공학대학교
TECH UNIVERSITY OF KOREA

이대현 교수

LEC 04 복습

블루프린트 (클래스) 설계 삼요소

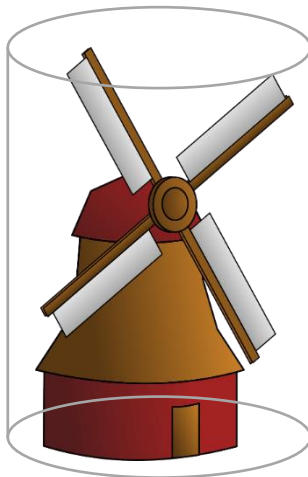
- 블루프린트 클래스를 설계할 때, 세가지 내용을 담아야 함.
 - 액터가 어떻게 보일 것인가? → 외형
 - 액터는 어떤 식으로 행동하는가? → 행위, 로직
 - 액터의 물리적인 특성은? → 물리, 존재영역(예. 충돌영역)



풍차의 외형



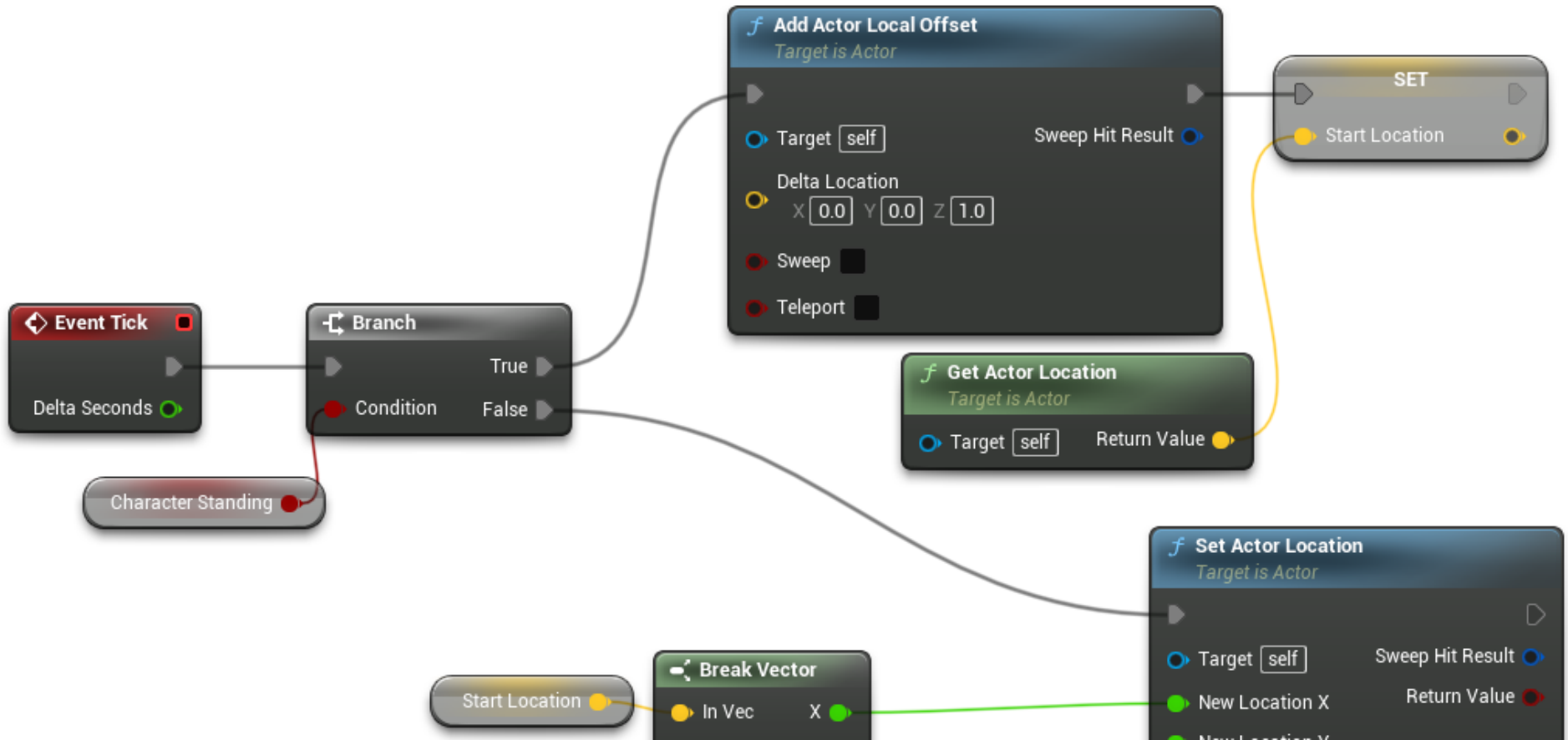
날개의 회전 로직



풍차의 충돌 영역

블루프린트 설계 요령

- 그대로 사용할 수 있는 블루프린트가 있으면?
 - 새로 만들지 말고, 기존 블루프린트를 사용
- 비슷한 블루프린트가 있으면?
 - 뺄 것도 있고, 추가할 것도 있으면? " 복제 " 한 후, 수정.
 - 추가할 것만 있으면? " 파생(또는 상속) " 한 후, 기능 추가 구현.
 - 기능을 추가할 경우, "컴포넌트"로 이미 기능이 구현된 것이 있는가 확인!!
- 완전히 새로운 기능을 갖는 블루프린트라면?
 - "Actor" 를 베이스 클래스로 한, 블루 프린트 클래스 새롭게 제작.



학습 내용

- **컨스트럭션 스크립트란?**
- **컨스트럭션 스크립트의 실행 구조**

- **실습 - GPS 블루프린트 제작**
- **실습 - 덤불 자동 심기 블루프린트 제작**
- **실습 - 담장 자동 생성 블루프린트 제작**

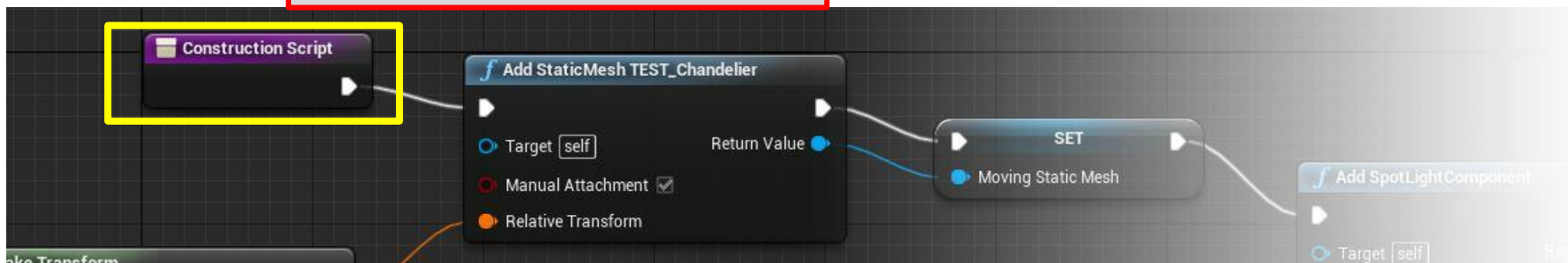
컨스트럭션 스크립트(Construction Script)

- 블루프린트 클래스의 구성을 프로그래밍적으로 하기 위한 스크립트.
- 블루프린트 안에 컴포넌트들을 효과적으로, 자동적으로 담음.

가 100 가 . 가

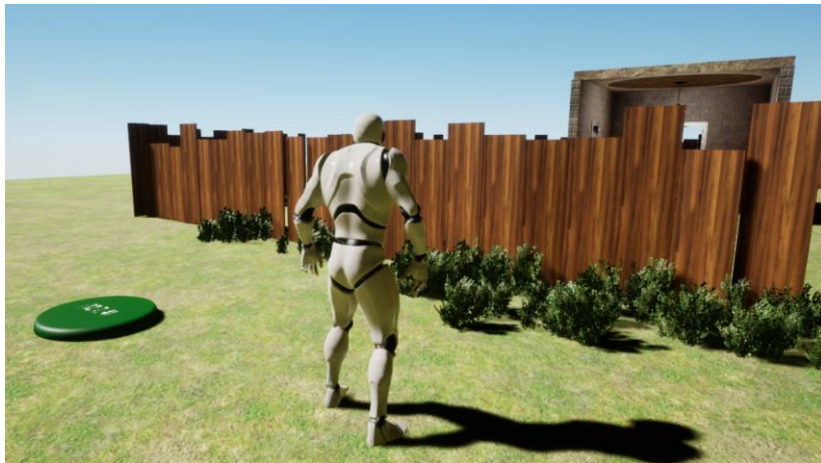
C++ ()가 가 ? .

Construction Script 노드를
시작점으로 하는 함수를 작성



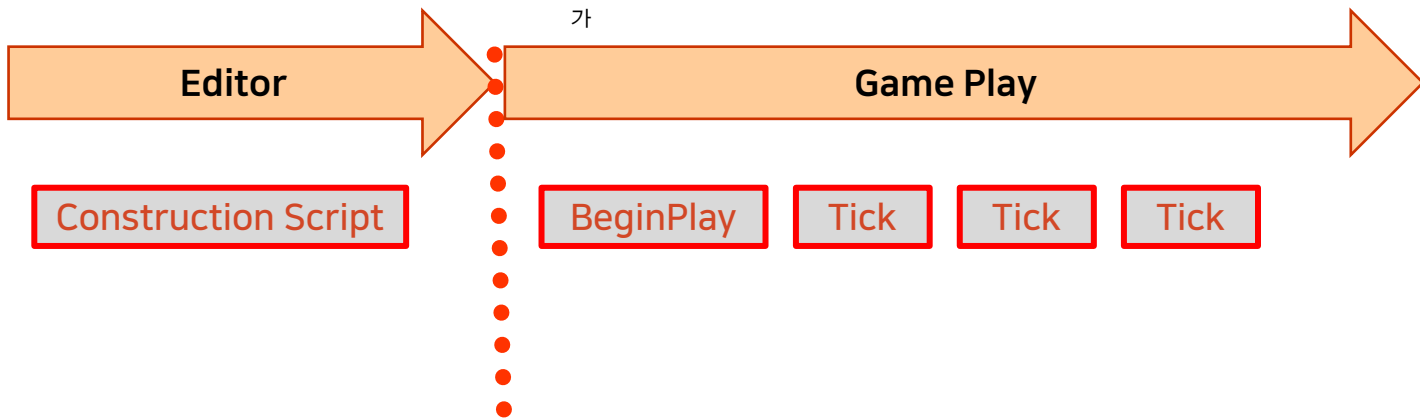
컨스터렉션 스크립트 활용

- 대량의 컴포넌트들로 구성된 블루프린트 클래스의 개발
 - 예) 수풀, 담장
- 유연한 블루프린트 클래스의 개발
 - 각기 다른 속성을 갖는 액터들을 생성할 수 있음.
 - 예) 키가 다른 적군 NPC
- 게임 플레이 시작 전에 준비되어야 할 요소를 갖추는데 활용
- 레벨 구성의 자동화



컨스트럭션 스크립트의 실행

- 컨스트럭션 스크립트는 에디터 상에서만 실행되고 반영됨.
 - 액터가 생성되는 시점(레벨 상에 액터를 끌어다가 놓을 때), 또는 액터의 속성 변경이 일어날 때(이동, 회전 등등)마다 실행됨.
 - 실제로 게임 플레이가 시작되면, BeginPlay event 가 호출되어, 기본적인 초기화를 수행하고, 이어서 게임 루프에서 반복적으로 계속해서 Tick event를 호출함.





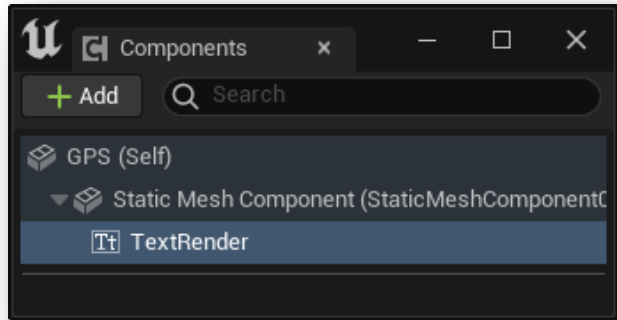
실습

GPS 블루프린트

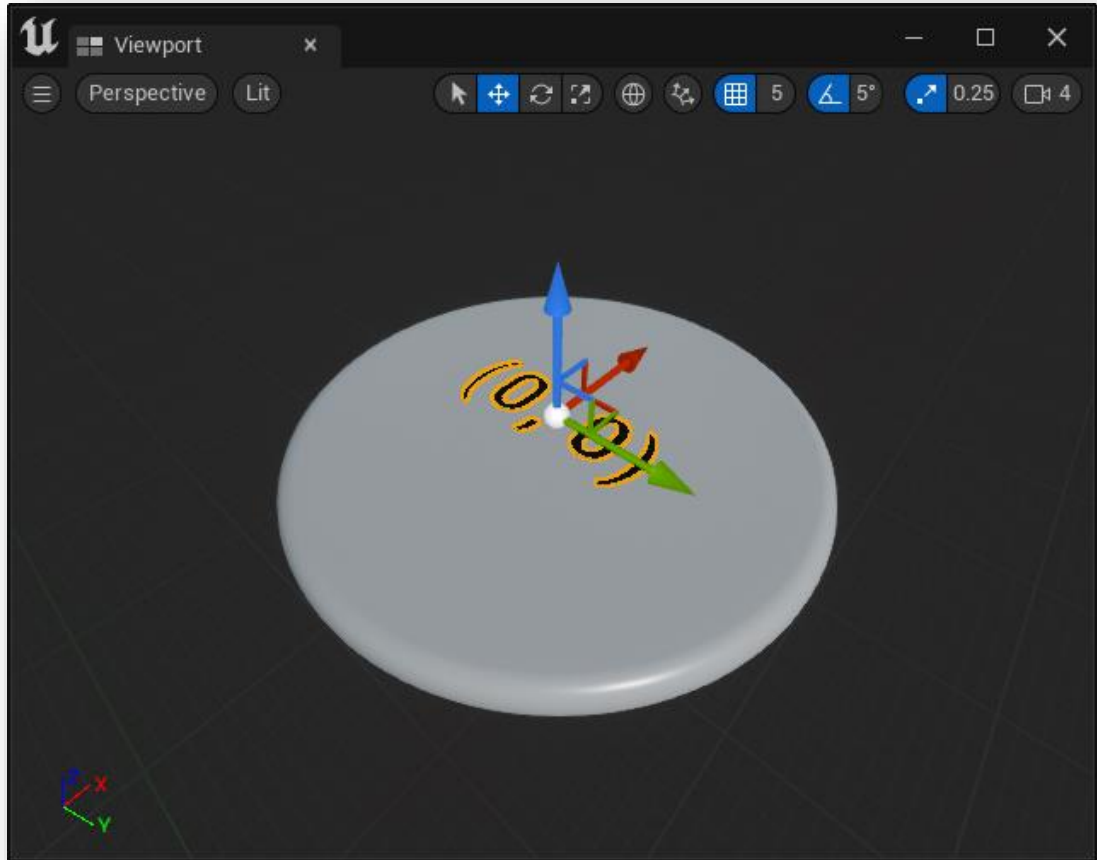
실습 목표

- 컨스트럭션 스크립트를 이용한 액터의 위치 표시.
- 컨스트럭션 스크립트의 역할과 구동 시점 이해.

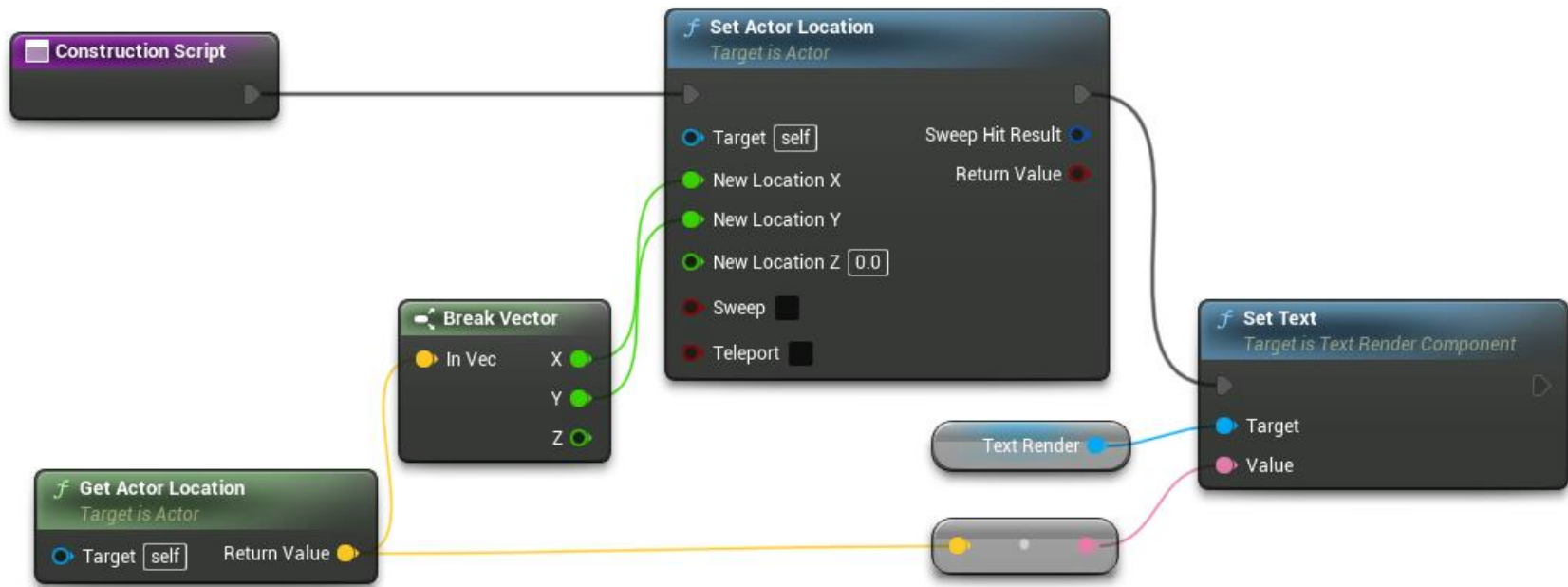




속성	값
Location	0, 0, 15
Rotation	180, 90, 0
Text	(0, 0)
Horizontal Alignment	Center
Vertical Alignment	Text Center
World Size	20

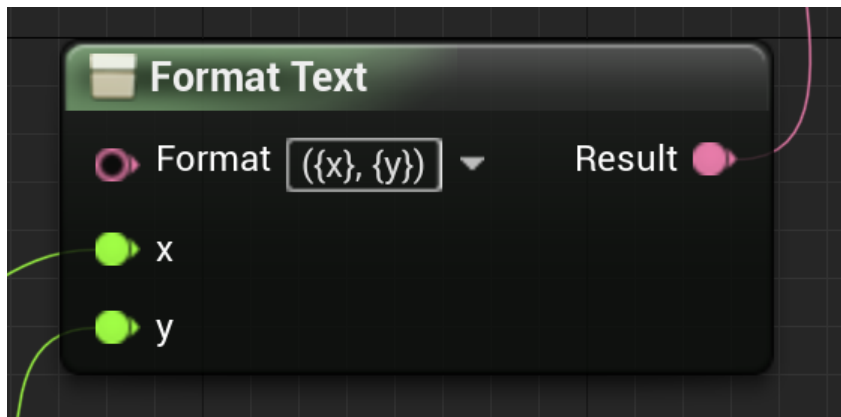


Construction Script



텍스트 출력 형식 설정 - Format Text 노드 함수

- 텍스트를 정해진 형식에 맞춰서 출력할 수 있음.
- {} 기호로 입력핀을 정의함.



x=80이고, y=55 인 경우,
(80, 55)가 출력됨.

Event Graph 안에 넣으면 어떤 차이가 있을까?

Text 와 String

FText

FText represents a display string. Any text you want to display to the user should be handled with FText. The FText class has built-in support for localization, and can handle text content that is localized and stored in a lookup table, as well as text that is localized at runtime, such as numbers, dates, times, and formatted text. Even text that does not need to be localized can be handled with FText. This includes user-entered content such as a player's name, and any text displayed with Slate. FText does not offer any mutation functions, because making changes to display strings is a very unsafe operation.

- [FText Reference Guide](#)

FString

Unlike FName and FText, **FString** is the only string class that allows for manipulation. There are many methods available for string manipulation, including case changing, excerpting substrings, and reversing. FString can be searched, modified, and compared against other strings. However, these manipulations can make FString more expensive than the immutable string classes.

- [FString Reference Guide](#)



실습

숲 덩불 자동 심기(BushPainter)

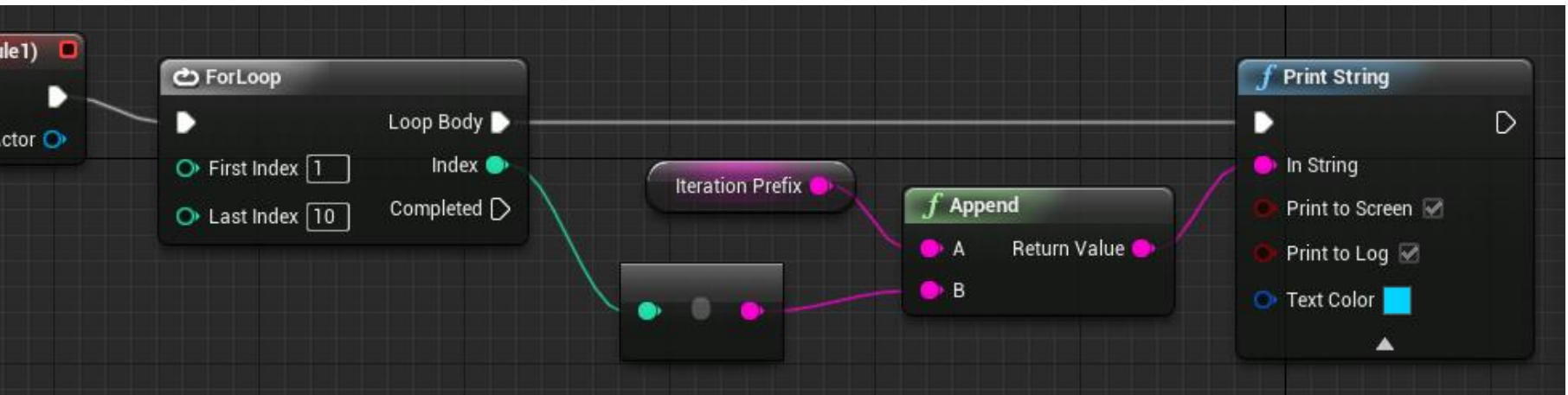
실습 목표

- 덤불을 효과적으로 심는 블루프린트 제작
- 컨스트럭션 스크립트를 활용한 액터 생성의 자동화 방법 이해.



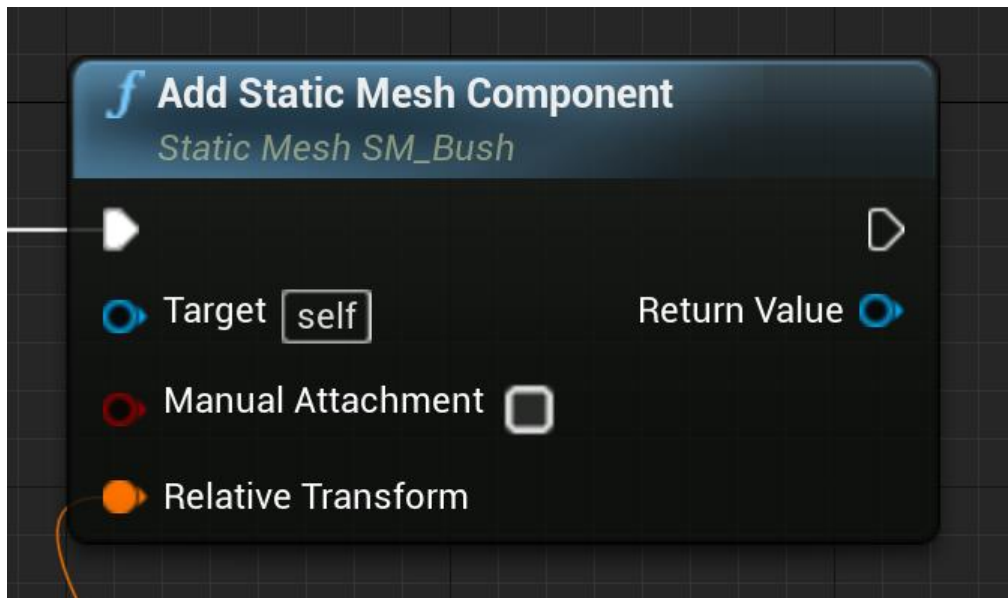
For Loop 노트

- 어떤 작업을 반복적으로 수행.
- 시작 인덱스부터 끝 인덱스까지 인덱스를 증가.
- 반복 횟수가 너무 크면 성능에 영향을 끼칠 수 있음.



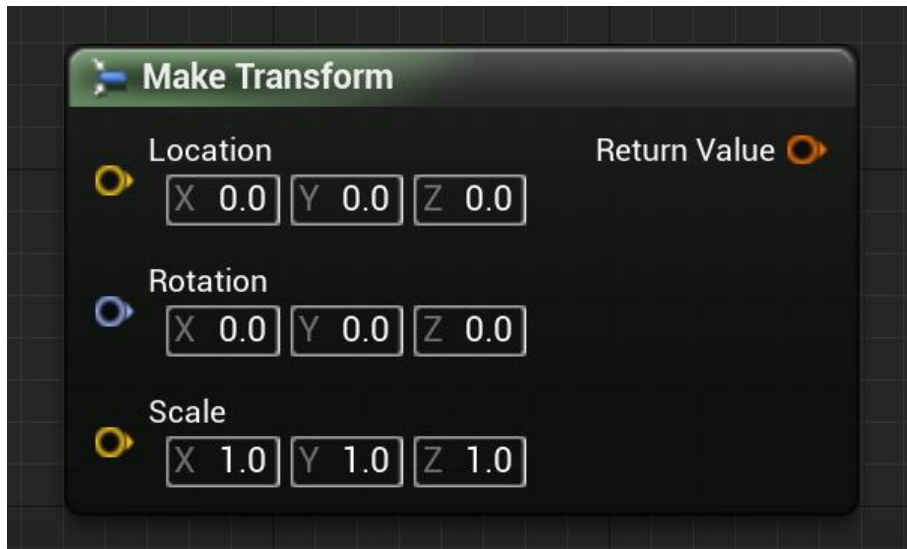
Add Static Mesh Component 노드

- 액터에 스태틱 메시를 추가



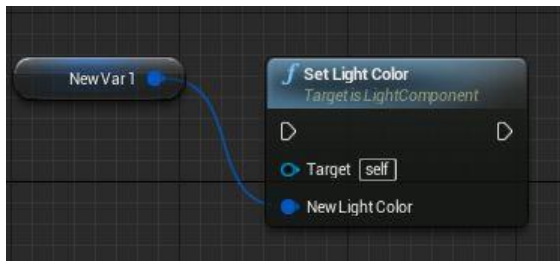
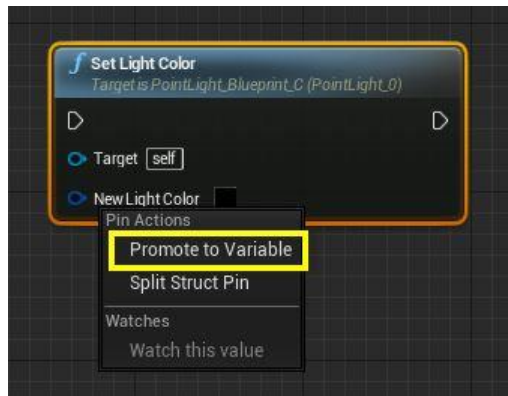
Make Transform 노드

- 이동, 회전, 스케일 값을 모아서 3D 변환값을 계산.



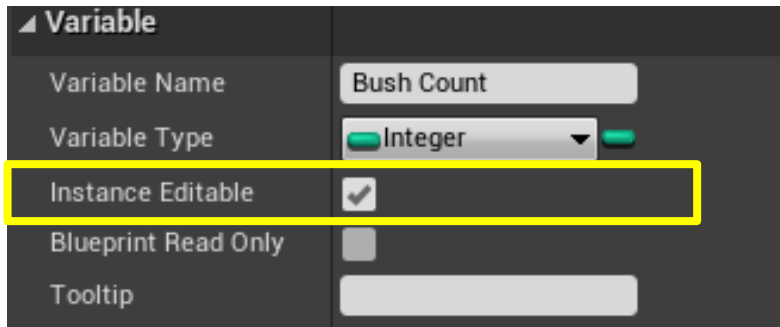
변수 승격

- 변수를 만드는 편리한 방법
- 핀을 클릭 또는 드래그한 후, “Promote to Variable” 선택



변수의 에디터 노출

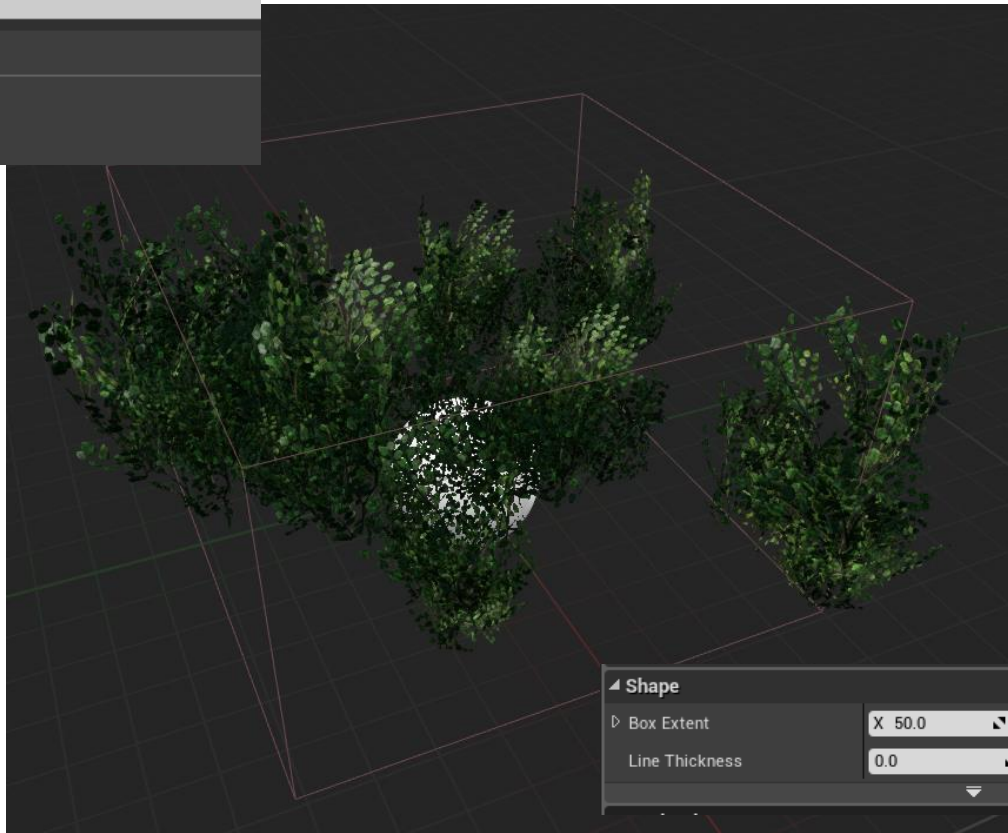
- 변수의 옵션 중, “Instance Editable” 을 체크.
- 에디터 상에서 변수가 노출되면, 액터마다 그 값을 독립적으로 설정할 수 있음.



Components

+ Add Component ▾ Search

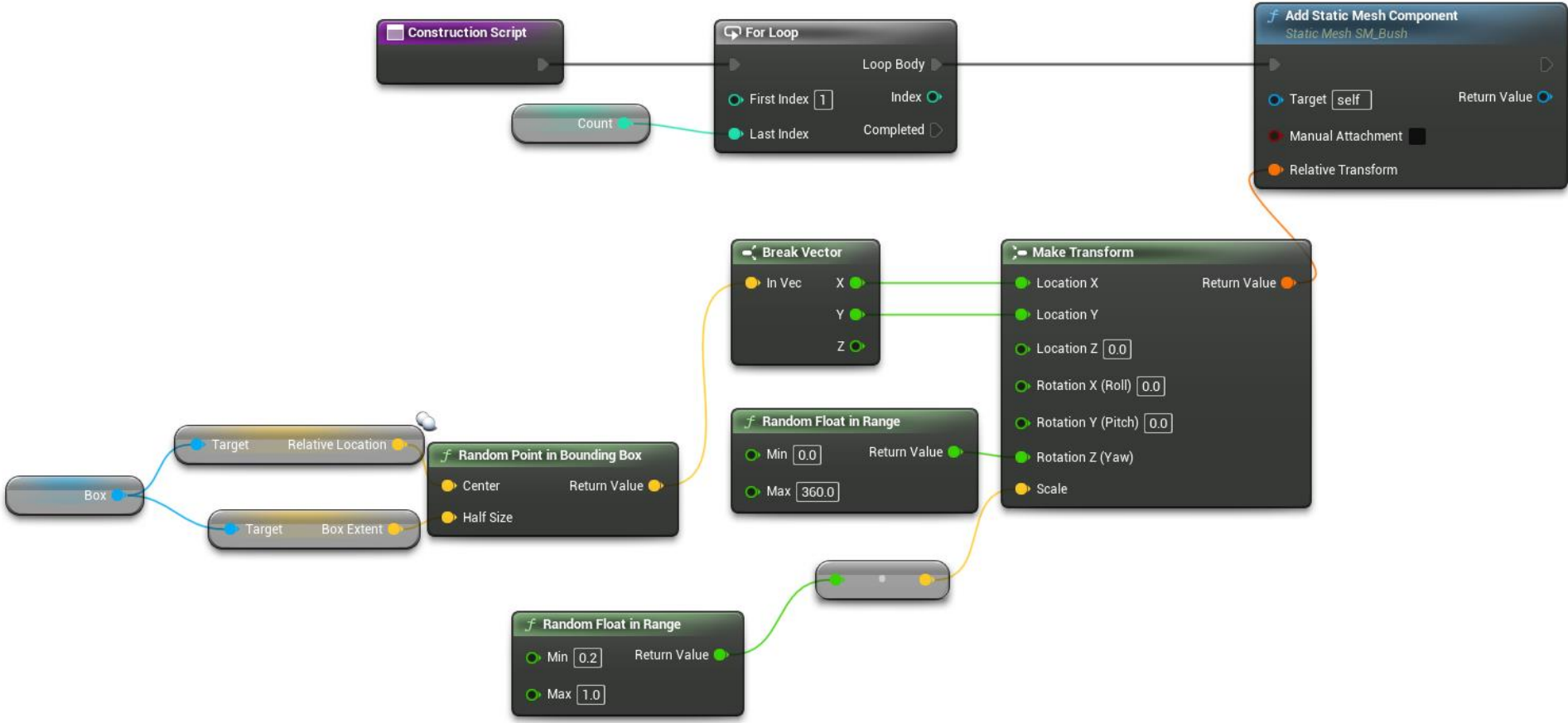
- BushPainter(self)
- DefaultSceneRoot
 - BushArea

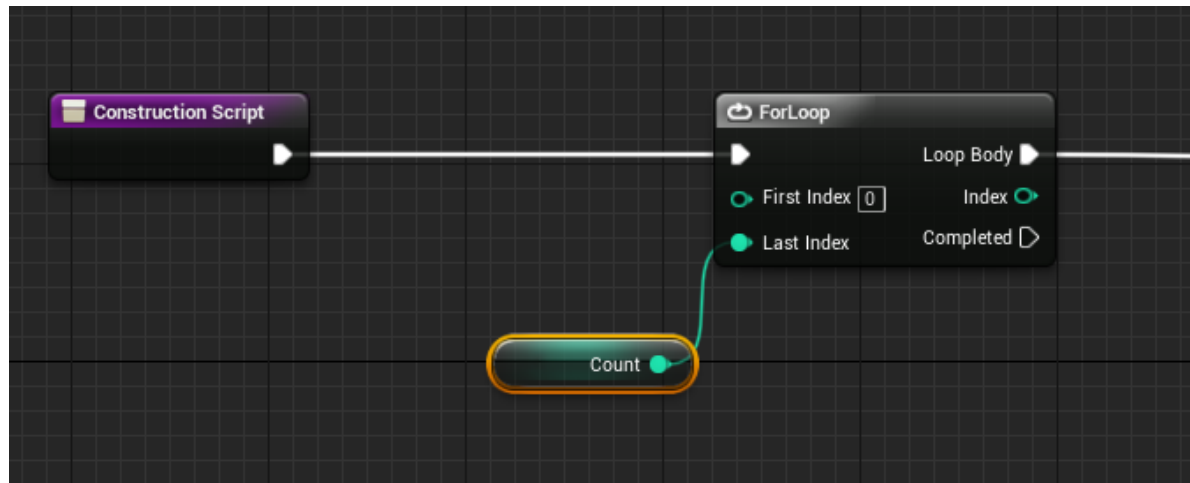
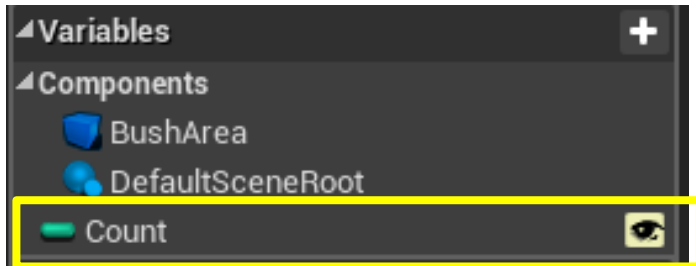


Shape

Box Extent X 50.0 Y 50.0 Z 32.0

Line Thickness 0.0





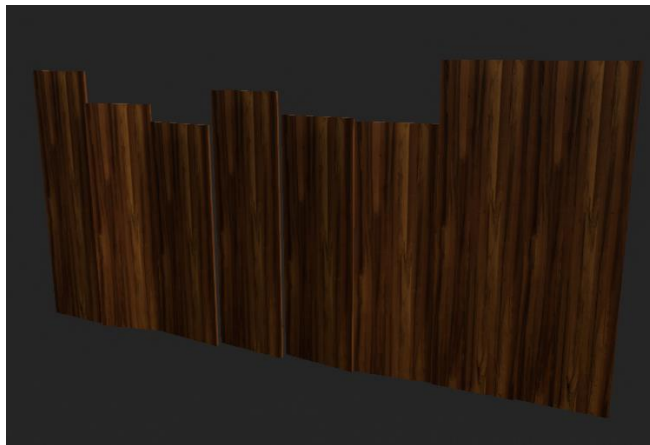


실습

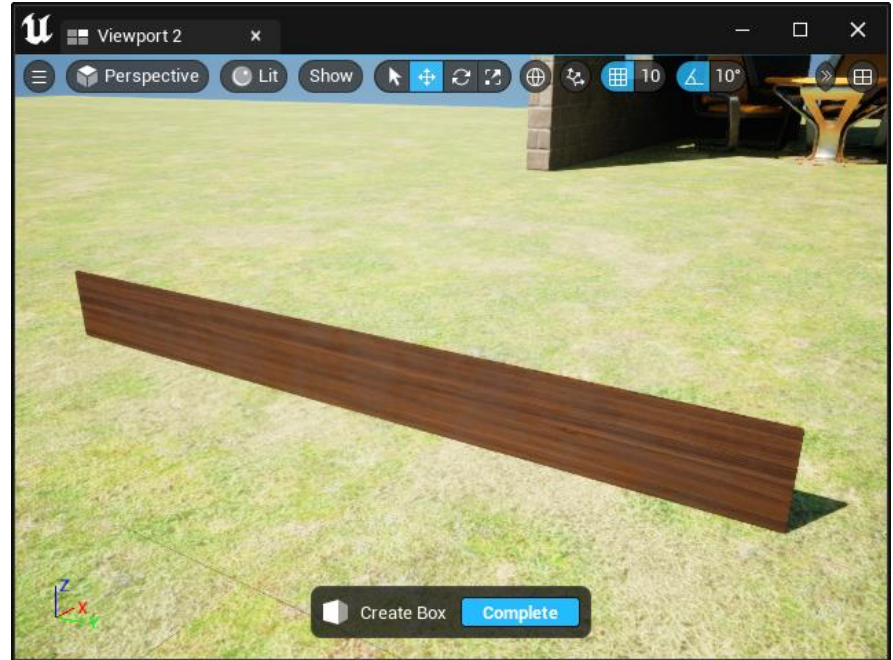
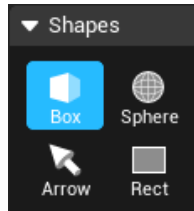
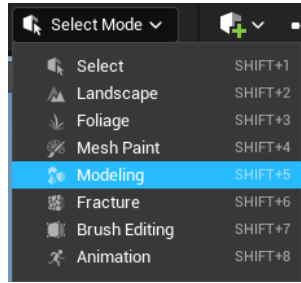
담장 자동 생성 AutoFence 제작

실습 목표

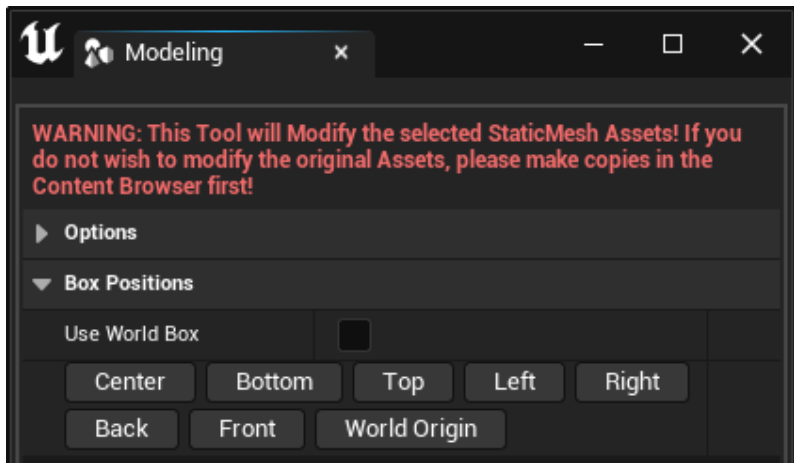
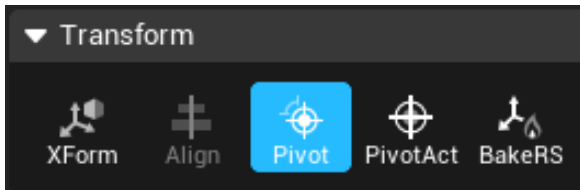
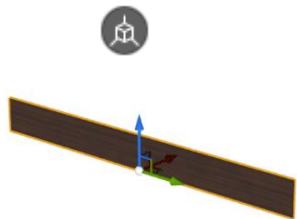
- 판자를 여러 개 모아서 담장을 구성
- 담장의 끝 위치를 자유롭게 조절



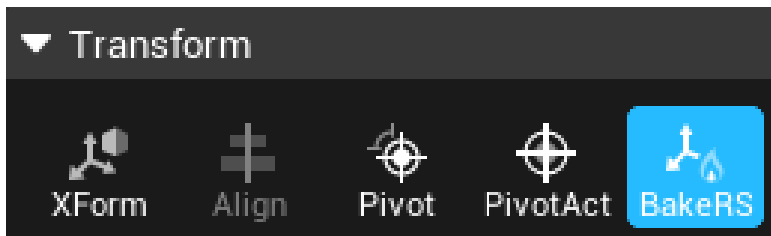
나무 패널 모델



피봇 조절 (위치)



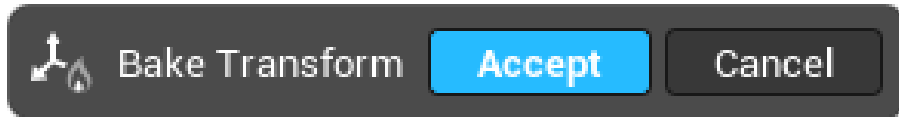
피봇 조절 (회전)



BakeRS -

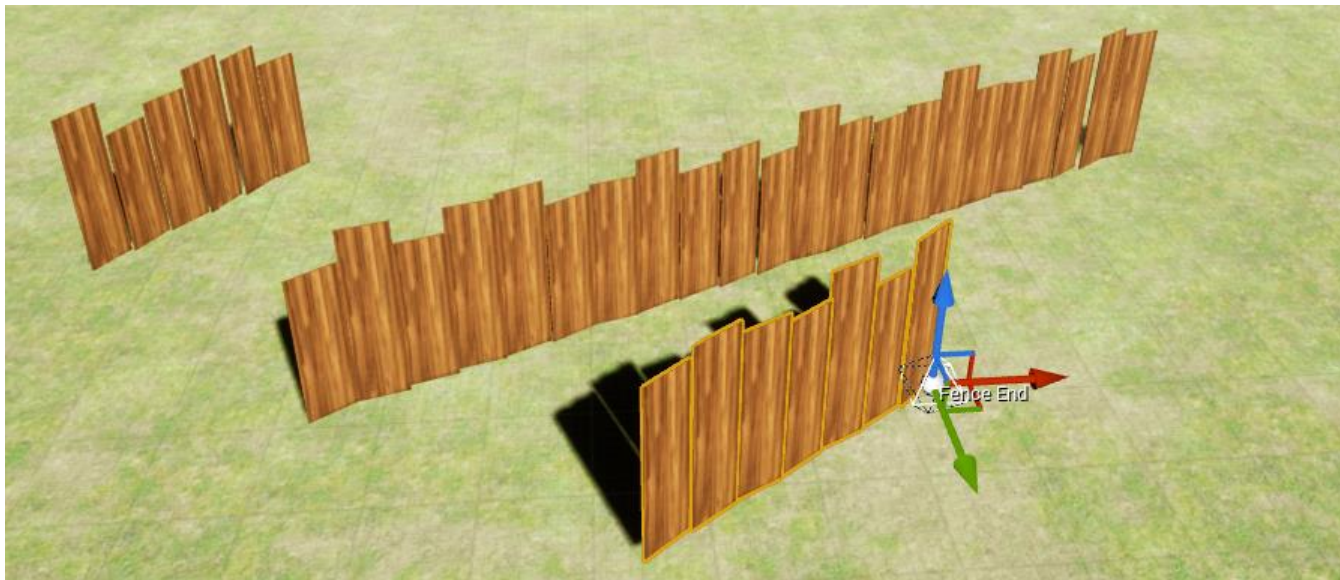
,

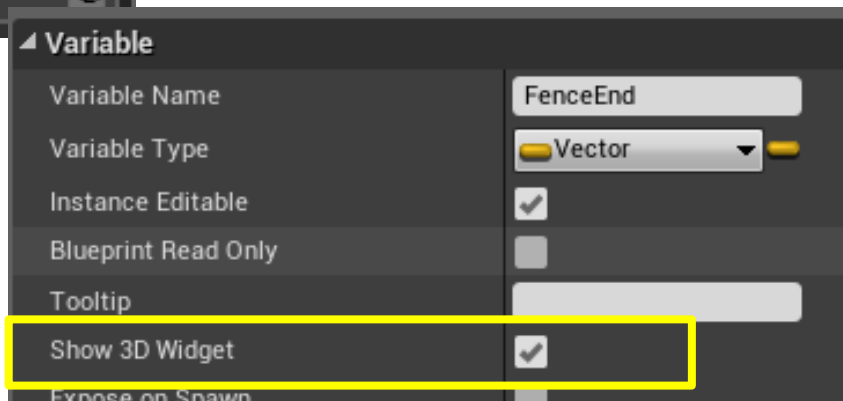
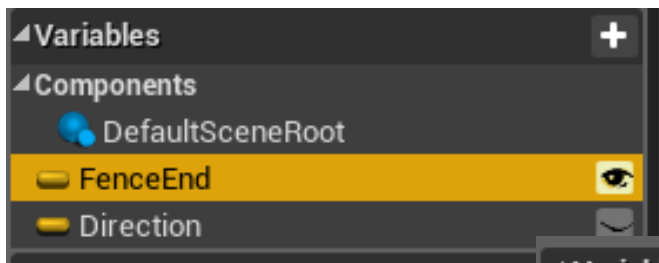
x, y, z



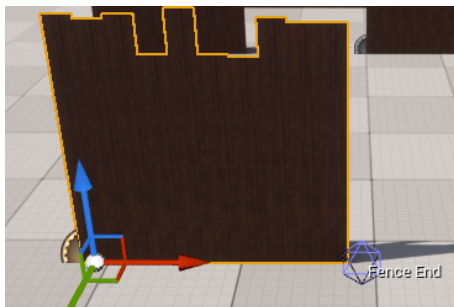
3D 위젯

- 컨스트럭션 스크립트에서 디자이너가 목적으로 하는 3D 위치를 실시간으로 지정하기 위해 사용.
- 담장의 길이와 방향을 시각적으로 조절하면서 생성 가능.





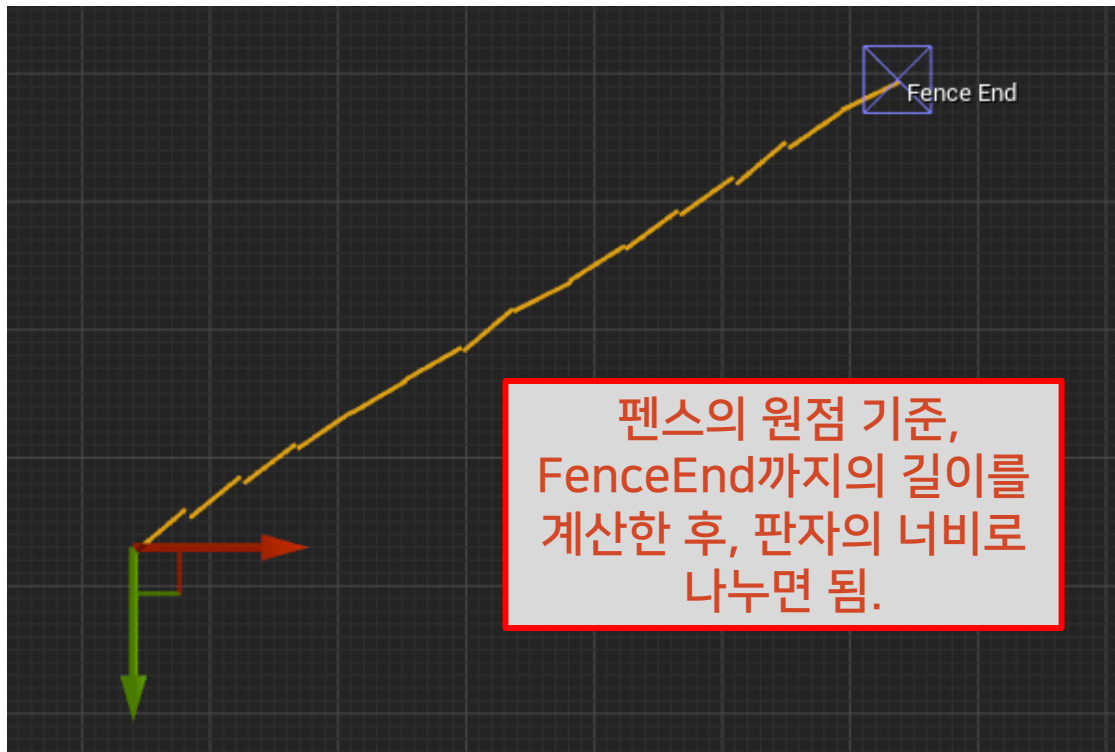
3



Fence End

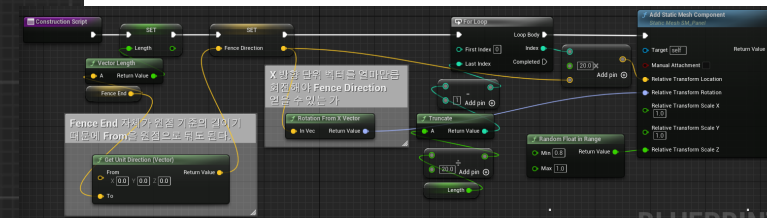
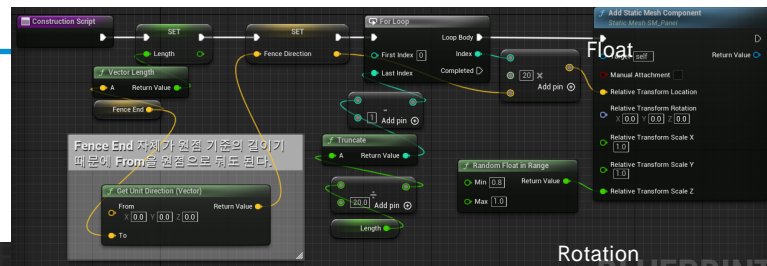
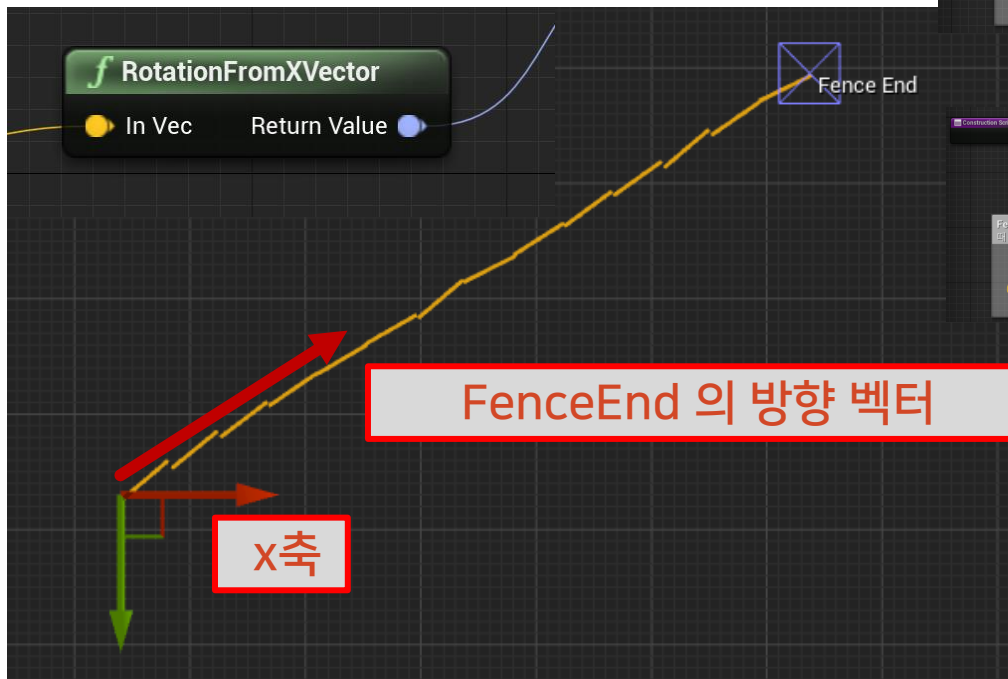
가 .

펜스에 들어갈 판자 개수 계산



RotationFromXVector 노드 함수

- X 축을 기준으로 했을 때, 어떤 벡터의 회전량

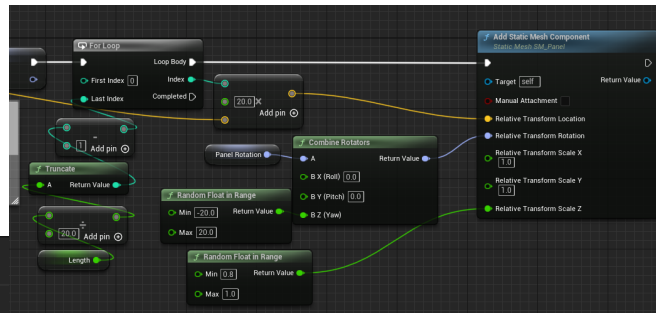
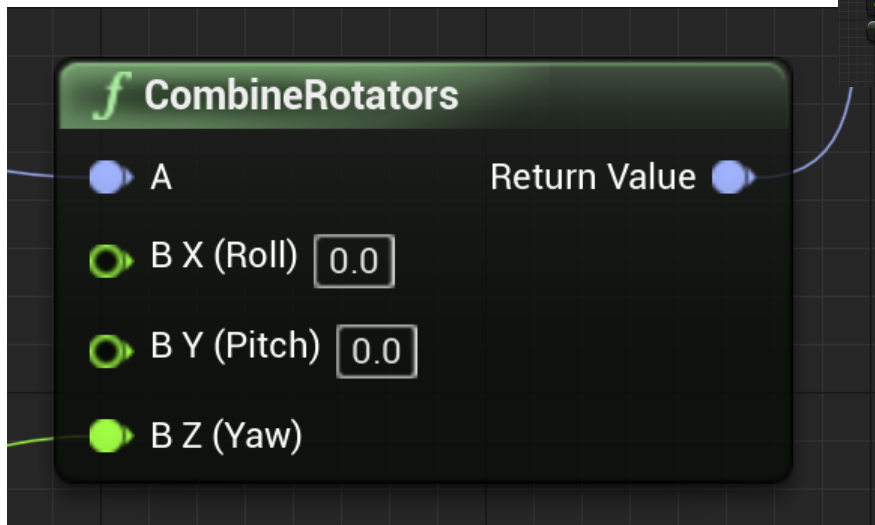


Rotation From X Vector Rotator

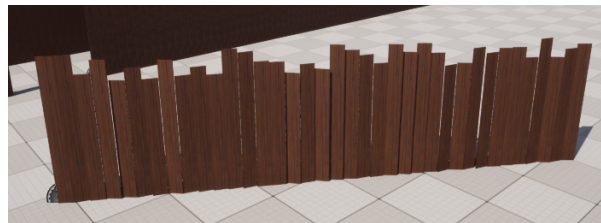
Vector.

CombineRotators

- 두개의 회전의 결합 결과를 한 개의 회전값으로 계산



Panel Rotation 가 Combine Rotation -20~20
Combine Rotation



Construction Script

