게임 엔진

LEC 03 블루프린트

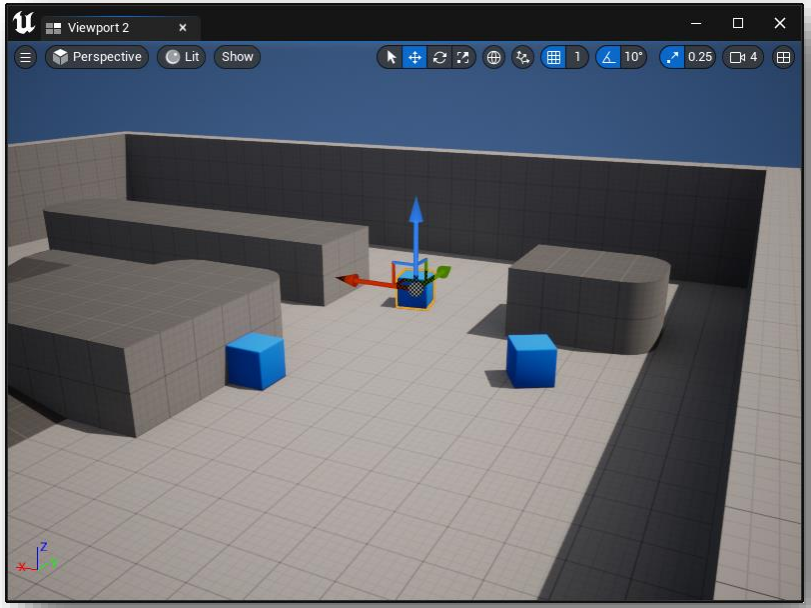
한국공학대학교 TECH UNIVERSITY OF KOREA 이대현 교수

학습 내용

- 블루프린트 개요
- 블루프린트 구성 요소
- 블루프린트 설계 요령
- 블루프린트 제작 실습

액터 움직이기

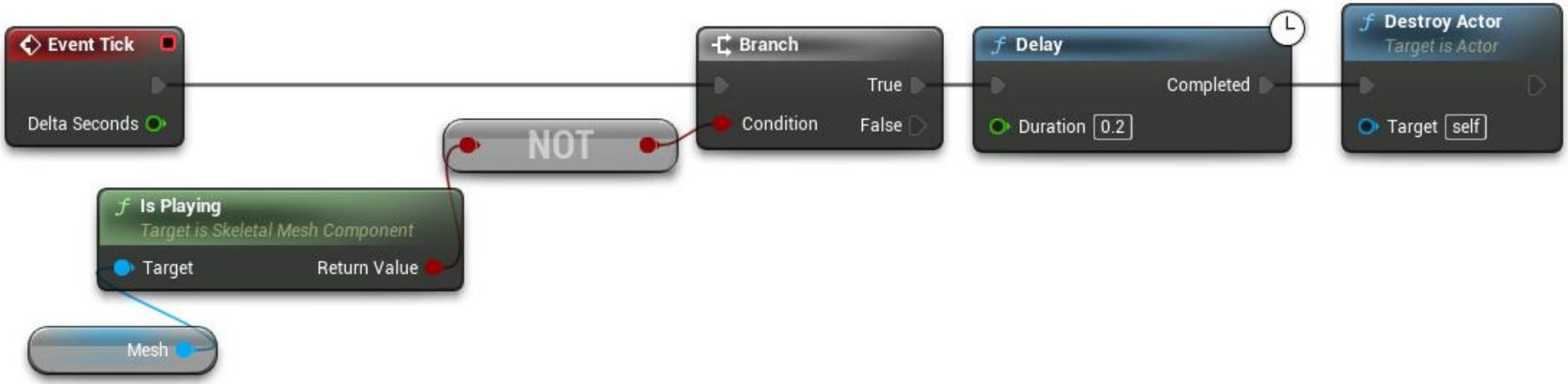
- 정육면체 모양의 액터가 공중에서 위아래로 움직이면서 떠 있게 하려면?
  - 액터의 운동 로직을 프로그램으로 구현하면 됨.
  - C++ 과 같은 프로그래밍 언어를 사용.




```
for (int t = 0; t < 200; t++)
{
    float z = 100.0 * sin(t/100*2*3.141592) + 400.0;
    cube->set_z(z);
}
```

블루프린트(Blueprint)

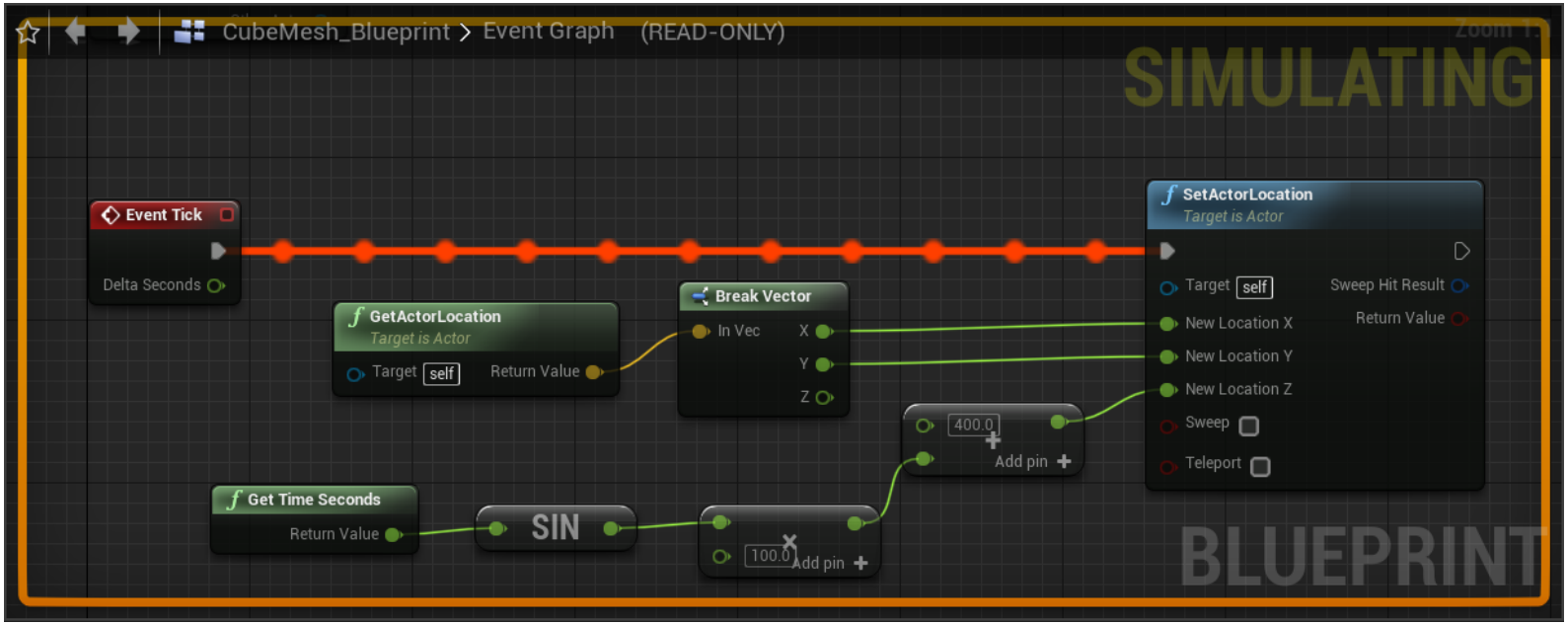
- 언리얼 엔진의 비주얼 스크립팅 언어
- 언리얼 엔진이 제공하는 가장 강력한 도구 - UE4 부터 본격적으로 사용됨.
  - 프로그래머뿐만 아니라 기획자와 디자이너도 쉽게 사용 가능.
  - C++과 연계한 개발 확장 가능.



“비주얼” 언어

시각적으로 보이는 언어

- 노드 기반 인터페이스
  - 노드와 와이어로 구성된 그래프를 이용하여 복잡한 로직을 “쉽게 " 프로그래밍할 수 있음.
  - 시각적으로 프로그램의 실행 흐름을 쉽게 파악할 수 있음.
  - 실행 흐름을 실시간으로 확인하면서 디버깅이 가능함.



“스크립팅” 언어

- 인터프리터의 장점을 지님.
  - 컴파일 시간이 필요하지 않음.
  - 바로 실행되므로, 디버깅이 용이함.

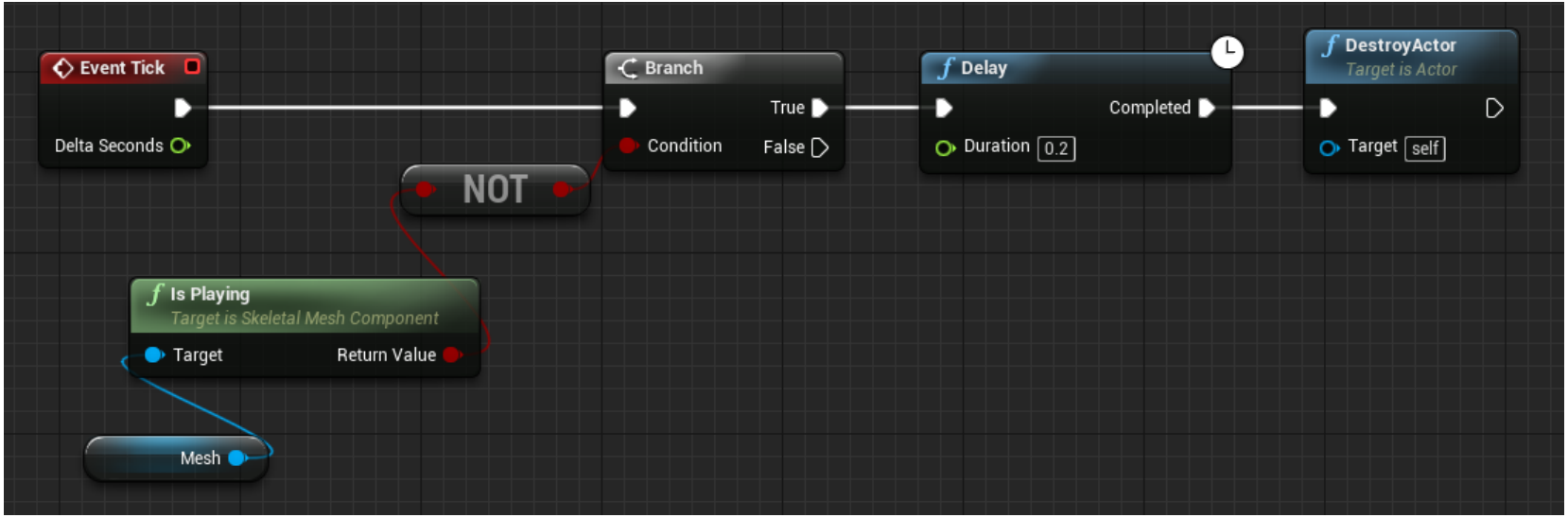
- 단점?

열배에서 스무배 정도까지 성능이 차이 난다.

  - VM 기반으로 실행되기 때문에, C++ 결과물에 비해 성능이 현저하게 떨어짐.
  - 언리얼 엔진의 모든 기능을 다 블루프린트만으로 활용하기는 어려움.

블루프린트 그래프의 기본 구조

- 기본 구성 요소
  - 노드 – 작업
  - 와이어 – 작업 순서 흐름, 또는 데이터 입출력

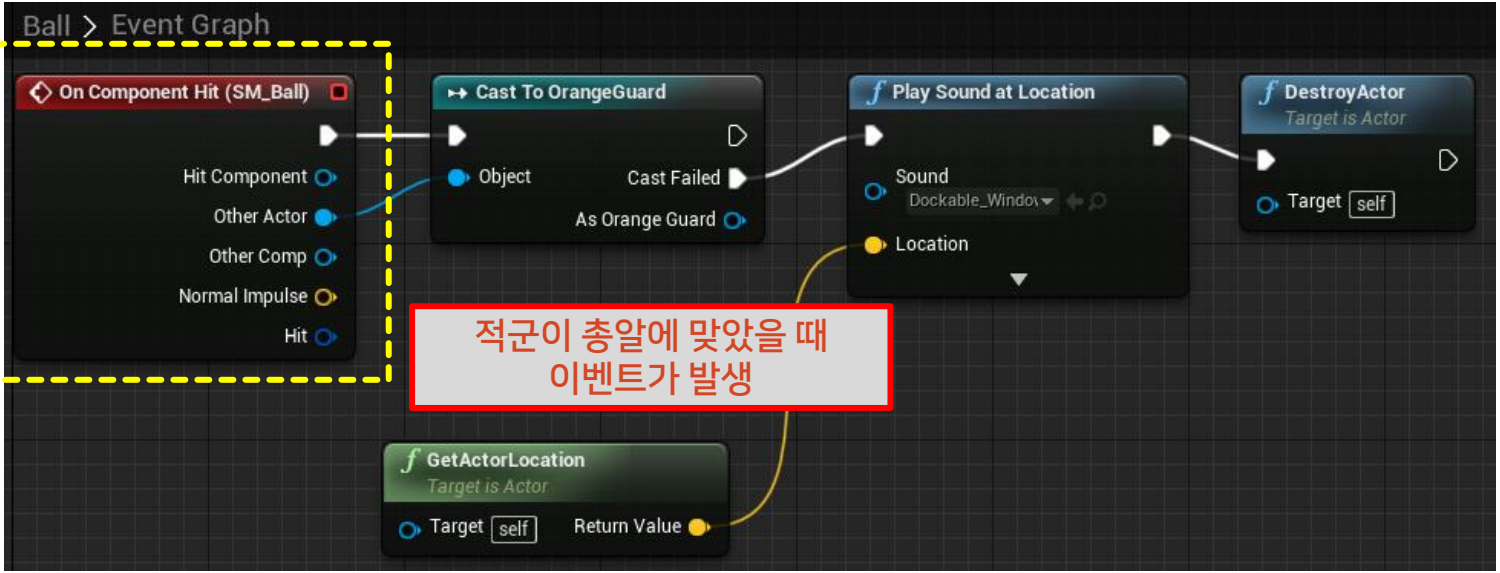


노드(Node)

- 블루프린트의 기본 단위 – 둥근모서리 사각형 모양
- 이벤트, 함수호출, 흐름제어동작, 변수 등을 나타냄.

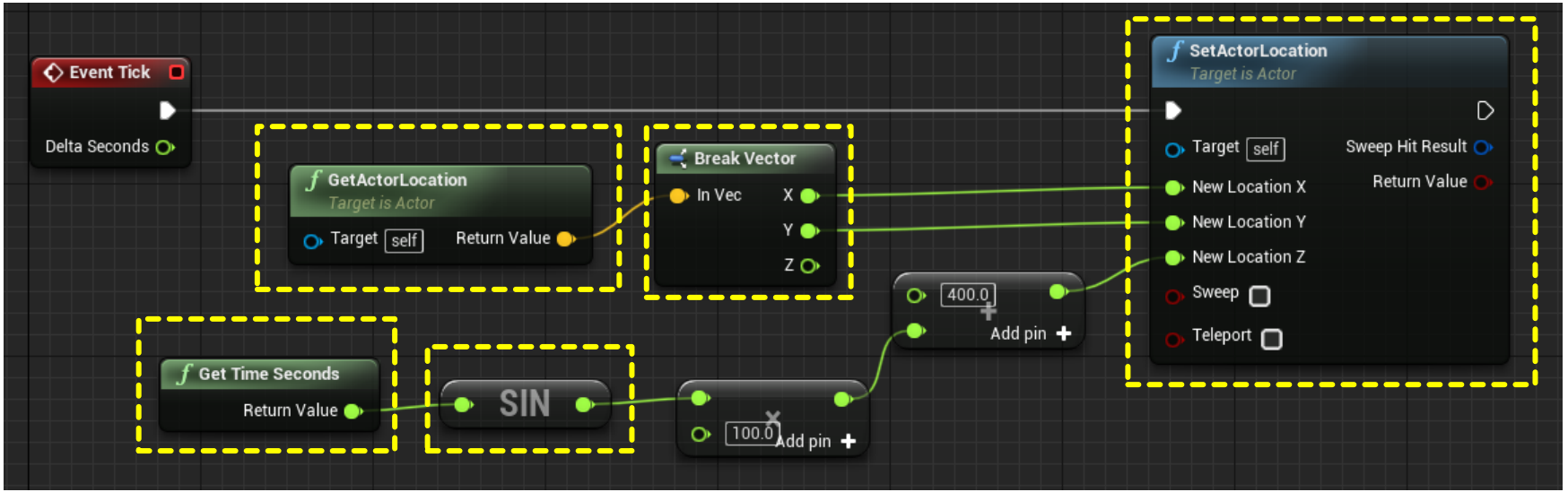
이벤트 노드

- 어떤 사건이 일어났음을 알림 – 총 피격, 제한시간 만료 등.
- 사건에 일어남에 따라 필요한 일들을 처리.



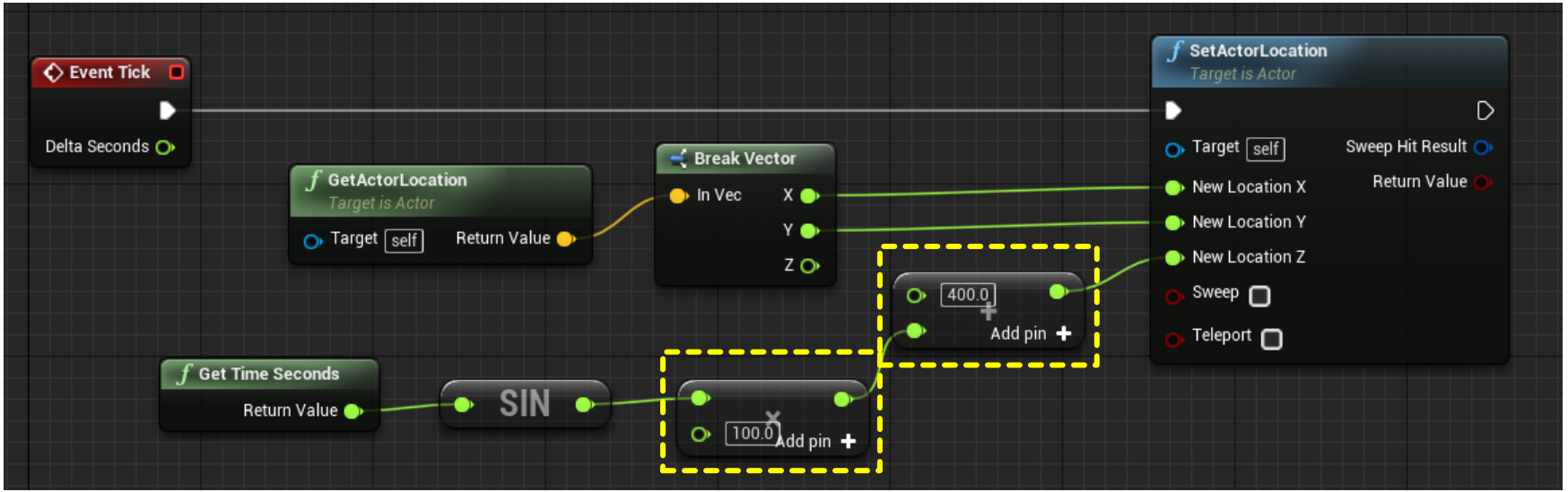
함수 노드

- 입력된 데이터를 처리하여, 그 결과를 반환함.



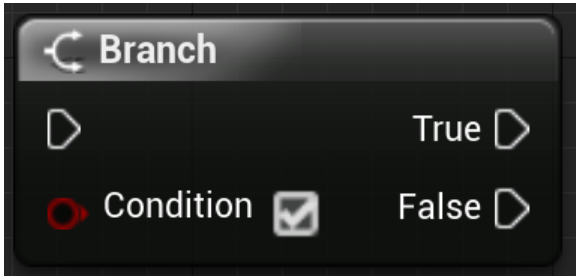
연산 노드

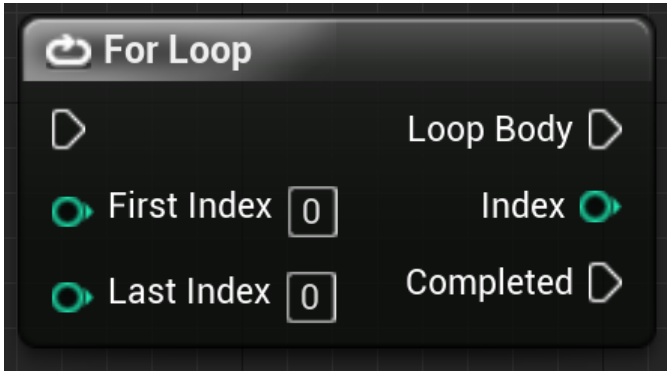
- 사칙 연산과 대소 비교 계산을 처리.



제어 노드

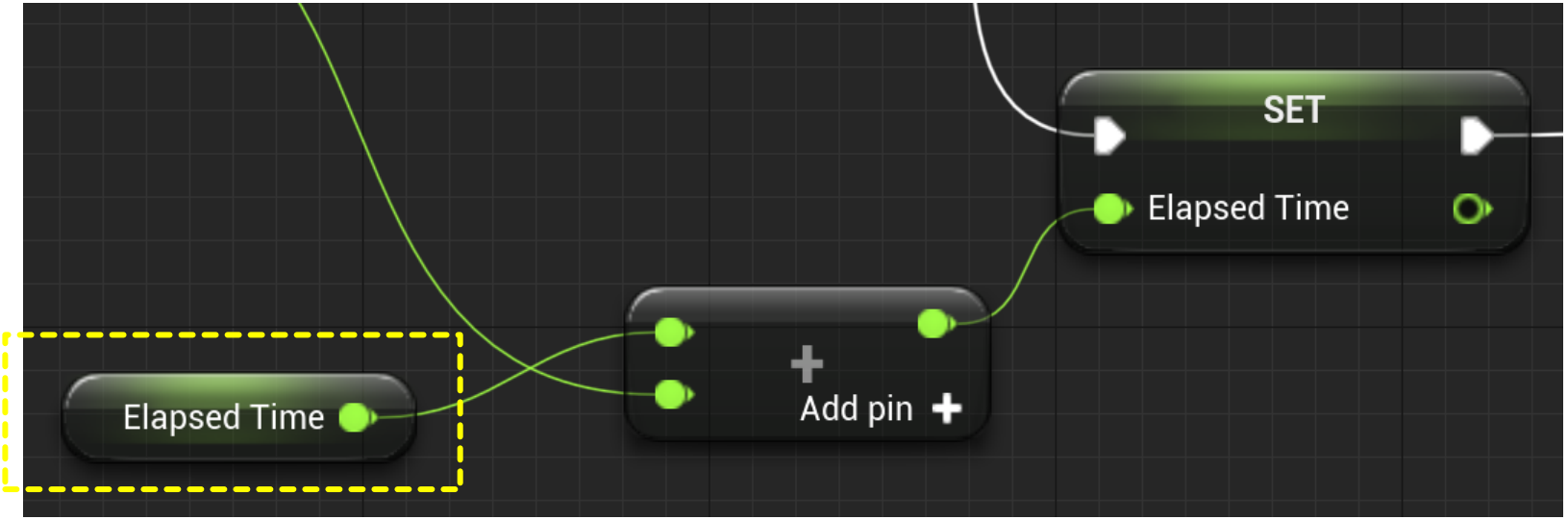
- 조건에 따라서 프로그램의 실행 흐름을 조절하는 노드.





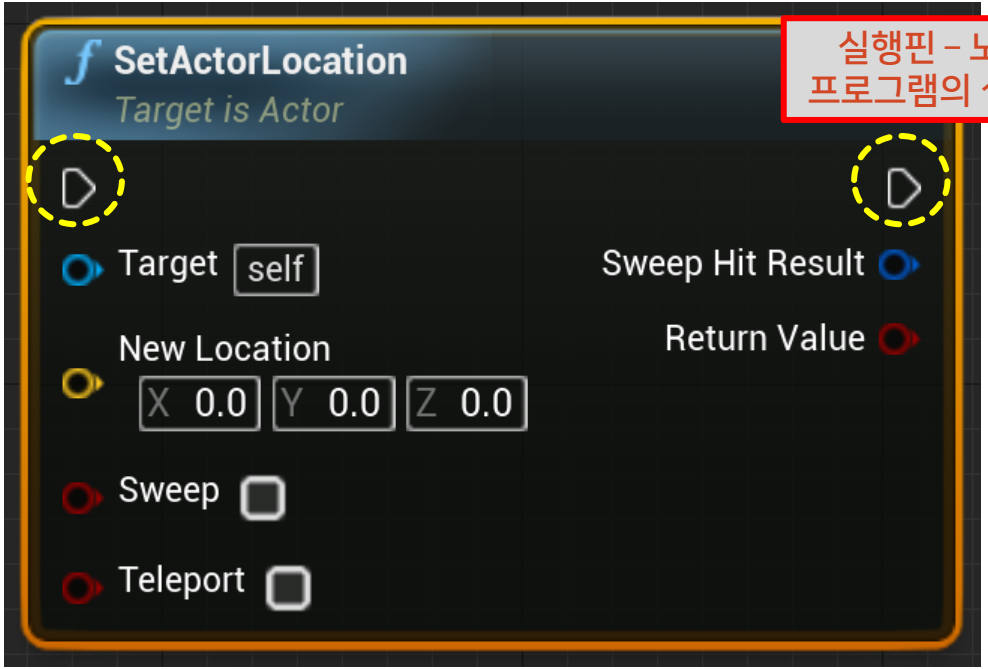
변수 노드

- 변수 – 어떤 값을 저장하는 공간으로써, 이름으로 구별할 수 있음.
- 변수값을 가져오는 노드



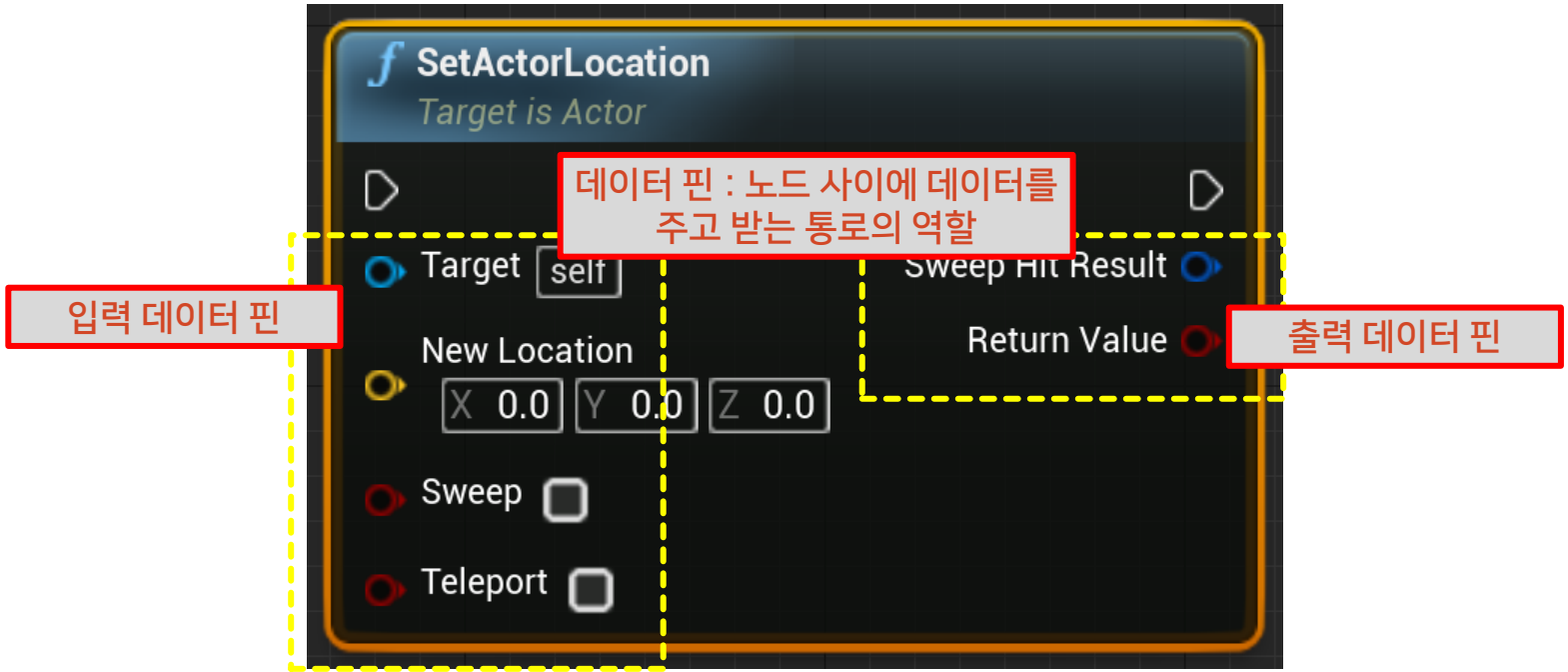
핀(Pin)

- 노드의 양쪽에 위치하는 연결점



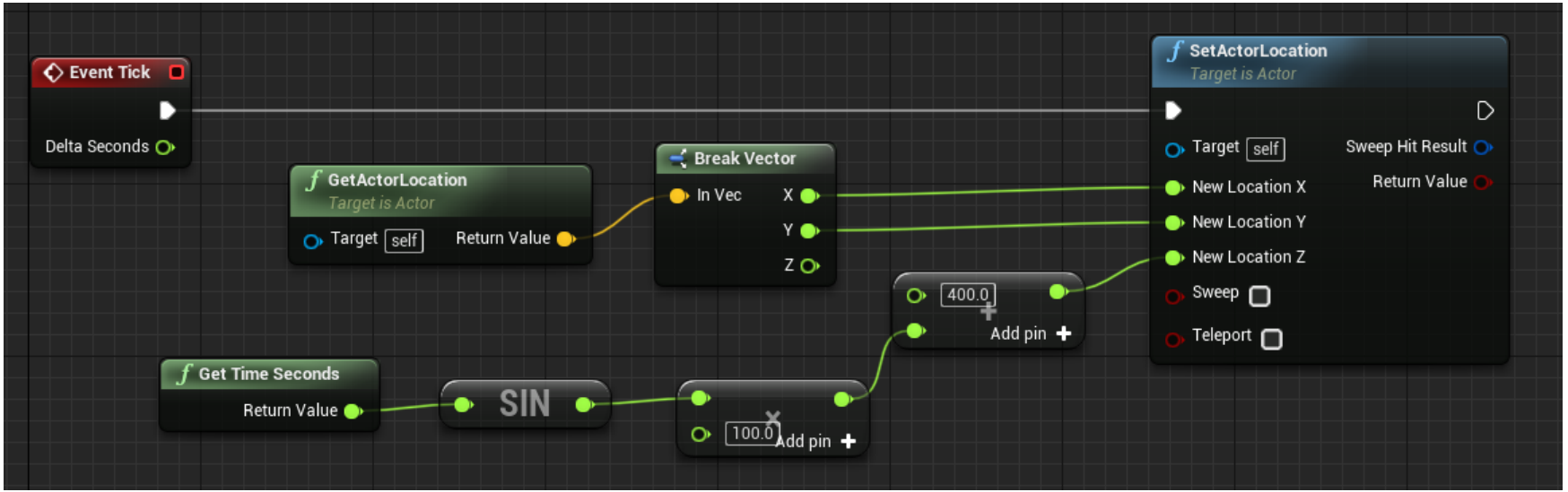
핀(Pin)

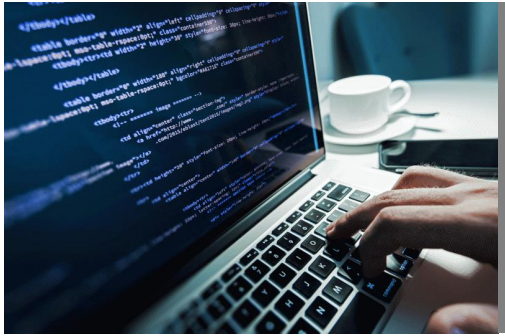
- 노드의 양쪽에 위치하는 연결점.



와이어(Wire)

- 핀 사이를 연결하는 선.
- 실행 흐름 또는 데이터의 흐름를 나타냄.
- 실행 흐름 – 흰색
- 데이터 흐름 – 데이터형에 의해 정의된 색



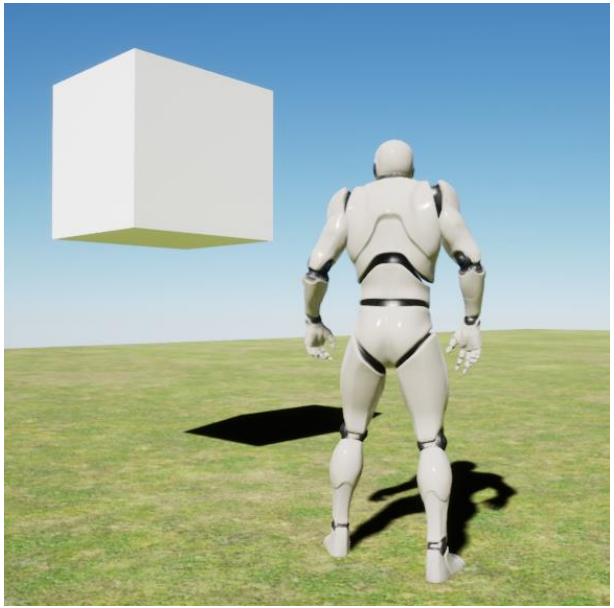


실습

액터 상하 이동과 회전

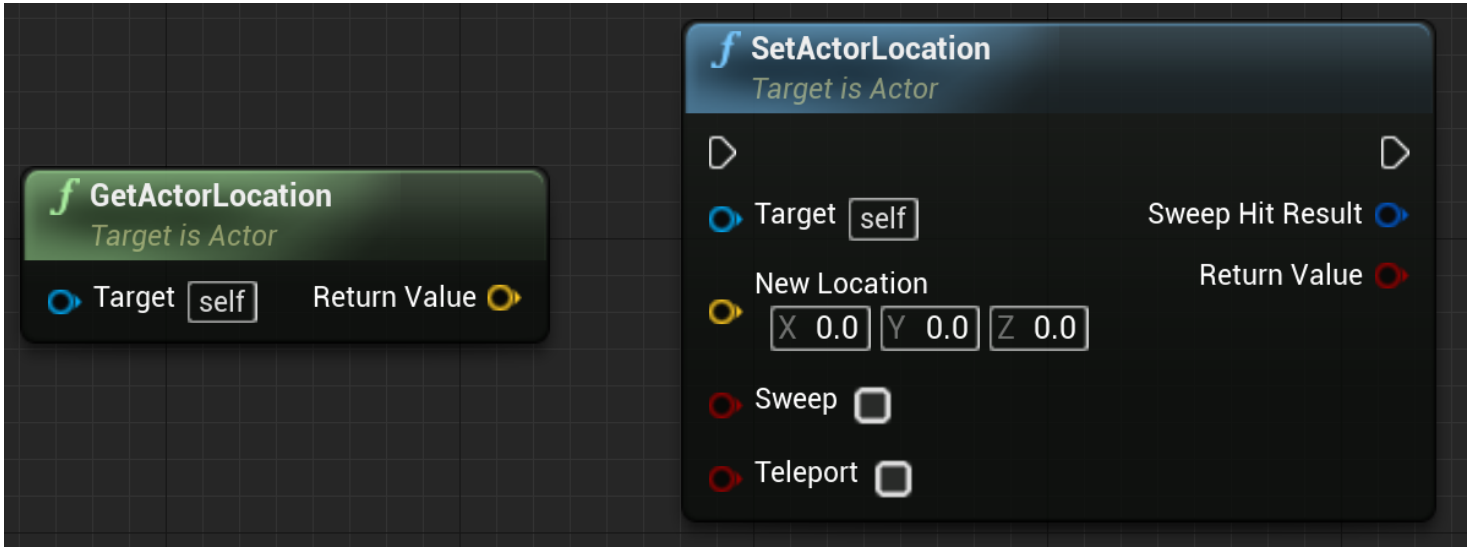
실습 목표

- 상하 운동하는 큐브 액터의 블루프린트 작성



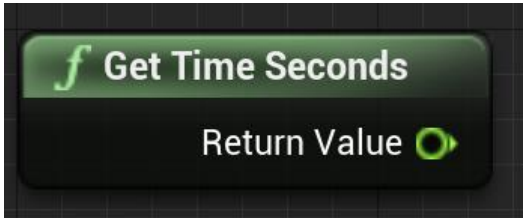
액터의 위치 관련 노드 함수

- GetActorLocation – 현재 액터의 3차원 좌표값을 획득.
- SetActorLocation – 액터의 3차원 좌표값을 설정.



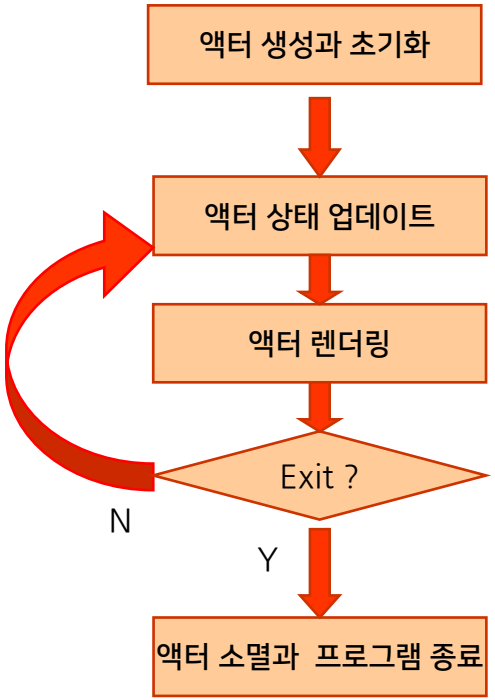
시간 획득 함수

- Get Time Seconds – 프로그램 실행 시작 후, 현시점까지의 경과 시간을 측정.



게임 루프

- 게임과 같이 가상 세계를 구현하는 인터랙티브 콘텐츠가 공통적으로 갖고 있는 프로그램 실행 구조

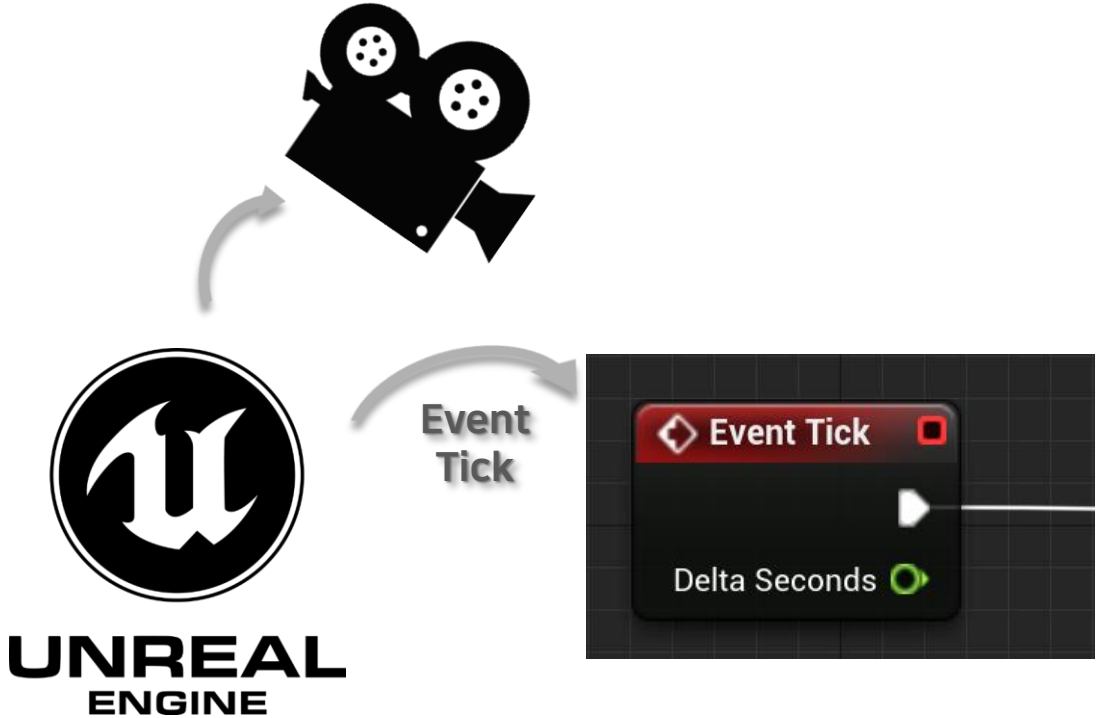


Stop Motion Animation

By Matthew850 at English Wikipedia - Transferred from en.wikipedia to Commons., Public Domain, https://commons.wikimedia.org/w/index.php?curid=2052925

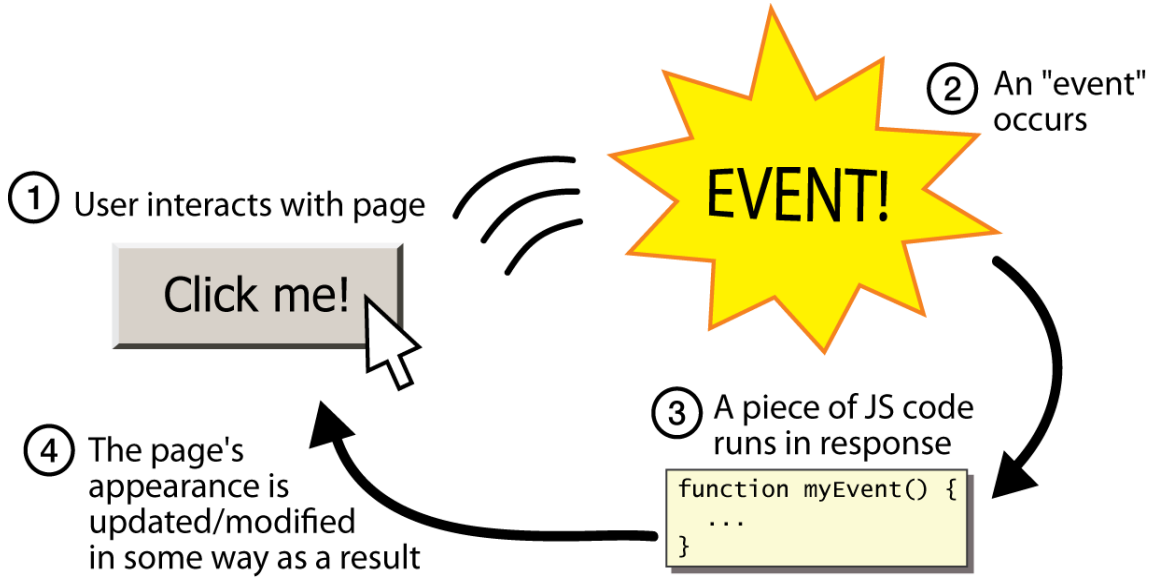
작업 절차

By Estarcido - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=56831605


UNREAL
ENGINE
Event
Tick
Event Tick
Delta Seconds

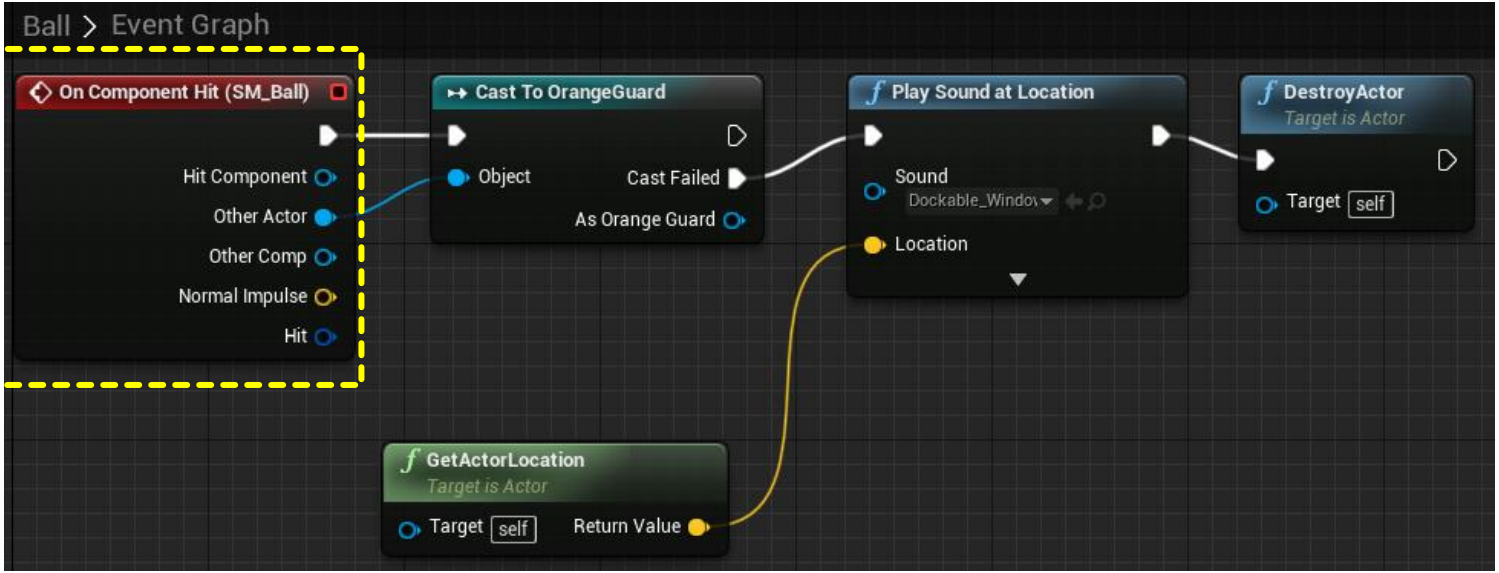
Event Driven Programming

- 이벤트(Event) - 상태의 변화
- 이벤트가 발생했을 때, 그에 따른 액션을 수행.

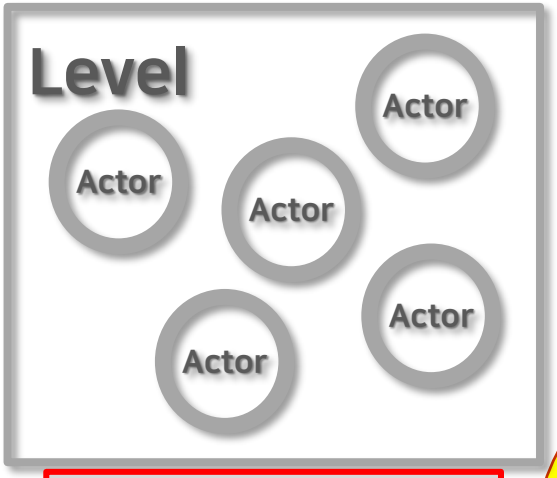


이벤트 기반 프로그래밍 (Event Driven Programming)

- 이벤트 – 어떤 상태의 변화가 일어났음을 알림. Ex. 총 피격, 제한시간 만료 등.
- 이벤트에 따른 필요한 일들을 처리.



언리얼 엔진 게임 루프



레벨 상에 배치된 액터들은
이미 생성된 상태임.

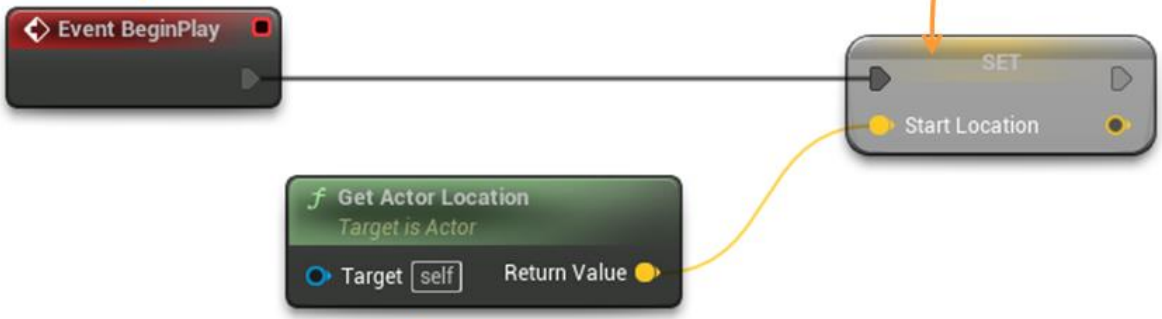
```
for each actor:
   actor->BeginPlay()

loop
{
   get inputs
   for each actor:
      actor->Tick()
   render()
}
```

각각의 액터들에 대해서, 액터의
초기화를 하기 위해 BeginPlay
Event가 호출됨.

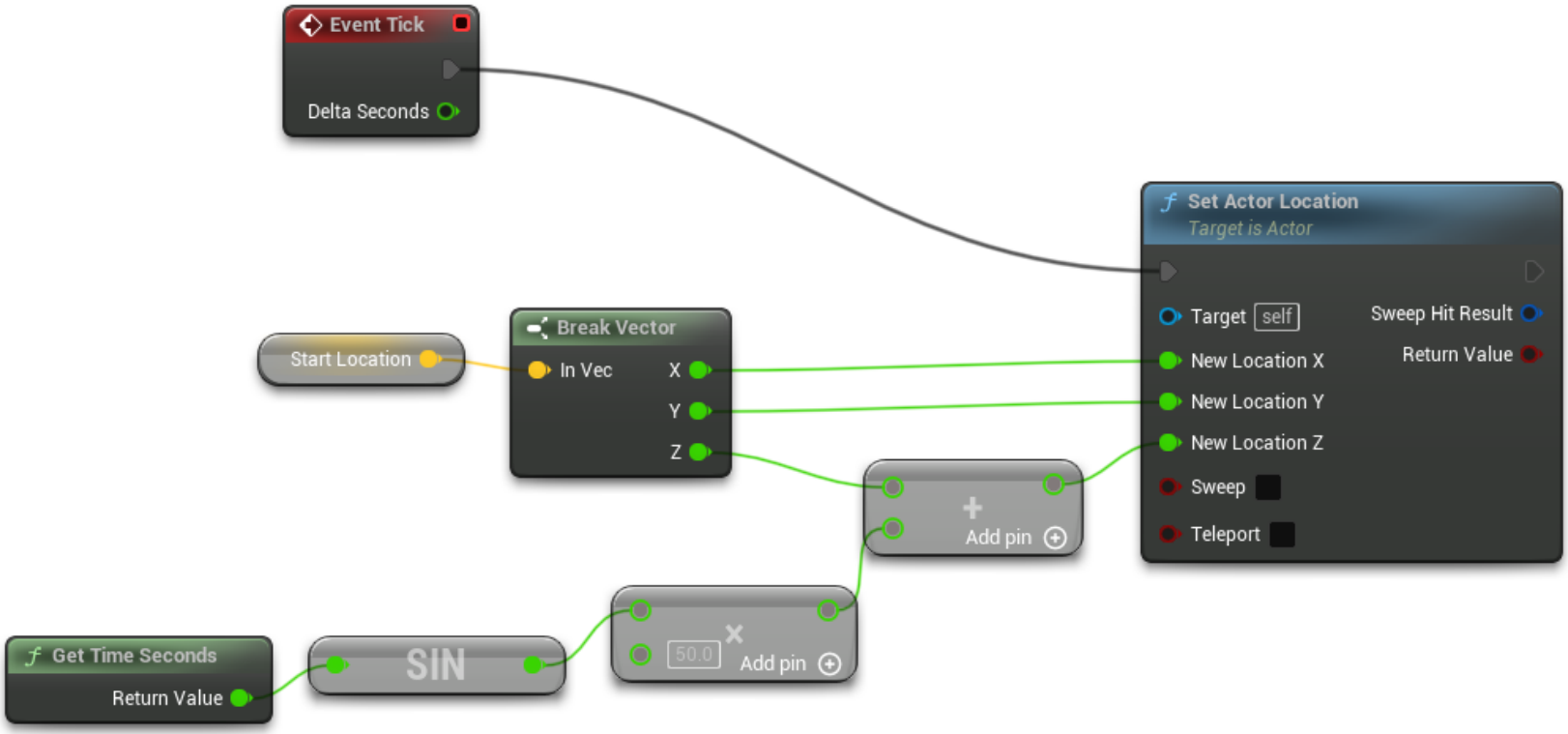
메인 게임 루프가 반복되면서, 각각의
액터들에 대해서 Tick Event를
반복적으로 호출함.

게임 플레이가 시작되면, 언리얼 엔진은 BeginPlay 이벤트를 보냅니다.



Start Location 이라는 이름의 변수에 액터의 초기 위치를 저장합니다.

액터의 현재 위치를 가져옵니다. 레벨에서 액터가 놓인 위치가 다 다르기 때문에, 이 위치를 어딘가 저장해놓아야 합니다.


Event Tick
Delta Seconds
Set Actor Location
Target is Actor
Target
self
New Location X
New Location Y
New Location Z
Sweep
Teleport
Sweep Hit Result
Return Value
Start Location
Break Vector
In Vec
X
Y
Z
Add pin
Get Time Seconds
Return Value
SIN
50.0
Add pin

블루프린트 클래스(Blueprint Class)

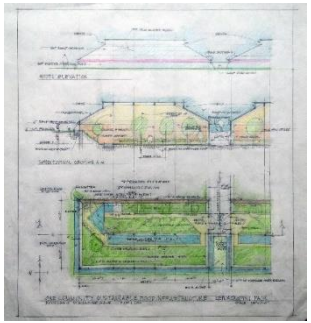
- 가장 핵심적인 블루프린트 유형. 그냥 블루프린트 라고도 부름.
- 새로운 종류의 액터를 설계하고 생성하는데 사용됨.
- 객체지향언어의 “클래스”와 동일한 개념.

붕어 Blueprint Class

붕어 Actor 들

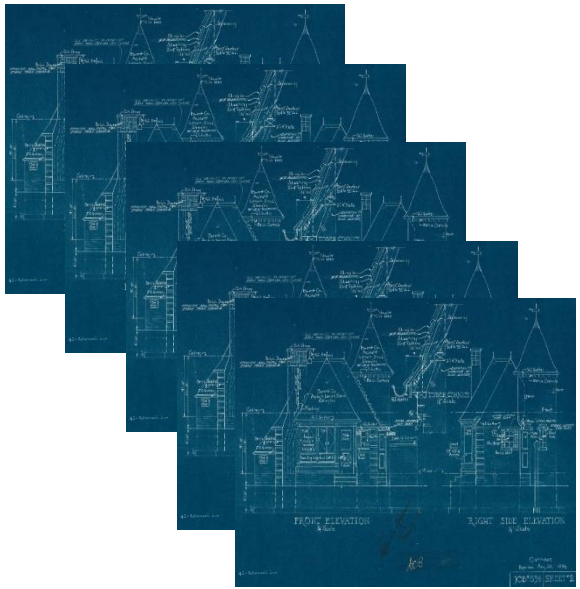
블루프린트(Blueprint) - 청사진

- 설계도를 복사한 사진
- 원본 설계도를 쉽게 복사해서 배포할 수 있음.



원본 설계도

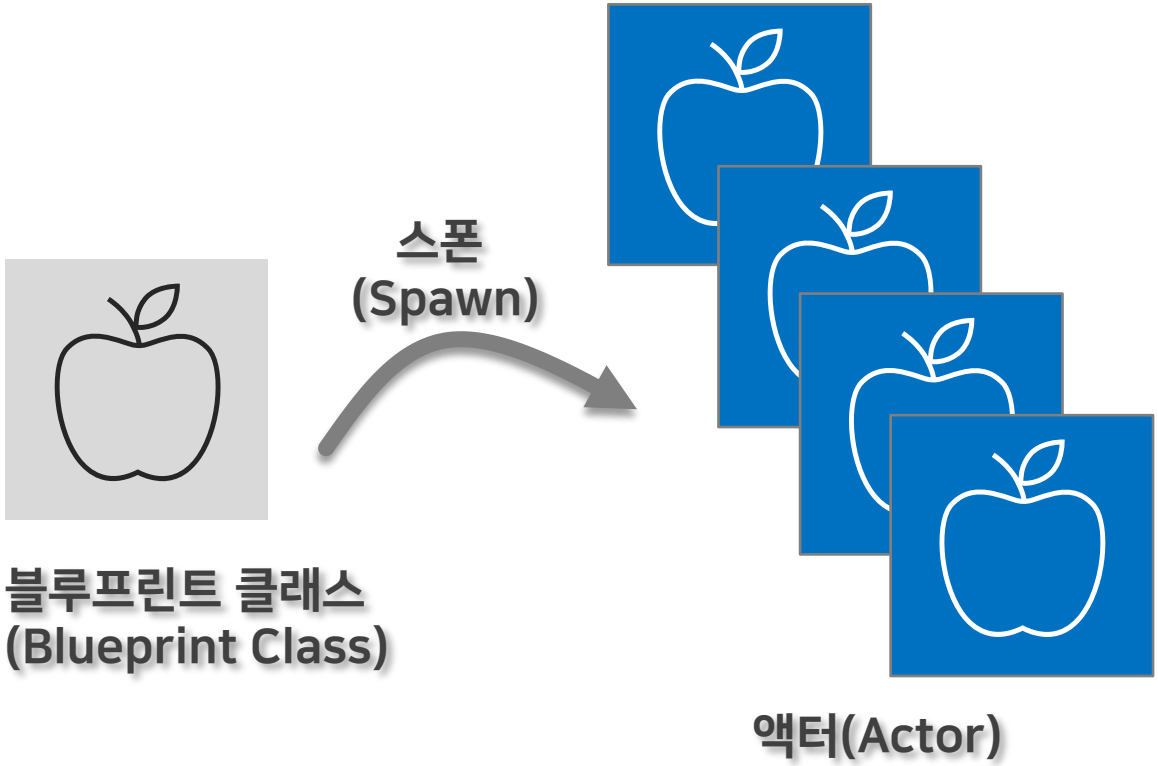
현상



복사된 설계도 청사진들

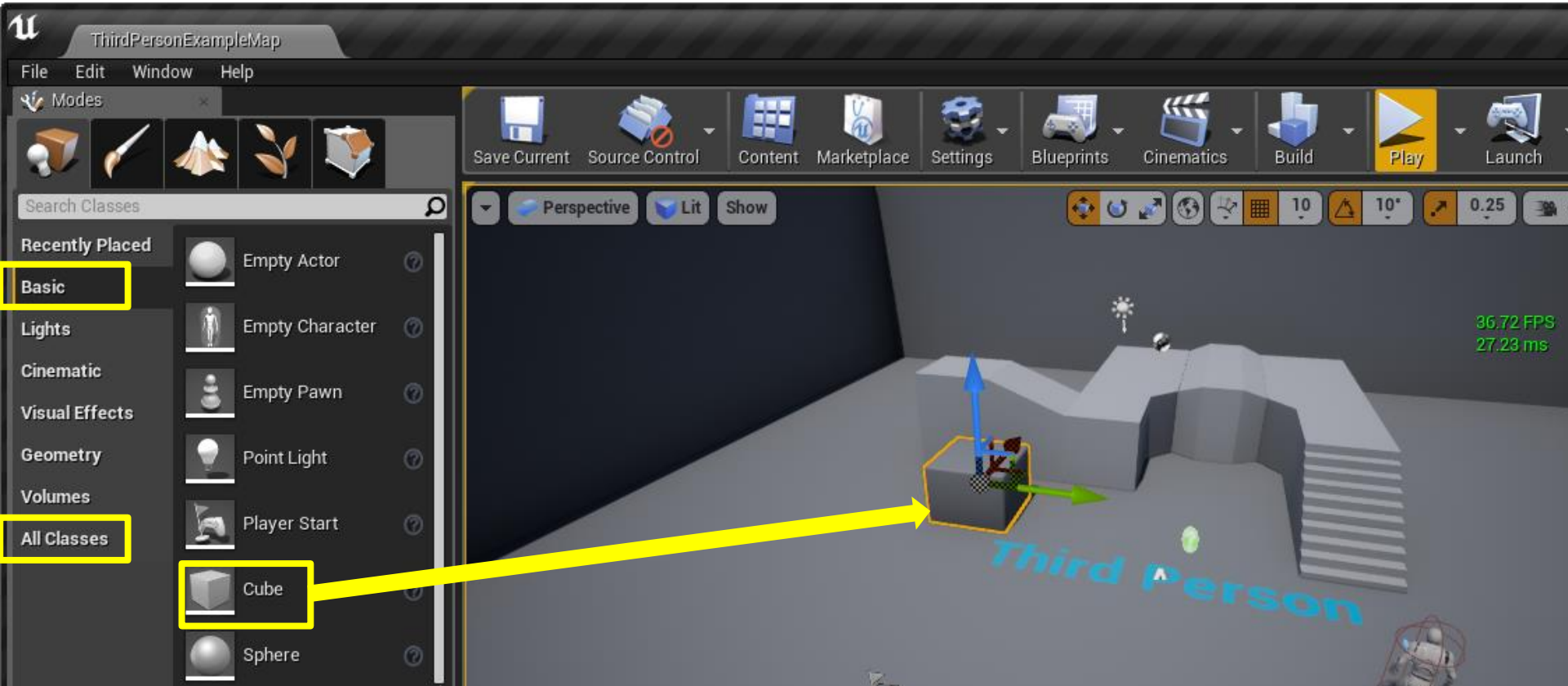
https://en.wikipedia.org/wiki/Blueprint#/media/File:Joy_Oil_gas_station_blueprints.jpg

블루프린트 클래스와 액터



블루프린트 클래스의 활용

- 블루프린트 클래스를 Level에 배치하면, 액터(인스턴스)가 생성됨.



모양이 조금 다른 붕어빵을 만드려면?

기존 붕어빵 틀을 수정
보완하여 사용함.

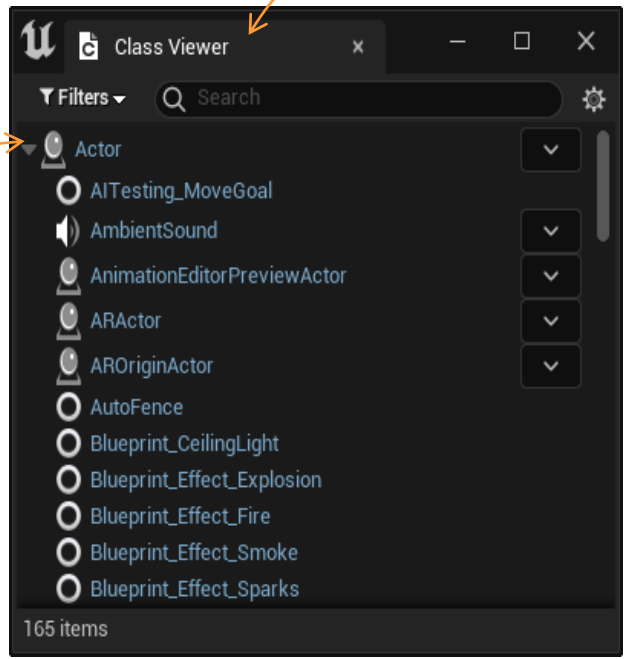
부모

상속

자식

언리얼 엔진은 다양한 틀을 제공합니다.

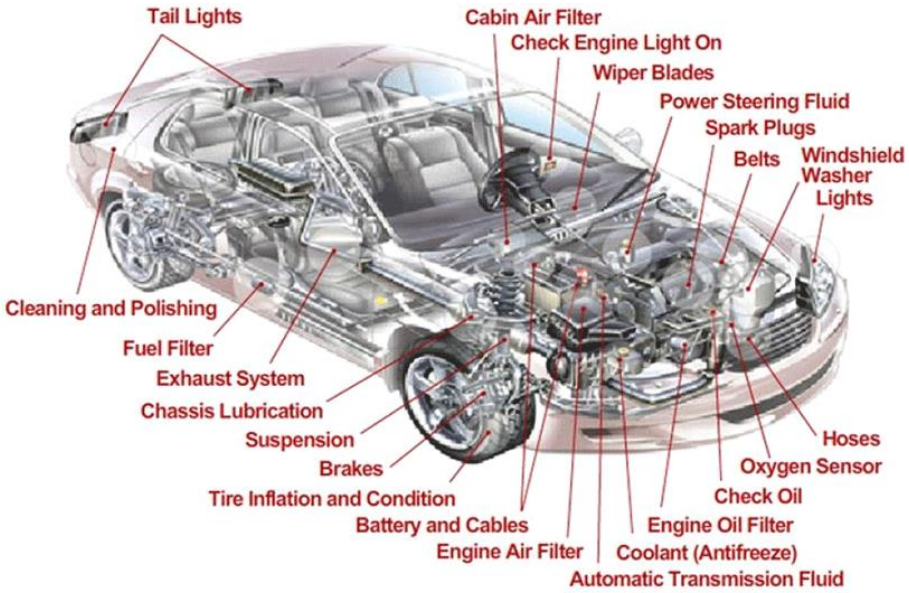
클래스 뷰어를 통해 클래스들의 상속 관계, 계층 구조 등을 한눈에 볼 수 있습니다.



Actor 클래스는 상속 계층 구조에서 가장 맨 위에 있는 최상위 클래스입니다.

언리얼 엔진은 틀 안에 담을 수 있는 다양한 부품도 제공합니다.

- 다수의 부품들을 조립함.



컴포넌트(Component)

- "A Piece of Functionality" : 기능(외형, 로직, 물리)을 구현하는 부품.
- 블루프린트 안에 추가시키면 기능 구현이 됨.
- 직접 기능 구현을 하기 전에, 기존 부품이 있는 지 확인 필요.
- 언리얼 엔진은 다양한 컴포넌트들을 제공하고 있음.

실제로는 상속받는 방법을 잘 안 쓴다.
자식에서 부모의 기능을 필요로 안 하는 부분이 많을 수 있고,
거기서 문제가 발생할 수 있기 때문에.
그래서 부품(컴포넌트)를 조합하는 방식으로 사용한다.

AI 컴포넌트 ›

AI 인지에 사용되는 AI 관련 컴포넌트 및 폰 감각에 대한 설명입니다.

오디오 컴포넌트 ›

AudioComponent 는 사운드 인스턴스의 생성 및 제어에 사용됩니다.

카메라 컴포넌트 ›

카메라 컴포넌트와 스프링 암 컴포넌트에 대한 설명입니다.

라이트 컴포넌트 ›

언리얼 엔진 4 에서 사용할 수 있는 여러가지 Light 컴포넌트에 대한 설명입니다.

무브먼트 컴포넌트 ›

캐릭터든 프로젝타일이든, 이동에 관련된 모든 것은 무브먼트 컴포넌트를 사용합니다.

내비게이션 컴포넌트 ›

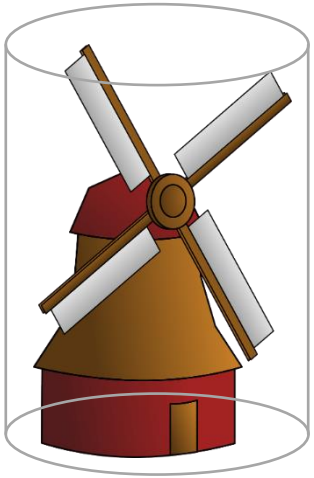
볼륨의 모양을 사용하여 선택된 AreaClass 를 내비메시에 적용할 수 있습니다.

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

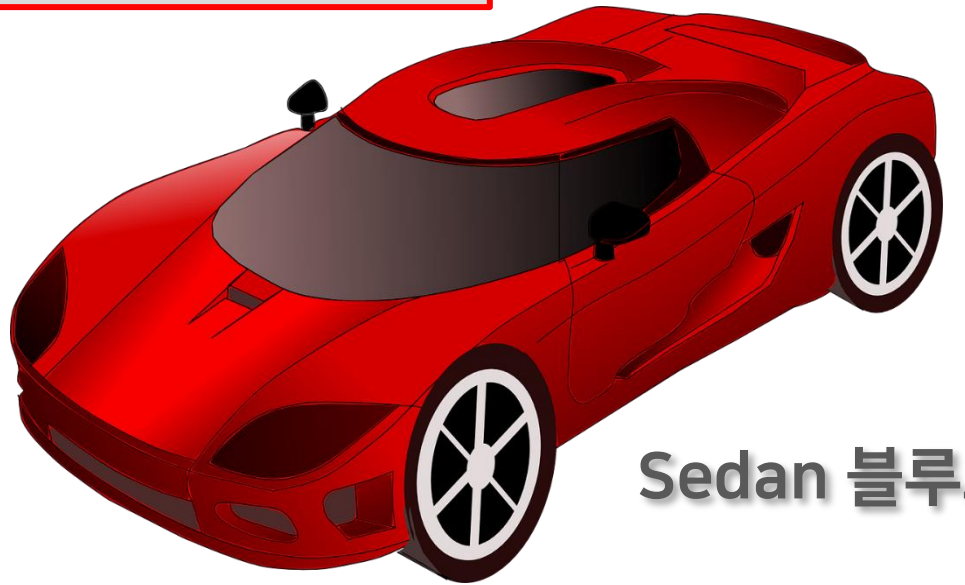
블루프린트 설계 요령

- 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.

"Sedan" 이라는 블루 프린트 클래스가 이미 있다고 가정.

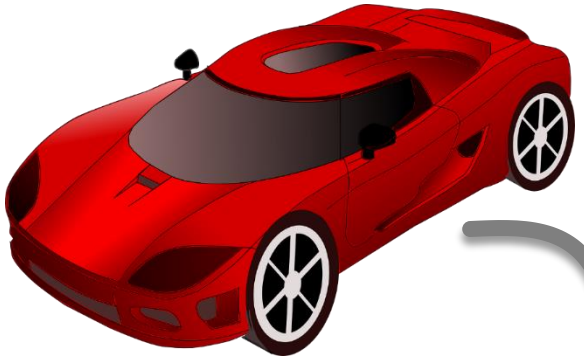


Sedan 블루프린트

"SportsCar" 블루 프린트
클래스를 만들고 싶으면?

SportsCar 블루프린트

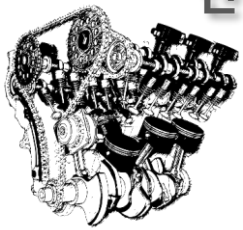
공통점: 자동차, 색상
차이점: 지붕

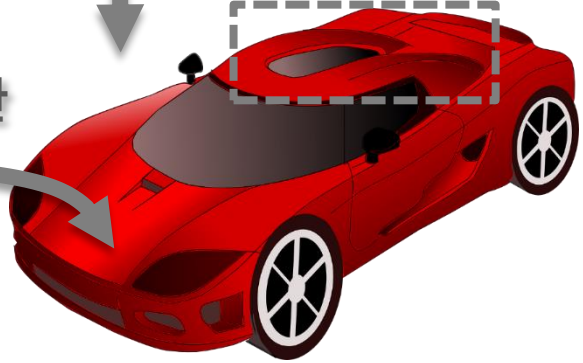


복제

지붕 절개

엔진 교환

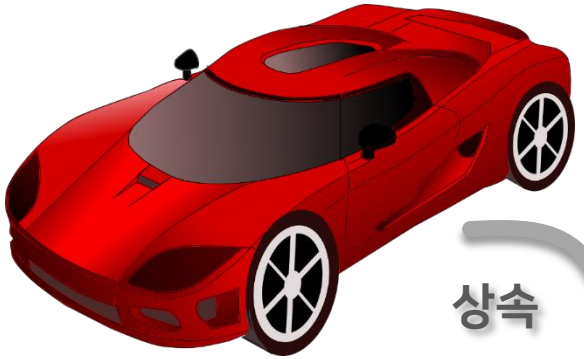




SportsCar 블루프린트

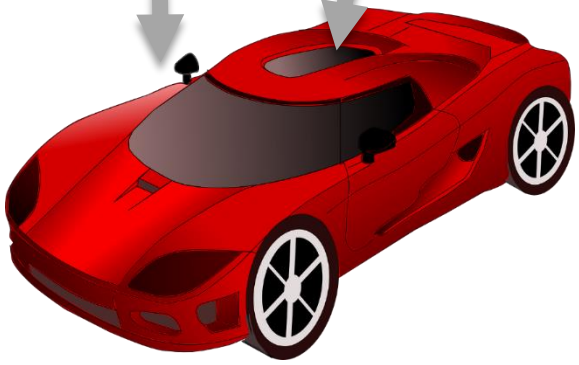
"FlyingCar" 블루 프린트
클래스를 만들고 싶으면?

FlyingCar 블루프린트

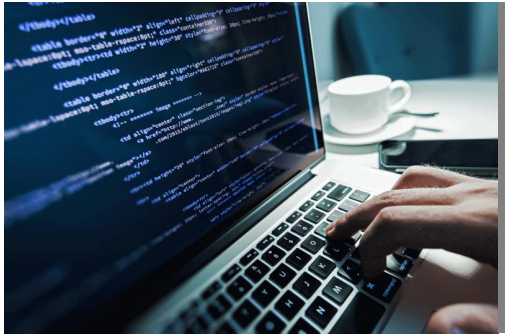


상속

프로펠러 추가



FlyingCar 블루프린트



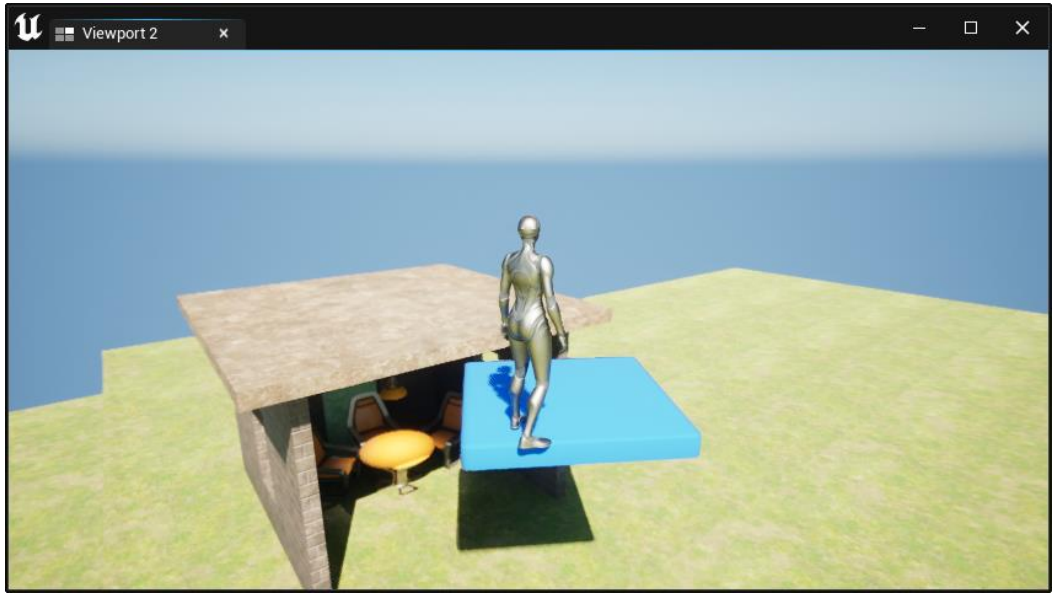
실습

이동 발판 블루 프린트 제작

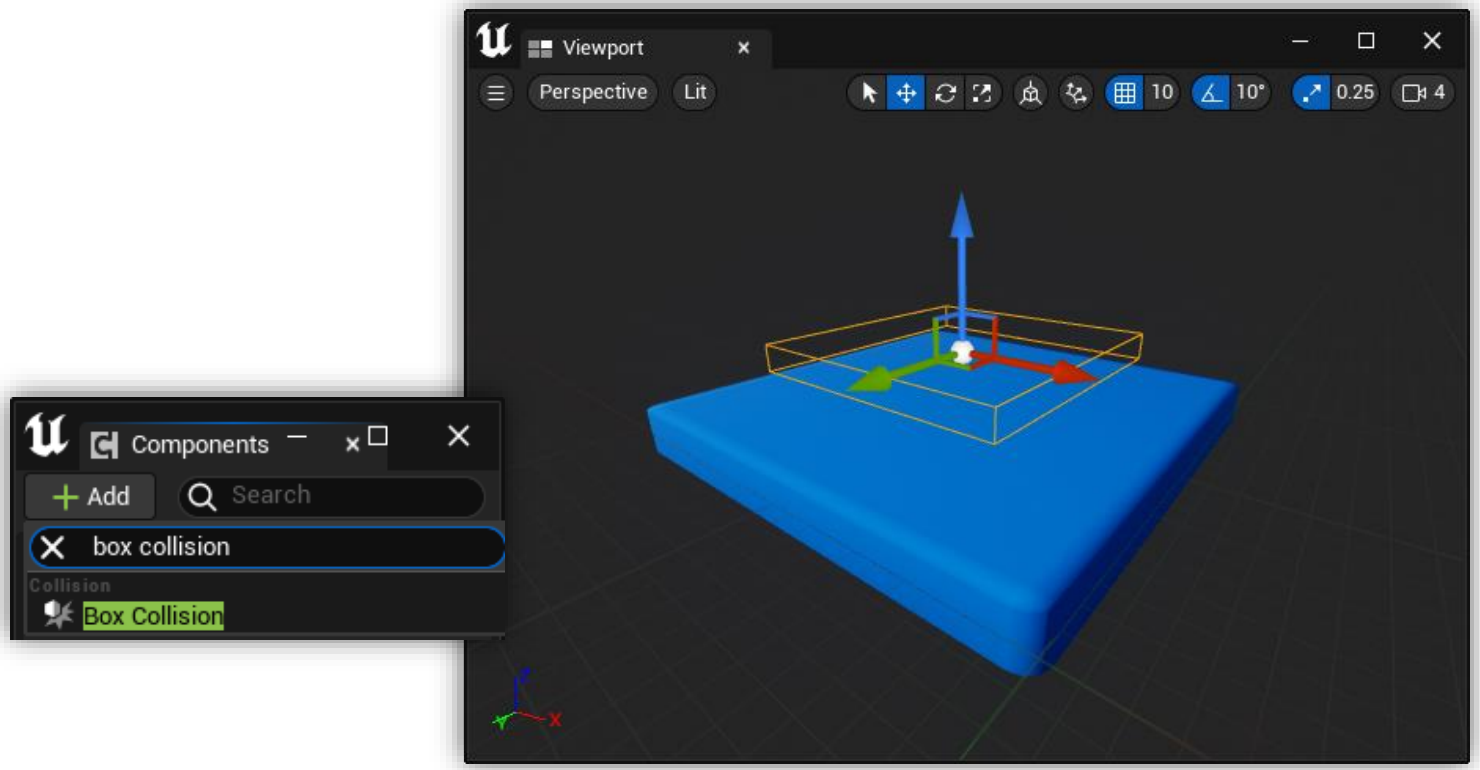
실습 목표

- 기존 블루프린트의 복제를 통한 블루프린트 과정 이해.

- 이동 발판의 기능
  - 둥둥 떠 있음.
  - 캐릭터가 발판 위에 오르면, 위로 이동.
  - 캐릭터가 나가면 이동을 멈춤.

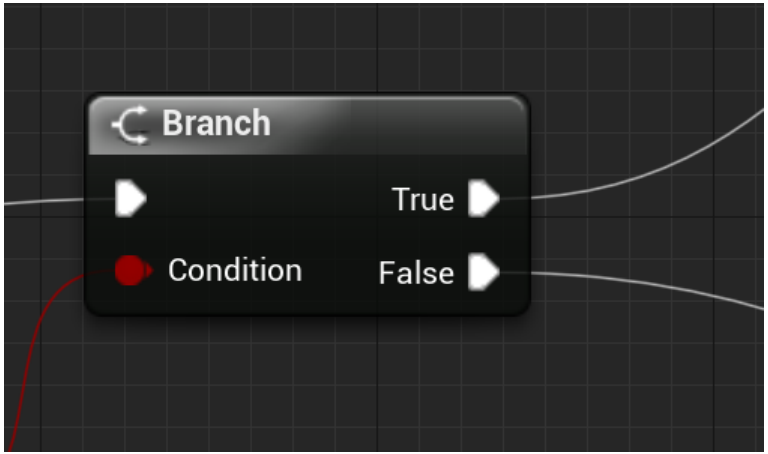


충돌 체크 컴포넌트


Viewport
Perspective
Lit
10
10°
0.25
4
Components
Add
Search
box collision
Collision
Box Collision

분기(Branch) 노드

- 입력 조건에 따라 실행 흐름을 선택

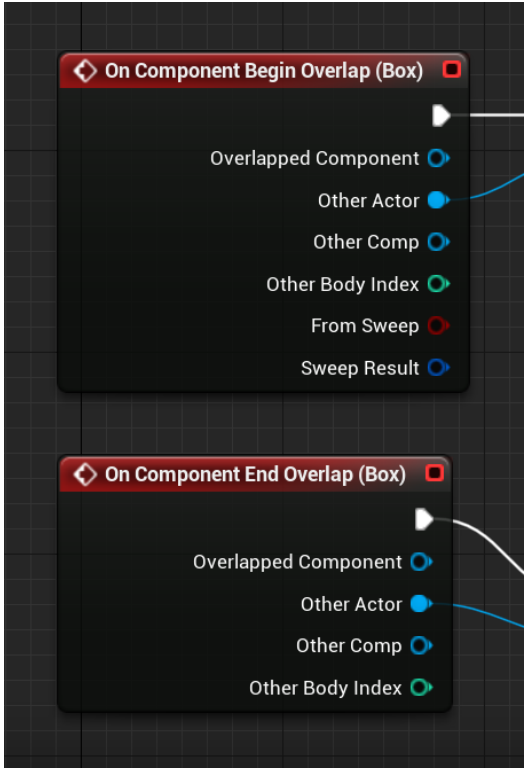


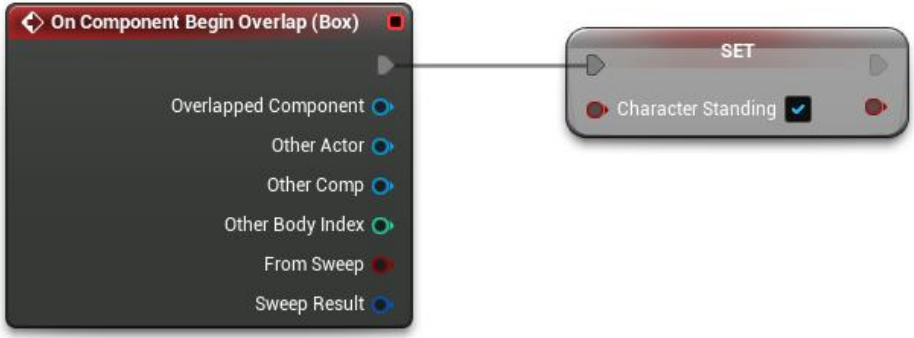
노드함수 AddActorWorldOffset

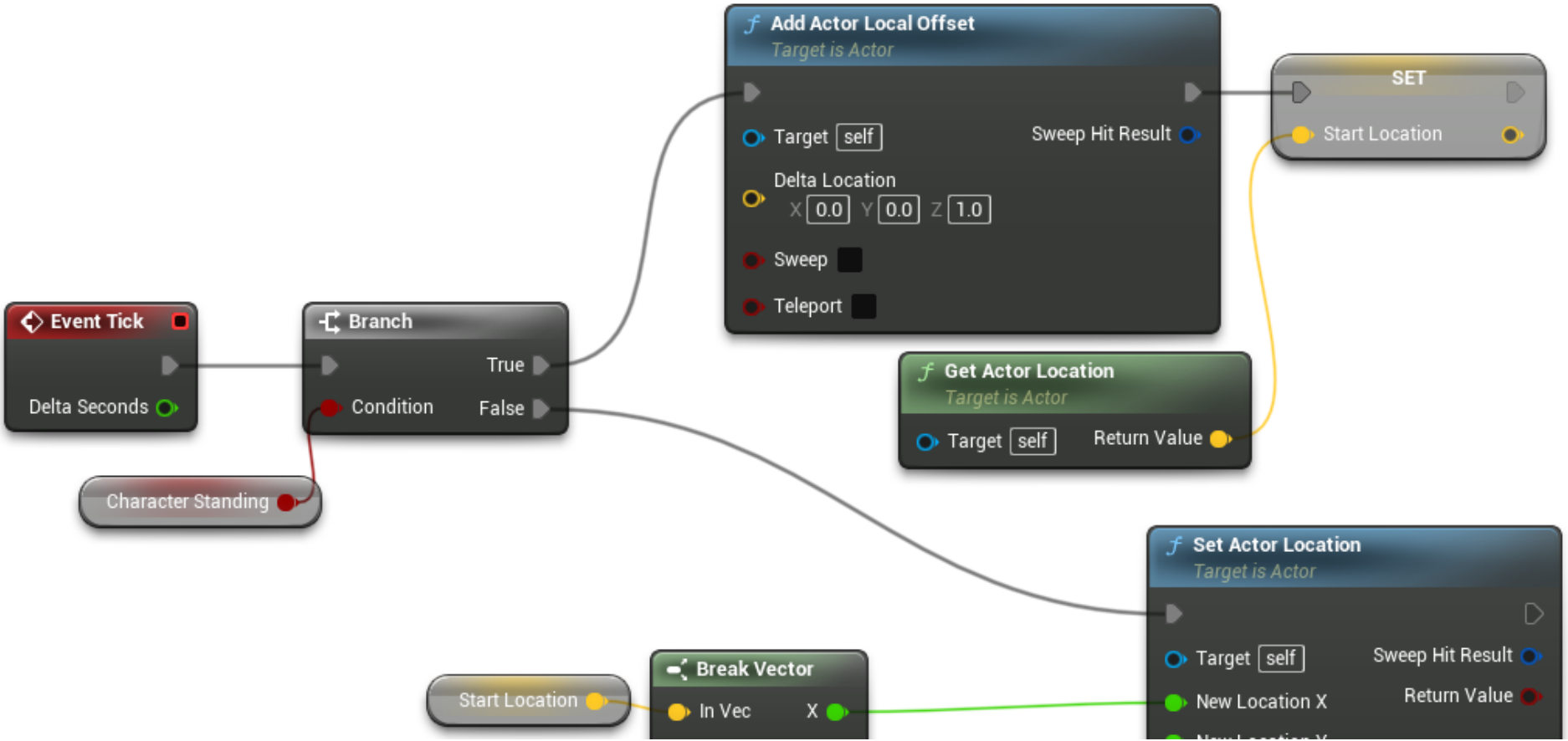
- 액터의 현재 위치를 기준으로, 상대값만큼 이동시킴.

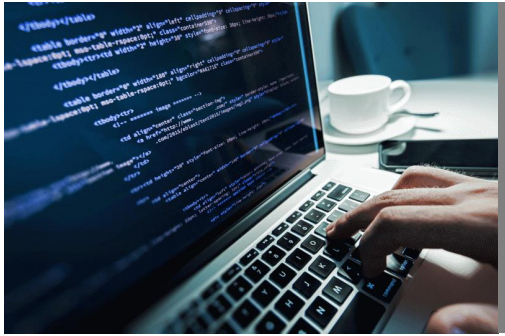
컴포넌트들이 서로 겹치는 상황 판단 방법

- BeginOverlap, EndOverlap 이벤트 활용






Add Actor Local Offset
Target is Actor
Target self
Sweep Hit Result
Delta Location
X 0.0 Y 0.0 Z 1.0
Sweep
Teleport
SET
Start Location
Event Tick
Delta Seconds
Branch
True
Condition
False
Character Standing
Get Actor Location
Target is Actor
Target self
Return Value
Set Actor Location
Target is Actor
Target self
Sweep Hit Result
New Location X
Return Value
Start Location
Break Vector
In Vec
X



실습

하우스 블루 프린트 제작

실습 목표

- 액터 병합을 통한 하우스 메쉬 생성.
- 액터들을 컴포넌트로 활용한 블루프린트 클래스 설계.

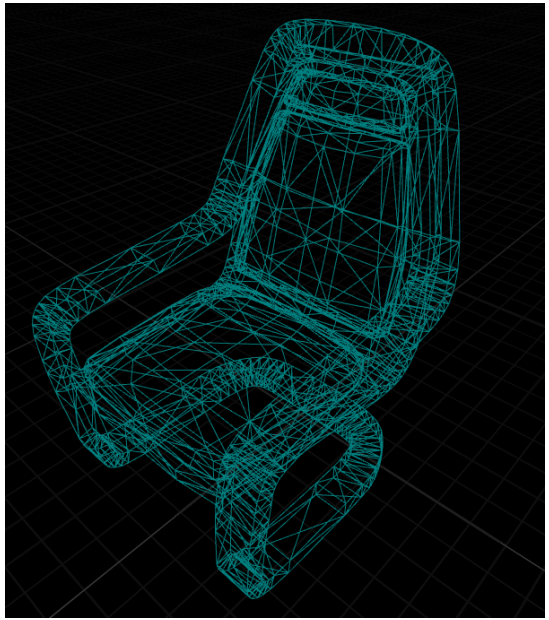
- 하우스의 기능
  - 전등이 설치된 집.
  - 캐릭터가 들어가면 전등 ON.
  - 캐릭터가 나오면 전등 OFF.

블루프린트 설계

- 그대로 사용할 수 있는 블루프린트 있는가? NO
- 비슷한 블루프린트가 있는가? NO
  -
- 새로운 기능의 블루프린트 !!
  - “Actor”를 Base Class 로 하는 블루 프린트 클래스 제작.

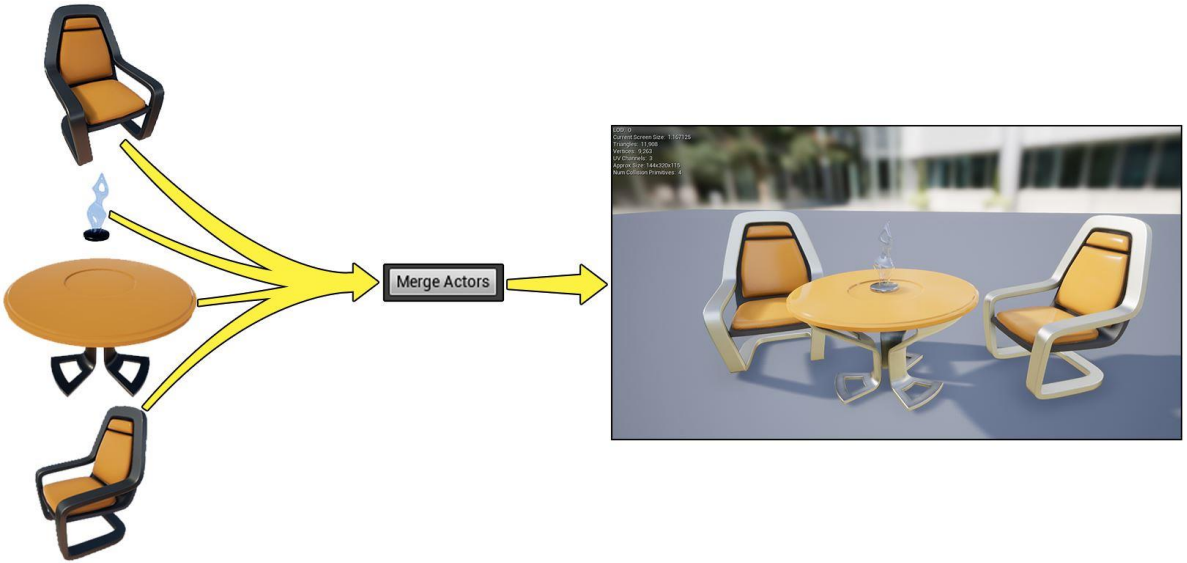
스태틱 메시(Static Mesh)

- 수많은 다각형(polygon)으로 구성된 3D 모델
- 3D 월드 렌더링의 기본 단위



액터 병합(Actor Merging)

- 여러 개의 스태틱 메시들을 하나의 액터로 병합.



노드함수 SetVisibility

- 가시성을 조정.