

# **언리얼 패스트 트랙** 워크샵 2

## **워크샵 2:** 블루프린트, 파워업, 그리고 장애물 살펴보기

### 📕 학습 포인트

#### 테스트 드라이브(75분)

- 블루프린트 심화 학습
- 블루프린트 만들기
- 다양한 블루프린트 이해하기
- 블루프린트 형변환

#### 그랑프리(1시간)

- 파워업과 장애물 만들기
- 블루프린트에서 이벤트 디스패처 사용하기
- 다양한 변수 만들기

#### 오프로딩 및 토론(75분)

- 블루프린트 처음부터 만들기
- 기존 콘텐츠 수정하기

## 💮 테스트 드라이브

2주 차에서는 블루프린트와 작동 원리를 자세히 살펴봅니다. 이번 주 언리얼 온라인 학습 강좌 제목은 '블루프린트 핵심 개념'입니다. 강좌는 <u>https://www.unrealengine.com/en-US/onlinelearning-courses/blueprints---essential-concepts</u> 에서 확인할 수 있습니다. 블루프린트는 언리얼 엔진에서 쓰이는 시각적 C++ 스크립팅 시스템으로, 반드시 사용법을 알아야 할 중요한 툴입니다. 블루프린트는 코딩 경험이 없어도 배울 수 있습니다. 수강을 시작하면 강사와 함께 직접 강좌 내용을 따라 할 수 있도록 언리얼 엔진 프로젝트를 다운로드하라는 안내가 표시됩니다.

이 문서에는 강좌가 처음 나온 이후로 일부 영상에 새로 추가된 팁과 변경 내용이 포함되어 있습니다. 필요한 경우 영상마다 팁과 변경 사항이 타임 스탬프로 표시되어 있으므로 쉽게 확인할 수 있습니다. 타임스탬프 표시가 없다면 마지막까지 기다렸다가 읽으면 됩니다.





#### 3: 블루프린트란?

**2:57**: 여기서 수정 중인 레벨은 **Content** → **Fundamentals** → **Blueprints** → **EssentialConcepts** → **StartAssets** 폴더에서 확인할 수 있습니다. 레벨 이름은 '**EssentialConcepts\_Start**'입니다.

#### 13: 추가 참고 자료

퀴즈를 푼 다음 추가 참고 자료를 눌러 강좌를 완료하고 배지를 획득하세요!



이번 주에는 3인칭 플랫포머에 파워업과 불을 추가해 보겠습니다. 이 기능을 활용하면 점점 더 강해지는 플레이어에 맞게 레벨을 바꾸고 피해야 할 장애물을 늘릴 수 있습니다.

먼저 파워업을 추가합니다.

- I.
   본 강좌용으로 다운로드한 FundamentalsClasses 프로젝트에서 아래와 같은 블루프린트를 선택합니다. 블루프린트는

   Content → Fundamentals → Blueprints → EssentialConcepts → FinishedAssets → Blueprints.
  - a. BP\_Pickup\_Parent
  - b. BP\_Pickup\_Child\_Coin
  - c. BP\_Pickup\_Child\_Cupcaket
  - d. **BP\_Pickup\_Child\_Health**
- 2. 위 블루프린트들을 선택한 뒤 **애셋**을 우클릭하고 '**애셋 액션 → 이주**'를 선택합니다.
- 3. 블루프린트가 참조하는 모든 애셋은 자동으로 **애셋 리포트**에 모아집니다. **확인**을 클릭합니다.
- 4. 만들어 둔 3인칭 플랫포머의 **Content** 폴더로 이동한 다음, **폴더 선택**을 클릭합니다.
- 5. 제대로 옮겨졌는지 확인하고 FundamentalsClasses 프로젝트를 닫습니다. 이제 이 프로젝트는 필요 없습니다.
- 신에 자손 블루프린트 세 개를 추가하고 제대로 작동하는지 테스트합니다. 실행하면 추가한 것들이 작아져서 캐릭터에 들어가고, 화면 좌측 상단에 메시지가 표시되는지 확인합니다.
- 7. 블루프린트가 제대로 작동하는 걸 확인했으면 씬에서 삭제합니다. 제대로 옮겨졌는지 확인하기 위해 잠시 놔둔 것뿐입니다.

- 8. **BP\_Pickup\_Child\_Cupcake** 블루프린트를 엽니다.
- 9. 이벤트 그래프에서 Destroy Actor 노드를 삭제하고 Pickup Grabbed 노드와 Print String 노드 사이의 연결을 끊습니다. 핀을 Alt + **마우스 좌클릭**하면 노드 사이의 연결이 끊어집니다.
- 10. **Get Player Character** 노드를 배치합니다.
- 11. Get Player Character 노드의 Return Value 핀에서 선을 드래그하고 Get Character Movement 노드를 배치합니다.
- 12.
   Character Movement 핀에서 선을 드래그하고 Set Jump Z Velocity 노드를 배치합니다. Pickup 노드를

   Set Jump Z Velocity 노드에 연결합니다. 그리고 Set Jump Z Velocity 노드를 Print String 노드에 연결합니다.
- 13.
   Jump Z Velocity의 기본 설정은 600 입니다. 설정을 변경하지 않았다면 Jump Z Velocity를 1200으로 설정합니다. 설정을 변경했다면 Jump Z Velocity를 기존의 두 배로 설정합니다.



- 14. 컴파일하고 컵케이크 블루프린트를 씬에 드래그 앤 드롭합니다. 테스트해 보세요! 점프 높이가 두 배가 되어야 합니다.
- 15. Get Player Character 노드의 Return Value 핀에서 선을 드래그하고 검색창에 'character movement'를 입력하면 캐릭터의 움직임을 바꿀 수 있는 변수 목록이 나타납니다. 다른 변수를 시험해 보면서 어떤 파워업을 만들 수 있는지 확인하세요.

여기서 흔히 발생하는 문제가 하나 있습니다. 플레이어가 추락한 뒤 부활하면 파워업이 리스폰되지 않습니다! 파워업이 있어야만 레벨을 통과할 수 있다면, 플레이어가 부활할 때마다 파워업도 리스폰되어야 합니다. **이벤트 디스패처**라는 새로운 노드를 사용하면 됩니다. 이벤트 디스패처는 블루프린트 이벤트 그래프의 **변수** 항목 아래에 있습니다. 이벤트 디스패처는 현재 블루프린트가 다른 블루프린트에서 발생하는 이벤트를 감지할 수 있도록 하는 방법입니다. 첫 번째 그랑프리 강좌에서도 특수한 이벤트 디스패처를 사용한 적이 있습니다. 소멸하는 플레이어의 이벤트 디스패처에 맞게 부활하는 플레이어를 바인딩했죠. 플레이어가 소멸하면 이벤트 디스패처 **On Destroyed**가 감지 중인 모든 블루프린트에 메시지를 보냅니다. **Game Mode** 블루프린트에 **On Destroyed** 이벤트를 바운딩해놨으므로 이벤트 디스패처가 플레이어 소멸을 감지하고 새로운 플레이어를 스폰하는 것입니다. 다른 이벤트 디스패처를 만들어 보겠습니다.

- 1. **ThirdPersonGameMode** 블루프린트에서 화면 왼쪽 아래의 **내 블루프린트** 패널로 이동합니다. '이벤트 디스패처' 옆의 더하기 아이콘을 클릭합니다.
- 2. 이벤트 디스패처 이름을 **Player Fell**로 설정합니다.
- OnDestroyed\_Event\_0 노드와 Delay 노드 사이에 'Call Player 호출'이라는 노드를 추가합니다. 이 노드는 우측 상단 모서리에 봉투아이콘이 있는 걸 알 수 있습니다. 이벤트 디스패처를 내보내고 있다는 뜻입니다. 잊지 말고 입력 실행 핀을 다시 연결합니다.
- 4. 블루프린트를 컴파일 합니다.
- 5. **BP\_Pickup\_Parent** 블루프린트를 엽니다. 이벤트 그래프에서 흔들림과 회전을 나타내는 애니메이션 **Floating**, 그리고 플레이어가 트리거와 겹칠 때 작동하는 애니메이션 **Pickup**, 을 확인합니다. 플레이어가 사망하면 이 애니메이션 두 가지를 초기화해야 합니다.
- 좌측 하단의 내 블루프린트 패널, 변수 항목에서 컴포넌트 왼쪽의 삼각형을 클릭하면 모든 컴포넌트 목록이 나타납니다.
   목록에서 Floating 을 드래그 앤 드롭하고 Get을 선택합니다. 타임라인 애니메이션에 대한 레퍼런스가 생성됩니다.
- 7. **커스텀 이벤트 추가...**를 찾아서 선택합니다. 이벤트 이름을 **PowerupRespawn**으로 설정합니다.
- 8. Floating 핀에서 선을 드래그하고 Play from Start노드를 추가합니다. Play from Start 노드를 방금 만든 PowerupRespawn 이벤트에 연결합니다.

- 9.
   이제 Set Relative Scale 3D 노드를 추가해서 Play from Start 노드에 연결합니다. New Scale 3D의 X, Y, Z값을

   1, 1, 1로 설정합니다. 컴포넌트 패널에서 Pickup Mesh를 이벤트 그래프로 드래그 앤 드롭하고 타깃으로 연결합니다.
- 10. 마지막으로는 콜리전을 변경해야 합니다. On Component 노드를 찾아서 삭제합니다. Set Collision Enabled 노드로 대체합니다. Pickup Trigger 메시를 다시 연결하고 New Type 을 'No Collision'으로 설정합니다. 이제 콜리전 컴포넌트가 삭제되지 않겠지만, 재활성화되지 않는 한 콜리전이 다시 처리되지 않을 뿐입니다.





11. 생성한 커스텀 이벤트 끝에 Set Collision Enabled 노드를 추가합니다. New Type은 'Query Only'로 설정해야 합니다.

PowerupRespawn     Custom Event	<b>f</b> Play from Start Target is Timeline Component	<b>f</b> Set Relative Scale 3D Target is Scene Component	<b>f</b> Set Collision Enabled Target is Primitive Component
•	• •		— • c
	🗢 Target	Target	
Floating		New Scale 3D	New Type
	Pickup Mesh	× 1.0 Y 1.0 Z 1.0	Query Only (No Physics Collision) -
		Piakun Triggar	
		Pickup Trigger	

- 12.
   거의 다 됐습니다. 이제 이벤트만 바인딩하면 됩니다. Event BeginPlay 노드에 연결된 Set Playback Position

   노드에서 ThirdPersonGameMode에 형변환 노드를 배치합니다. ThirdPersonGameMode에 형변환 노드의

   Object 핀에서 선을 드래그하고 Get Game Mode 노드를 배치합니다. ThirdPersonGameMode에 형변환

   노드에서 Assign Player Fell 노드를 배치합니다. Bind Event to Player Fell 노드의 타깃 핀을

   ThirdPersonGameMode에 형변환 노드의 As Third Person Game Mode 핀에 연결합니다.
- 13. 마지막으로 PlayerFell\_Event\_0 노드에서 Powerup Respawn 노드를 배치합니다. 블루프린트를 컴파일합니다.



이제 플레이어가 추락할 때마다 이벤트 디스패처가 켜지고, 이 사실을 감지한 부모 블루프린트가 커스텀 이벤트 PowerupRespawn 을 작동합니다.

파워업을 다 추가했으니 이제 피해야 할 장애물이 더 필요합니다! 게임에 불을 추가해봅시다. StarterContent 폴더에서 'P\_Fire'를 검색합니다. 여기서 P는 '파티클 효과'를 뜻합니다.

- 1. 블루프린트를 생성하고 이름을 '**BP\_Fire**'로 설정합니다. 생성한 블루프린트를 열고 **Particle System** 컴포넌트를 추가합니다.
- 2. **디테일** 패널의 'Particles' 아래에서 Template을 'P\_Fire'로 설정합니다.
- 3. 블루프린트에 Sphere Collision 컴포넌트를 추가하고, 불을 포함할 수 있도록 위치와 크기를 설정합니다.
- 4. 블루프린트를 컴파일합니다.
- 5. 블루프린트를 맵으로 드래그해서 불을 확인합니다.
- 6. **ThirdPersonCharacter** 블루프린트를 엽니다.
- 7.**컴포넌트** 패널에서 CapsuleComponent를 선택한 뒤 On Component Begin Overlap (CapsuleComponent)<br/>노드를 이벤트 그래프에 추가합니다.
- 8. CapsuleComponent가 불과 겹치는지 확인해야 합니다. 새로 만든 이벤트에서 선을 드래그하여 BP\_Fire에 형변환 노드를 생성한 다음, Overlap 이벤트의 Other Actor 핀을 Cast 노드의 Object 핀에 연결합니다.
- 9. 변수를 추가해 보겠습니다. 내 블루프린트 패널의 변수 항목에서 더하기 아이콘을 클릭합니다. 변수 이름을 'InFire'로 설정합니다.
- 10. **디테일** 패널에서 **변수 유형**을 Boolean으로 설정합니다. 사용할 수 있는 변수 유형은 많지만, 지금은 부울 변수가 필요합니다. 블루프린트를 컴파일합니다.
- 11. Cast 노드에 InFire 부울 변수를 설정하겠습니다. 변수 목록에서 InFire를 선택하고 이벤트 그래프에 드래그 앤 드롭합니다. Set을 선택하고 InFire 옆의 박스에 체크하여 변수를 true로 설정합니다.
- 12. **OnFire**라는 부울 변수를 생성합니다. 변수를 만든 뒤 블루프린트를 컴파일하는 걸 잊으면 안 됩니다.
- 13. Set InFire 노드에서 Branch 노드를 추가합니다. branch는 IF 조건문과 같습니다. OnFire 변수를 이벤트 그래프에 드래그 앤 드롭하고 Get을 선택합니다. OnFire 핀을 Branch 노드의 Condition 핀에 연결합니다.

		→ Cast To BP_Fire			- C Branch		
					SET		True D
Overlapped Component	0						
Other Actor	•	Object	Cast Failed [)	In Fire		Condition	False []
Other Comp	•		As BP Fire 🔿				
Other Body Index	0						
From Sweep					On Fire		
Sweep Result	•						

- 14. Branch 노드의 False 핀에서 선을 드래그하고 Spawn Emitter Attached 노드를 추가합니다.
- 15.**컴포넌트** 패널에서 Mesh를 선택해 이벤트 그래프로 드래그한 다음, Mesh 핀을 Spawn Emitter Attached 노드의<br/>Attach to Component 핀에 연결합니다.
- 16. Attach to Component 핀 아래에 Attach Point Name 핀이 있습니다. 여기에 핀을 연결하는 대신, 'None'을 클릭하고 'spine\_03'을 입력합니다. 불을 붙일 스켈레톤의 소켓입니다.
- 17. Location Type을 'Snap to Target, Keep World Scale'로 설정하고 Auto Destroy와 Auto Activate에 체크합니다.
- 18. Spawn Emitter Attached 노드에서 와이어를 드래그하고 Set OnFire 노드를 생성합니다. 박스를 체크합니다.

	True D	<b>f</b> Spawn Emitter Attached			
dition False	False 🗅 🗕	•	•	SET	D
		Emitter Template     P_Fire	Return Value 🔿	💿 On Fire 🛃	
	sh 🔶	Attach to Component			
		• Attach Point Name spine_03			
		◆ Location X 0.0 Y 0.0 Z 0.0			
		O Rotation X 0.0 Y 0.0 Z 0.0			
		Scale			
		Cocation Type Snap to Target, Keep World Scale ▼			
		👝 Auto Destroy 🛃			
		Pooling Method			
		👝 Auto Activate 💌			

- 19. 이제 플레이어가 불에서 나가면 어떻게 될지 설정해 봅시다. 다시 **컴포넌트** 패널에서 **Capsule**Component를 선택합니다. 이번에는 **On Component End Overlap (CapsuleComponent)** 노드를 이벤트 그래프에 추가합니다.
- 20. **BP\_Fire에 형변환** 노드를 생성하고 앞서 했던 것과 같은 방식으로 **On Component Begin Overlap** 노드에 붙입니다. 그리고 **Set InFire** 노드를 만드는데, 이번에는 박스를 체크하지 않습니다. 이제 **InFire**가 **false**로 설정됐습니다.
- 21. Set InFire 노드에서 Retriggerable Delay 노드를 배치하고 Duration을 4로 설정합니다.
- 22. Retriggerable Delay 노드에서 Branch 노드를 배치한 다음, Branch 노드의 Condition 핀을 Get InFire 노드에 연결합니다.
- 23. Branch 노드의 False 핀에서 Set OnFire 노드를 생성하고, OnFire가 false로 설정되도록 박스를 체크하지 않습니다.
- 24. 마지막으로 Spawn Emitter Attached 노드의 Return Value 핀에서 선을 드래그하고 Deactivate 노드를 배치합니다. Deactivate 노드를 Set OnFire 노드에 연결합니다. 블루프린트를 컴파일 합니다.



- 25. 레벨에 불을 배치하고, 그 위로 지나가면 어떻게 되는지 테스트합니다.
- 26. 파워업과 불을 레벨에 추가하고, 플레이어에게 새로 생긴 파워와 장애물에 맞게 디자인을 바꾸세요. 다음 워크샵에서는 불이 피해를 주도록 할 겁니다.





오프로딩 및 토론

**콘텐츠 브라우저**에서 Indoor라는 레벨을 검색합니다. 이 레벨에는 스탠드밖에 없는 텅 빈 데모 방이 있습니다. 지금까지 배운 기능을 써보고, 나만의 블루프린트 데모를 만들어 보세요. EssentialConcepts\_Start 레벨에서 다양한 노드와 컴포넌트를 사용해 보고, 프로젝트에 포함된 다른 레벨도 둘러보시기 바랍니다. 여러 블루프린트 사이에서 노드 네트워크를 복사 붙여넣기하고, 새로운 것을 시도해 보고, 부숴 보고, 다시 조합해 보세요. 강좌에 소개된 블루프린트 대부분은 Content → Fundamentals → Blueprints → EssentialConcepts → FinishedAssets → Blueprints 폴더에 포함되어 있습니다. Content → DemoRoom → BluePrint. 폴더에서 다른 블루프린트도 확인할 수 있습니다. 완벽한 작품을 만드는 게 아니라 블루프린트 시스템을 탐구하는 게 요점입니다.

데모 방에서 다양한 블루프린트를 직접 만들고, 팀원들에게 보여준 뒤 함께 논의해 보세요. 항상 그렇듯이 토론은 **3D**를 중심으로 진행하면 좋습니다.

- **어려움(Difficulties)**: 이번 주 워크샵에서 가장 어려웠거나 헷갈렸던 부분은 뭐였나요?
- 발견(Discoveries): 데모 블루프린트를 만드는 과정에서 발견한 흥미롭거나 멋진 사실은 어떤 것인가요?
- **꿈(Dreams)**: 이번 주 워크샵을 진행하면서 어떤 메커니즘이나 시스템이 떠올랐나요? 지금 알고 있는 지식으로 아이디어를 실현할 방법을 논의해 보세요.



