
Unreal Engine 4 게임 개발

Toy Battle

강현구

00

목차

1. 개발 목표
2. 개발 현황
3. 개발 정리

01

개발 목표

Unreal Engine 4을 이용한 게임 목표

캐주얼한 느낌의 멀티 플레이 배틀 로얄 게임 개발



Unreal Engine 4을 이용한 게임 개발 내용

1. 인 게임 기능 구현
2. 옵션 기능 구현
3. 리슨 서버를 기반으로 한 멀티플레이 구현

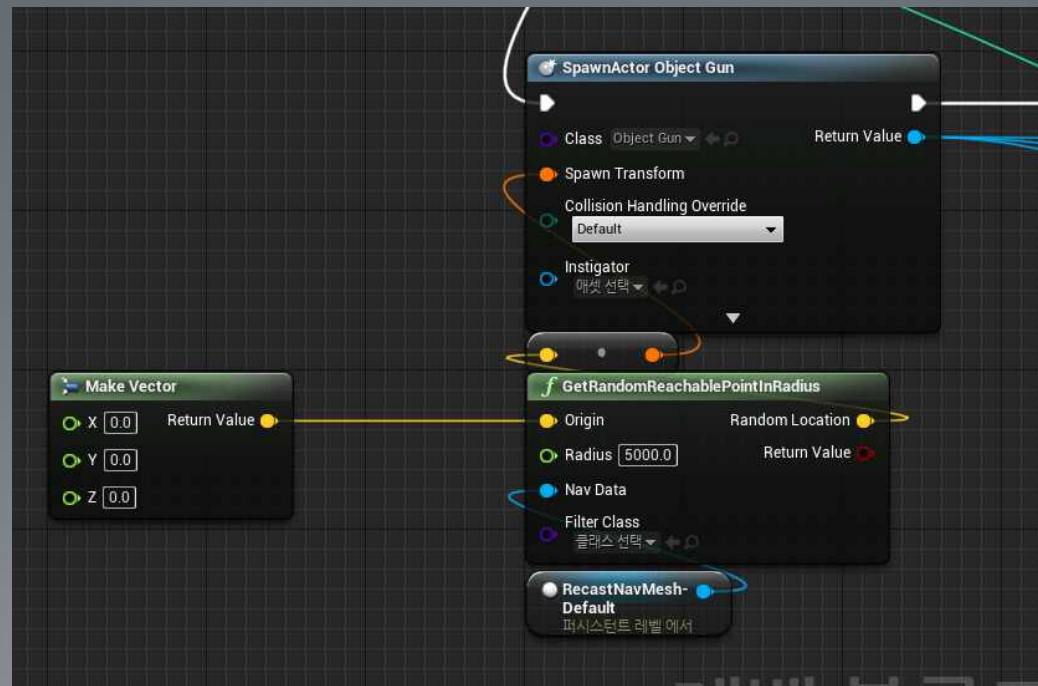
인 게임 기능 구현

1. RecastNavMesh를 사용하여 랜덤한 위치에 총 생성

RecastNavMesh의 정보를 활용하여 랜덤하게 총을 드랍합니다.

RecastNavMesh는 AI가 이동 가능한 범위를 나타내는 구역이며, 동적으로 처리되어 실시간으로 맵의 오브젝트에 반응하여 구역이 변합니다.

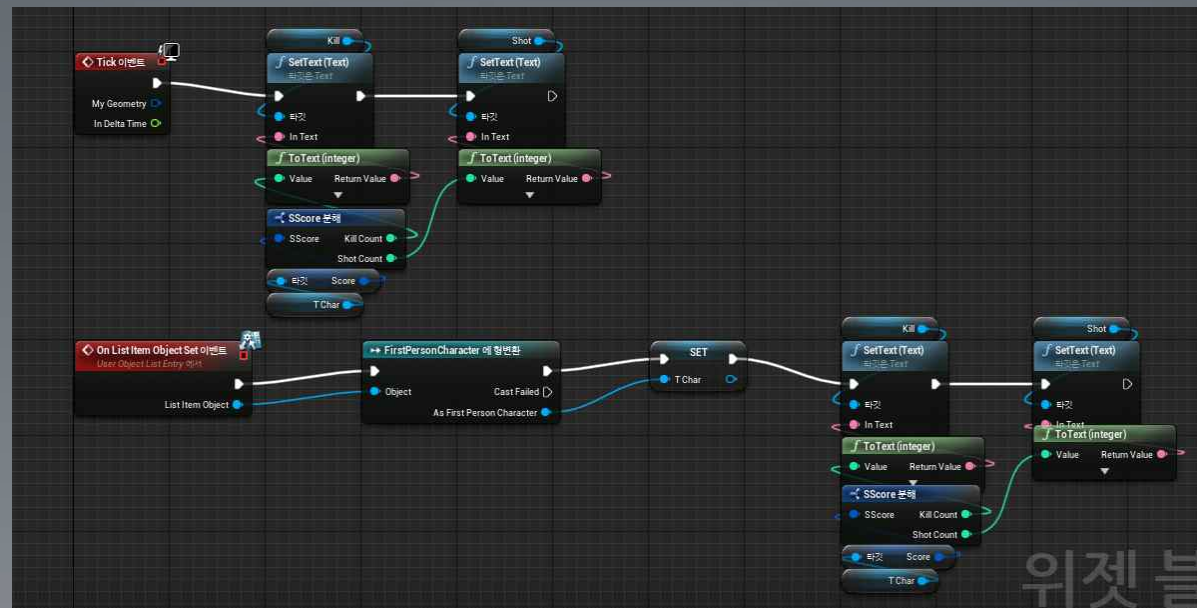
0, 0, 0을 중심으로 5000radius 범위 내에 랜덤한 위치에 총을 드랍합니다.



2. 게임 중 점수 창 구현

게임 중 실시간으로 유저의 정보를 가져와서 리스트 뷰로 점수창을 구현하였습니다.

아이템 리스트의 실시간 정보 변경

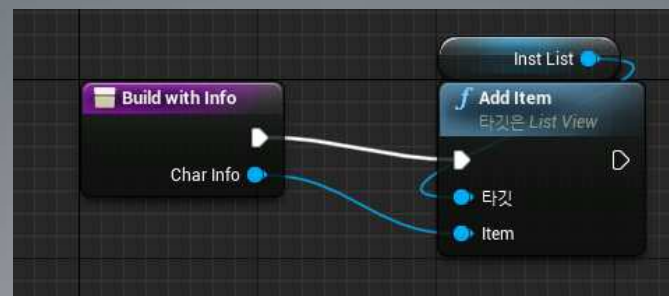
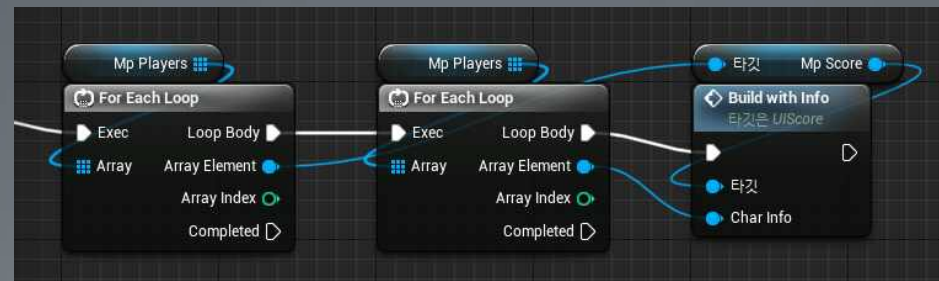


인 게임 기능 구현

2. 게임 중 점수 창 구현

리스트 뷰 아이템 등록

Game State에서 게임이 실행할 때 모든 플레이어들의 정보를 입력해준다.



인 게임 기능 구현

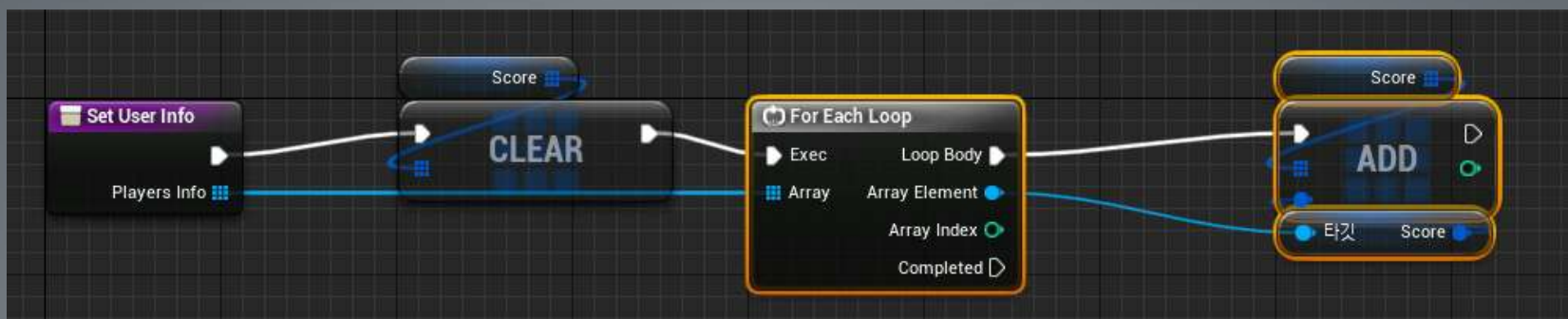
4. GamelInstance를 사용한 게임 종료 후 결과창 생성

PlayGame 종료 시에 결과창 출력에 필요한 정보들을 GamelInstance에 저장하고
EndGame 시작 시에 캐릭터 오브젝트를 생성하여 생성된 캐릭터 오브젝트에 정보들을 입력
하여 입력된 캐릭터 오브젝트를 리스트 뷰의 아이템으로 등록한다.

결과창 출력에 필요한 정보들을 GamelInstance에 저장하는 함수

Score는 유저의 정보에 관련된 정보로 이후 결과창 출력에 필요한 정보를 추가/삭제를 쉽게
하기 위해서 사용하였습니다.

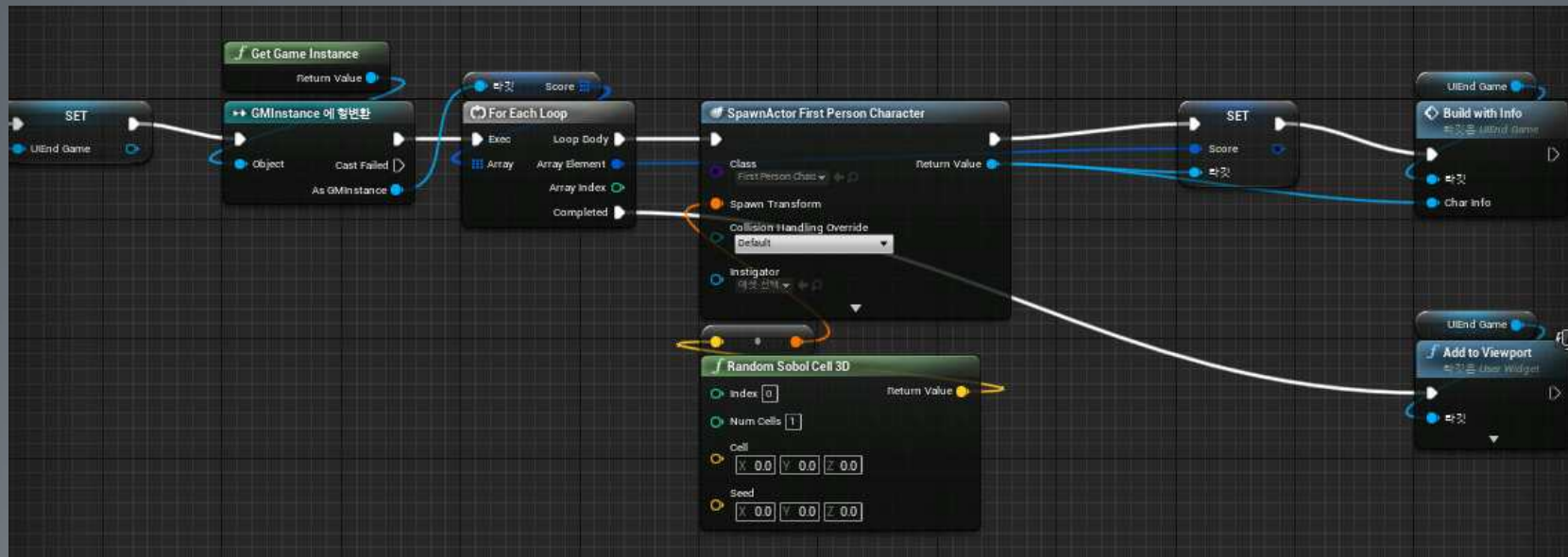
CLEAR를 하는 이유는 GamelInstance는 CLEAR하지 않으면 정보가 꾸준히 저장되기 때문
에 CLEAR하는 것으로 이전에 남은 정보를 삭제합니다.



4. GameInstance를 사용한 게임 종료 후 결과창 생성

EndGame 레벨에서 결과창에 출력하는 코드

리스트 뷰의 아이템으로 등록하기 위해서는 Object형태로 바꿔줄 필요가 있어서 캐릭터를 새로 스폰하여 그 안에 정보를 담아서 리스트 뷰의 아이템으로 등록하였습니다.

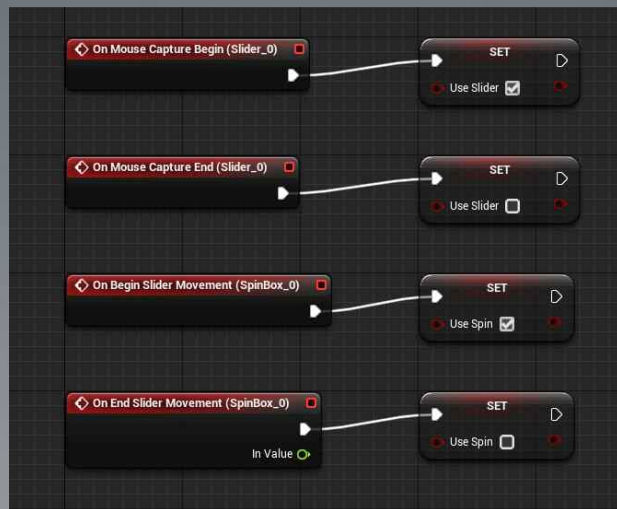


옵션 기능 구현

1. Camera의 PostProcess를 이용한 화면 밝기 조절

Get Player Pawn으로 Pawn의 정보를 FirstPersonCharacter로 형변환하여 FirstPersonCharacter의 FirstPersonCamera의 PostProcess의 LLens|Bloom의 Intensity 값을 변경하는 것으로 화면의 밝기를 조정한다.

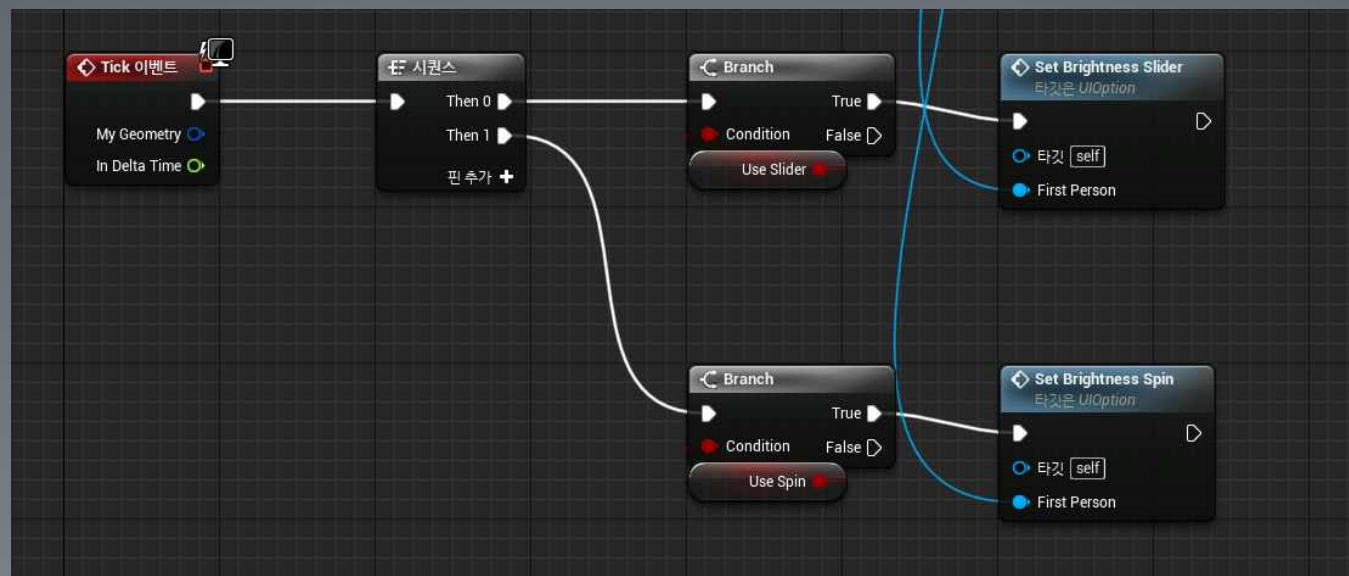
Slinder와 SpinBox의 값이 변경되고 있는 지 유무를 확인한다.



옵션 기능 구현

1. Camera의 PostProcess를 이용한 화면 밝기 조절

실시간으로 Slinder와 SpinBox의 값이 변경되는 지 확인하여 변경되고 있을 경우 밝기를 조정한다.

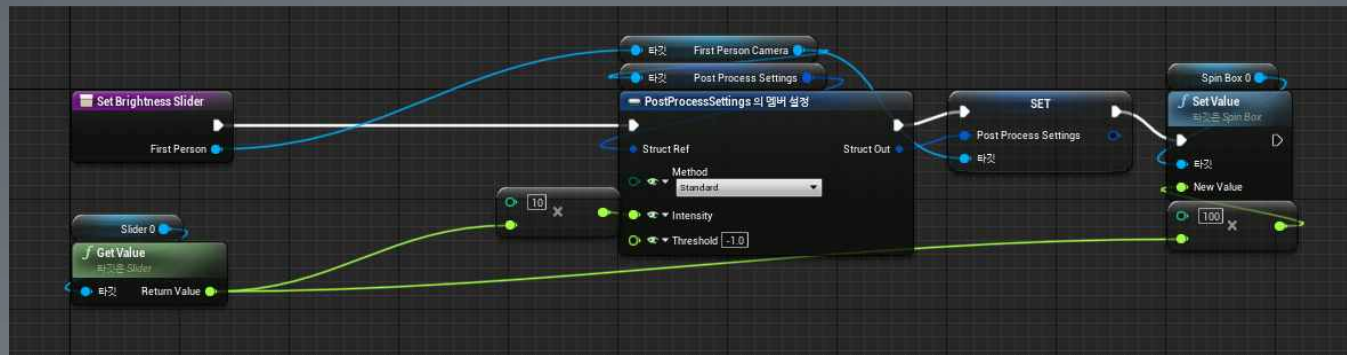


옵션 기능 구현

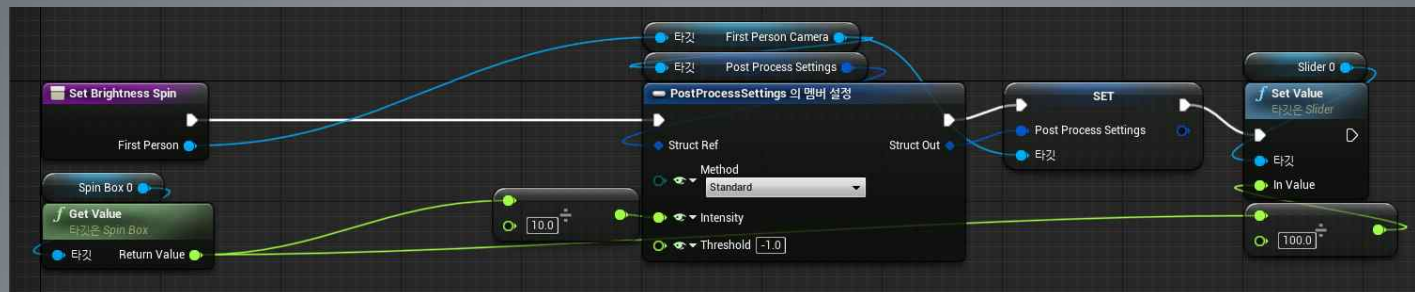
2. UI의 Slinder와 SpinBox의 동기화

UI의 Slinder와 SpinBox의 값을 이용해 화면 밝기 및 음량을 조정하며, Slinder와 SpinBox의 값이 항상 일정한 비율로 표시되게 하기 위해 동기화 한다.

Slider의 값으로 화면 밝기를 조절하는 함수



SpinBox의 값으로 화면 밝기를 조절하는 함수

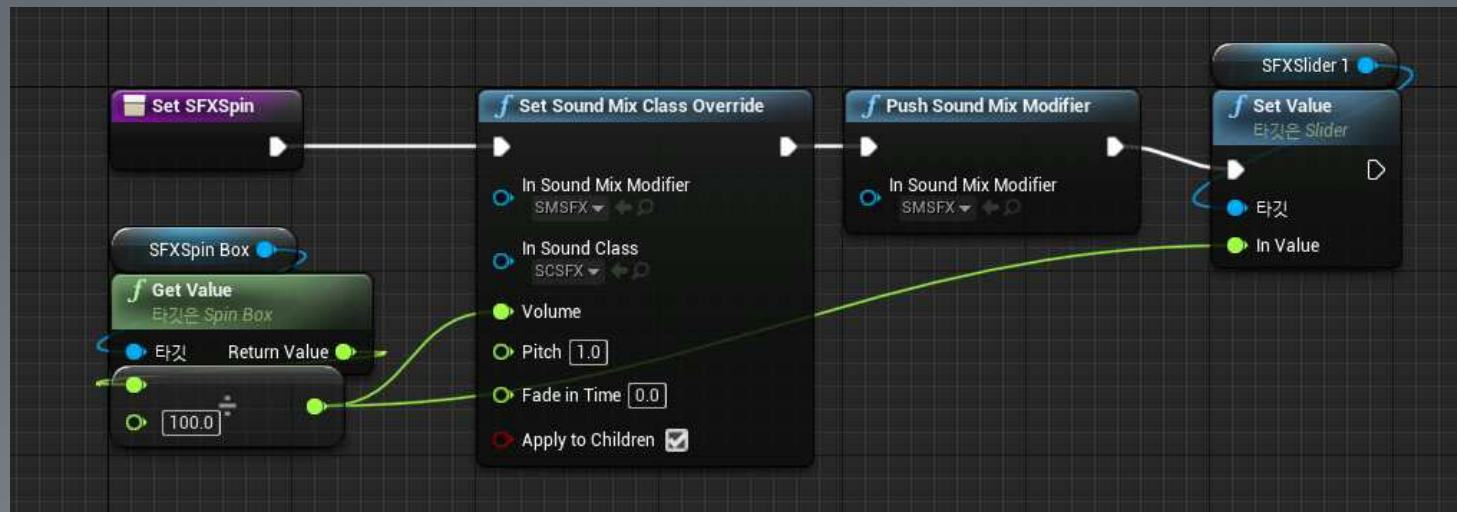


옵션 기능 구현

3. Sound Mix Class와 Sound Master Class를 사용하여 음량 조절

UI의 Slinder와 SpinBox의 값을 이용해 음량의 크기 값을 입력받는다.
 입력받은 값으로 Set Soun Mix Class Override 함수를 사용하여 음량 Volume에 적용한다.
 적용 후 Slinder와 SpinBox의 값을 동기화한다.

Slider의 값으로 SFX 음량을 조절하는 함수

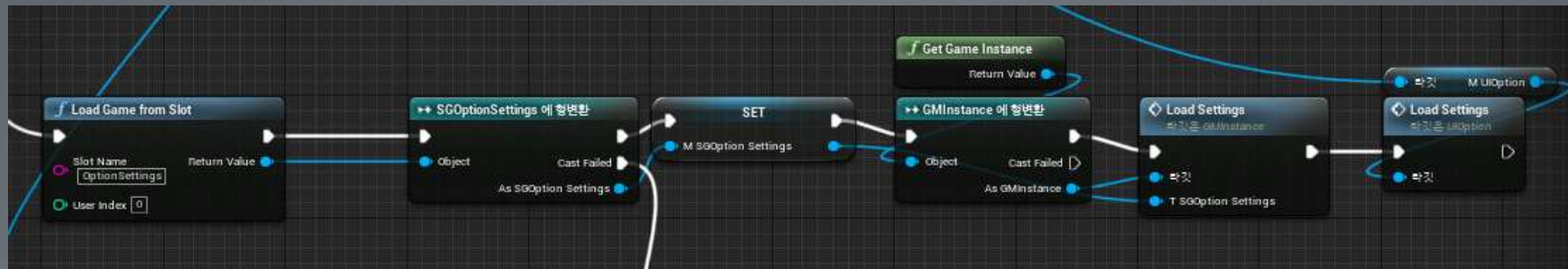


옵션 기능 구현

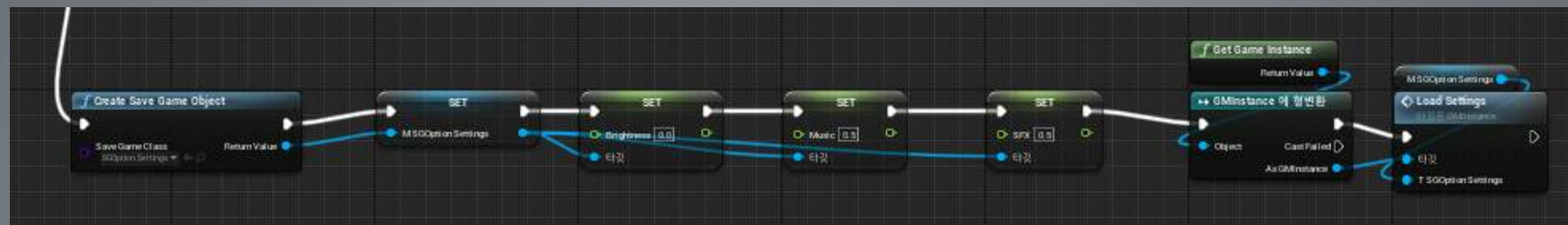
4. 옵션 값 불러오기 및 저장 구현

Load Game from Slot함수와 Async Save Game to Slot함수를 사용하여 파일을 불러오거나 저장한다.

Load Game from Slot함수를 쓰는 과정에서 저장된 옵션 값이 있을 경우 불러온 옵션 값을 옵션 UI에 세팅한다.



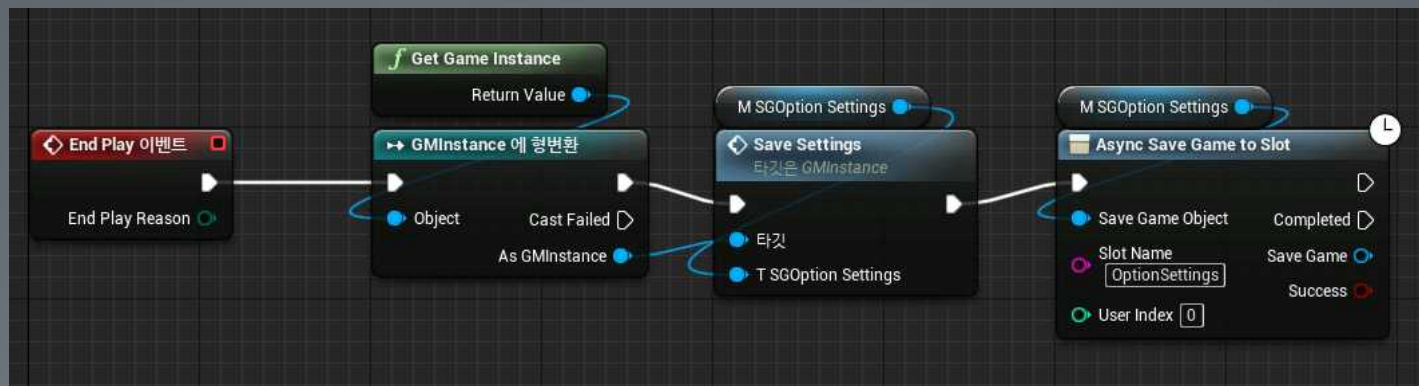
Load Game from Slot함수를 쓰는 과정에서 저장된 옵션 값이 없을 경우 옵션 값을 새로 생성하여 초기화 후 옵션 UI에 세팅한다.



옵션 기능 구현

4. 옵션 값 불러오기 및 저장 구현

타이틀에서 게임 종료 시 Async Save Game to Slot 함수를 사용하여 파일을 저장한다.



리슨 서버를 기반으로 멀티플레이 환경 구현

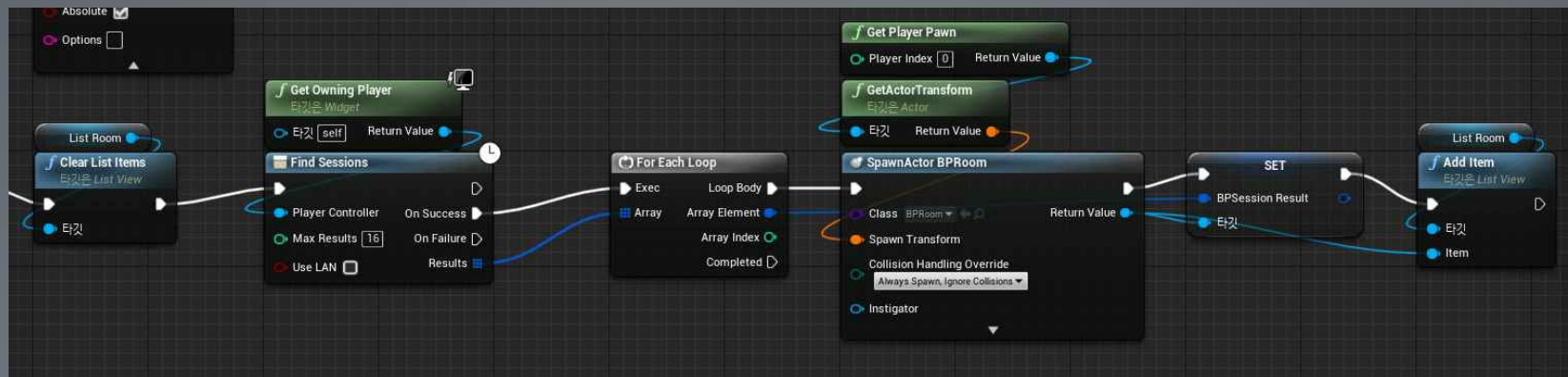
1. Find Session 함수를 사용하여 Session 찾기 및 목록 출력 기능 구현

기존에 있던 Session의 목록을 비워낸다.

Find Session 함수를 사용하여 Session의 정보들을 가져온다.

Session의 정보들을 순차적으로 BPRoom이라는 액터에 담아서 ListRoom에 추가한다.

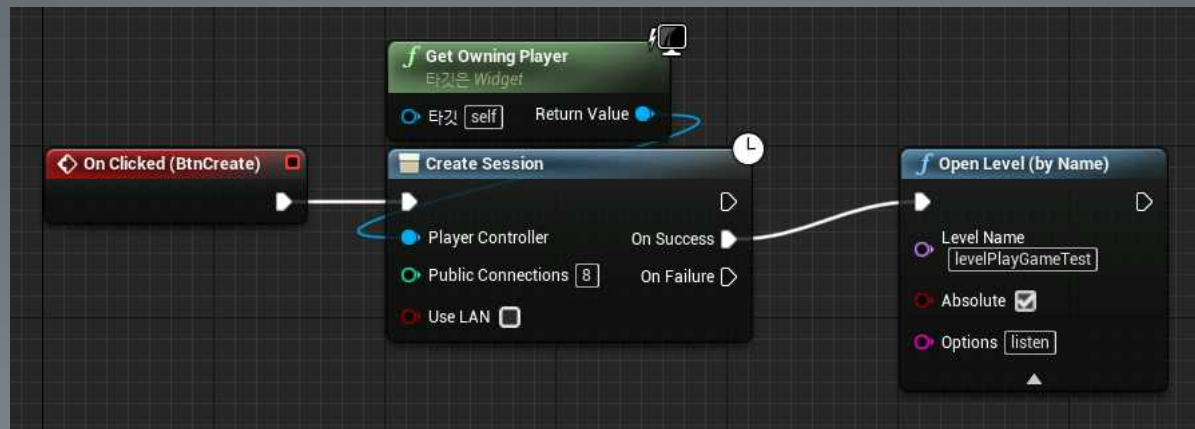
추가된 BPRoom의 정보를 토대로 화면에 방의 목록이 출력된다.



리슨 서버를 기반으로 멀티플레이 환경 구현

2. Create Session 함수를 사용하여 Session 생성 및 Join Session 함수를 사용하여 Session 접속 구현

Create Session 함수를 사용하여 Session을 생성한다.
생성된 Session은 listen 서버이며, 서버는 Session을 생성한 생성자가 담당한다.



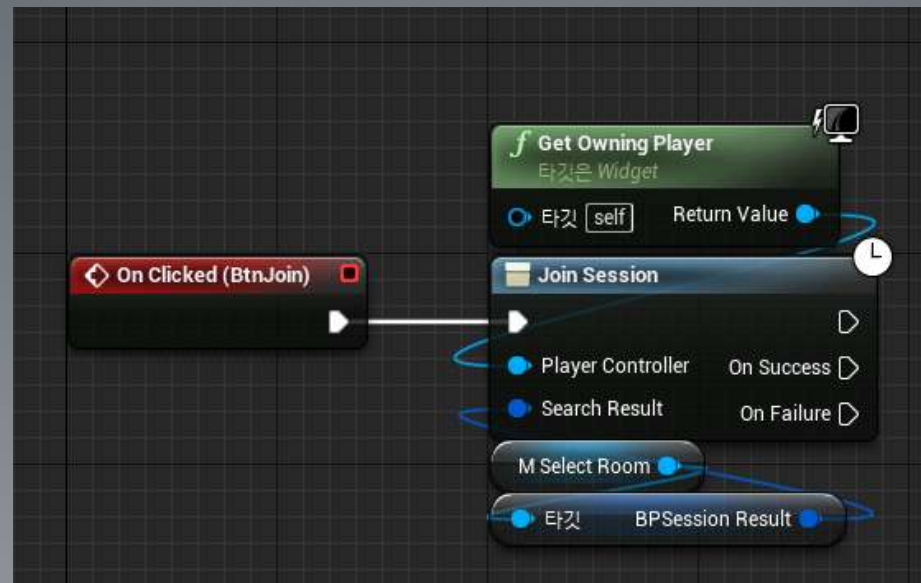
리슨 서버를 기반으로 멀티플레이 환경 구현

2. Create Session 함수를 사용하여 Session 생성 및 Join Session 함수를 사용하여 Session 접속 구현

방 목록 중 참가할 방을 클릭하여 선택한다.

선택된 방의 Session 정보를 토대로 Join Session 함수를 사용하여 해당 Session에 접속한다.

접속된 Session은 listen 서버이며, 참가자는 클라이언트 역할만을 담당한다.



개발 후 자기 평가

초기의 목표인 멀티 플레이와 캐주얼 한 배틀 로얄이라는 게임의 틀은 잡히게 되었습니다. 하지만 시간 분배의 문제로 완성도가 미흡했으며, C++을 사용하여 코딩을 하지 못 하여서 아쉬웠습니다. 만약 시간이 더 있었다면 블루 프린트로 만든 내용을 토대로 C++로 변경하여 블루 프린트와 C++을 혼합한 개발을 하고 싶었습니다.

감사합니다