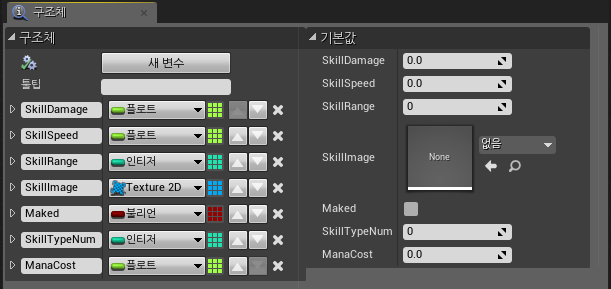
|  |  |  |
| --- | --- | --- |
|  | | |
|  | **프로젝트 실무** |  |
|  | 개인 기술문서 |  |
| http://www.inu.ac.kr/mbshome/mbs/inu/images/0801_img_1.jpg | | |

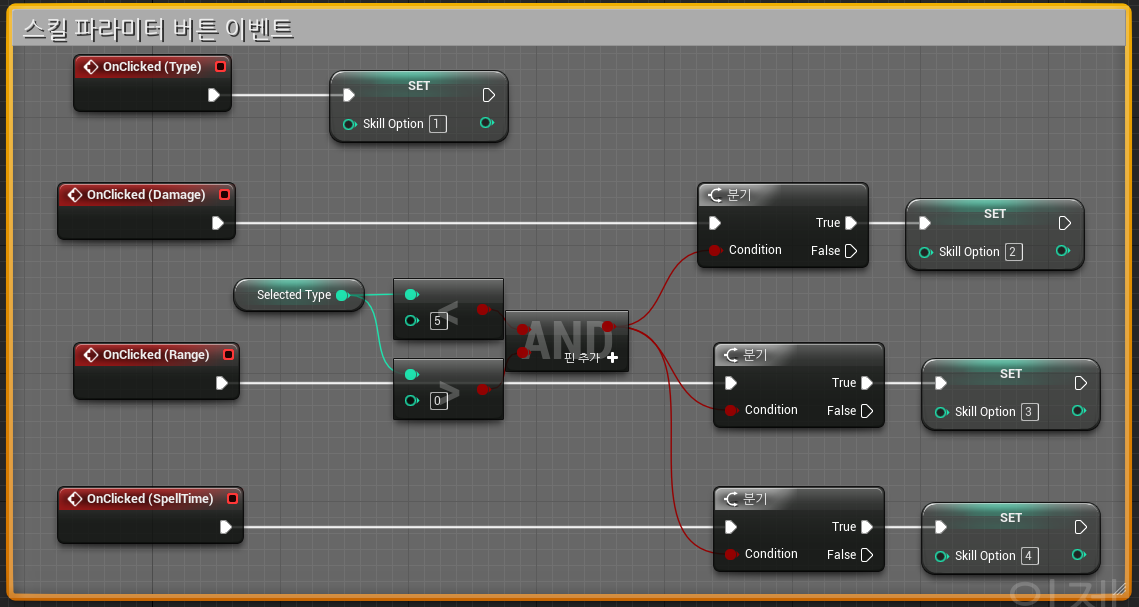
목차

1. 스킬 커스터마이징
2. UI
3. 맵 디자인
4. 상성 구현, 피아 구분
5. 파티클 추가

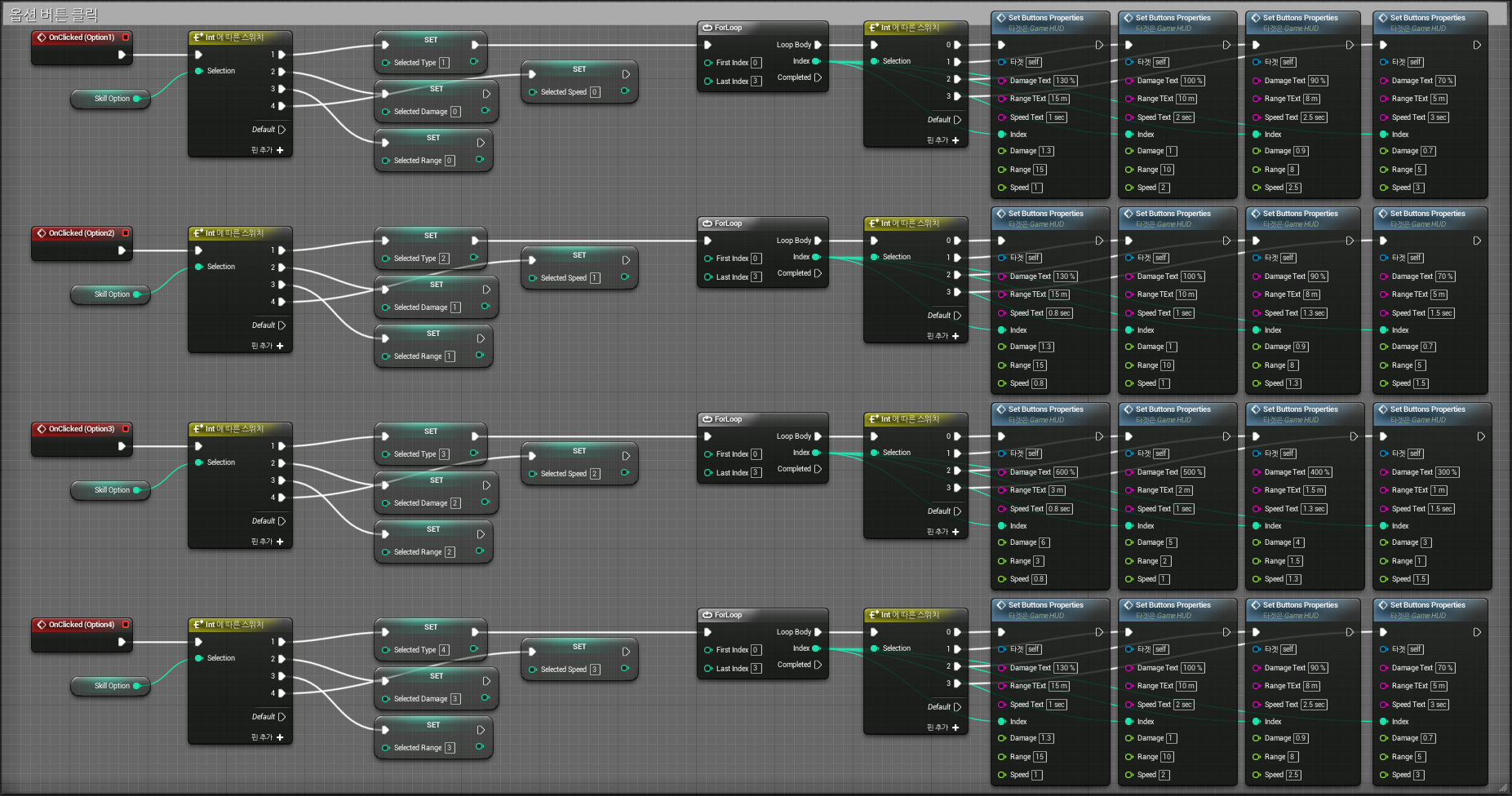
* 타워 커스터마이징
  + 플레이어가 직접 타워를 제작하여 사용하는 시스템
  + 타워의 공격 타입(유형)의 선택에 따라 설정 가능한 파라미터 값이 변경됨
  + 제작한 타워를 등록하면 퀵슬롯에 등록되어 핫키로 사용 가능
  + HUD 블루프린트에서 스킬을 제작하여 플레이어 캐릭터의 퀵슬롯에 저장한다.



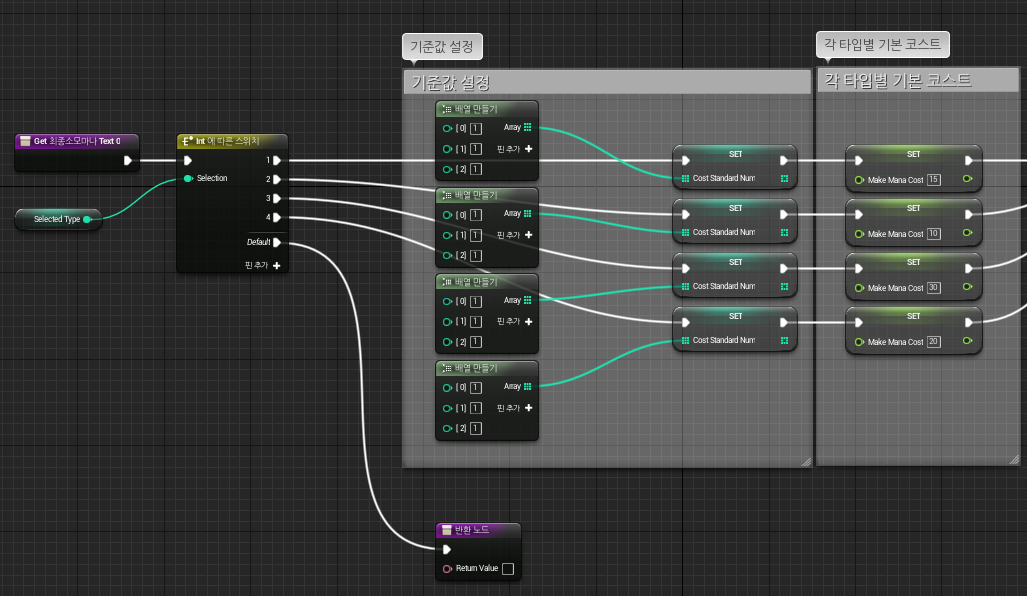
* SkillStruct 구조체를 만들어 타워의 정보를 저장.
  + SkillDamage: 비율
  + SkillSpeed: 초단위
  + SkillRange: 미터 단위
  + SkillImage: 퀵슬롯에 등록되는 이미지
  + Maked: 퀵슬롯에 타워가 등록되어 있는지 확인하는 불리언 변수



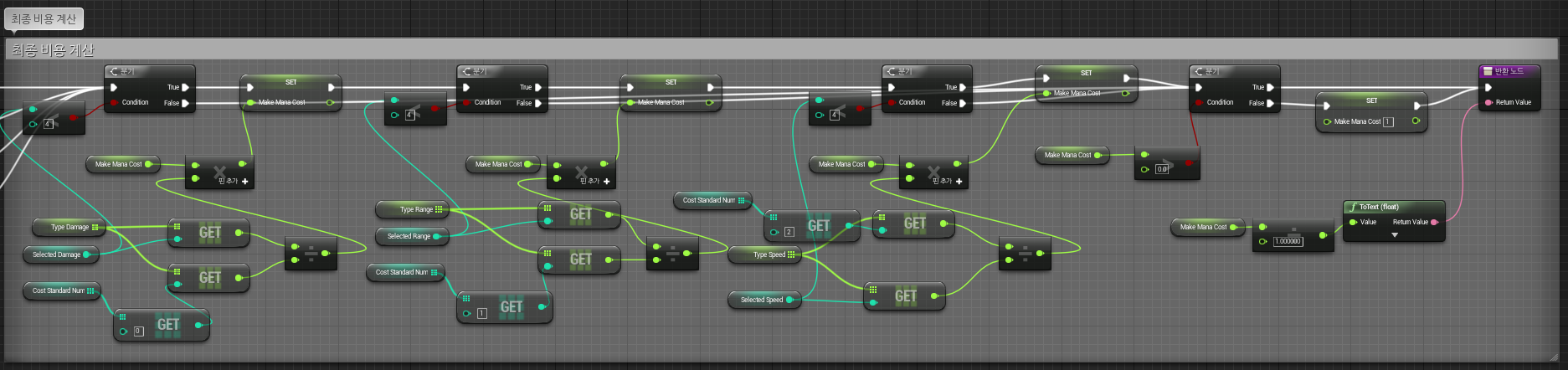
* 각 파라미터 타입별 버튼 이벤트
  + 타입을 설정하지 않으면 데미지, 사정거리, 공격속도 버튼 클릭 이벤트 비활성화



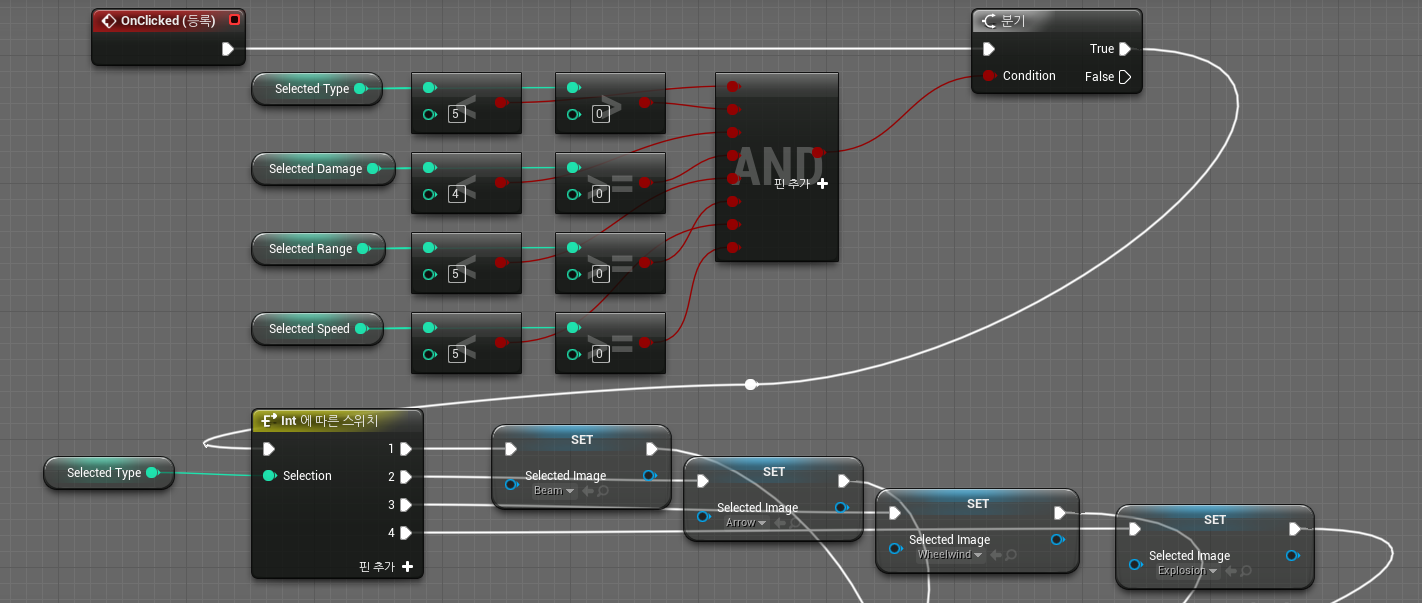
* 옵션버튼 클릭 이벤트가 발생하면 해당 파라미터 값 선택
  + SetButtonsProperies에서 각 타입별 설정가능한 파라미터 값들을 배열에 저장

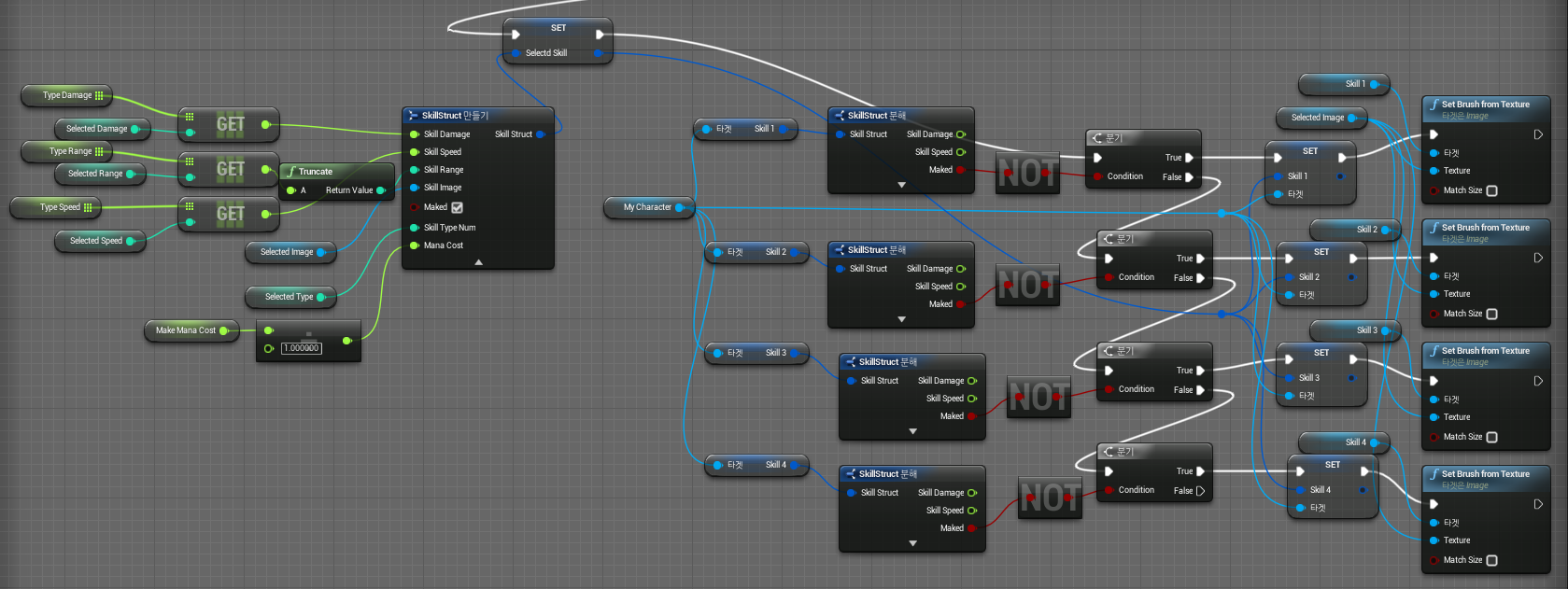


* 각 스킬 유형별로 비용 계산의 기준이 되는 기본값 설정
  + 설정한 값과 기본값의 비율을 이용하여 최종 코스트 계산
  + 배열의 0, 1, 2에는 순서대로 공격력, 사정거리, 공격속도의 기본값이 들어감
* 각 타입별 기본 코스트 설정
  + 밸런스 표를 토대로 기본값 설정.

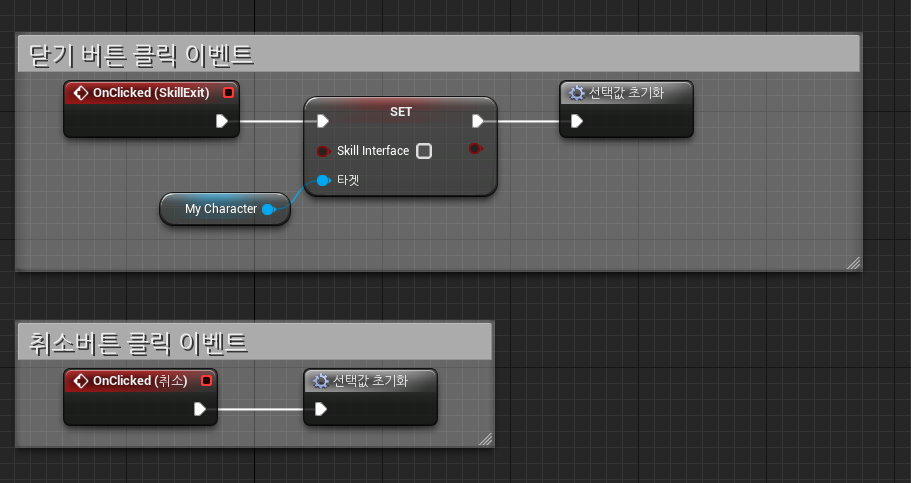


* 유저가 설정 파라미터에 따라 비용 계산
  + 불리언 변수를 이용하여 플레이어가 설정한 파라미터 값만 받아서 비용 계산
  + 정해진 공식을 이용하여 기본값과의 비율에 따라 비용 계산
  + 계산된 값은 타워 제작창에 표시되어 확인하면서 비용 확인하면서 제작 가능

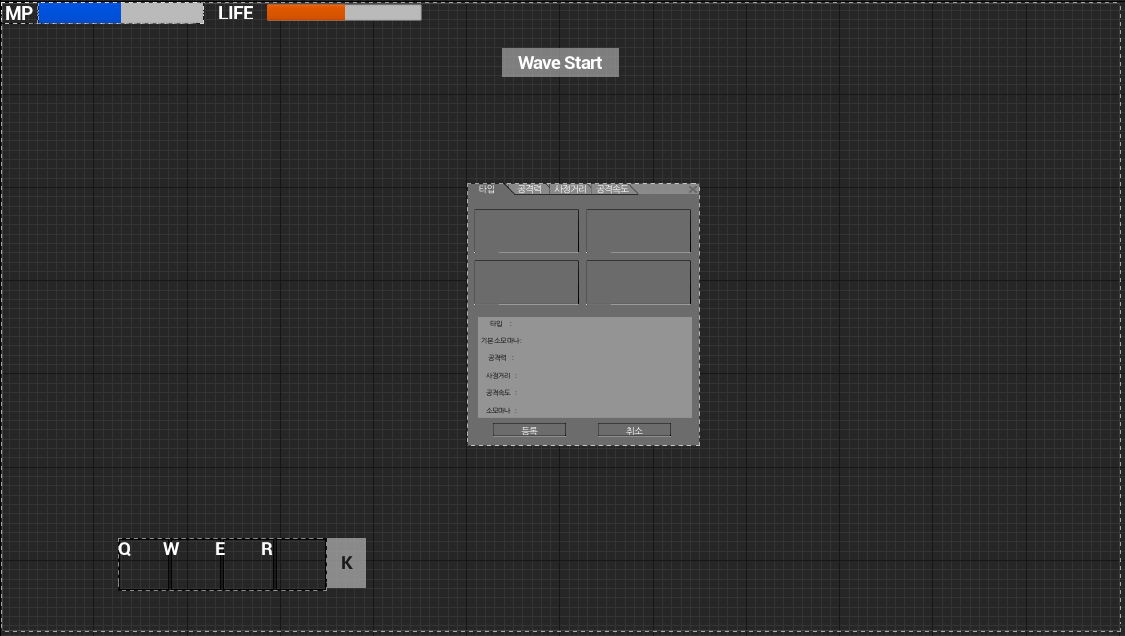




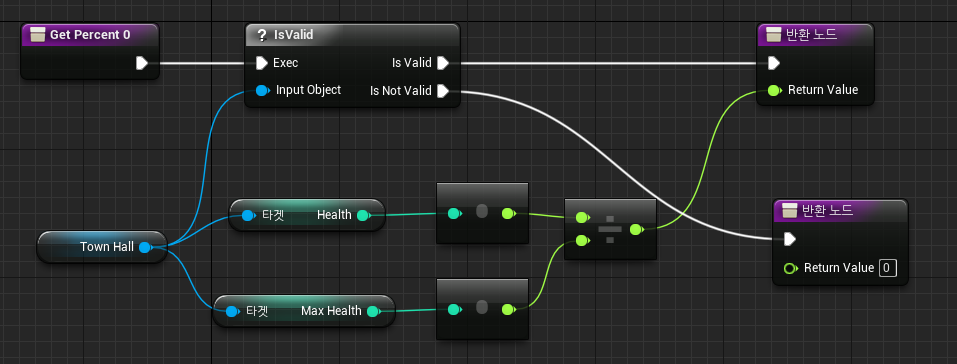
* 스킬 등록버튼 이벤트
  + 모든 파라미터 값이 설정되어 있는지 확인하여 True일 때에만 등록 진행
  + 선택한 타입에 따라 이미지 저장
  + 선택한 값들로 TowerStruct 배열을 만들어서 비어있는 퀵슬롯에 등록
  + 퀵슬롯이 가득 차있으면 등록 안됨
  + 등록된 퀵슬롯의 이미지를 저장한 이미지로 바꿈
  + 퀵슬롯에 스킬이 등록되어 있는지는 TowerStruct의 Maked 불리언을 사용



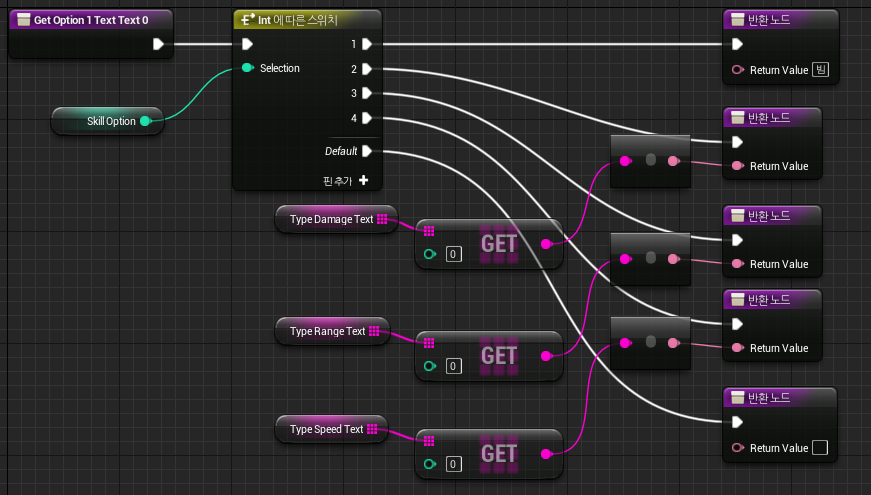
* 스킬창 닫기 버튼과 취소 버튼을 눌렀을 때에는 플레이어가 선택했던 파라미터 값들을 초기화



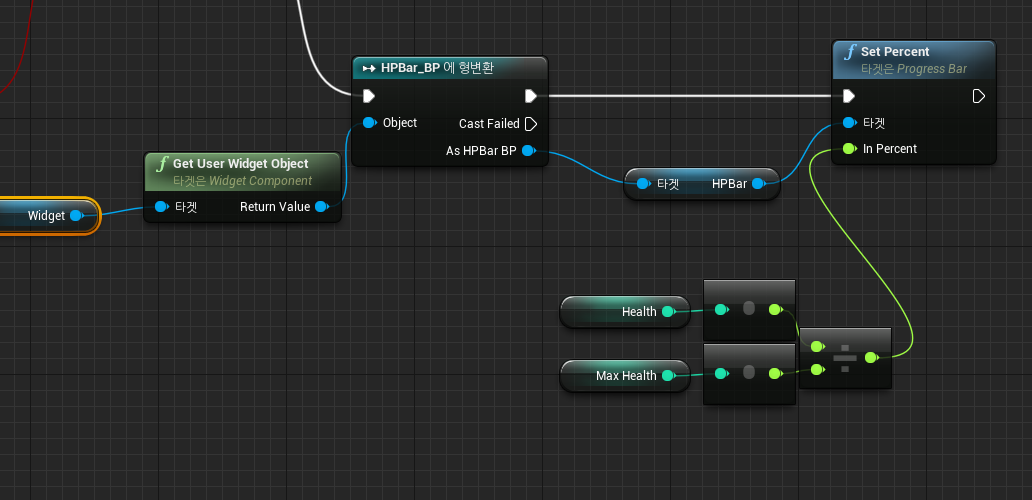
* UI
  + HUD 블루프린트를 이용하여 제작
  + 스킬등록창 UI는 포토샵을 이용하여 제작함
  + 스킬등록창의 회색으로 된 패널에는 플레이어가 현재 설정 중인 타워의 정보를 표시
  + 각 버튼별로 클릭 이벤트에 따른 기능 구현
  + MP와 LIFE는 플레이어 캐릭터의 MP와 회관의 Health 값을 불러온다.



* GetLifePercent 함수
  + 회관의 최대 체력과 현재 체력의 비율을 반환하여 LIFE바를 채운다.
  + 회관이 적의 공격에 의하여 제거되었을 때에는 0을 반환한다.



* Get Option Text 함수
  + SkillOption 변수로 현재 선택한 파라미터에 대한 정보를 받아 텍스트를 반환
  + Type을 제외한 변수들은 플레이어가 설정한 Type에 따라 배열에서 값을 받아 텍스트로 반환



* 적군 체력 표시
  + HPBar 위젯블루프린트를 만들어서 적 캐릭터의 블루프린트에 컴포넌트로 추가한다.
  + 위의 그림과 같이 현재 체력이 변화할 때 마다 바의 퍼센트를 갱신하여 표시한다.

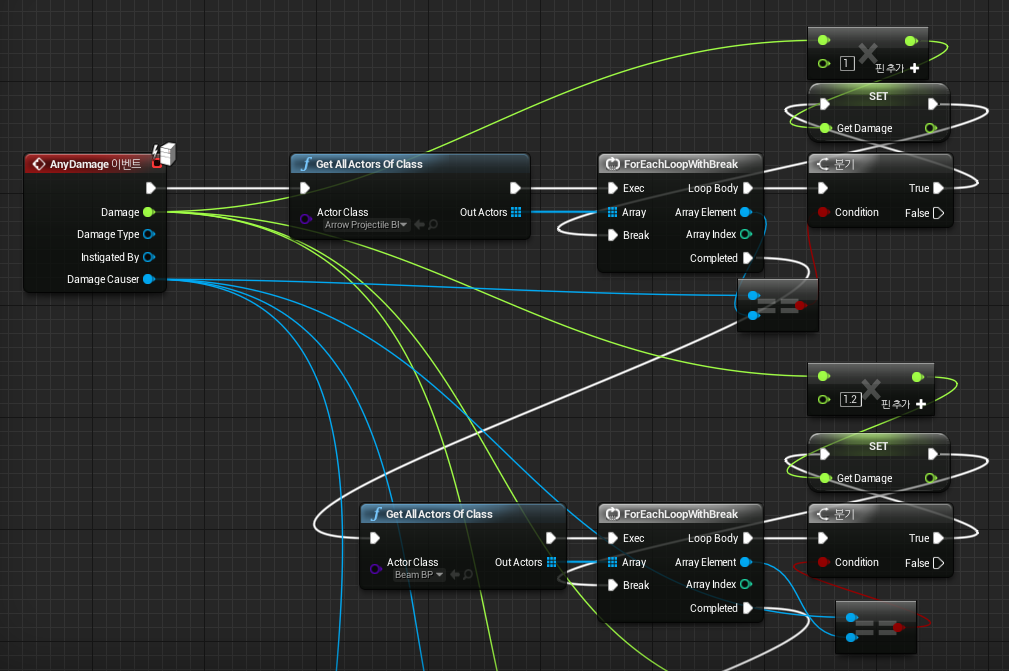


(최종 맵 전경)

* 맵 디자인
  + 최종 맵의 적 유닛이 오는 길은 갈림길이 없는 일자형 길
  + 범위 공격과 관통형 공격이 사용 가능하게 직선의 긴 구간과 길이 꺾이는 구간을 만들어줌
  + BlockBolume을 이용하여 네비게이션 맵을 설정
  + BlockBolume이 플레이어 캐릭터와 부딪히지 않게 콜리전 설정

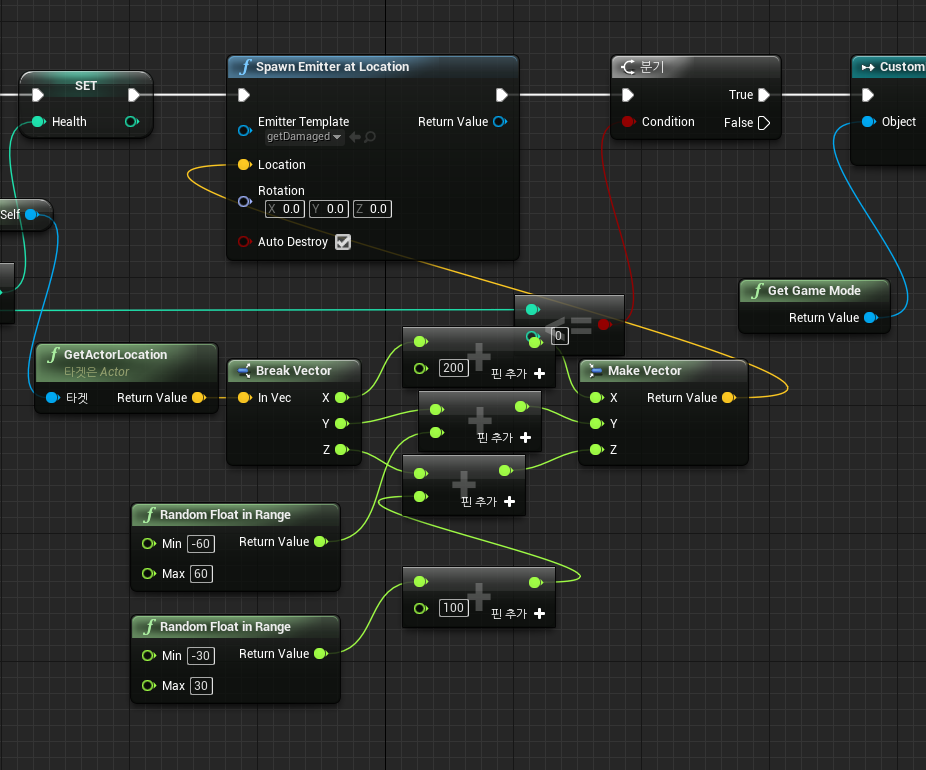
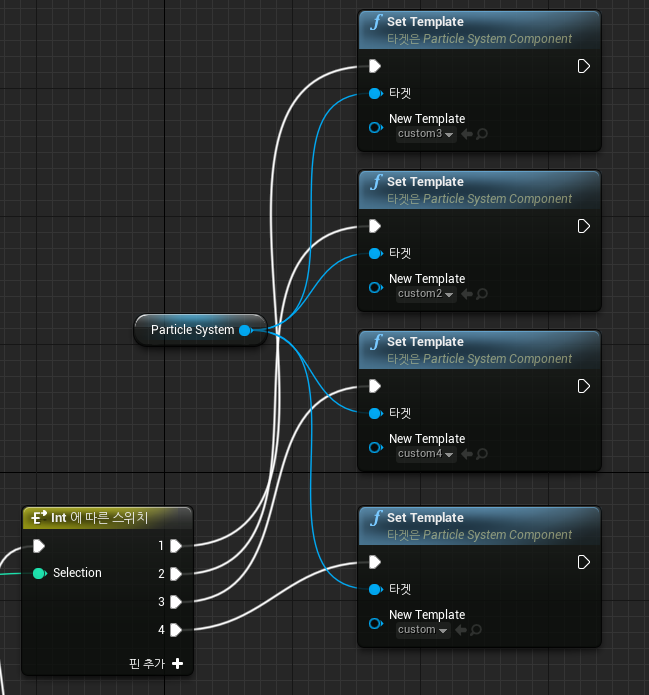


타워가 발사하는 발사체에서 데미지를 전달하는 ApplyDamage



적 캐릭터 BotGrunt에서 데미지를 받았을 때 발생하는 AnyDamage이벤트

* 상성
  + 발사체 블루프린트의 Apply Damage의 Damage Causer에 자기 자신을 넣어서 전달
  + 적 캐릭터 블루프린트의 AnyDamage이벤트 Damage Causer에서 발사체의 정보를 확인할 수 있음
  + GetAllActorsOfClass로 현재 레벨에 생성된 액터 중 해당하는 클래스(화살, 빔, 폭발, 휠윈드)로 생성된 액터의 정보를 모두 확인하여 데미지를 준 액터와 비교함
  + 데미지를 준 액터의 정보가 확인되면 정보에 따라 상성을 적용하여 받는 데미지 값을 조정함



* 파티클 추가
  + 타워를 공격 타입별로 구분 가능하게 파티클을 설정함
  + 몬스터가 회관 공격시 적정한 위치에 이펙트 생성
    - 피격위치에 생성시키지 않는 이유는 한곳에 공격이 집중되면 이펙트도 한 곳에 집중되어 되려 부실해 보일 수 있음
  + 구해온 파티클이 사용하고자 하는 것과 달라 기존에 스터디했던 자료를 토대로 변형하여 적용