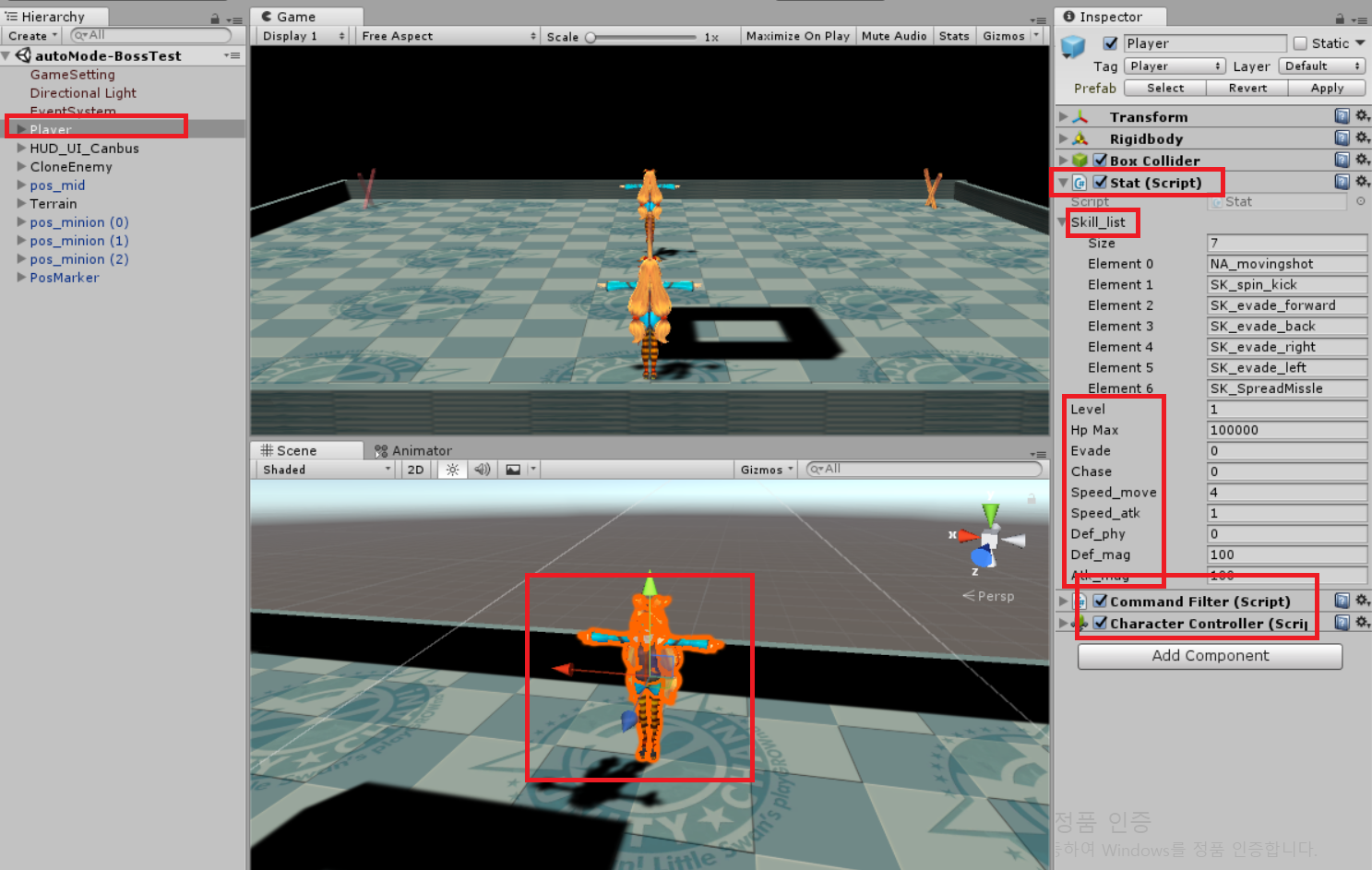
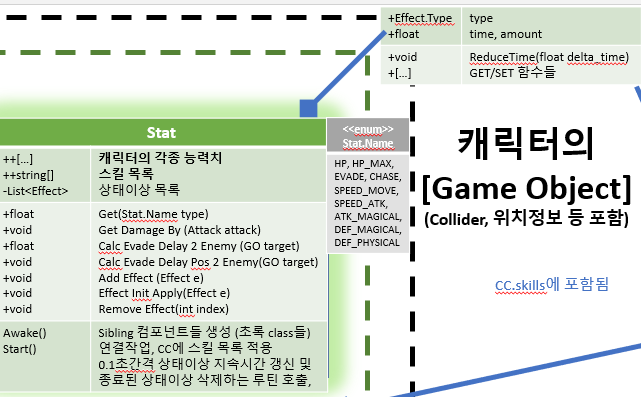
교수님 수업에서 배운 정상적인 UML을 만들어 보여드리려 했으나, 유니티 프로젝트와 잘 안 맞는 부분이 많아 불가피하게 파워포인트로 많은 부분을 작업했습니다. 다음은 **UML을 변형시켜야 했던 이유들과, 변형된 도면을 읽는 방법**에 대한 설명입니다.

**1.게임오브젝트와 컴포넌트**



유니티의 모든 시스템은 Game Object를 중심으로 작동합니다. (이하 GO) 지금 가운데 화면에 붉은 박스로 표시된 캐릭터는 [Player]라는 이름을 가진 GO입니다.

GO는 자신의 부속 Component로 여러가지 특성을 가질 수 있습니다. 우측 창을 보면 그 목록을 볼 수 있습니다. 위치, 속도, 무게, 크기, 충돌체크 등 여러 기능을 하는 Component들도 있지만, Script Component가 3개 부착되어 있습니다. Stat, Command Filter, Character Controller. 이는 변형된 클래스 다이어그램에서 아래와 같이 표현했습니다.

검은색 점선이 GO를 뜻하며, GO 안에 배치된 Class는 해당 GO의 Component로 사용됨을 의미합니다.

배치 다이어그램과 클래스 다이어그램이 혼합된 형태이나, 어느 GO에 배치되었는지가 중요해 이와 같이 도면을 구성했습니다.

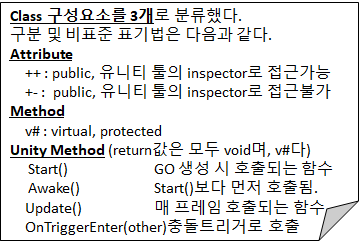
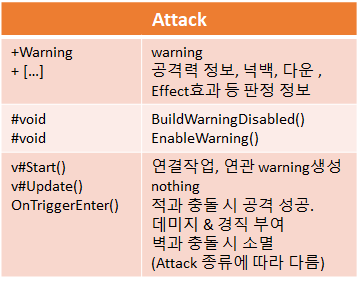
**2. Awake(), Start(), Update(), On Trigger Enter()**

Class Diagram에 모든 attribute와 method를 표기하진 않았고, 중요 함수들만 표기했습니다. 또한 1인개발에 개발중 설계변경이 잦다 보니 대부분의 attribute를 public으로 사용합니다.

**유니티 메소드는** Unity.MonoBehavior로부터 상속받은 virtual method들을 의미합니다.

**일반적인 method들과 사용방식이 아예 다르므로 따로 표기했습니다**.

아래는 UML상에 표기된 클래스의 샘플과 주석입니다. 다음과 같은 해석이 가능합니다.



Attribute Warning warning과 기타 변수들이 public

Method 2개의 virtual void 함수 보유.

Unity Method Attack클래스를 컴포넌트로 가지는 GO가 생성되면,

warning을 생성하고, 클래스간 레퍼런스 연결작업을 한다.

virtual이므로 상속 후 변형가능.

Update()함수는 초당 60회 호출되며, Attack에서는 특별한 행동을 하지 않는다.

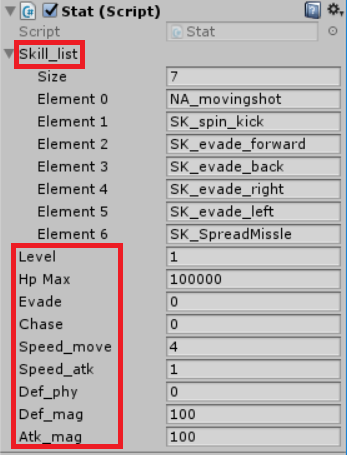
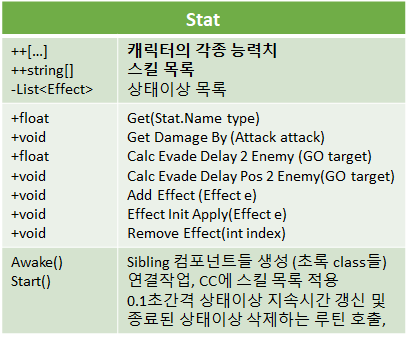
하지만 Attack을 상속받은 Child에서는 뭔가 할 예정

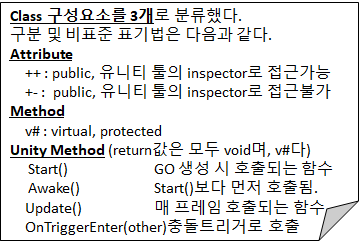
다른 Collider와 충돌 시, 만약 상대가 적이라면 공격관련 함수를 호출한다.

OnTriggerEnter는 추가적으로 상속, 변형할 계획이 없다.

사실 On Trigger Enter()도 Unity.MonoBehavior로부터 상속받은 함수지만, 더 이상 virtual로 사용할 계획이 없기 때문에 v#을 표기하지 않았습니다.

**3. Unity Inspector에서 변형가능한 ++attribute**

****아래는 ++표기를 덧붙인 class의 예시입니다.



+는 public이고, ++는 public임과 동시에, Unity Inspector에서 수정가능한 value라는 뜻입니다.

오른쪽은 Inspector창에서 수정 가능한 데이터를 보여주며, 이 캐릭터는 7개의 스킬을 보유하고, 능력치가 아래와 같이 세팅 되어 있음을 나타냅니다.

Unity Inspector상에서 설정한 변수는 코드 내에서 지정한 초기값보다 더 높은 우선권을 가집니다. 같은 타입의 GO들이 서로 다른 변수를 이용해 동작하도록 할 때 유용하게 사용되는 기능이나, 남발하다 보면 스크립트와 인스펙터가 서로 꼬이는 문제가 발생해 따로 표기했습니다.

**04. 시퀀스 다이어그램**

UML 시퀀스 다이어그램으로 5개 이상의 Class간 복합적 움직임을 표현함과 동시에 GO내 component배치까지 표현하기 어려워 02~05번과 같이 변형된 시퀀스 다이어그램을 사용했습니다. 화살표를 1,2,3,4 순서대로 읽으시면 기존 시퀀스 다이어그램과 크게 다르지 않습니다.