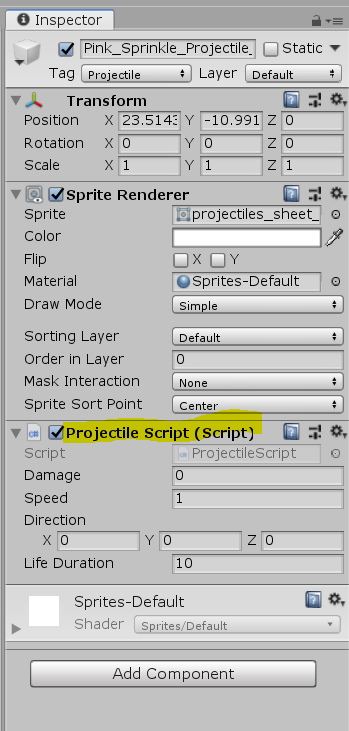
**공개SW프로젝트 8조 진행상황 #2**

[발사체 만들기]

여러가지 발사체로 판다를 공격 할 것이므로, 일반적인 발사체 스크립트를 작성하여 재사용함. 기본적인 발사체의 요구사항은 다음과 같다.

1. 발사체는 직선으로 움직인다.
2. 발사체는 판다에게 입힐 수 있는 데미지 정보를 가지고 있다.

기본 발사체 스크립트 ProjectileScript.cs 파일을 참고문헌 저자의 오픈소스 깃헙 페이지에서 다운받아 사용하였다. 앞으로 모든 기본 스크립트는 오픈소스 파일을 활용하기로 한다. 모든 소스파일은 **MIT 라이선스**인것을 확인하였다. 즉, 수정 및 배포, 상업적 판매또한 자유롭다. (링크: <https://github.com/PacktPublishing/Getting-Started-with-Unity-5.x-2D-Game-Development>)



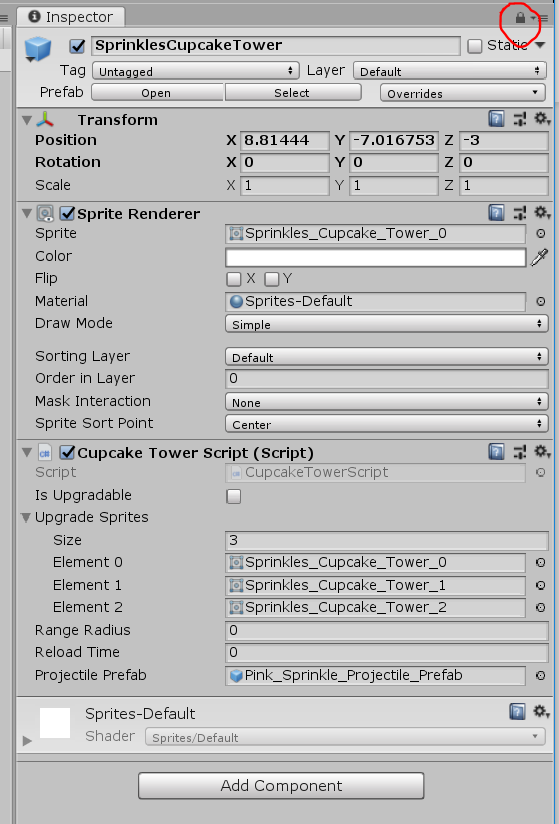
▲ 발사체 prefab에 ProjectileScript를 컴포넌트로 추가한 모습. public으로 선언한 변수들을 inspector에서 조작할 수 있다.

[컵케이크 타워 만들기]

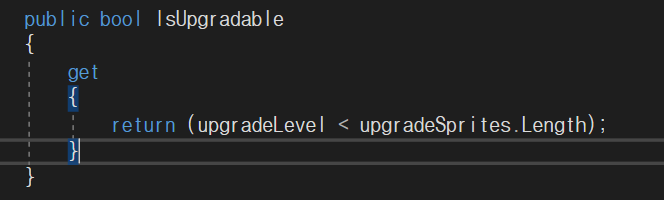
컵케이크 타워가 해야 할 일은 다음과 같다.

1. 일정 범위에 있는 판다를 감지하여 발사체를 발사한다.
2. 발사체의 종류와, 범위 내의 어떤 판다를 쏠 지 결정하는 알고리즘이 따로 있다.
3. 컵케이크 타워는 발사체를 재장전하기 위한 일정 시간이 필요하다.
4. 컵케이크 타워는 더 좋은 타워로 업그레이드 될 수 있으며, 스탯이 증가하고 모양이 바뀐다.

오픈소스 자료의 CupcakeTowerScript.cs소스를 활용한다.



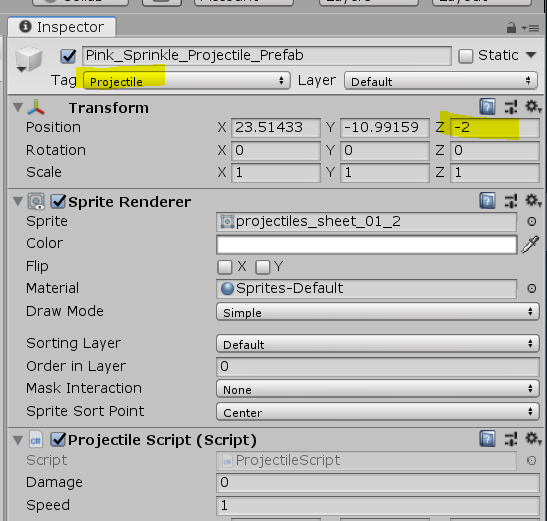
위와 같이 Sprinkles Cupcake Tower Prefab을 만들었다. inspector창에서 현재 선택 항목을 고정시키려면 우측 상단의 자물쇠 아이콘을 클릭하면 된다.



컵케이크 타워의 업그레이드 관련 변수인 isUpgradable을 좀 더 편리하게 다루기 위해, 클래스 내에서 IsUpgradable 프로퍼티를 정의하고 위의 이미지와 같이 getter 함수를 작성하였다.

[발사체 Prefab 생성]

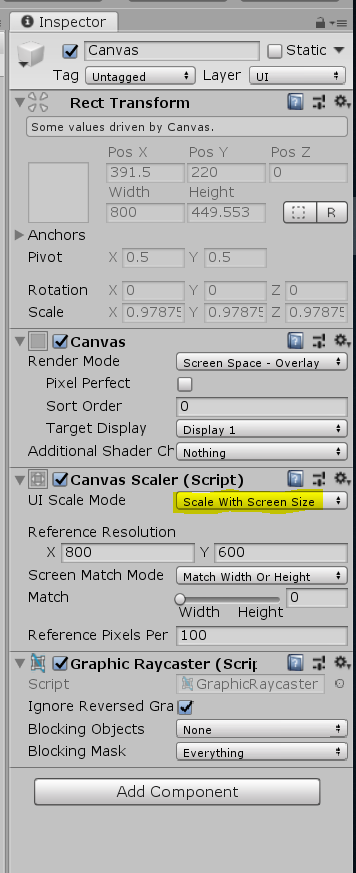
위와 같은 방법으로 다양한 종류의 발사체 prefab들을 생성한다.



위의 이미지와 같이, Tag는 미리 지정해 준 Projectile tag로 설정하고, z-buffering 표에서 정해두었던 z값인 -2를 모든 발사체 prefab에 설정해준다.

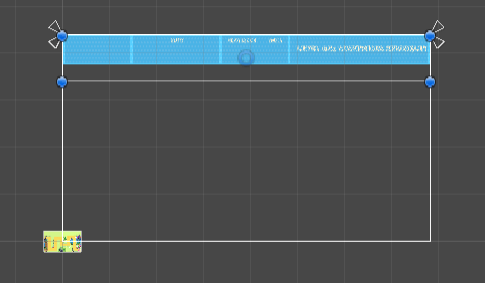
[기본 UI배치]

UI Canvas를 만들고, image 요소들을 추가하여 UI를 배치한다.



여기서 중요한 것은, 위의 이미지와 같게 Canvas의 Canvas Scaler의 **UI Scale Mode를 Scale With Screen Size로 설정**해 주어야 한다는 것이다. 참고자료에는 나와있지 않은 정보이므로 유의한다.

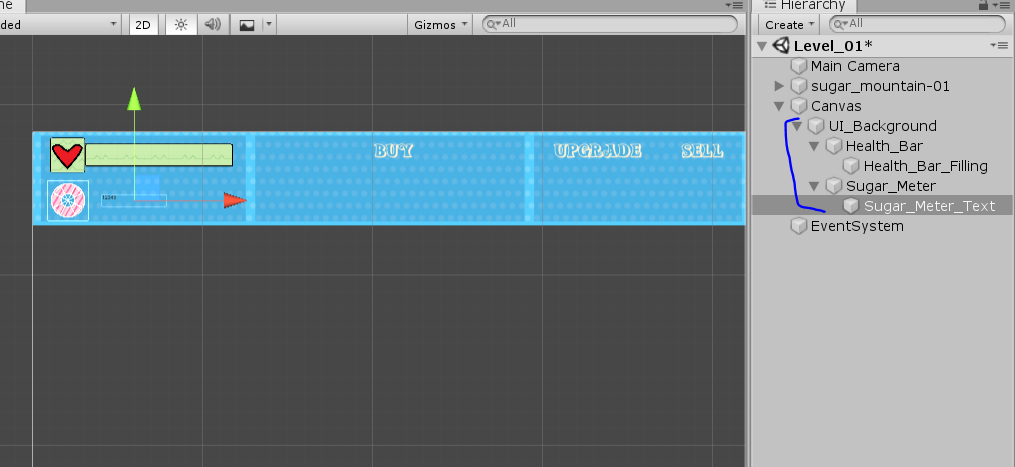
기본 메뉴바를 다음과 같이 Canvas에서 배치한다.



게임 실행 시 화면 사이즈에 관계 없이 UI가 스케일링 되는 것을 확인할 수 있다.



플레이어의 남은 HP를 표시하는 Health Bar와 슈거의 양을 표시하는 Sugar Meter UI들 또한 배치한다. Parent-Child 관계를 이용하여 하나의 UI요소가 종속된 UI요소를 이동 및 사이즈조절 시에 알맞게 따라가도록 한다.



위의 이미지에서 Hierarchy 패널을 살펴보면 UI요소들 간의 관계를 확인할 수 있다.