

# 업적시스템

남정웅

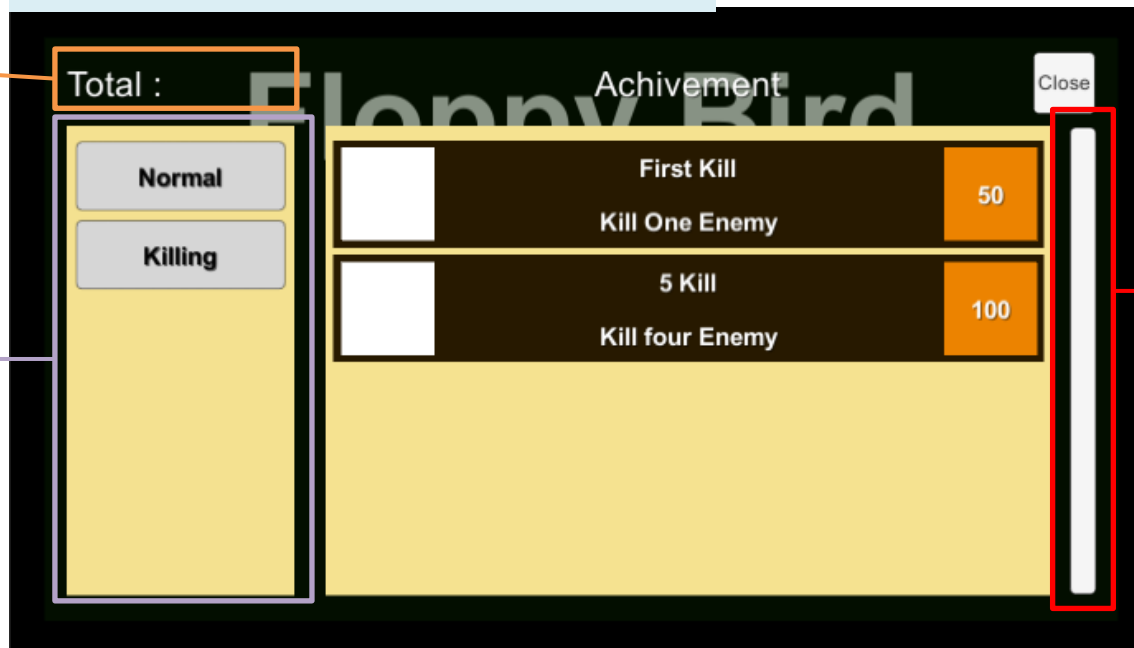
다운로드(GitHub)주소 : <https://github.com/hehza90/-Unity-AchievementSystem.git>

# 1. 개요

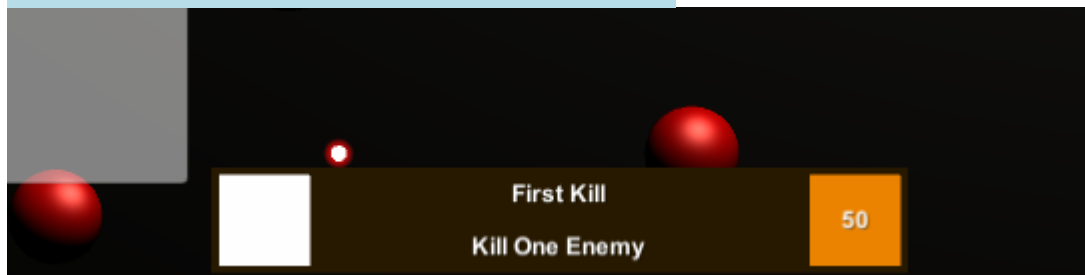
UI를 직접 편집 가능한 업적 창    인트로에서 열 수 있음

전체 업적 점수

카테고리 목록



게임 도중 업적 달성시 알림

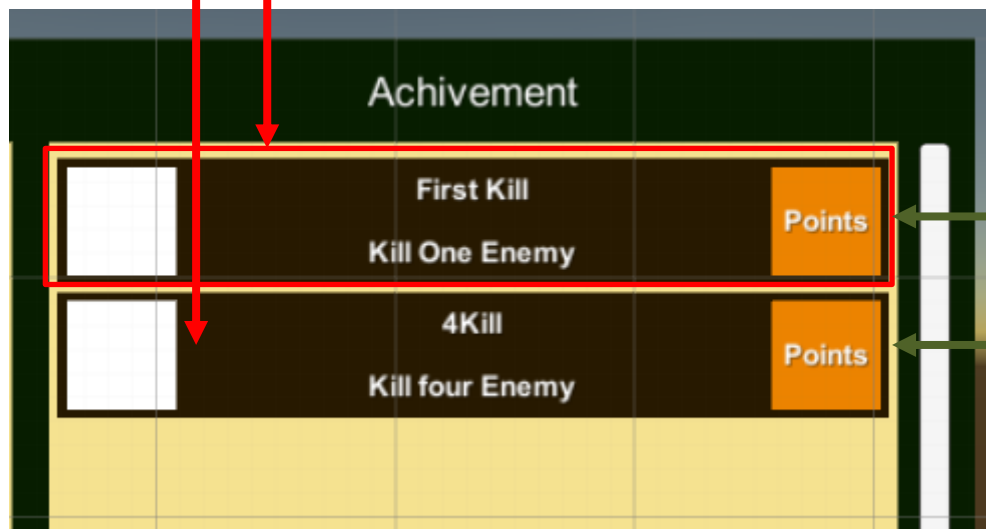


## 2. 구조

Ex) 지정된 수의 적을 처치했을 때 해금되는 업적

Unity5 에서 업적 Prefab 배치 후 UI 요소 편집

Prefab



컴포넌트로 추가 &  
변수에 수치 입력

코드로 작성

class Achievement

상속

업적종류에 따라  
다양한 자식 클래스  
를 만듦

지정된 업적들에게  
알림(옵저버 패턴)

class AchievementManager

업적을 해금시키는 함수  
업적 해금을 팝업 시키는 함수 등  
(Achievement에서 실행)을 포함

AchievementManager의 함수 실행

죽었을 때  
Mediator을 통해  
함수 실행

class Enemy

class Mediator

### 3. Enemy에서 Mediator

※ 코드는  
Asset/Script/AchievementSystem  
에서 확인 가능 합니다

Enemy의 Die()에서 Mediator의 함수 실행

#### Class Enemy

```
public override void Die()
{
    base.Die();
    //Mediator에서 필요한 함수 실행
    Mediator.Singleton.WhenEnemyDead(this.gameObject.GetComponent<Enemy>());

    //재사용 여부에 따라 Destroy Or Disable
    ObjectPoolingManager.Singleton.DestroyOrDisable(this.gameObject);
}
```

#### Class Mediator

```
//Enemy가 죽었을때 DeathEvent에 필요한 여러함수 실행
public void WhenEnemyDead(Enemy deadEnemy)
{
    //ScoreManger에서 Enemy에 지정된 스코어 만큼 스코어를 올리는 함수
    ScoreManger.Singleton.GetEnemyScore(deadEnemy);
    //AchievementManager의 적 처치횟수를 세는 함수
    AchievementManager.Singleton.CountTotalEnemyKillCount();
}
```

WhenEnemtDead(...)에서는 적이 죽었을 때  
실행 되어야 할 다른 class의 함수를 실행

## 4. AchievementManager와 Achievement

### class AchievementManager

```
public int TotalEnemyKill=0; //데이터 - 전체 적 처치 수
public event Action NotifyAnyEnemyKilled; //데이터가 갱신 될때 특정 Achievement에 알림

public void CountTotalEnemyKillCount() //데이터를 갱신하고 NoifyAnyEnemyKilled();
{
    TotalEnemyKill++;
    PlayerPrefs.SetInt("TotalEnemtKill", TotalEnemyKill); //데이터 갱신
    if (NotifyAnyEnemyKilled!= null)
    {
        NotifyAnyEnemyKilled(); //업적들에게 알림
    }
}
```

class Mediator을 통해 실행

### class Achievement

← 업적 해금, 진척도 갱신, 해금됨 알림 등  
모든 업적의 공통된 기능을 정의

↑ 상속

### Ex) class TotalKillEnemy

← 업적의 종류에 따른 다양한 class

```
protected override void Awake()
{
    base.Awake();
    if (unlocked == false) //언락이 안되어 있을때만
    {
        //원하는 데이터를 다루는 곳에 이벤트 등록
        AchievementManager.Singleton.NotifyAnyEnemyKilled += () => RenewProgress();
    }
}
```

진척도가 목표를 넘  
어 섰다면  
AchievementManag  
er의 업적 해금 함수  
실행

진척도 갱신 함수를 등록함으로써 데이터가 변할 때 Achievement에서 알 수 있음