

```

66
67
68
69  /* 결합이 많이 되면 리팩토링(코드 개선)시 왜 힘든지에 대한 분석
70
71  // 감사님의 말을 종합해보자면 지금 이 applySkillEffect 매소드에는
72  // 이미 3가지 기능을 하고 있기 때문에 위에 startGame 메소드 부분에서 코드를 찔 때 비슷한 형식(gmArr[id].applySkillEffect)으로 짜고싶으나
73  // for 부분에서 플레이어 수만큼 루프를 도는 기능이 중복이 되기 때문에 비슷한 형식으로 짜기 어려워졌다.
74  // applySkillEffect 매소드가 세가지 기능을 담고 있기 때문에
75  // 이것을 다시 리팩토링 하기에는 아마 크게 많은 기능이 없어서 당장은 힘들지 않다고 생각되나 그것 자체가 엔티티 오염의 시작이고
76  // 만약에 이 기능들을 세부적으로 나눴을때 새로이 추가된 기능이나 코드를 작성할 때 기존 엔티티 오염이 덜 하고
77  // 이와 같은 것처럼 더 코드를 깔끔한 형식으로 찔 수 있다는 측면에서 봤을때
78  // 결합이 많이 되면 이러한 것들을 방해하기 때문에 반대로 힘들어 진다고 말하는 것 같다.
79
80

```

```

81  //그럼 applySkillEffect 매소드에 for문을 없애고 i에 id를 넣고 이 매소드를 GameManager나 Player에 옮겨야 될까나?
82  // 그럼 id는 매소드 안에서 0으로 초기화 하는게 맞나?
83  public void applySkillEffect () {
84      int tmp;

```