

# Design Pattern

## Final Report

담당교수 : 박용범

학과 : 소프트웨어학과

학번 : 32204077

이름 : 정성원

## # 기본 가정

챔피언은 일종의 캐릭터를 의미하며 한 게임당 플레이 할 수 있는 챔피언의 수는 하나이다. 현재 리그오브레전드 게임은 150개의 챔피언이 존재한다. 그 중 사용이 가능한 챔피언의 개수를 3개로 구현할 것이며, 그 종류는 각각 케일(Kayle), 사일러스(Sylas), 탈론(Talon)이다.



## # 챔피언과 스킬



챔피언은 모두 레벨이 오를 때마다 증가하는 HP,마나, 공격력, 주문력 수치들을 가지면서 (맨 왼쪽에서 확인 가능) 모든 챔피언들은 동일하게 패시브와 스킬(Q,W,E,R)을 가지고 있음.

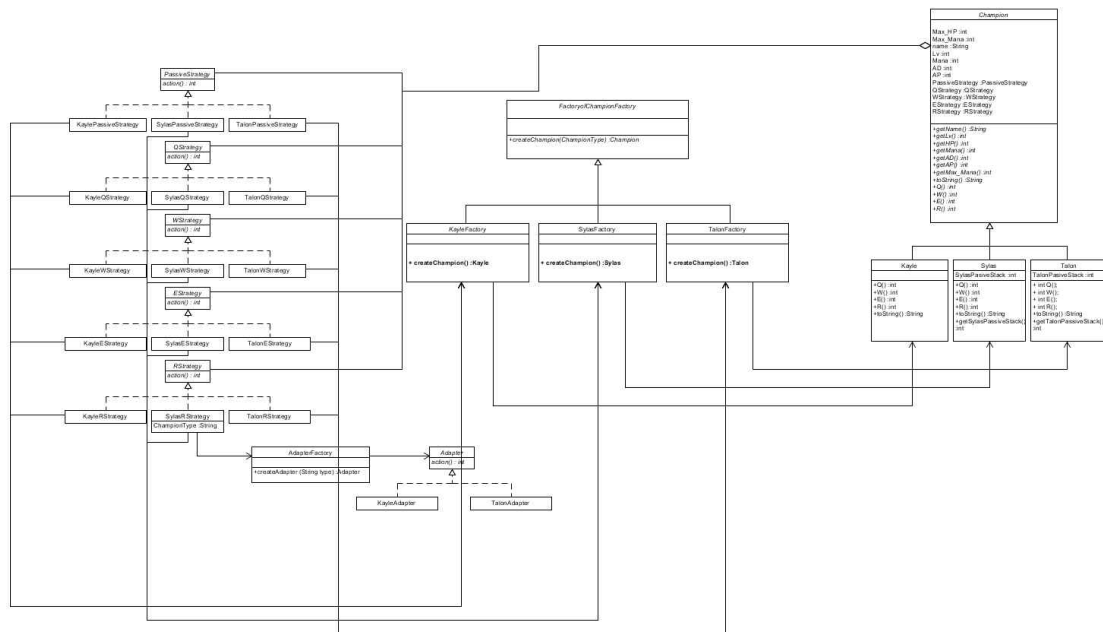
이 스킬들의 데미지는 챔피언이 가진 공격력(AD), 주문력(AP) 수치들에 가중치를 입혀 계산한다

일반적으로 스킬을 사용하면 마나를 소모하며, 적 챔피언에게 데미지를 줄 수도, 자신의 HP를 회복하는 등 다양한 기능이 발휘된다.

## # 사용 패턴

1. *Factory Method Pattern* (챔피언(캐릭터) 생성)
2. *Strategy Pattern* (일반적인 챔피언(캐릭터)들의 스킬들 구현)
3. *Adapter Patern* (구현하려는 챔피언(캐릭터)중 하나인 Sylas의, 특수한 스킬 구현을 위한 패턴)

## # Class Diagram



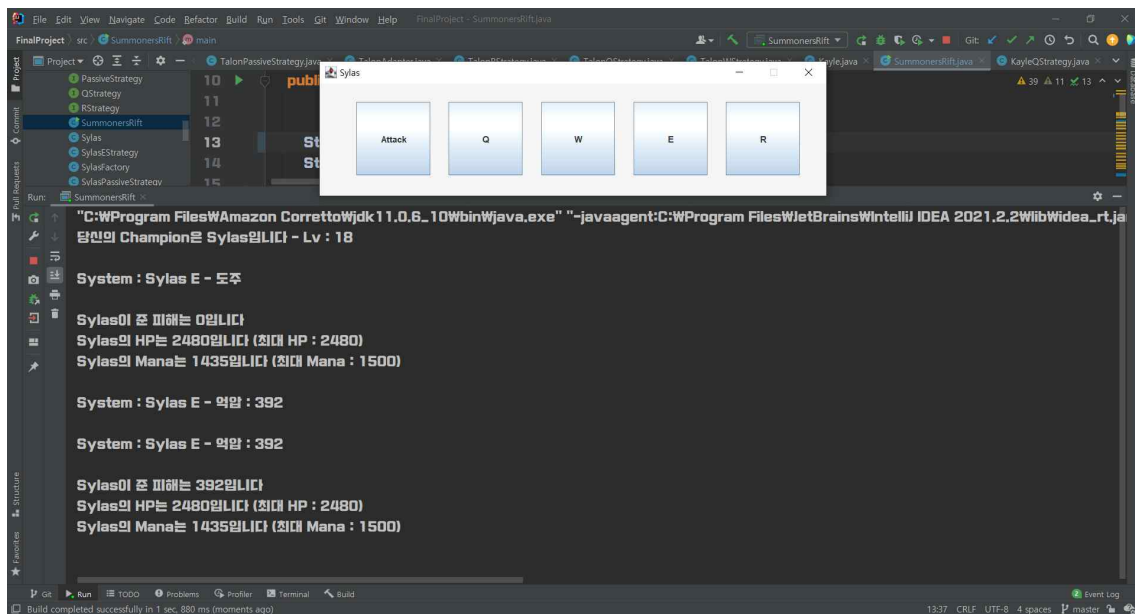
좌측부터 **Strategy Pattern**, **Adapter Pattern**, **Factory Method Pattern**  
가시성을 위하여 추가적으로 Class Diagram PDF를 첨부.

총 클래스 개수 : 34개

총 코드 길이 : 1150줄

## # 프로그램의 동작

ChampionType을 String으로 받아서, 해당 Champion의 스킬 인터페이스를 GUI로 구현한다. 이 때 GUI에 focus를 넣은 후 키보드 키 A,Q,W,E,R를 누르면 해당 기본 공격이나 스킬의 데미지, 그리고 상태가 콘솔 창에 출력되어 나온다.



사일러스의 E 스킬을 최초 시전 했을 때(Sylas E - 도주). 원하는 방향으로 돌진 후, E를 재시전 했을 때 (Sylas E - 억압) 사슬을 던져 적으로 이동하여 데미지를 주는 스킬을 GUI의 KeyListener를 이용하여 추상화하여 구현한 모습 .

## 1. Strategy Pattern

QStrategy.java

```
public interface QStrategy {  
    int action(Champion champion);  
}
```

Q, W, E, RStrategy의 action 메소드는 스킬 데미지를 리턴하는 메소드이며, 스킬 데미지는 Champion이 가진 고유 공격력이나 주문력 수치에 가중치를 곱하여 계산된다. 챔피언들의 마나, 공격력, 주문력 등 여러 수치를 가져오기 위해 매개변수를 Champion 객체로 설정한다.

Champion.java

```
public abstract class Champion {  
  
    PassiveStrategy PassiveStrategy;  
    QStrategy QStrategy;  
    WStrategy WStrategy;  
    EStrategy EStrategy;  
    RStrategy RStrategy;  
  
    public Champion(PassiveStrategy PassiveStrategy, QStrategy QStrategy, WStrategy WStrategy,  
        EStrategy EStrategy, RStrategy RStrategy)  
    {  
        this.PassiveStrategy = PassiveStrategy;  
        this.QStrategy = QStrategy;  
        this.WStrategy = WStrategy;  
        this.EStrategy = EStrategy;  
        this.RStrategy = RStrategy;  
    }  
}
```

모든 Champion들은 Q, W, E, R 스킬을 가진다. Champion들의 같은 Q,W,E,R 스킬을(같은 기능)을 각기 Champion마다 다른 전략으로 구현해야한다. 이 말은 Concrete한 Champion객체들은 Concrete한 Strategy 객체들의 집합으로 구성되어있다는 말로 이해할 수 있다. 예를 들자면, Champion을 상속받은 Talon 객체는 Talon만의 Passive, Q, W, E, R Strategy들을 가진다는 의미.

이를 위해 Champion들의 구성요소인 Strategy 객체들을, 슈퍼클래스인 Champion의 생성자로 초기화시켜주는 생성자를 정의하고, Concrete한 Champion 객체들에서 super 메소드를 이용하여 처리하게 하였다.

어떻게 보면 setStrategy Method를 따로 구현하지 않고 한꺼번에 생성자에 구현한 것인데, 이 이유는 뒤에 있을 Factory Method Pattern의 구현을 더 쉽게 하기 위함임.

```
public abstract int Passive();  
  
public abstract int Q();  
  
public abstract int W();  
  
public abstract int E();  
  
public abstract int R();
```

또한 각기 다른 Strategy들의 action 메소드를 실행시킬 Passive, Q, W, E, R 메소드를 abstract로 선언하고, Concrete한 Champion 객체들로 하여금 오버라이딩하여 사용할 수 있게 하였다.

## Talon.java

예컨데, 이제 Champion을 상속받는 Talon이라는 Concrete한 Champion 객체를 살펴보자.

```
public Talon(PassiveStrategy PassiveStrategy, QStrategy QStrategy, WStrategy WStrategy,
             EStrategy EStrategy, RStrategy RStrategy) {
    super(PassiveStrategy, QStrategy, WStrategy, EStrategy, RStrategy);

    Max_HP = 2203;
    Max_Mana = 1006;
    name = "Talon";
    Lv = 18;
    HP = 2203;
    Mana = 1006;
    AD = 140;
    AP = 0;
}
```

super메소드를 이용해 Strategy들을 Concrete한 Strategy(Talon의 Passive,Q,W,E,R) Strategy 들로 초기화해주고, 상속받은 Champion의 능력치 변수들을 초기화시켜준다

abstract로 정의된 Champion의 Passive, Q, W, E, R 메소드들을 오버라이딩한다.

```
@Override
public int Q() {
    if (this.getMana() < 40) {
        System.out.println("System : 마나가 부족합니다");
        return 0;
    } else {
        int damage = QStrategy.action(champion: this);
        this.Mana -= 40;
        TalonPassiveStack++;
        return damage;
    }
}
```

챔피언의 현재 마나를 읽어와서, 현재 마나량이 마나소모량인 40보다 큰 값이라면 QStrategy의 action 메소드를 수행하면, 생성자에서 초기화된 Concrete한 Strategy 객체의 action 메소드가 수행된다. 현재의 사진으로는 TalonQStrategy에 있는 action메소드가 실행된다.

TalonQStrategy.java

```
public class TalonQStrategy implements QStrategy{
    int damage = 0;

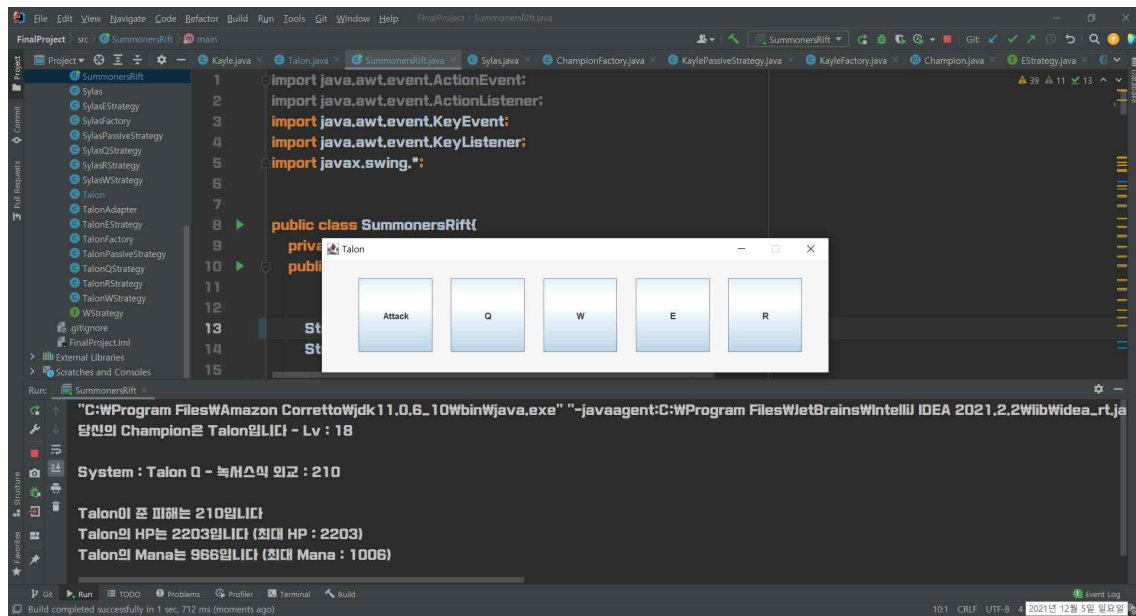
    @Override
    public int action(Champion Talon) {

        damage = (int) (Talon.getAD() * 1.5);
        System.out.println("System : Talon Q - 녹서스릭 외교 : "+damage);
        System.out.println();
        return damage;
    }

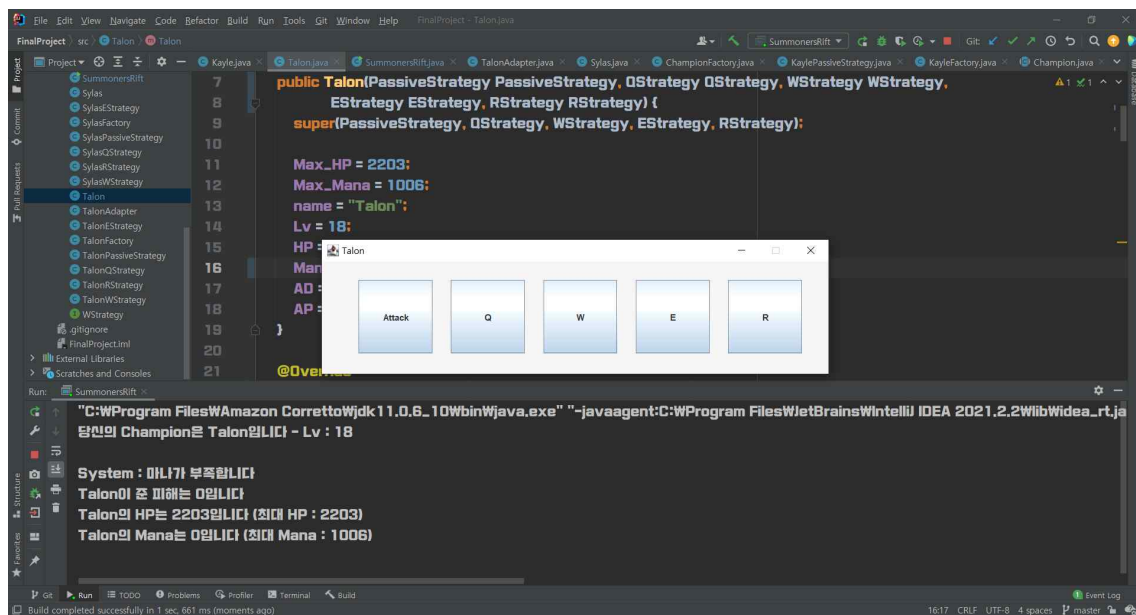
    public String toString(){
        return "탈론이 대상에게 도착해 물리 피해를 입힙니다." ;
    }
}
```

Talon의 Q스킬, Q스킬의 데미지는 Talon의 공격력 수치의 150%의 데미지를 준다.





실행화면 1 : 키보드 자판 'Q'를 눌러 40의 마나를 사용하여 210의 데미지를 주었다.



실행화면 2 : 키보드 자판 'Q'를 눌렀지만, Talon의 마나가 부족하여 스킬 시전이 되지 않는다.

## 2. Factory Method Pattern

Concrete한 Champion들을 만드는 Factory(KayleFactory, SylasFactory, TalonFactory)를 만드는 Factory인 FactoryofChampionFactory를 만든다. FactoryofChampionFactory 클래스는 abstract로 정의된 createChampion 메소드를 가지는 abstract class이다

```
public abstract class FactoryofChampionFactory {  
    public abstract Champion createChampion();  
}
```

KayleFactory.java

Concrete한 Champion들을 만드는 Factory는 각각 KayleFactory, SylasFactory, TalonFactory이며 abstract로 정의되었던 createChampion 메소드를 오버라이딩하여 각각의 Champion(Kayle, Sylas, Talon) Factory들에 맞게 구현하고, 이 메소드를 이용하여 생성자의 인자에 Concrete한 Strategy를 등록시킨 Kayle, Sylas, Talon 객체를 각각 반환하는 역할을 한다.

```
public class KayleFactory extends FactoryofChampionFactory{  
    public KaylePassiveStrategy createPassive() { return new KaylePassiveStrategy(); }  
    public KayleQStrategy createQ() { return new KayleQStrategy(); }  
    public KayleWStrategy createW() { return new KayleWStrategy(); }  
    public KayleEStrategy createE() { return new KayleEStrategy(); }  
    public KayleRStrategy createR() { return new KayleRStrategy(); }  
    public Kayle createChampion(){  
        return new Kayle(this.createPassive(),this.createQ(), this.createW(),this.createE(),this.createR());  
    }  
}
```

## ChampionFactory.java

```
public Champion createChampion(String type) {
    Champion tmp = null;
    switch (type) {

        case "Kayle":

            FactoryofChampionFactory KayleFactory = new KayleFactory();
            Champion Kayle = KayleFactory.createChampion();
            tmp = Kayle;
            break;

        case "Sylas":

            FactoryofChampionFactory SylasFactory = new SylasFactory(OpponentChampion);
            Champion Sylas = SylasFactory.createChampion();
            tmp = Sylas;
            break;

        case "Talon":

            FactoryofChampionFactory TalonFactory = new TalonFactory();
            Champion Talon = TalonFactory.createChampion();
            tmp = Talon;
    }
}
```

ChampionFactory class에서는 Type에 따라 Kayle, Sylas, Talon을 만드는 FactoryofChampionFactory를 상속받은 KayleFactory, SylasFactory, TalonFactory 객체를 생성한다. Concrete한 Factory들의 createChampion메소드의 리턴값인 Kayle, Sylas, Talon 객체중 Type에 일치하는 Champion 객체를 Main으로 리턴시키는 역할을 하는 것이 ChampionFactory.java 파일의 역할이다.

### # 실행화면



Example - Sylas 생성 : toString 메소드를 Override하여, 생성된 Champion 객체를 출력시켰을 때의 모습이다.

### 3. Adapter Pattern

지금까지 3개의 챔피언들을 구현하였는데, 그중 사일러스(Sylas)라는 챔피언의 R스킬(궁극기)은 다른 챔피언들의 R스킬(궁극기)를 가져와서 사용하는 스킬이다.

프로그램의 동작을 설명할 때 사용했던 사일러스의 E스킬의 예처럼, (최초시전 - 도주, 재시전 - 억압) 사일러스의 R스킬도 최초 시전했을 때, 재 시전했을 때로 나눌 수 있다.



그림 1 : 최초 시전을 할 경우 원하는 챔피언의 R스킬을 가져와 사용가능한 상태로 설정한다.

그림 2 : R스킬을 재시전할 경우 가져온 R스킬을 사용한다

SylasRStrategy.java

```
public int action(Champion sylas) {  
  
    if (status == 0) {  
  
        System.out.println("System : Sylas R - 강탈");  
        ChampionFactory factoryofChampionFactory = new ChampionFactory(ChampionType);  
        Champion champion = factoryofChampionFactory.createChampion(ChampionType);  
  
        System.out.println("System : "+ champion.getName()+"의 궁극기를 강탈하였습니다");  
        status++;  
        return 0;  
  
    }else if (status == 1) {  
        AdapterFactory AdapterFactory = new AdapterFactory(sylas, ChampionType);  
        Adapter Adapter = AdapterFactory.createAdapter(ChampionType);  
  
        damage += Adapter.action();  
        status = 0;  
  
    }  
  
    return damage;  
}
```

1) 최초시전을 할 때 (status ==0)일 때

RStrategy에서는 ChampionFactory를 다시 불러와서 상대 ChampionType을 String객체로 받아와 Dummy로 쓰일 Champion객체를 만든다. Strategy가 아닌 Champion 객체를 만드는 이유는, 사일러스의 스킬 데미지를 조정하기 위한 Dummy Champion의 수치들이 필요하기 때문이다.

2) 재시전을 할 때 (status == 1) 일 때

Dummy로 만들어진 Champion을 이용하여

1. 이 Dummy Champion의 R스킬을 시전하고,
2. 가져온 R스킬의 데미지를 Sylas의 계수로 다시 계산해주는 역할이

바로 Adapter의 역할이다.

이를 위해 각각의 Concrete한 Champion 형으로의 다운캐스팅 과정이 수반된다. Concrete

한 Adapter들은 abstract class인 Adapter를 상속받으며, 이 Concrete Adapter들의 생성은 Adapter Factory를 만들어 사용하였다. SylasRStrategy.java에서는 AdapterFactory의 createAdapter 메소드를 통해, 상대의 ChampionType에 따라 KayleAdapter와 TalonAdapter중 상대의 ChampionType과 일치하는 Adapter를 생성해 반환하고, Adapter의 action 메소드를 수행한다.

#### AdapterFactory.java

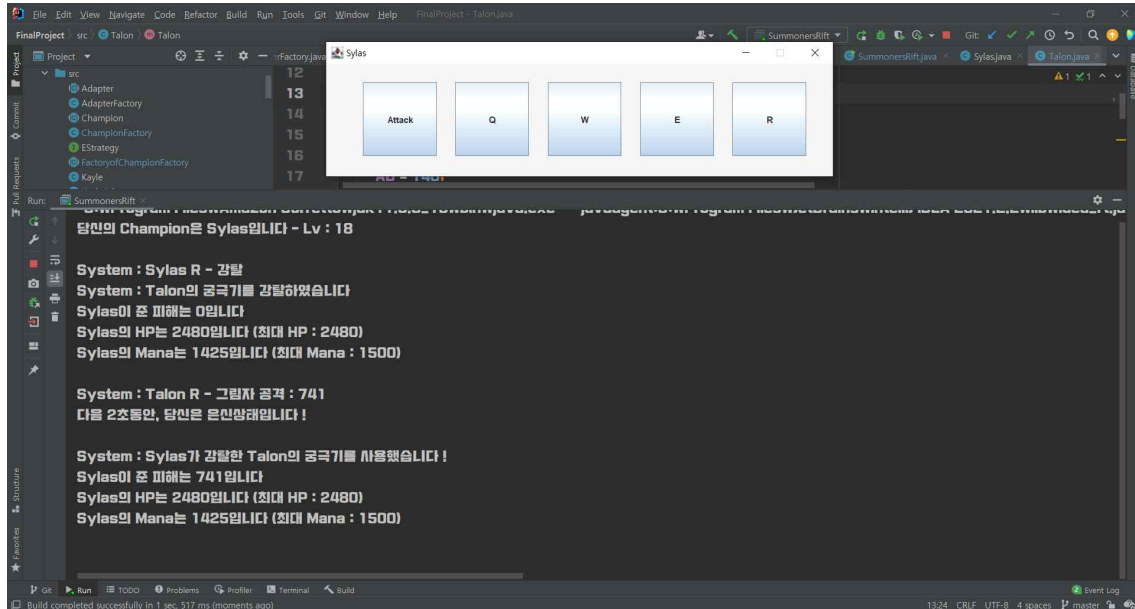
```
public Adapter createAdapter (String type) {
    Adapter tmp = null;
    switch (type) {
        case "Kayle":
            KayleAdapter KayleAdapter = new KayleAdapter(Sylas, factoryOfChampionFactory.createChampion( type, "Kayle" ));
            tmp = KayleAdapter;
            break;
        case "Talon":
            TalonAdapter TalonAdapter = new TalonAdapter(Sylas, factoryOfChampionFactory.createChampion( type, "Talon" ));
            tmp = TalonAdapter;
            break;
    }
    return tmp;
}
```

#### TalonAdapter.java

```
@Override
public int action() {
    Talon Talon = (Talon) champion; // 다운캐스팅
    Talon.AD = (int)(0.6 * Sylas.getAD()); // 계수 조정
    damage += Talon.R();
    Talon.Mana = Talon.getMax_Mana();
    System.out.println("System : Sylas가 강탈한 Talon의 궁극기를 사용했습니다!");
    return damage;
}
```

Dummy로 만들어진 Talon의 Mana가 떨어지면, 스킬 시전을 할 수 없으므로 R()가 실행된 이후 DummyTalon의 마나를 최대 마나로 초기화시켜준다.

## # 실행화면



최초 시전 : Sylas가 Talon의 R스킬을 가져왔기 때문에, 준 피해는 0이 누적되었으며, 최초 시전시 소비되는 마나 75가 차감된 모습이다.

재시전 : 가져온 Talon의 R스킬을 사용한다. TalonAdapter의 스킬 데미지 재계산을 통해 741이라는 데미지를 주었으며, 마나는 최초시전시 차감하였기 때문에 이전의 양 그대로 유지된다.




부록 1 : 챔피언 및 스킬 소개 ( 추가자료 : 동영상 첨부 )

Kayle




Passive : 거룩한 승천




챔피언 및 스킬 레벨이 오를수록 케일의 공격이 강화됩니다.

- 1레벨: 케일이 기본 공격 시 5초 동안 공격 속도가 6 +0.01 주문력)% 상승하며<sup>[1]</sup> 이 효과는 최대 5회 중첩됩니다. 최대 중첩 시 회열 상태가 되며 이동 속도가 8% 상승합니다.
- 6레벨: 공격 사거리가 525까지 증가합니다.
- 11레벨: 회열 상태에서 공격 시 화염파를 발사해 마법 피해를 입힙니다.
- 16레벨: 케일이 영구히 회열 상태를 유지하며 공격 사거리가 625까지 늘어납니다.

화염파는 치명타로 적중할 수 있으며 마법 아이템 효과가 적용됩니다. 피해량은 화염주문검 스킬 레벨에 따라 증가합니다.

 525 / 625<sup>[1]</sup>

 15 / 20 / 25 / 30 / 35 (+0.1 추가 공격력) (+0.25 주문력) - 화염파



Q : 광휘의 일격



케일이 처음으로 적을 맞으면 멈추는 천상의 검을 발사합니다. 검은 대상과 그 뒤에 있는 적들에게 **마법 피해**를 입히고 2초 동안 **둔화**시키며 4초 동안 **15%의 방어력**과 **마법 저항력**을 감소시킵니다.<sup>[13]</sup>

 70 / 75 / 80 / 85 / 90

 900

 12 / 11 / 10 / 9 / 8

 60 / 100 / 140 / 180 / 220 **(+0.6 추가 공격력)** **(+0.5 주문력)**


 26 / 32 / 38 / 44 / 50%

시전 시간: 0.25

투사체 속도: 1600

폭발 피해 범위: 195


W : 천상의 축복




케일이 자신과 대상 아군 챔피언에게 빛을 불어넣어 **체력**을 회복하고 2초 동안 **이동 속도**를 상승시킵니다.

---


대상을 지정하지 않고 이 스킬을 사용하면 범위 내에서 자신의 총 체력 대비 가장 피해를 많이 입은 아군을 치유합니다.




90 / 100 / 110 / 120 / 130




900



15

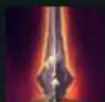







60 / 90 / 120 / 150 / 180 (+0.3 주문력)




+ 24 / 28 / 32 / 36 / 40 (+ 0.08 주문력)%

E : 화염주문검

<b>기본 지속 효과:</b> 케일의 기본 공격이 <b>마법 피해</b> 를 추가로 입힙니다.	
	<b>사용 시:</b> 케일의 다음 기본 공격 사거리가 증가하며 대상이 잃은 체력에 해당하는 <b>마법 피해</b> 를 추가로 입힙니다.  이 스킬에는 <b>적중 시 효과</b> 및 <b>마법 아이템 효과</b> 가 적용되며 <b>정글 몬스터</b> 에게는 최대 400의 추가 피해를 입힙니다.
	이 공격은 케일이 11레벨에 도달하면 대상에게 <b>적중 시 폭발</b> 하며 주변 적에게 피해를 입힙니다.
 15 / 20 / 25 / 30 / 35 <b>(+0.1 추가 공격력)</b> <b>(+0.2 주문력)</b>	
 525 / 625	 8 / 7.5 / 7 / 6.5 / 6
 대상이 잃은 체력의 8 / 9 / 10 / 11 / 12 <b>(+0.02 주문력)%</b>	





R : 신성한 심판



케일이 아군 챔피언 한 명을 일정 시간 동안 무적 상태로 만듭니다. 그런 다음 대상 주위 지역을 정확히 주변 적들에게 **마법 피해**를 입힙니다.

---


케일은 이 스킬을 사용하는 동안 기본 공격을 가하거나 스킬을 사용할 수 없지만 이동할 수 있습니다.

 100 / 50 / 0	 900	 160 / 120 / 80
무적 지속 시간: 2 / 2.5 / 3		
 200 / 350 / 500 (+1.0 추가 공격력) (+0.8 주문력)		
피해 범위: 500		

Sylas




Passive : 패트리사이트 폭발




스킬 사용 후 사일러스가 패트리사이트 폭발을 1회 충전합니다. (최대 충전량 3회) 다음 기본 공격 시 충전량을 하나 소모하여 사슬을 휘두르며 적중당한 대상에게 **마법 피해**를 입히고 대상 주변의 모든 적에게 **감소한 마법 피해**를 입힙니다.


충전이 하나라도 되어 있으면 사일러스의 공격 속도가 125% 증가합니다.



(+1.3 총 공격력) (+0.25 주문력) - 사일러스의 대상에 대한 피해량

 (+0.4 총 공격력) (+0.2 주문력) - 사일러스의 대상이 아닌 주변 적에 대한 피해량

Q : 사슬 후려치기



사일러스가 사슬을 후려쳐 **마법 피해**를 입히고 1.5초 동안 **둔화**시킵니다. 사슬이 교차하는 지점은 폭발해 **마법 피해**를 추가로 입힙니다.

폭발 범위 안에 있는 미니언과 정글 몬스터는 40%의 피해를 입습니다.

 55	 775	 10 / 9 / 8 / 7 / 6
 40 / 60 / 80 / 100 / 120 (+0.4 주문력) - 후려치기 피해량		
 70 / 125 / 180 / 230 / 290 (+0.9 주문력) - 폭발 피해량		
 15 / 20 / 25 / 30 / 35%		

W : 국왕시해자



사일러스가 마법의 힘으로 적에게 돌진해 **마법 피해**를 입힙니다. 챔피언에게 사용하면 사일러스가 **잃은 체력에 비례해 체력**을 회복합니다.  
(체력이 40%일 때 최대 회복량 적용)

 60 / 70 / 80 / 90 / 100











 400

 13 / 11.25 / 9.5 / 7.75 / 6


 70 / 105 / 140 / 175 / 210 (+0.9 주문력)

 25 / 50 / 75 / 100 / 125 (+0.4 주문력)  
~ 50 / 100 / 150 / 200 / 250 (+0.8 주문력)

E : 도주 / 억압

	사일러스가 지정 위치로 돌진한 후 3.5초 동안 <b>재사용</b> 을 준비합니다.	
	재사용 시: 사일러스가 사슬을 던져 적에게 격중하면 사슬을 끌어당겨 적 방향으로 이동하며 <b>마법 피해</b> 를 입히고 0.35초 동안  공중으로 띄워 올립니다.	
 65	 400	 14 / 13 / 12 / 11 / 10
 -	 800	 3.5
 80 / 130 / 180 / 230 / 280 (+1.0 주문력)		

R : 강탈





사일러스가 적 챔피언의 궁극기를 강탈해 적과 똑같이 사용합니다. 효과는 사일러스의 궁극기 레벨과 능력치에 비례합니다.


적의 궁극기를 강탈하면 해당 궁극기 재사용 대기시간의 200%(사일러스의 재사용 대기시간 감소 적용)만큼 재사용 대기시간이 적용되어 사일러스가 그동안 해당 적의 궁극기를 다시 강탈할 수 없습니다.(최소 40초)

---

사일러스가 순수한 공격력 계수를 주문력 계수로 전환해 추가 공격력 1당 주문력 0.4, 총 공격력당 주문력 0.6씩을 얻습니다.

 75

 950

 100 / 70 / 40

Talon



Passive : 검의 최후




탈론이 챔피언이나 대형 정글 몬스터에게 스킬을 사용해 피해를 입히면 6초 동안 중첩을 적용합니다.(최대 3회 중첩)

3회 중첩된 대상에게 탈론이 기본 공격을 가하면 2초에 걸쳐 물리 피해를 추가로 입힙니다.

정글 몬스터에게는 지속 피해로 120%의 피해를 입힙니다.

75 ~ 255<sup>[6]</sup> (+2.0 추가 공격력)

Q : 녹서스식 외교




탈론이 대상에게 도약해 **물리 피해**를 입힙니다. 근접 공격이 가능한 거리에서 사용하면 대신 치명타가 적용되어 **150%의 물리 피해**를 입힙니다.

이 스킬로 대상을 처치하면 **체력**을 회복하고 재사용 대기시간의 50%를 돌려받습니다.

*근접 공격 피해량은 치명타 피해량 증가에 영향을 받습니다.*

 40	 575 / 175 (치명타 판정)	 8 / 7.5 / 7 / 6.5 / 6
 65 / 85 / 105 / 125 / 145 (+1.0 추가 공격력)		
 97.5 / 127.5 / 157.5 / 187.5 / 217.5 (+1.5 추가 공격력)		
 10 ~ 70 		

W : 갈퀴손




탈론이 부메랑 단검을 여러 개 던져 **물리 피해**를 입힙니다. 이후 단검이 돌아오며 **물리 피해**를 입히고 1초 동안 **둔화**시킵니다.




*이 스킬은 정글 몬스터에게 150%의 피해를 입힙니다.*

 55 / 60 / 65 / 70 / 75	 900	 9 / 8.5 / 8 / 7.5 / 7
 투척 시 피해량: 40 / 50 / 60 / 70 / 80 (+0.4 추가 공격력)		
 회수 시 피해량: 50 / 80 / 110 / 140 / 170 (+0.8 추가 공격력)		
 40 / 45 / 50 / 55 / 60%		

E : 그림자의 길




탈론이 가장 가까운 지형이나 구조물 위로 도약해 뛰어넘습니다. 한 번 넘어간 지형은 일정 시간 동안 다시 넘을 수 없습니다.

 800	 2
 160 / 135 / 110 / 85 / 60 - 벽 넘기 불가 지속시간	



R : 그림자 공격









탈론이 사방에 검을 던져 **물리 피해**를 입히고, 이동 속도가 상승하며 최대 2.5초 동안 투명 상태가 됩니다. 투명 상태가 끝나면 검이 탈론에게 돌아오며 다시 **물리 피해**를 입힙니다.

탈론이 기본 공격이나 **녹서스식 외교** 스킬로 투명 상태를 해제하면 검이 탈론 대신 탈론의 대상에게 날아갑니다.

---

투명 상태의 유닛은 포탑이나 **절대 시야**로만 모습이 드러납니다.

 100	 550	 100 / 80 / 60
 90 / 135 / 180 (+1.0 추가 공격력) (타격당 피해량)		
 180 / 270 / 360 (+2.0 추가 공격력) (총 피해량)		
 + 40 / 55 / 70%		

부록 2 : 모든 기본공격, 스킬 실행화면

Kayle

Passive 구현

Kayle의 패시브는 공격사거리와 속도가 늘어나는 것이기 때문에 따로 구현하지 않았음

Q스킬 구현



70의 마나를 소비하여 321의 데미지를 주는 스킬이다.



W스킬 구현

**당신의 Champion은 Kayle입니다 - Lv : 18**

**System : Kayle W - 천상의 축복 : 207**

**현재 최대 체력으로 회복되었습니다**

**Kayle의 HP는 2045입니다 (최대 HP : 2045)**

**Kayle의 Mana는 1090입니다 (최대 Mana : 1180)**

90의 마나를 소비하여 207만큼 체력을 회복하는 스킬이다. **현재 체력 + 회복량이  $\geq$  최대 체력**이라면, 최대 체력으로 회복시킨다.

**당신의 Champion은 Kayle입니다 - Lv : 18**

**System : Kayle W - 천상의 축복 : 207**

**당신의 체력은 207만큼 회복되었습니다**

**Kayle의 HP는 1707입니다 (최대 HP : 2045)**

**Kayle의 Mana는 1090입니다 (최대 Mana : 1180)**

Kayle의 체력을 1500으로 설정한다면 **현재 체력 + 회복량이  $<$  최대체력**이므로, 207만큼의 체력을 회복한 과정이 HP에 반영된다.

E스킬 구현

**당신의 Champion은 Kayle입니다 - Lv : 18**

**System : Kayle E - 화염주문검 : 223**

**Kayle이 준 피해는 223입니다**

**Kayle의 HP는 2045입니다 (최대 HP : 2045)**

**Kayle의 Mana는 1180입니다 (최대 Mana : 1180)**

마나를 소비하지 않고 223의 데미지를 주는 스킬이다.

R스킬 구현

**System : Kayle R - 진정한 심판 : 465**  
**다음 2초동안, 당신은 무적입니다 !**

**Kayle이 준 피해는 465입니다**  
**Kayle의 HP는 2045입니다 (최대 HP : 2045)**  
**Kayle의 Mana는 1080입니다 (최대 Mana : 1180)**

100의 마나를 소모하여 다음 2초동안 무적상태에 들어간 후, 무적상태가 해제하면 상대에게 465의 피해를 주는 스킬이다.

Sylas

Passive 구현



Q,W,E,R 스킬을 쓰지않고 기본공격을 할 시 패트리사이트 폭발 Stack이 충전되지않아(최대 3회) 112의 피해를 주는 반면, Q스킬을 쓰고 난 후 기본공격을 할 시에는 Sylas의 패시브인 패트리사이트 폭발 효과가 적용되어 기본공격의 데미지가 173으로 바뀌게 된다.

Q스킬 구현

**당신의 Champion은 Sylas입니다 - Lv : 18**

**System : Sylas Q - 사슬 후려치기 : 425**

**Sylas이 준 피해는 425입니다**

**Sylas의 HP는 2480입니다 (최대 HP : 2480)**

**Sylas의 Mana는 1445입니다 (최대 Mana : 1500)**

55의 마나를 소비하여 425의 데미지를 주는 스킬이다.

W스킬 구현

**System : Sylas W - 국왕시해자 : 310**

**Sylas W - 국왕시해자 : 현재 최대 체력으로 회복되었습니다**

**Sylas이 준 피해는 310입니다**

**Sylas의 HP는 2480입니다 (최대 HP : 2480)**

**Sylas의 Mana는 1400입니다 (최대 Mana : 1500)**

최대 체력에서 W를 시전하면, 적에게 데미지를 주는 동시에 자신은 회복을 하는데, 사일러스의 **현재 체력 + 회복량**이  $\geq$  최대체력이라면, 최대 체력으로 회복시킨다.

**System : Sylas W - 국왕시해자 : 310**

**Sylas W - 국왕시해자 : 당신의 체력은 169만큼 회복되었습니다**

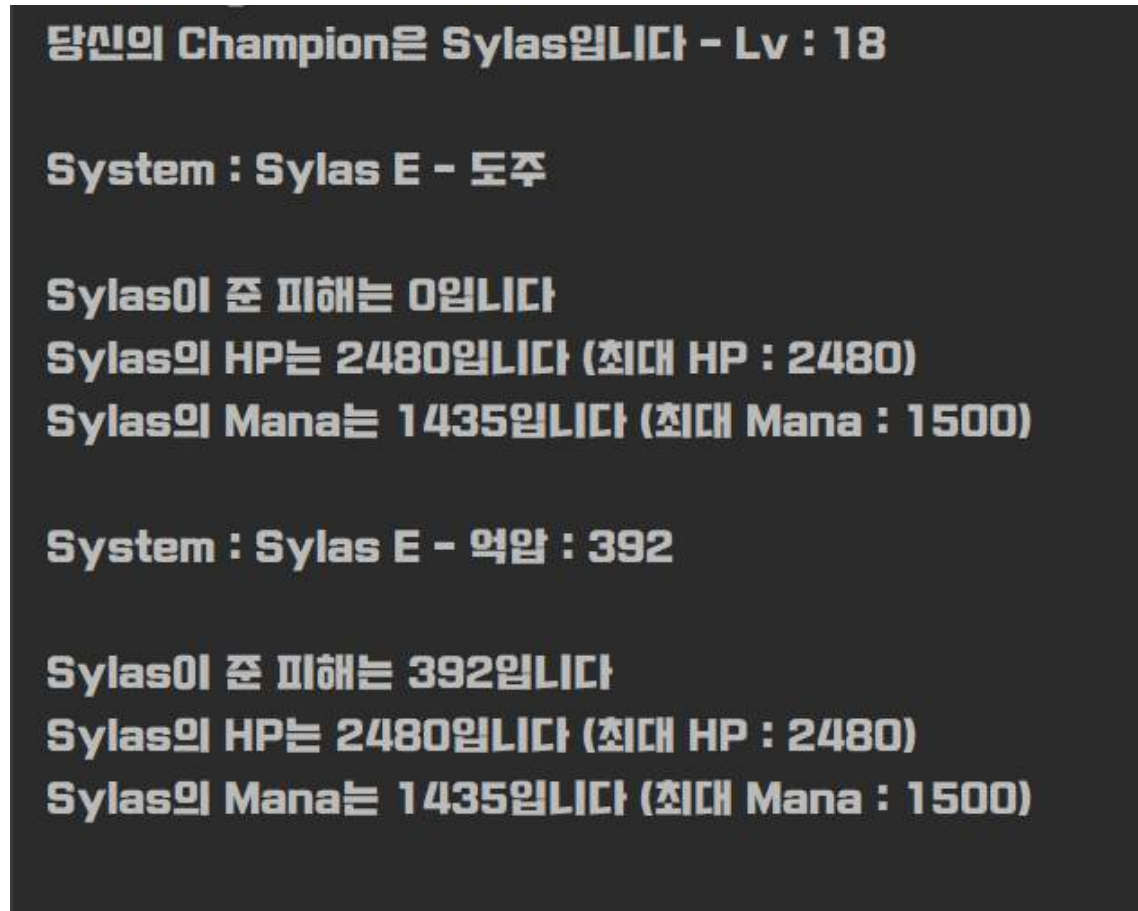
**Sylas이 준 피해는 310입니다**

**Sylas의 HP는 1669입니다 (최대 HP : 2480)**

**Sylas의 Mana는 1400입니다 (최대 Mana : 1500)**

사일러스의 체력을 1500으로 설정한다면 **현재 체력 + 회복량**이  $<$  최대체력이므로, 100의 마나를 소비하여, 적에게 310의 데미지를 주는 동시에 169만큼의 체력을 회복한다.

E스킬 구현



앞에서 이미 설명했듯이, 사일러스의 E 스킬을 최초 시전 했을 때(Sylas E - 도주) 65의 마나를 소비하여 원하는 방향으로 돌진 후, E를 재시전 했을 때 마나를 소비하지 않고 (Sylas E - 억압) 사슬을 던져 적으로 이동하여 392의 데미지를 준다.



R스킬 구현

**당신의 Champion은 Sylas입니다 - Lv : 18**

**System : Sylas R - 강탈**

**System : Talon의 궁극기를 강탈하였습니다**

**Sylas이 준 피해는 0입니다**

**Sylas의 HP는 2480입니다 (최대 HP : 2480)**

**Sylas의 Mana는 1425입니다 (최대 Mana : 1500)**

**System : Talon R - 그림자 공격 : 741**

**다음 2초동안, 당신은 은신상태입니다 !**

**System : Sylas가 강탈한 Talon의 궁극기를 사용했습니다 !**

**Sylas이 준 피해는 741입니다**

**Sylas의 HP는 2480입니다 (최대 HP : 2480)**

**Sylas의 Mana는 1425입니다 (최대 Mana : 1500)**

앞에서 이미 설명했듯이, 사일러스의 R 스킬을 최초 시전 했을 때 75의 마나를 소비하여 상대 챔피언의 R(궁극기)스킬을 강탈하여 사용가능한 상태로 만든다.

R 스킬을 재시전시 최초 시전시 빼앗은 챔피언의 궁극기(Talon의 R스킬)을 마나소모 없이 사용한다.



Talon

Passive 구현

**당신의 Champion은 Talon입니다 - Lv : 18**

**Talon이 준 피해는 140입니다**

**System : Talon Q - 녹서스릭 외교 : 210**

**Talon이 준 피해는 350입니다**

**Talon의 HP는 2203입니다 (최대 HP : 2203)**

**Talon의 Mana는 966입니다 (최대 Mana : 1006)**

**System : Talon W - 갈퀴손 : 418**

**Talon이 준 피해는 768입니다**

**Talon의 HP는 2203입니다 (최대 HP : 2203)**

**Talon의 Mana는 891입니다 (최대 Mana : 1006)**

**System : Talon Passive - 검의 최후 : 280**

**Talon이 준 피해는 1048입니다**

Q,W,R 스킬을 쓰지않고 기본공격을 할 패시브 : 검의최후 Stack이 3회 충전되지않아(Stack != 3) 140의 피해를 주는 반면, Q,W스킬을 쓰고 난 후(Stack == 3) 기본공격을 할 시에는 Talon의 패시브인 검의 최후 효과가 적용되어 기본공격의 데미지가 280으로 바뀌게 된다.

Q스킬 구현

**당신의 Champion은 Talon입니다 - Lv : 18**

**System : Talon Q - 녹서스릭 외교 : 210**

**Talon이 준 피해는 210입니다**

**Talon의 HP는 2203입니다 (최대 HP : 2203)**

**Talon의 Mana는 966입니다 (최대 Mana : 1006)**

40의 마나를 소비하여 210의 데미지를 주는 스킬이다.

W스킬 구현



75의 마나를 소비하여 418의 데미지를 주는 스킬이다.

E스킬 구현



Talon이 마나소모없이 지형지물을 넘어다닐 수 있는 스킬이며, 0의 데미지를 반환한다

R스킬 구현



100의 마나를 소모하여 다음 2초동안 은신상태에 들어간 후, 은신 상태를 해제하면 상대에게 960의 피해를 주는 스킬이다.

앞서 보았던 Sylas의 R - ‘강탈’ 스킬로 가져간 Talon의 R스킬 데미지하고는 다른 값인 960의 피해를 준다는 것을 확인할 수 있다 (TalonAdpater가 정상적으로 작동한다는 의미)