

〈Unity Physics 2D Extensions〉

1. 개요

Physics 2D Extensions는 **Unity** 기본 2D 물리 시스템(**BoxCollider2D**, **CircleCollider2D**, **Rigidbody2D** 등)을 보완해, 더 정교한 타격 판정(**Hitbox**) 및 피격 판정(**Hurtbox**)을 구현할 수 있도록 도와주는 확장 기능입니다.

전투 게임, 액션 RPG, 슈팅 게임 등에서 공격/피격 판정 시스템을 손쉽게 관리할 수 있습니다.

2. 주요 기능

-히트박스(Hitbox) / 허트박스(Hurtbox) 구현

- 공격 범위와 피격 범위를 분리해 세밀한 전투 판정 가능
- Trigger 기반 충돌 처리 강화
- 단순 Collider 충돌이 아닌 공격 판정 전용 시스템 지원

-확장 가능 구조

- 다양한 무기/스킬/캐릭터의 판정을 손쉽게 추가 가능

-디버깅 기능

- Scene 뷰에서 히트박스/허트박스를 시각적으로 확인

3. 기본 구조

```
public class Hitbox : MonoBehaviour
{
    public int damage = 10;

    private void OnTriggerEnter2D(Collider2D other)
    {
        Hurtbox hurtbox = other.GetComponent<Hurtbox>();
        if (hurtbox != null)
        {
            hurtbox.TakeDamage(damage);
        }
    }
}

public class Hurtbox : MonoBehaviour
{
    public int health = 100;

    public void TakeDamage(int damage)
    {
        health -= damage;
        Debug.Log($"{gameObject.name} 피격! 남은 체력: {health}");
    }
}
```

4. 활용 예시

4.1 격투 게임

- 캐릭터마다 공격 애니메이션 프레임에 맞춰 히트박스 생성

- 상대 캐릭터의 히트박스와 충돌 시 데미지 판정

4.2 액션 RPG

- 무기 스윙 범위를 CircleCollider2D Hitbox로 설정

- 적의 피격 범위를 Hurtbox로 설정

4.3 슈팅 게임

- 총알의 Collider를 Hitbox로 설정

- 플레이어/적에게 Hurtbox 부착

5. 구현 절차

5.1 캐릭터 Prefab 구성

- ‘Player’ – Rigidbody2D + Animator
- ‘Enemy’ – Rigidbody2D + Hurtbox

5.2 무기 / 스킬 Prefab 구성

- SwordSwing, Bullet- Hitbox 컴포넌트 추가

5.3 OnTriggerEnter2D로 판정 처리

- Hitbox – 공격 발생

- Hurtbox – 데미지 수신

5.4 디버그 및 최적화

- Gizmos로 히트박스 표시
- 필요시 Object Pooling 적용

6. 장점

- 정밀한 타격 판정 가능
- Unity Physics 2D와 완벽 호환
- 확장성이 뛰어나 다양한 장르 적용 가능

7. 단점 / 주의사항

- Collider2D + Rigidbody2D 의존 - 성능 이슈 발생 가능
- 프레임 단위 판정 관리 필요 (애니메이션 이벤트 활용 권장)
- 기본 Physics보다 관리할 오브젝트가 많아질 수 있음

8. 결론

Physics 2D Extensions는 단순 충돌 판정을 넘어서 격투 게임 수준의 정밀한 공격/피격 시스템을 구축할 때 필수적인 도구입니다.

Unity 2D 프로젝트에서 전투를 강화하려면 반드시 고려할 만한 확장 기능입니다.