

# 게임 클라이언트 이력서

도전하고 행동하는 개발자 배성식입니다.

EMAIL: [rabae0985@gmail.com](mailto:rabae0985@gmail.com)

# 목차

1. 소개
2. Witch One
3. Stars Of Prey
4. 49 Defense
5. Phage
6. Bon Appetit
7. Earth Dive
8. Running Game

# 기술 스택 및 학력

학력:

- 2018~2024: 청강문화산업대학교 게임학과 졸업(3.76/4.5)
- 2015-2018: 세명고등학교(포항)

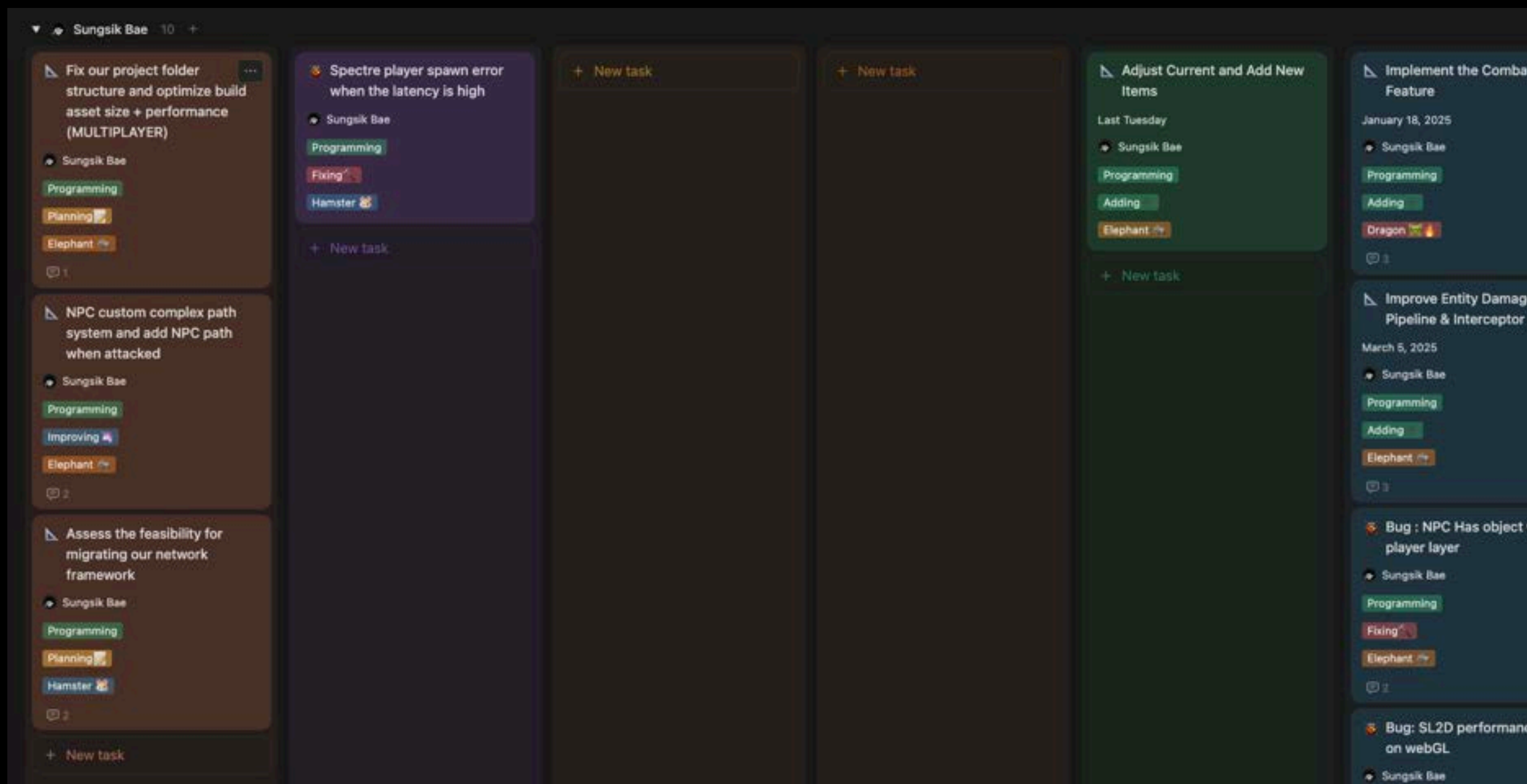
수상 내역:

- 경상북도 학생 정보올림피아드(공모) [은상] / 2015
- 무한도전 장학 프로젝트 게임 제작 경진대회 [2등] / 2018
- 2023 청강 알발 페스타 [금상] / 2023

| 구분                    | Skill  |
|-----------------------|--|
| 언어                    | C++, C#  |
| Framework/<br>Libaray | PUN2, UniTask, UniRx(R3), VContainer,<br>DOTween, Odin Inspector,<br>MessagePipe |
| 엔진                    | 유니티 엔진   |
| 기타                    | Notion, Slack  |

# 작업하는 방식

## 티켓 관리



### Acceptance Criteria

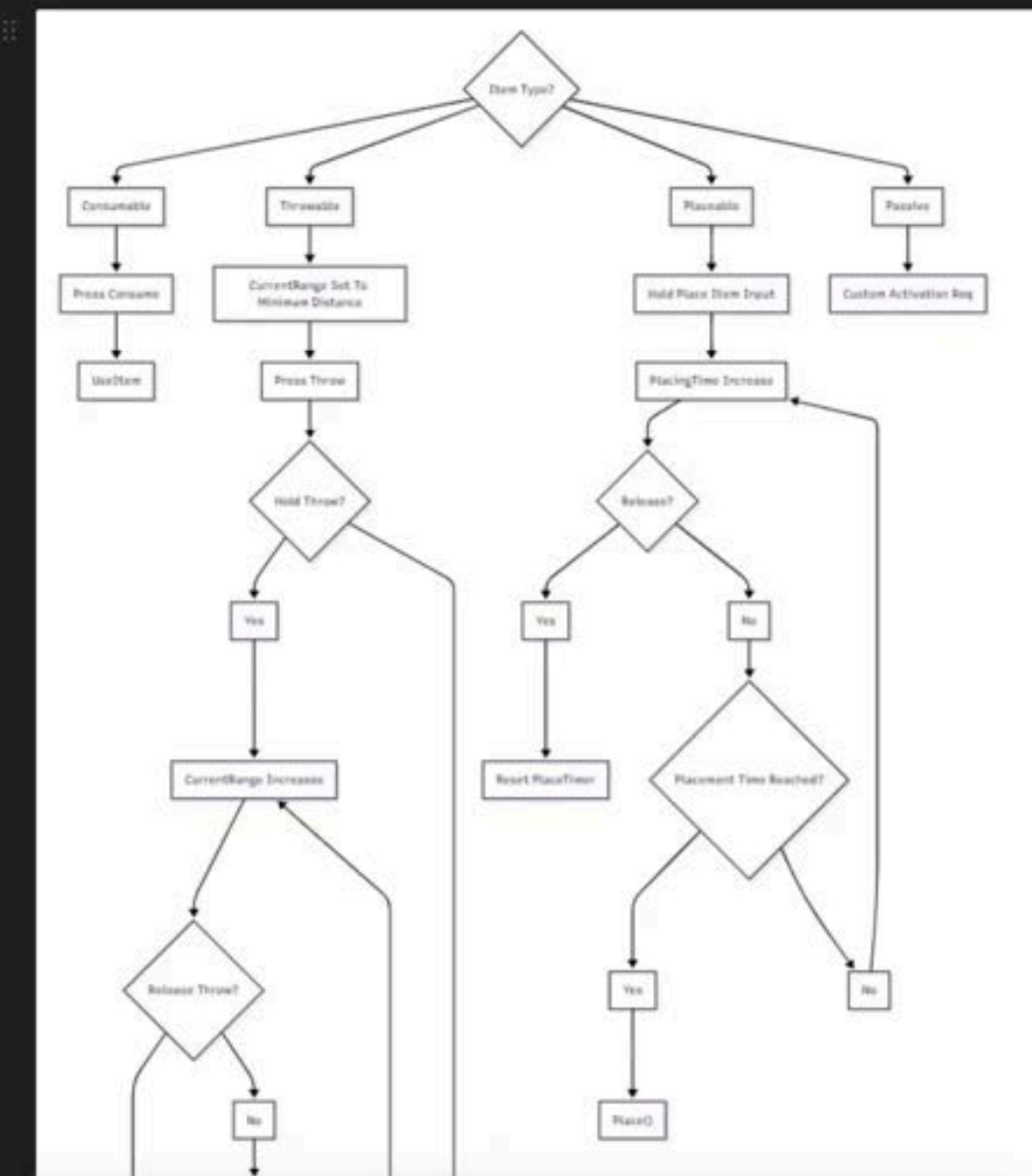
The criteria must be met to consider this deliverable to be completed.

[Add a Criteria](#)

- ☒ Items Are picked-up-by-Pressing-the-"Interact"-Key
- ☒ Items need to be accessed through the Contextual interaction
- ☐ Throwable Items Mechanic returned
  - ☒ Press + Hold "Use" key for up to 1s
    - ☒ Throws Current throwable item in Player's facing direction
    - ☒ Distance is Min To Max throw distance based on charge ti
    - ☒ Display throwing Arc and Landing Point while being charg
  - ☐ Throwable Type Distinction
    - ☒ "Flat Speed" throwables are constant speed, in a flat traje
      - ☒ Move at speed until distance reached
      - ☒ Hits anything it comes into contact with while in air (s
    - ☐ "Arc" Throwables are Arc-Based and use a time-based m
      - ☒ Time To land after being thrown is scaled from 0.1 (or s
      - ☒ Scale relation is based on Charge-Time (ThrowDis
    - ☐ Only Stops at land position or **if hitting High Walls (Ho**
- ☐ Stone Item Returned
  - ☒ Throwable (Flat)
  - ☐ Distinct Sound Event called when landing or hitting somet
  - ☒ Deals 30 Damage when Hitting a character
  - ☒ 25% Chance to 1-2s Stun character it hits
  - ☒ "Damaged" Variable Value increases Upon Hitting Something:
    - ☒ 20-75% Chance of Breaking when thrown based on 0-MaximumAllowedDamaged-Variable
  - ☐ Blast Pouch Item Returned
    - ☒ Throwable (Arc)
    - ☐ Sound Event Called When Exploding
    - ☒ Deals 50 damage in a 3Unit radius when landing

### Logs

- @April 10, 2025




Wiki: [https://drive.google.com/file/d/14P\\_9D1VjgbOa7wmZomrkUs6sLt-ZAUFO/view?usp=sharing](https://drive.google.com/file/d/14P_9D1VjgbOa7wmZomrkUs6sLt-ZAUFO/view?usp=sharing)



# 작업하는 방식

## PR 작성 및 리뷰

 Labae commented 3 weeks ago · edited ▾

### Pull Request Summary

#### Description

Implements comprehensive utility items system including throwable mechanics, placeable items, and enhanced interaction systems. This PR brings back legacy items (Stone, Blast Pouch) with modern improvements and introduces new placeable items (Teleporter Rune, Trap Rune) to expand tactical gameplay options.

[Ticket](#)

[Video](#)

#### Check List


- ☒ contains breaking changes
- ☒ adds new feature
- ☒ modifies existing feature
- ☐ relies on other tasks
- ☐ non-coding changes


#### Change Log

This pull request makes the following changes:

##### Adds


- Contextual interaction system for item pickup using "Interact" key
- Throwable items mechanic with charge-based throwing (0- MaxChargeTime )
- Visual throwing arc and landing point indicator with real-time feedback
- Dual trajectory system:
  - Flat speed throwables (constant velocity, hits anything in path)
  - Arc-based throwables (time-scaled parabolic, selective collision)
- Stone item with 30 damage, 25% stun chance, and durability system

 Labae requested changes on Mar 7 View reviewed changes


 Labae left a comment

The functionality works properly :>

```
Assets/Scripts/Runtime/Character/NetworkCharacter.cs
...    @@ -150,14 +149,14 @@ public void RemoveCurse()
150    149         photonView.RPC(nameof(RPC_RemoveCurse), RpcTarget.All, photonView.ViewID);
151    150     }
152    151
153    - public void GenerateSoundWaveEvent(SoundWaveType waveType)
152    + public void GenerateSoundWaveEvent(SoundWaveType waveType, string sfxParameterValue
```


 Labae on Mar 7


Is it impossible to generate footprints and sound waves not use RPC?  
<https://www.notion.so/teamstepgames/Refactor-the-soundwave-generation-and-detection-mechanism-planning-19e40abe716d8081b227eee4754f5896?pvs=4#1a840abe716d8097b9b6f897ef305e4>

 thilina27 (Thilina Madhusanka) on Mar 7 Author


For taking damage, DOT ticks are handle on client that taking damage. So we need to all.

For footstep, sound wave does not use RPC calls. Only the Footprints that uses RPC. I when we're not using RPC for footprint they seem to miss align on clients.




 Labae reviewed on Jun 21, 2024 View reviewed changes


```
Assets/Scripts/Runtime/Character/Detector/InteractableDetector.cs
38    + private void Start()
39    + {
40    +     _owner = GetComponentInParent<CharacterFacade>();
41    +     _owner.Input.OnInteract.AddListener(PlayerInteractionInputDetected);
```

 Labae on Jun 21, 2024


It's not a big deal but We need to add the `RemoveListener` on `OnDestroy`.  
I'm not sure we should remove the listener because the `CharacterInput` instance will be deleted when `ForestVillage(GameplayScene)` is unloaded.

 Labae on Jun 21, 2024

In my opinion, Removing the listener makes sense.

 thilina27 (Thilina Madhusanka) on Jun 21, 2024 Author

I didnt remove the listener since they both get destroyed when leaving the map. I will remove it on destroy just to make sure. :)


 Reply...

5



# 작업하는 방식

## PR 작성 및 리뷰

 Labae commented 3 weeks ago · edited ▾

### Pull Request Summary

---

#### Description

Implements comprehensive utility items system including throwable mechanics, placeable items, and enhanced interaction systems. This PR brings back legacy items (Stone, Blast Pouch) with modern improvements and introduces new placeable items (Teleporter Rune, Trap Rune) to expand tactical gameplay options.

[Ticket](#)

[Video](#)

#### Check List


- ☒ contains breaking changes
- ☒ adds new feature
- ☒ modifies existing feature
- ☐ relies on other tasks
- ☐ non-coding changes


#### Change Log

This pull request makes the following changes:

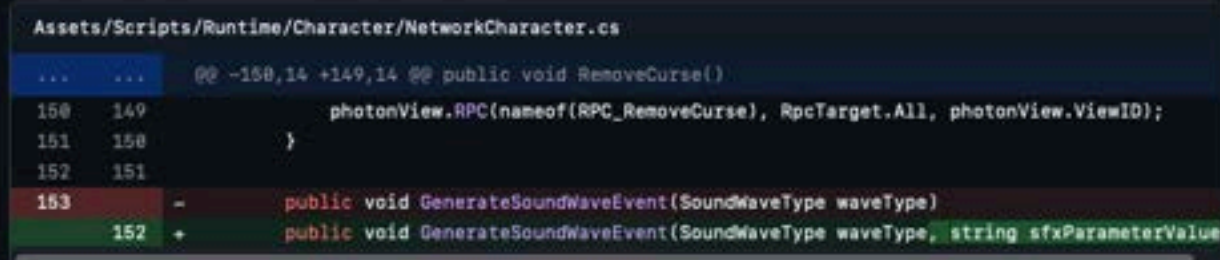
##### Adds

- Contextual interaction system for item pickup using "Interact" key
- Throwable items mechanic with charge-based throwing (0- MaxChargeTime )
- Visual throwing arc and landing point indicator with real-time feedback
- Dual trajectory system:
  - Flat speed throwables (constant velocity, hits anything in path)
  - Arc-based throwables (time-scaled parabolic, selective collision)
- Stone item with 30 damage, 25% stun chance, and durability system


 Labae requested changes on Mar 7 View reviewed changes

 Labae left a comment


The functionality works properly :>



```
Assets/Scripts/Runtime/Character/NetworkCharacter.cs
...    @@ -150,14 +149,14 @@ public void RemoveCurse()
150    149         photonView.RPC(nameof(RPC_RemoveCurse), RpcTarget.All, photonView.ViewID);
151    150     }
152    151
153    -     public void GenerateSoundWaveEvent(SoundWaveType waveType)
152    +     public void GenerateSoundWaveEvent(SoundWaveType waveType, string sfxParameterValue
```

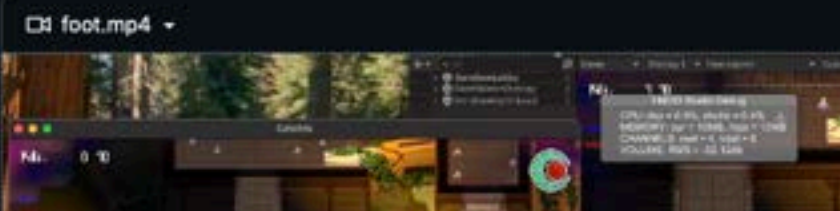
 Labae on Mar 7


Is it impossible to generate footprints and sound waves not use RPC?  
<https://www.notion.so/teamstepgames/Refactor-the-soundwave-generation-and-detection-mechanism-planning-19e40abe716d8081b227eee4754f5896?pvs=4#1a840abe716d8097b9b6f897ef305e4>

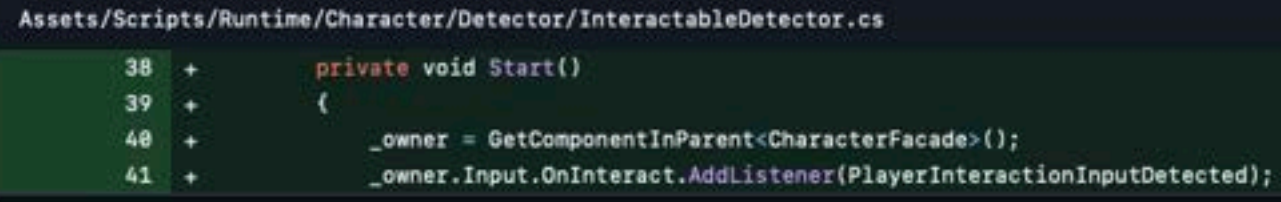
 thillina27 (Thilina Madhusanka) on Mar 7 Author

For taking damage, DOT ticks are handle on client that taking damage. So we need to all.


For footstep, sound wave does not use RPC calls. Only the Footprints that uses RPC. I when we're not using RPC for footprint they seem to miss align on clients.




 Labae reviewed on Jun 21, 2024 View reviewed changes




```
Assets/Scripts/Runtime/Character/Detector/InteractableDetector.cs
38    +     private void Start()
39    +     {
40    +         _owner = GetComponentInParent<CharacterFacade>();
41    +         _owner.Input.OnInteract.AddListener(PlayerInteractionInputDetected);
```

 Labae on Jun 21, 2024


It's not a big deal but We need to add the `RemoveListener` on `OnDestroy`.  
I'm not sure we should remove the listener because the `CharacterInput` instance will be deleted when `ForestVillage(GameplayScene)` is unloaded.

 Labae on Jun 21, 2024

In my opinion, Removing the listener makes sense.

 thillina27 (Thilina Madhusanka) on Jun 21, 2024 Author

I didnt remove the listener since they both get destroyed when leaving the map. I will remove it on destroy just to make sure. :)

 Reply...



# Witch One

기간 | 2024.01.14~

인원 | 9명(기획 1, 아트 3, 프로그래밍 4, 사운드 1)

도구 | Unity / PC

플러그인 | UniTask, Odin Inspector, DOTween, PUN2  
Jump Point Search(길찾기 사용)

주요 역할:

- 아이템
- 데미지 파이프라인
- 이펙트 시스템
- 애니메이션 생성 에디터



홈페이지: <https://witchone.io/>

영상: <https://youtu.be/Ftl-HL9nI5k?si=FY-qr53GWu5Razki>

# 아이템

## 아이템 종류

### 아이템 종류

- **소모** 아이템: 사용 시 소모되며 즉시 효과를 발휘 (체력 회복, 등)
- **던지기** 아이템: 목표 지점에 던져서 원거리 효과를 발휘 (피해, 등)
- **트리거** 아이템: 특정 조건 충족 시 자동으로 발동되는 수동형 아이템
- **설치** 아이템: 맵에 설치하여 지속적인 효과를 제공 (함정, 등)
- **조종** 아이템: 사용 시 조종 가능한 엔티티를 생성하여 플레이어가 직접 제어



플레이 영상

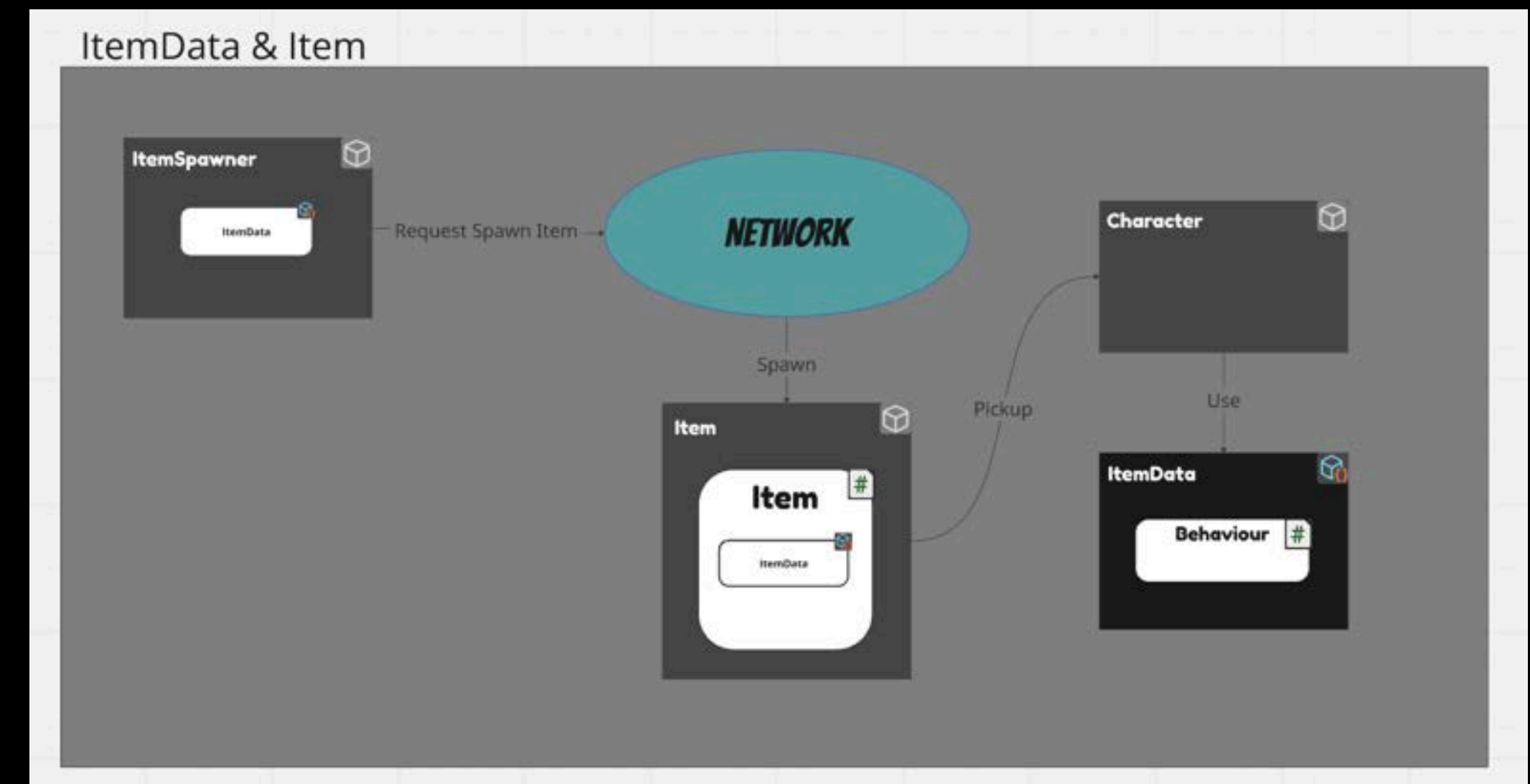


# 아이템

## 아이템 구조

### 설계 장점

- 관심사 분리: 데이터 정의와 실행 로직 분리로 코드 명확성 향상
- 협업 최적화: 기획자는 데이터 조정, 개발자는 동작 구현에 집중
- 메모리 효율: 동일 데이터를 여러 인스턴스가 공유하여 자원 절약
- 유지보수성: 데이터 또는 동작만 독립적으로 수정 가능
- 확장성: 새로운 아이템 타입 추가가 기존 시스템에 영향 최소화



코드 링크:

<https://gist.github.com/Labae/242f06b90d368850d0067005cded8af1>

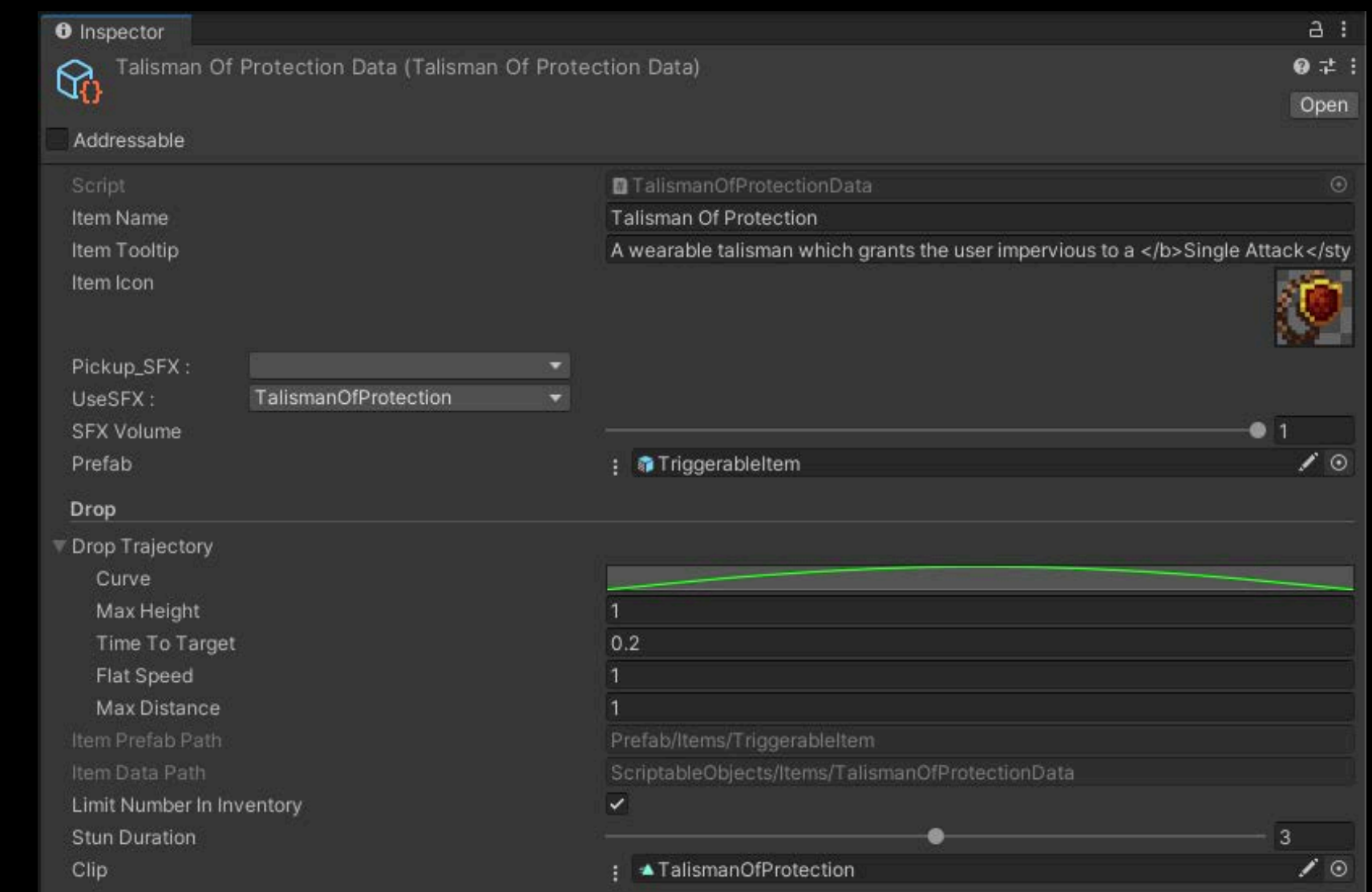
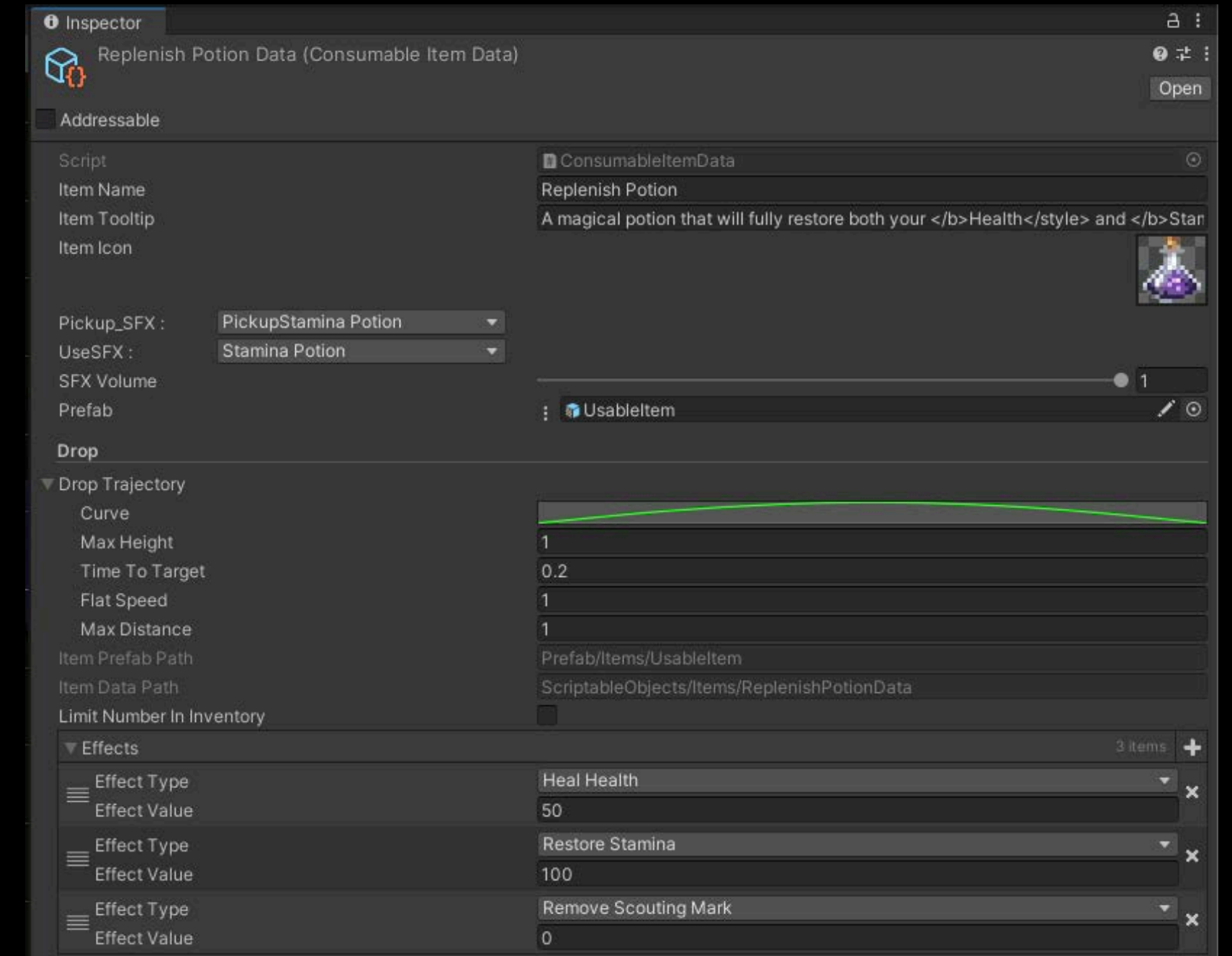
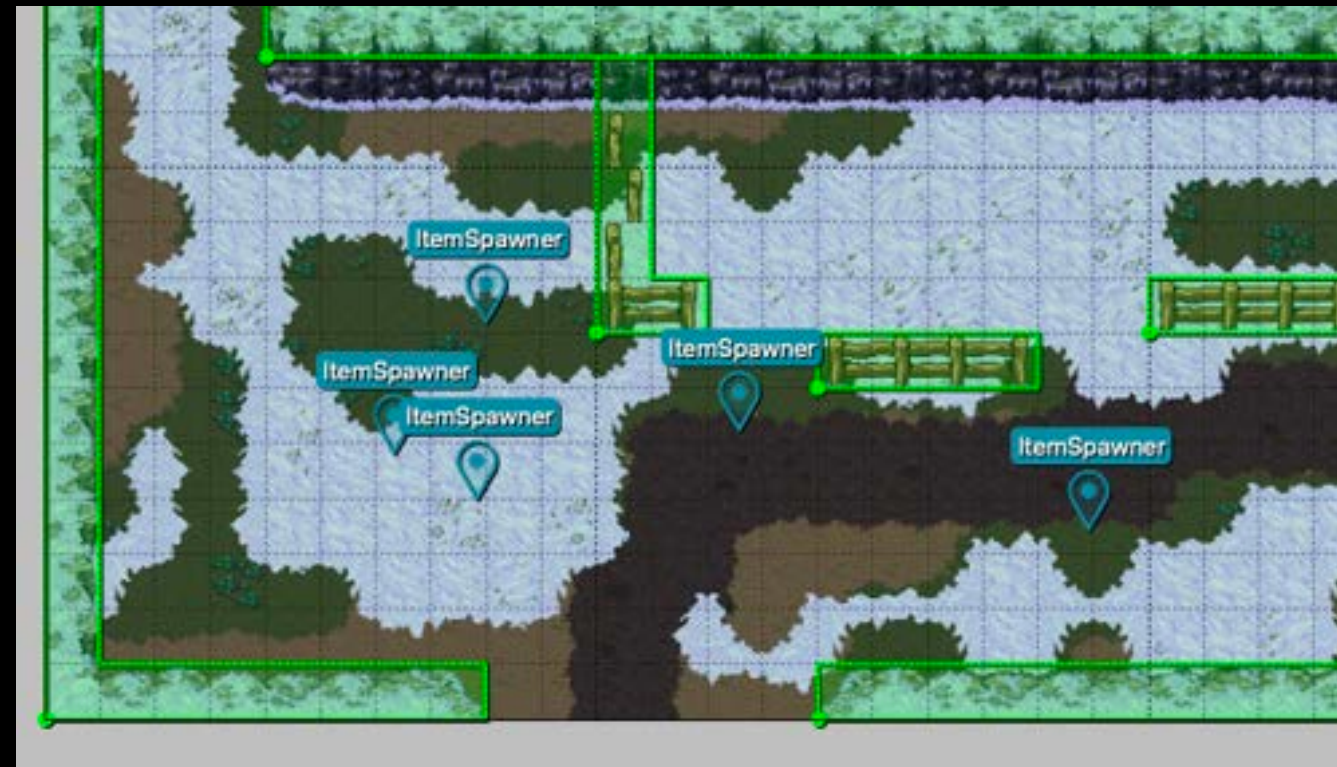
<https://gist.github.com/Labae/59423b07875f30cf2bead1bfb3c1e213>

# 아이템

## 실제 사례

### 실제 사례:

- 기획자들이 Tiled에서 아이템 종류, 위치등을 자유롭게 설정함.
- 프로그래밍적 지식이 없어도 Data에 접근하여 다양한 효과의 아이템을 구현함.
- **빠른 프로토타이핑**이 가능해 새로운 아이템 아이디어를 즉시 레벨에 적용 가능함.





# 데미지 파이프라인

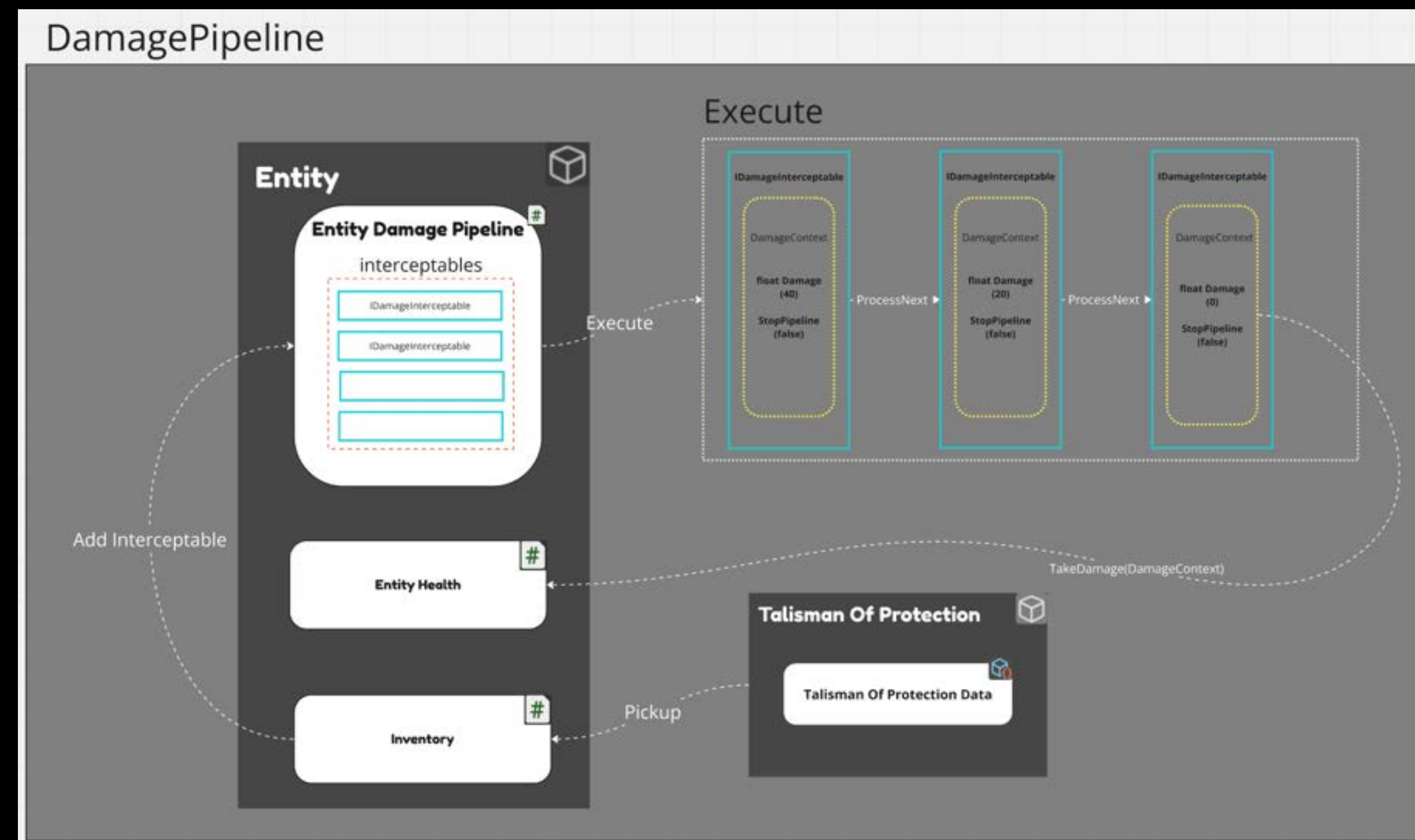
## 시스템 개요

### 데미지 파이프라인:

- 엔티티에게 들어오는 데미지를 중간에 가로채 다양한 효과를 적용할 수 있는 구조

### 책임 연쇄 패턴 구현

- 데미지 처리를 일련의 인터셉터 체인으로 구성
- 각 인터셉터가 순차적으로 데미지 컨텍스트를 처리하고 다음 단계로 전달
- 처리 과정 중 언제든지 파이프라인 중단 가능



# 데미지 파이프라인

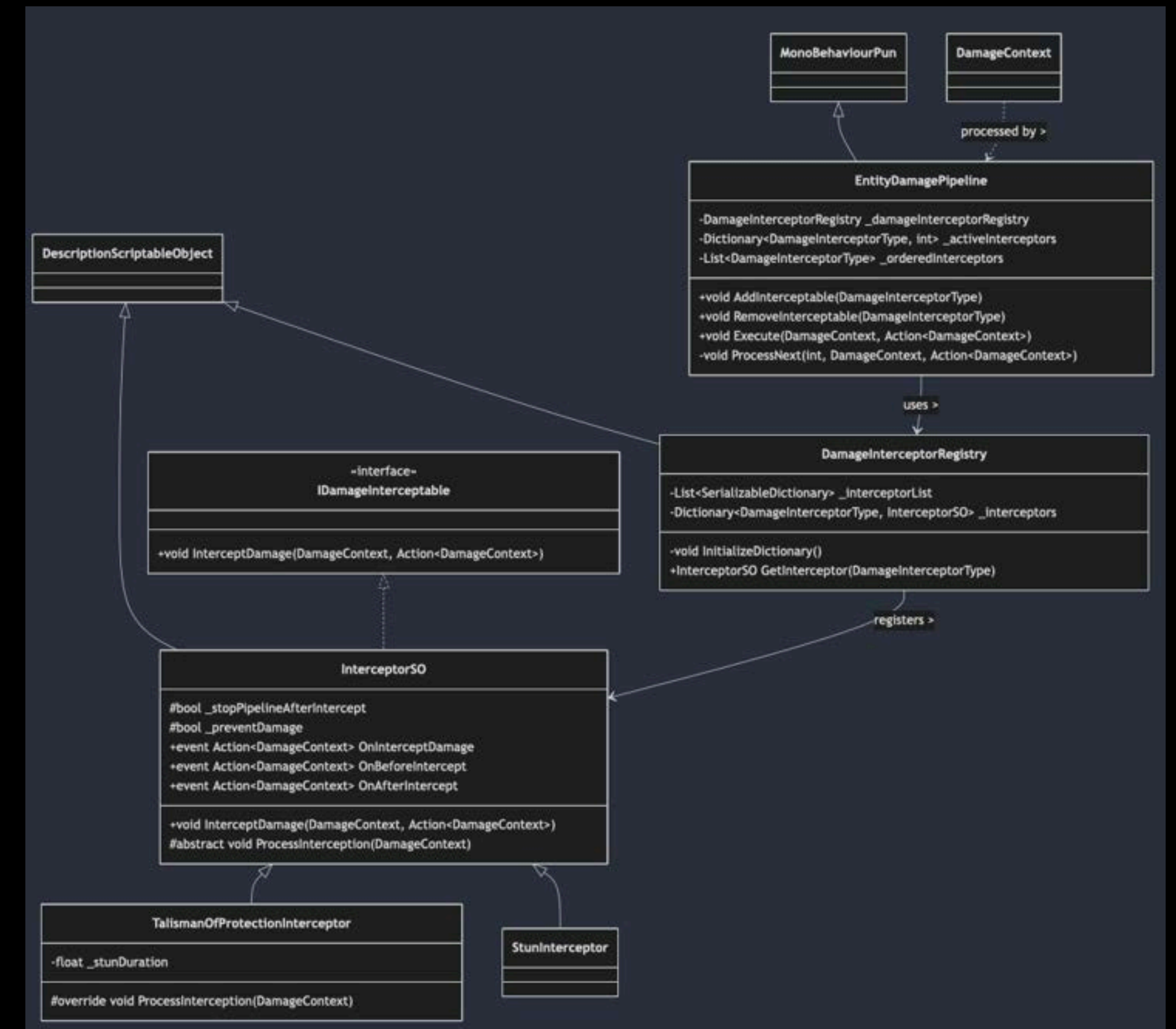
## 구조 및 설계

### 구조:

- InterceptorSO: ScriptableObject 기반 데이터 중심 인터셉터 정의
- Registry시스템: 인터셉터들을 중앙에서 관리하고 제공하는 레지스트리
- 체인 구성: Registry로부터 필요한 인터셉터들을 가져와서 동적으로 구성

### 설계 장점

- 모듈성: 새로운 데미지 처리 로직은 인터셉터 추가만으로 확장 가능
- 유연성: 인터셉터 순서와 조합으로 다양한 게임 메커닉 구현
- 디커플링: 데미지 발생과 처리 로직의 분리로 유지보수성 향상
- 재사용성: 동일한 인터셉터를 여러 엔티티나 상황에서 재사용 가능





# 데미지 파이프라인

## 사용 예시

### 실제 사용 예시

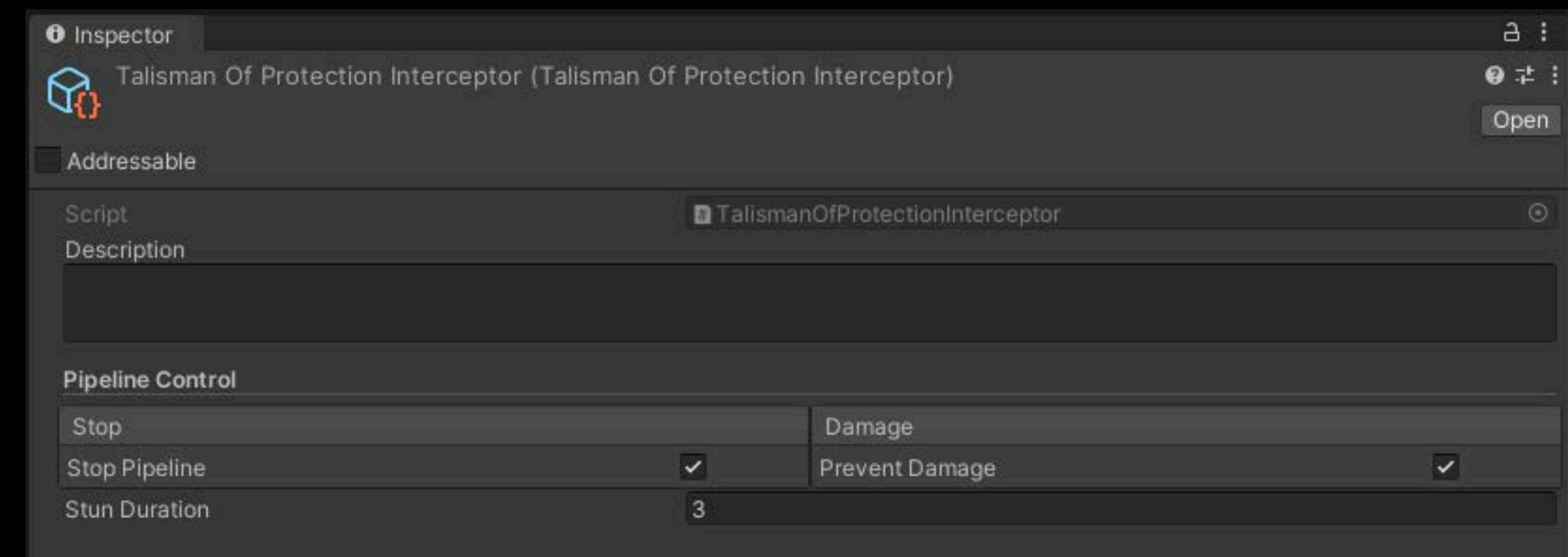
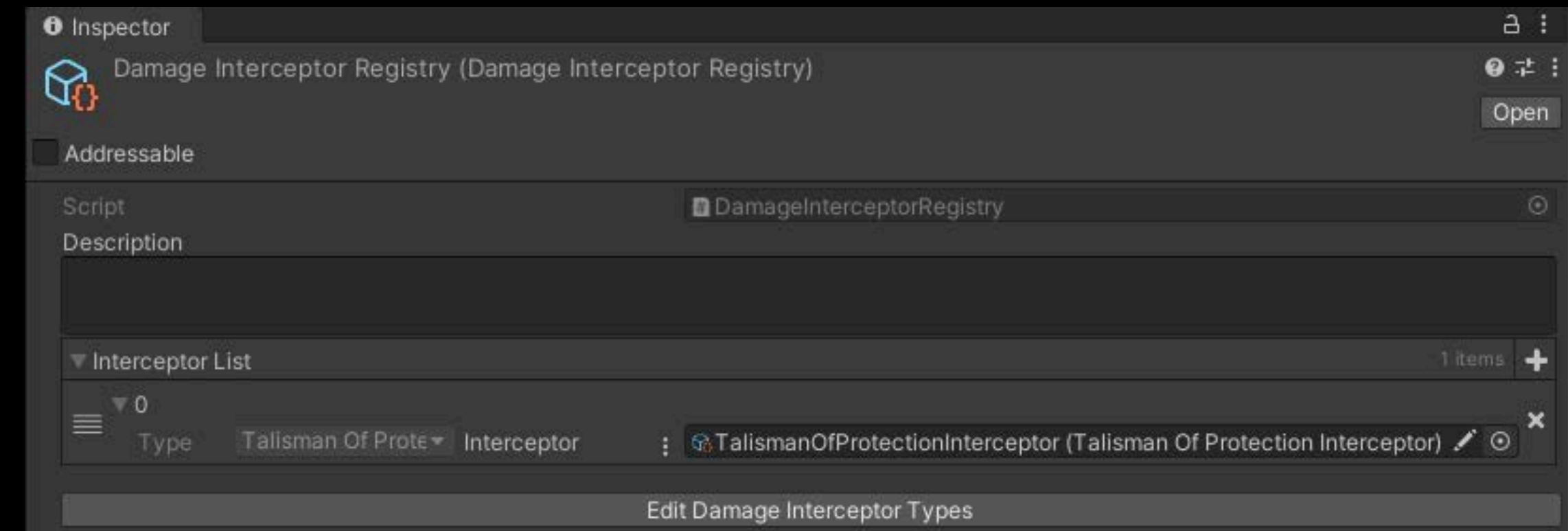
- 보호의 부적(Talisman of Protection): 데미지 차단 및 공격자에게 스턴 효과
- 아이템과 연동: TriggerableItemData가 IDamageInterceptable 구현하여 파이프라인 참여
- 동적 인터셉터 관리: 게임 중 아이템 효과에 따라 인터셉터 추가/제거

### 기획자 활용

- Data-Driven 설계: 기획자들이 InterceptorSO를 통해 코드 수정 없이 새로운 방어 메카닉 제작
- 빠른 프로토타이핑: 기존 인터셉터들의 조합으로 다양한 방어 시스템 구현
- 재사용성 극대화: 한 번 만든 인터셉터를 다양한 아이템과 캐릭터에 재활용

### 개발 생산성 개선

- 새로운 방어 효과 구현 시간 단축
- 버그 발생률 감소(검증된 인터셉터SO를 재사용)



# 이펙트 시스템

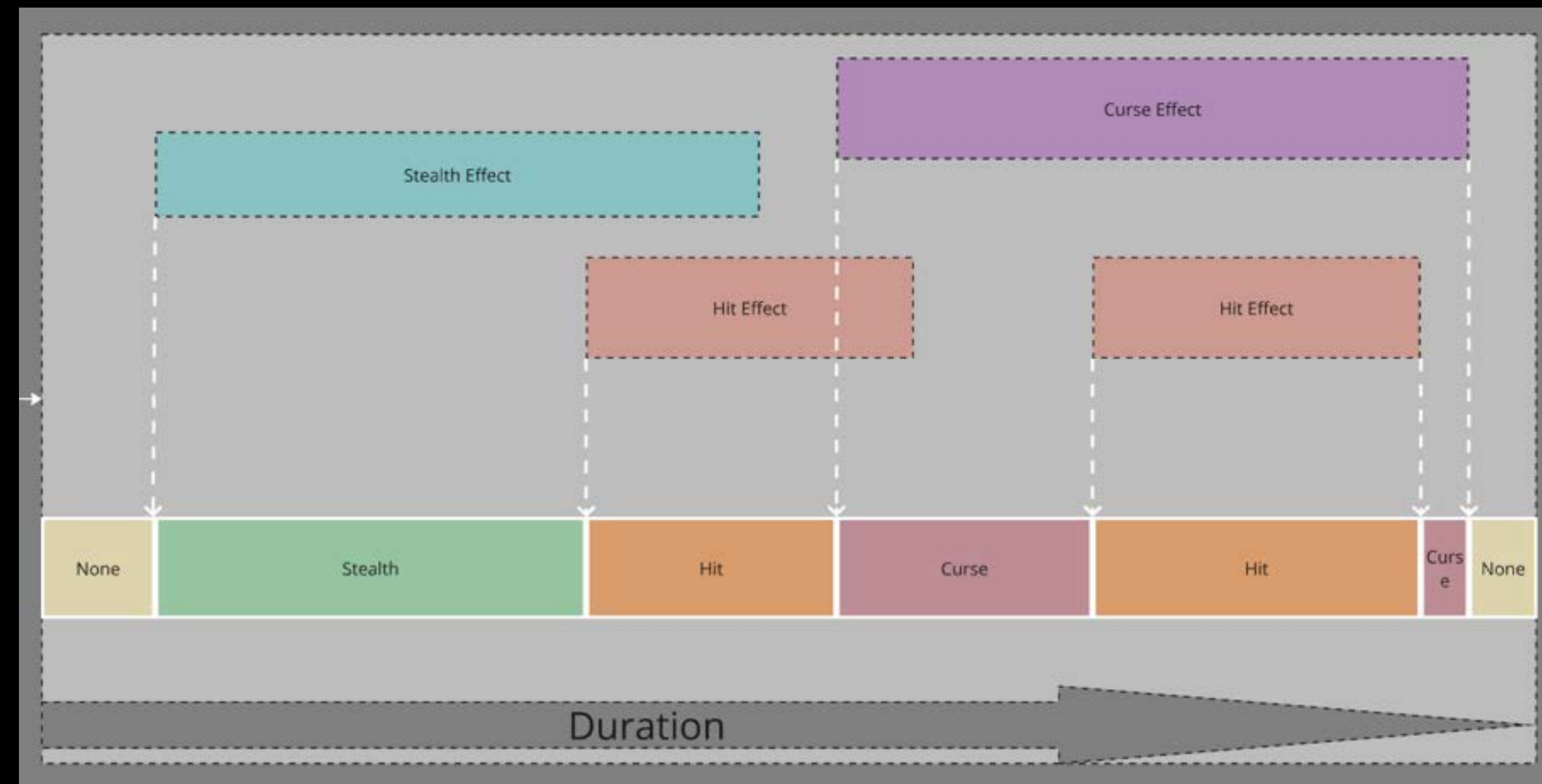
## 이펙트 시스템 개요

### 구현 기능:

- 통합 이펙트 관리: 셰이더, 이펙트를 **하나의 시스템**으로 통합 관리
- Data-Driven 이펙트: ScriptableObject 기반 코드 수정 없이 효과 제작
- 실시간 동기화: 멀티 플레이 환경에서 이펙트 상태 **동기화**
- 라이프사이클 관리: 자동 생성/해제 및 **메모리 누수 방지**

### 시스템 장점:

- 확장성: 새로운 이펙트 타입이나 엔티티 추가가 용이
- 재사용성: 이펙트 데이터 에셋으로 **일관된 효과 관리**
- 네트워크 효율: 최소한의 데이터로 시각 효과 동기화
- 동시성 관리:
  - UniTask 기반 **비동기 처리**
  - 취소 토큰을 통한 안전한 이펙트 중단
  - 중첩 이펙트 처리를 위한 **순서 관리 시스템**





# 이펙트 시스템

## 사용 예시

### 실제 구현 사례:

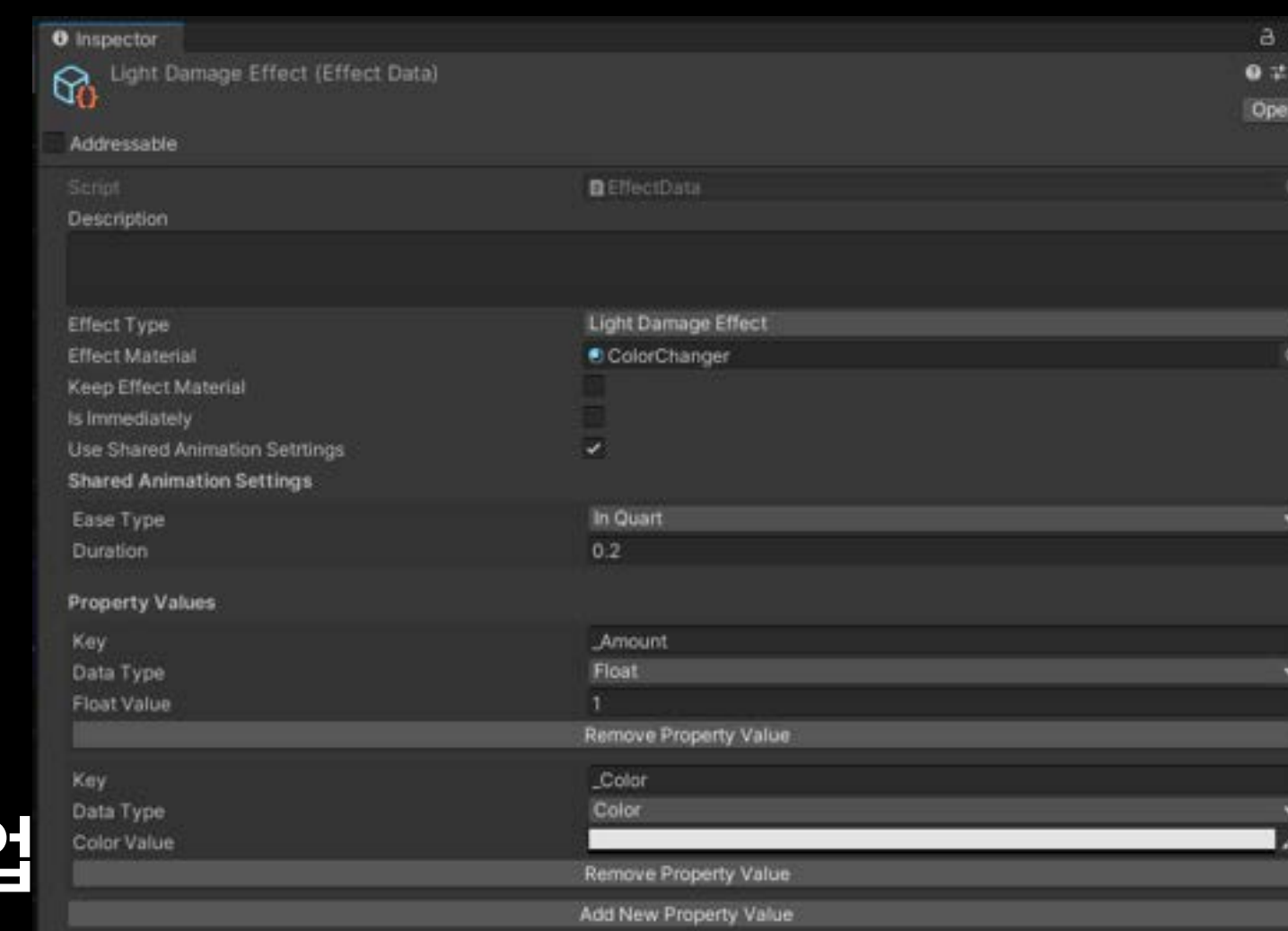
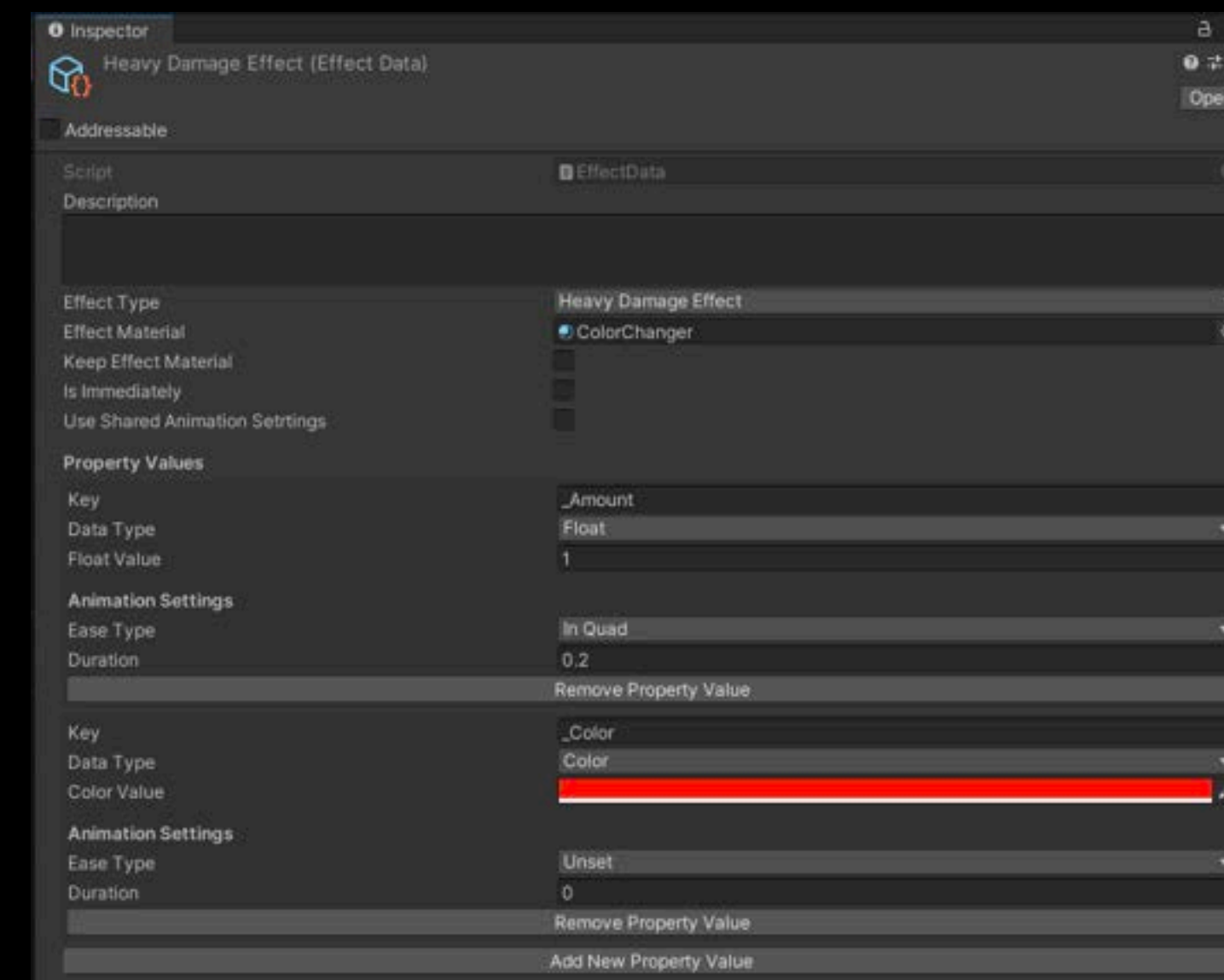
- 헤비 데미지 이펙트: 빨간색 플래시 + 화면 흔들림 효과
- 라이트 데미지 이펙트: 하얀색 플래시 + 부드러운 페이드
- 힐링 이펙트: 초록색 글로우

### 활용 방식:

- Material 기반 구성: 시각적 스타일 정의
- 프로퍼티별 애니메이션: PropertyValue를 통해 색상, 투명도, 강도 등 개별 제어
- 유연한 타이밍: Duration, EaseType 조합으로 다양한 느낌의 효과 연출
- 즉시 적용 옵션: 순간적인 효과와 점진적 효과 구분

### 기획자 활용 및 생산성 향상

- **코드 없는** 이펙트 제작: 기획자들이 EffectData 에셋만으로 새로운 시각 효과 제작
- 실시간 밸런싱: Duration과 Ease 값 조정으로 게임 내에서 확인 가능
- 아티스트 협업: EffectMaterial은 아티스트가, 애니메이션 설정은 기획자(아티스트)가 작업



# 애니메이션 생성 에디터

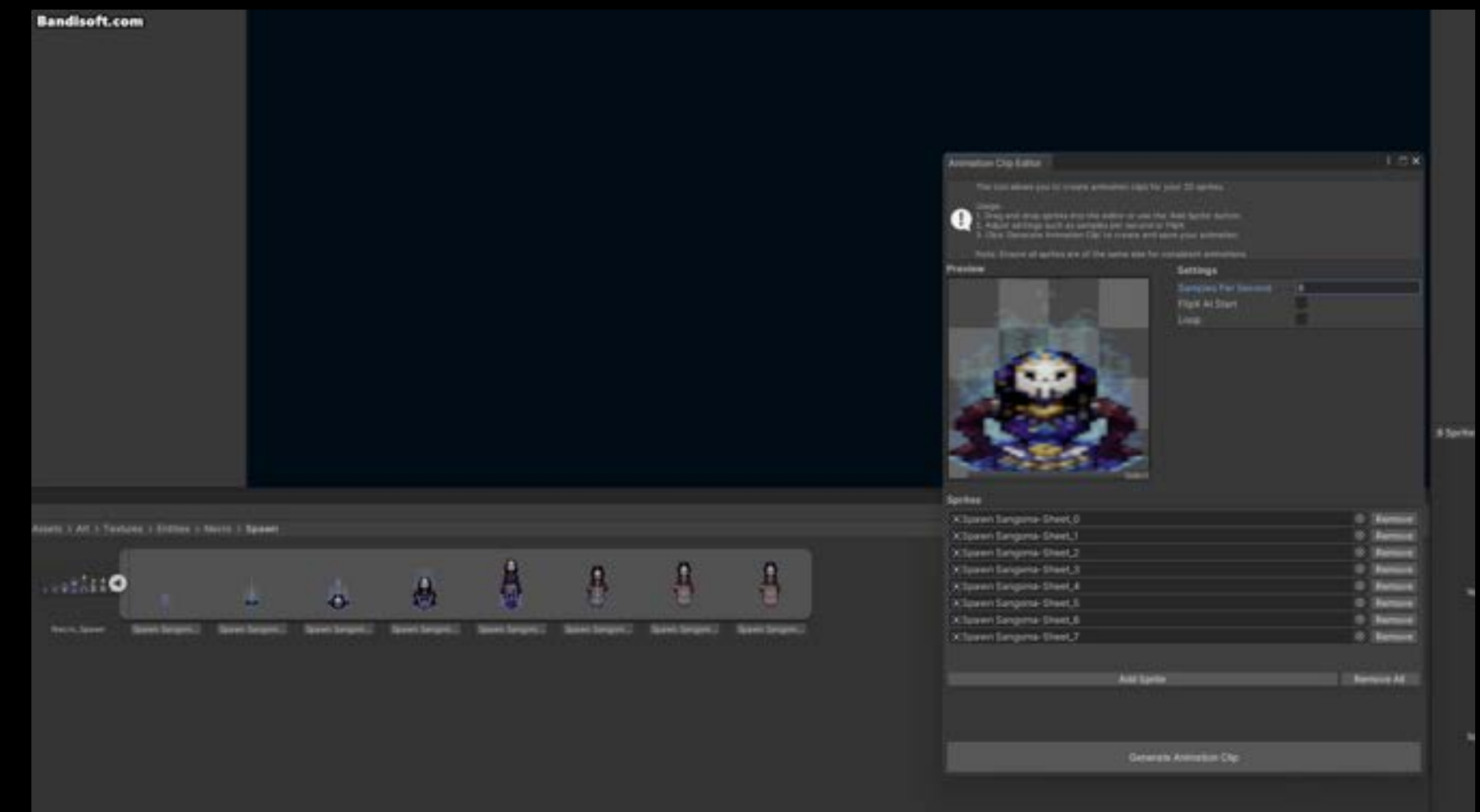
## 애니메이션 생성 에디터 개요

### 개발 배경 및 목적:

- 반복 작업 **자동화**: 다수의 애니메이션 클립을 수동으로 생성하는 비효율성 해결
- 통합 워크플로우: AnimationEvent 추가 과정을 포함한 애니메이션 제작 **파이프라인 통합**
- 생산성 향상: 아티스트와 기획자가 유니티 에디터의 지식 없이 애니메이션 시스템 활용 가능

### 에디터 확장 시스템

- Unity 에디터를 확장한 **전문 애니메이션 제작 도구**
- OdinInspector 기반으로 강화된 인터페이스 제공
- 일괄 처리 가능: 여러 스프라이트 시퀀스를 한 번에 애니메이션 클립으로 전환
- 이벤트 통합 관리: AnimationEvent **자동 삽입 및 관리** 시스템



영상: <https://youtu.be/ykAXNwdpzpk>



# 애니메이션 생성 에디터

## 시스템 장점 및 응용

### 확장성 및 유지보수성

- 모듈화된 에디터 도구로 새로운 기능 추가 용이
- 네이밍 컨벤션을 통한 체계적인 에셋 관리
- 에디터 전용 코드와 런타임 코드의 명확한 분리

### 실제 적용 사례

- 다양한 캐릭터 애니메이션 세트 빠른 제작
- 방향 기반 애니메이션 시스템 구현 간소화
- 프로토타입에서 최종 게임까지 일관된 애니메이션 파이프라인 유지



영상: <https://youtu.be/6p4YU6CG1k8>

# Stars Of Prey

기간 | 2023.03.01 ~ 2023.11.xx

인원 | 11명(기획 3, 아트 5, 프로그래밍 2, 사운드 1)

도구 | Unity / VR

플러그인 | VContainer(DI), UniRx, UniTask, Odin Inspector, DOTween

주요 역할:

- 보스 AI 구현
- 적기 구현
- 아군기 구현
- 보스 IK 구현
- Outline 구현

이 게임은 좋은 작품입니다.  
이 게임은 칭찬할것 밖에 떠오르지가 않네요.  
단점을 굳이 적어보자면 VR 기기가 없는 분들을 위한  
게임이 따로 나왔으면 하는 바람이 있다는점이 가장 큰 단점이겠네요.  
장점을 적어보자면 무진장 많이 있습니다.  
재미와 사운드 디자인, 레벨 디자인, 아트 디자인등  
여러모로 사랑스럽고 좋은 게임입니다.

리뷰

스토브: <https://store.onstove.com/ko/games/2650>

스팀: [https://store.steampowered.com/app/2782480/Stars of Prey VR/](https://store.steampowered.com/app/2782480/Stars_of_Prey_VR/)

소개 영상: <https://youtu.be/erzgW3xAFP0>

플레이 영상: <https://www.youtube.com/watch?v=erzgW3xAFP0>





# 보스 AI

## AI 시스템

### FSM을 선택한 배경

- 명확한 행동 패턴: 보스의 행동이 단계별로 **명확하게 정의**되어 있어 FSM의 상태 전환 모델과 완벽 매칭
- 예측 가능한 로직: **정해진 순서와 조건**에 따른 행동 변화로 상태 기반 관리가 최적

### 시스템 구현

- FSM 패턴: 모듈식 보스 AI 구현으로 각 행동을 독립적인 상태로 분리
- 팩토리 패턴: 상태 생성 및 관리 자동화로 새로운 보스 패턴 추가 간소화
- 독립적 행동 단위: 복잡한 보스 패턴을 작은 상태들의 조합으로 구현

### 설계 장점

- **유지보수성**: 각 상태가 독립적이어서 특정 행동 수정 시 다른 부분에 영향 없음
- **확장성**: 새로운 상태 추가만으로 보스 행동 패턴 확장 가능
- **가독성**: 보스의 전체 행동 흐름을 상태 다이어그램으로 직관적 파악



# 보스 IK 구현

## IK 시스템 개요

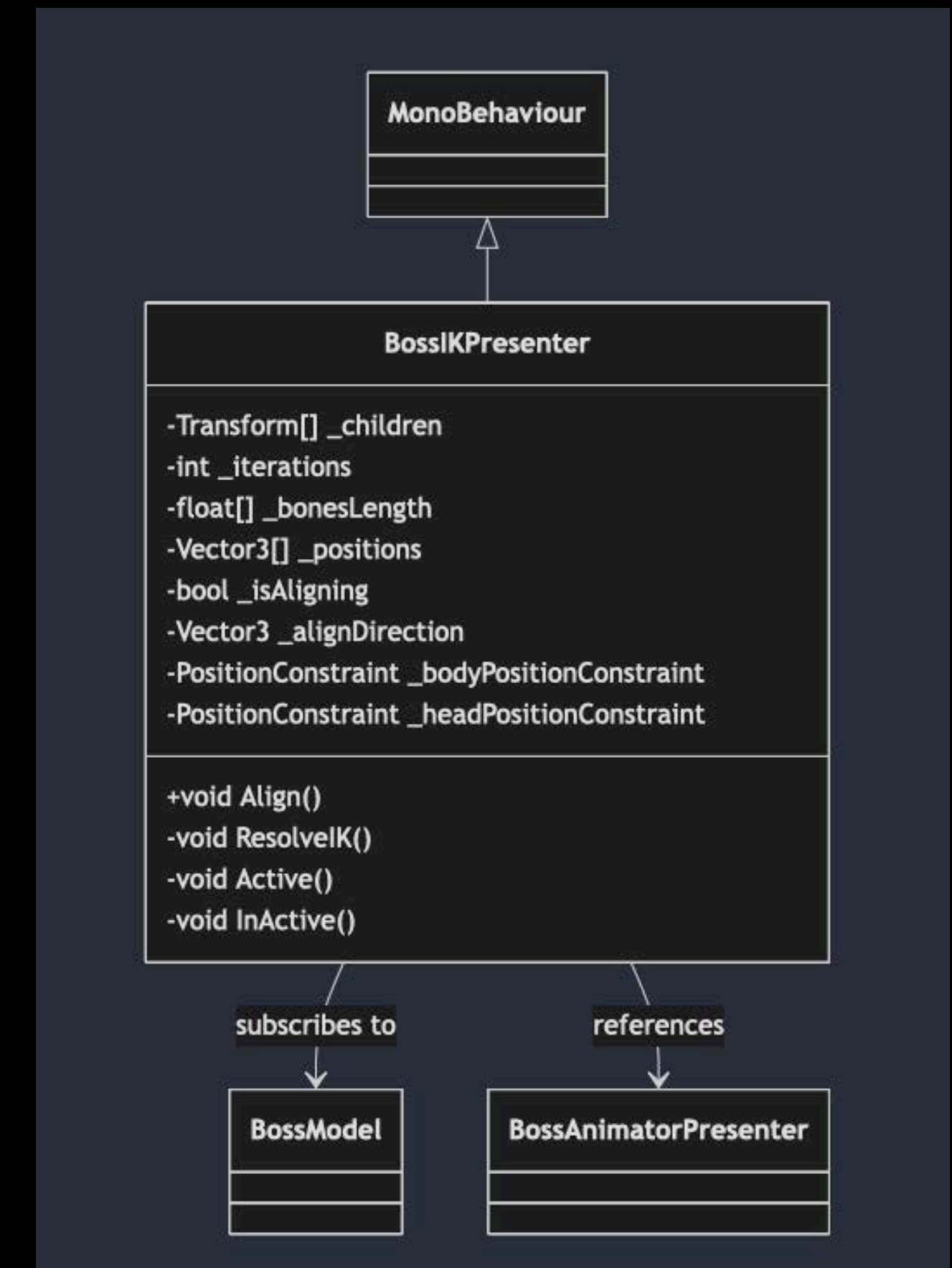
- 뱀형 보스의 자연스러운 움직임 구현
- 뼈대 간 물리적 연결성 유지
- 다양한 방향 전환 및 공격 모션 지원

## 핵심 알고리즘

- 반복적 위치 계산(**FABRIK** 기반)
- **뼈대 길이 제약** 조건 유지
- 루트 노드부터 끝단까지 위치 전파

## 기술적 특징

- 보스 상태에 따른 IK 활성화/비활성화
- 방향 정렬 기능으로 급격한 회전 구현
- 포지션 컨스트레인트와 연동된 머리 움직임



영상: <https://youtu.be/vNLis-0sWRM>

코드: <https://gist.github.com/Labae/2f238b762206385f5eee4b602ad961fa>



# 적기 구현

## 적기 개요

### 핵심 기능:

- 군집 이동 시스템: 다수의 적기가 협조적으로 움직이는 **Flocking 알고리즘** 구현
- BehaviourTree AI: 복잡한 적기 행동 패턴을 노드 기반으로 구성
- BT Editor: BT를 시각적으로 편집할 수 있는 커스텀 에디터 도구

### 군집 이동 구현:

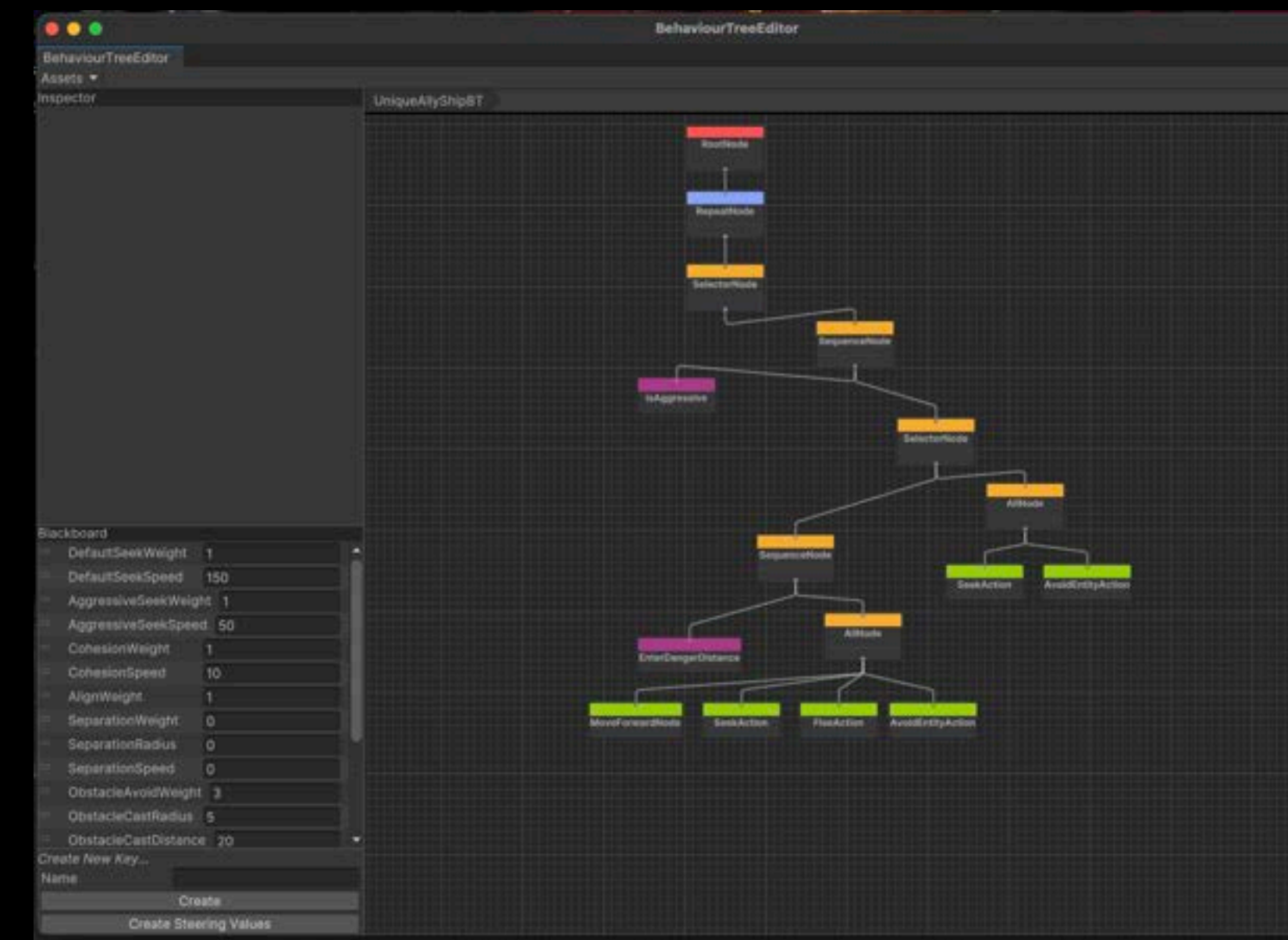
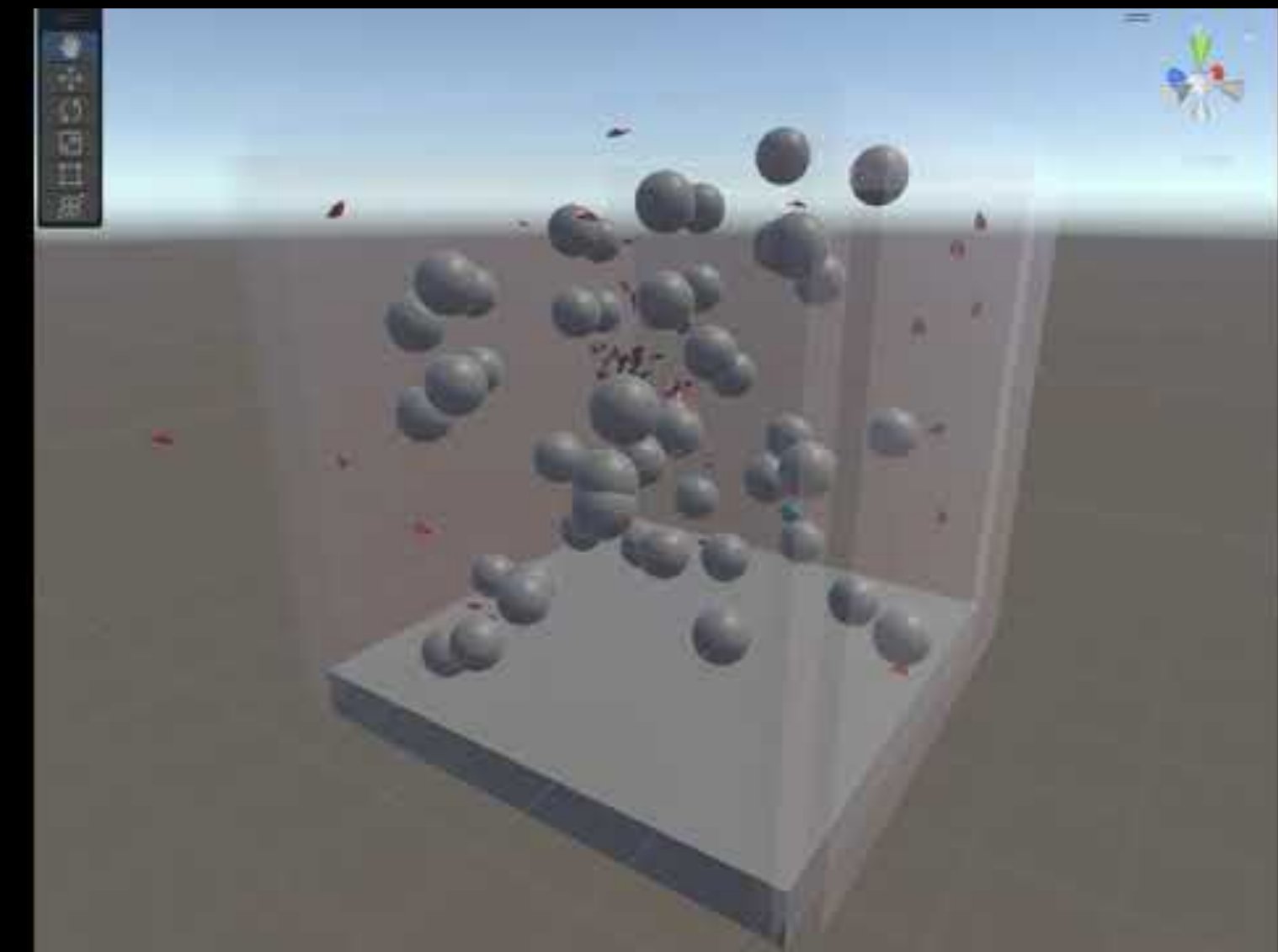
- 응집력: 무리 중심으로 모으려는 행동
- 분리: 개체 간 충돌 방지를 위한 거리 유지
- 정렬: 주변 개체들과 동일한 방향으로 이동
- 장애물 회피: 환경 오브젝트와의 충돌 방지

### BehaviourTree:

- 노드 기반 AI: Composite, Decorator, Action 노드로 구성된 계층적 행동 트리
- 재사용 가능한 행동: 공통 행동 패턴을 서브트리로 모듈화
- 동적 우선순위: 상황에 따른 행동 우선순위 자동 조정

### BT Editor:

- 시각적 편집: 드래그&드롭으로 직관적인 행동 트리 구성
- TheKiwiCoder가 작성한 에디터를 따라 구현



군집이동: <https://gist.github.com/Labae/b312f4eccb7d96d1d569b166db29d49b>

BT 구현: <https://gist.github.com/Labae/5c9217535304f0a1be3f8a87eda3f56f>



# 적기 구현

## 일반 적기 시스템

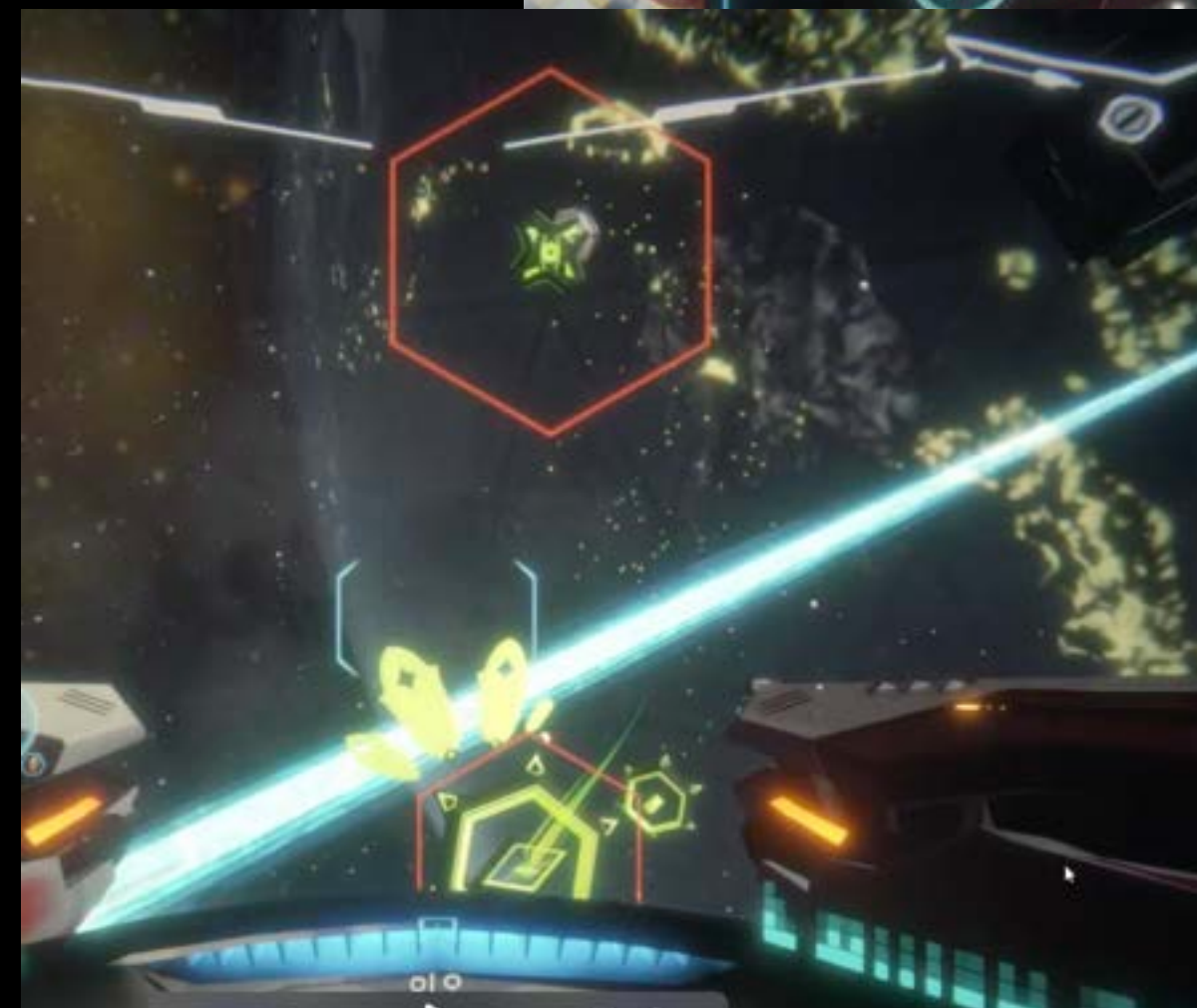
### Chaser 적기

- 그룹 행동 AI 기반의 추적형 적기
- 피로도 시스템을 통한 공격-휴식 패턴 구현
- 공격 타이머와 인내심 타이머로 자연스러운 행동 패턴



### Looker 적기

- 플레이어 시야 외곽에서 출현하여 기습 공격
- 시야각 계산 및 좌표 기하학 기반 출현 위치 선정
- 도착점 기반 정밀 이동 및 공격 패턴



영상: [https://youtu.be/Mn\\_3BQaPjII](https://youtu.be/Mn_3BQaPjII)



# 적기 구현

## 엘리트 적기 시스템

### 특화된 엘리트 타입

- BulletElite: 정밀한 탄환 발사, 빠른 회피 기동
- LaserElite: 지속적 레이저 빔으로 영역 제어
- MissileElite: 다중 크리스탈 무기로 광역 타격

### 고급 전투 패턴

- 적응형 거리 유지: 최적 공격 거리 계산 및 유지
- 충돌 회피 알고리즘: 충돌 예측 및 우회 기동
- 다단계 발사 시퀀싱: 복합 무기 시스템 교차 발사



영상: [https://youtu.be/1gU\\_ARY04GA](https://youtu.be/1gU_ARY04GA)



# 아군기 구현

## 아군기 시스템 개요

### 역할 기반 분류:

- 리더-팔로워: 편대를 이루어 협동 전투 수행, 리더기 중심의 전술적 움직임
- 유니크: 플레이어의 전투 지원 특화
- 포메이션: 게임 후반부 플레이어 주변 포진하여 방어 및 화력 지원

### 핵심 시스템:

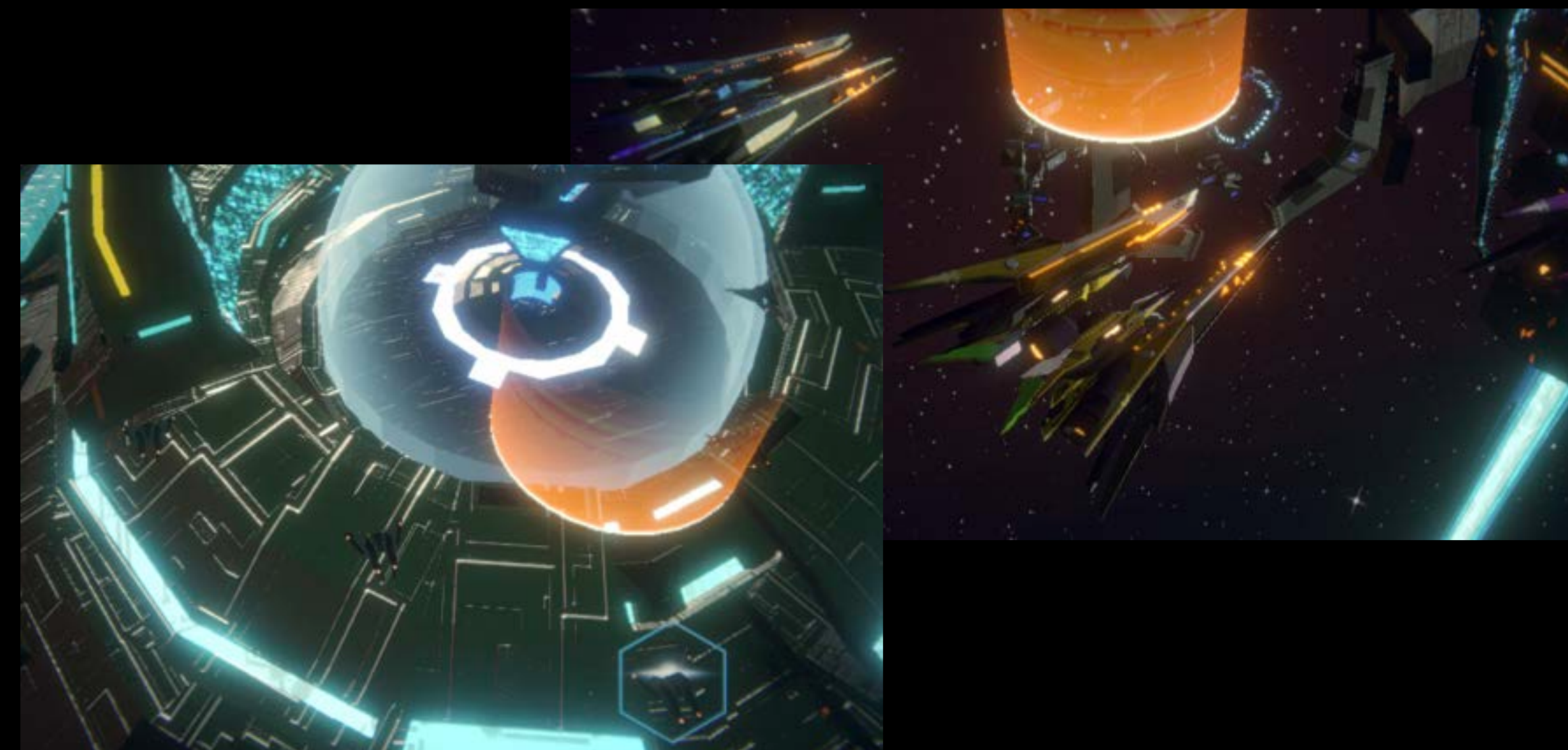
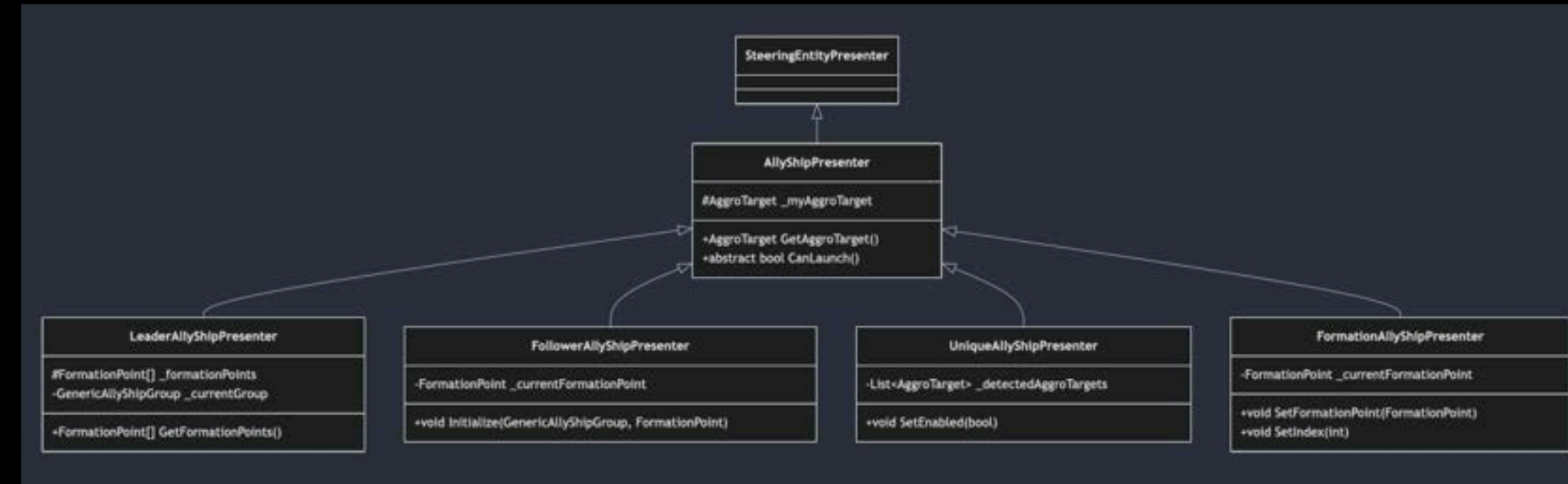
- 공유 타겟팅 시스템: **일관된 공격 대상** 관리로 효율적인 화력 집중
- 다양한 무기 시스템: 역할별 특화된 무기 체계 탑재

### 협동 메커니즘:

- 편대 전술: 리더기의 명령에 따른 조직적 공격 패턴
- 포지셔닝: 전투 상황에 따른 동적 위치 조정

### 플레이어 경험:

- 몰입감 증대: 유니크 아군기와의 대화로 캐릭터 애착 향상
- 시각적 임팩트: 대규모 편대 전투의 연출



코드: <https://gist.github.com/Labae/e84cddd9148a69c4b68a8f9860f2104f>

# 아군기 구현

## 편대 시스템 (리더-팔로워)

### 리더 아군기

- 적 탐지 및 타겟팅을 전담하는 **판단 주체**
- 포메이션 포인트를 제공하여 **팔로워 위치 정의**
- 미사일과 투사체 무기 동시 운용

### 팔로워 아군기

- 리더의 포메이션 포인트에 맞춰 위치 유지
- 리더 공격 시 **자동 연계 공격** 수행
- 리더 보호를 위한 데미지 우선 흡수

### GenericAllyShipGroup

- 리더-팔로워 관계 관리 컨테이너
- 통합 데미지 처리 및 이벤트 전파
- 팔로워 소실 시 리더 보호 메커니즘



영상: <https://youtu.be/OWKXbmws06Q>



# 아군기 구현

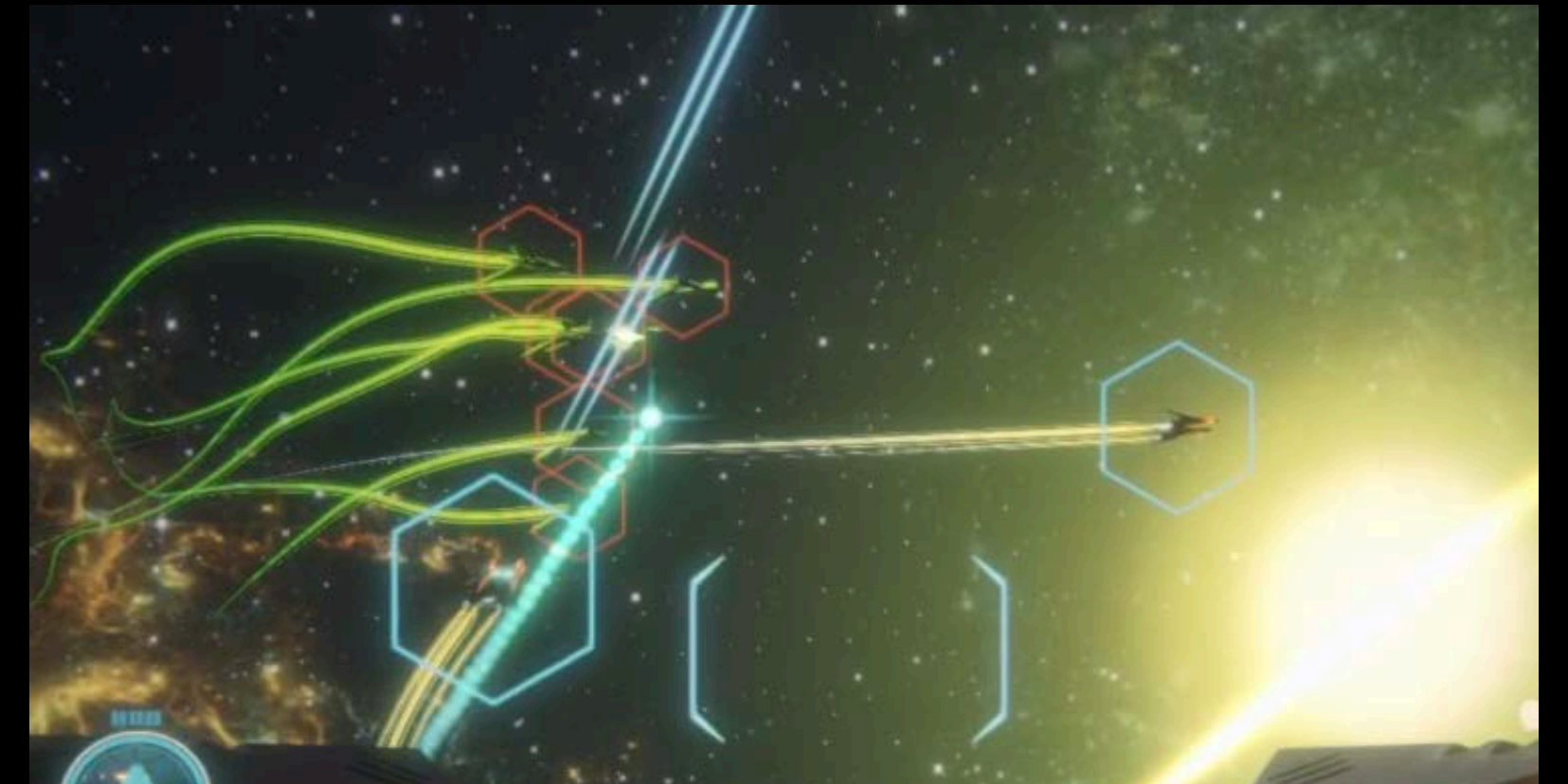
## 특수 유닛

### 유니크 아군기

- 게임 초반부터 등장하여 플레이어 지원
- 독립적 타겟팅과 전투 능력 보유
- 게임 진행에 따른 상태 변화 관리

### 포메이션 아군기

- 게임 3단계에서 플레이어 주변에 포진
- 플레이어 입력에 연동된 공격 패턴
- 보스전 QTE 시 집중 미사일 공격 수행



영상: [https://youtu.be/WJr2T\\_O7rr0](https://youtu.be/WJr2T_O7rr0)



# Outline 구현

## URP 커스텀 렌더 파이프라인 확장

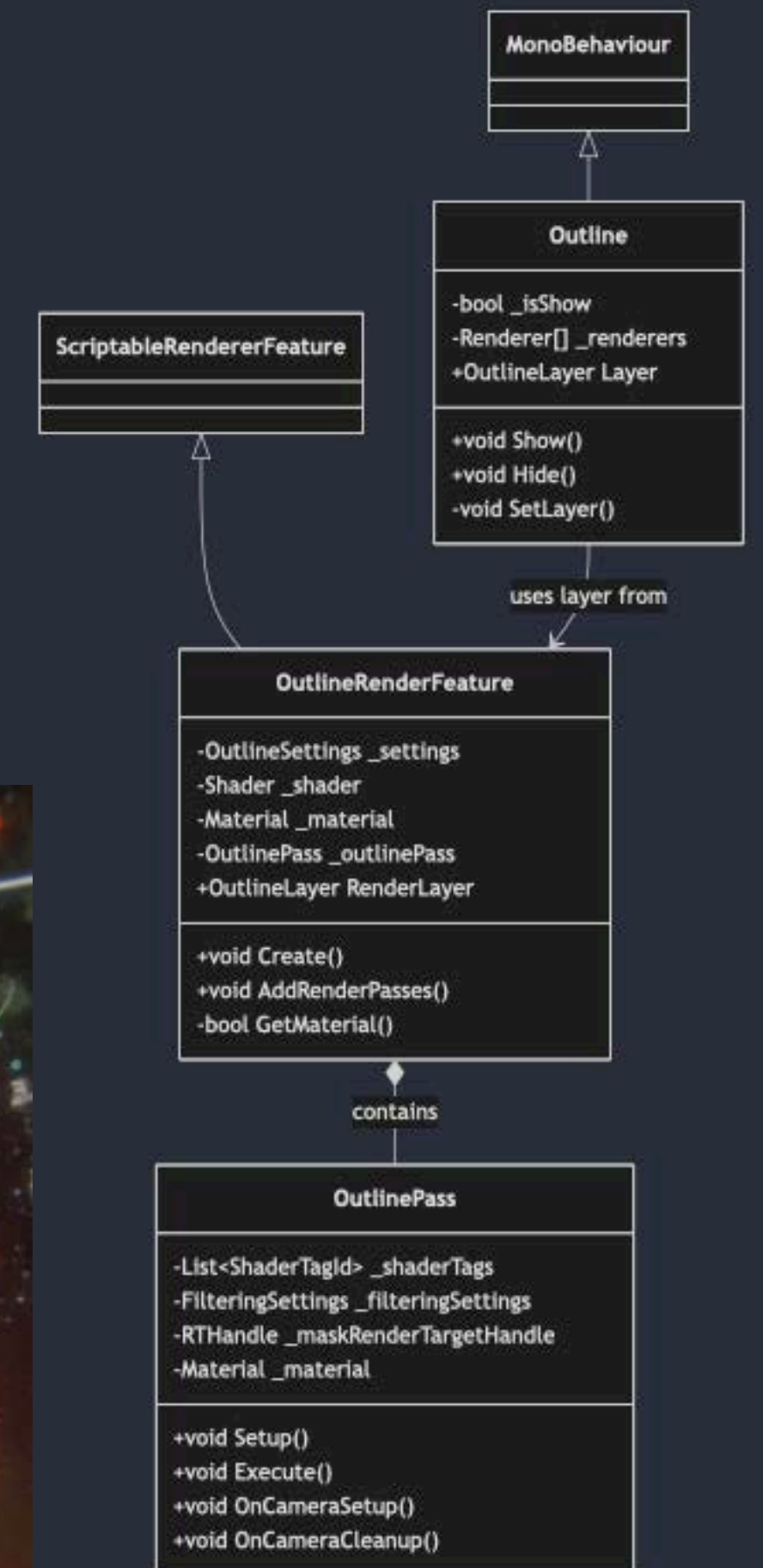
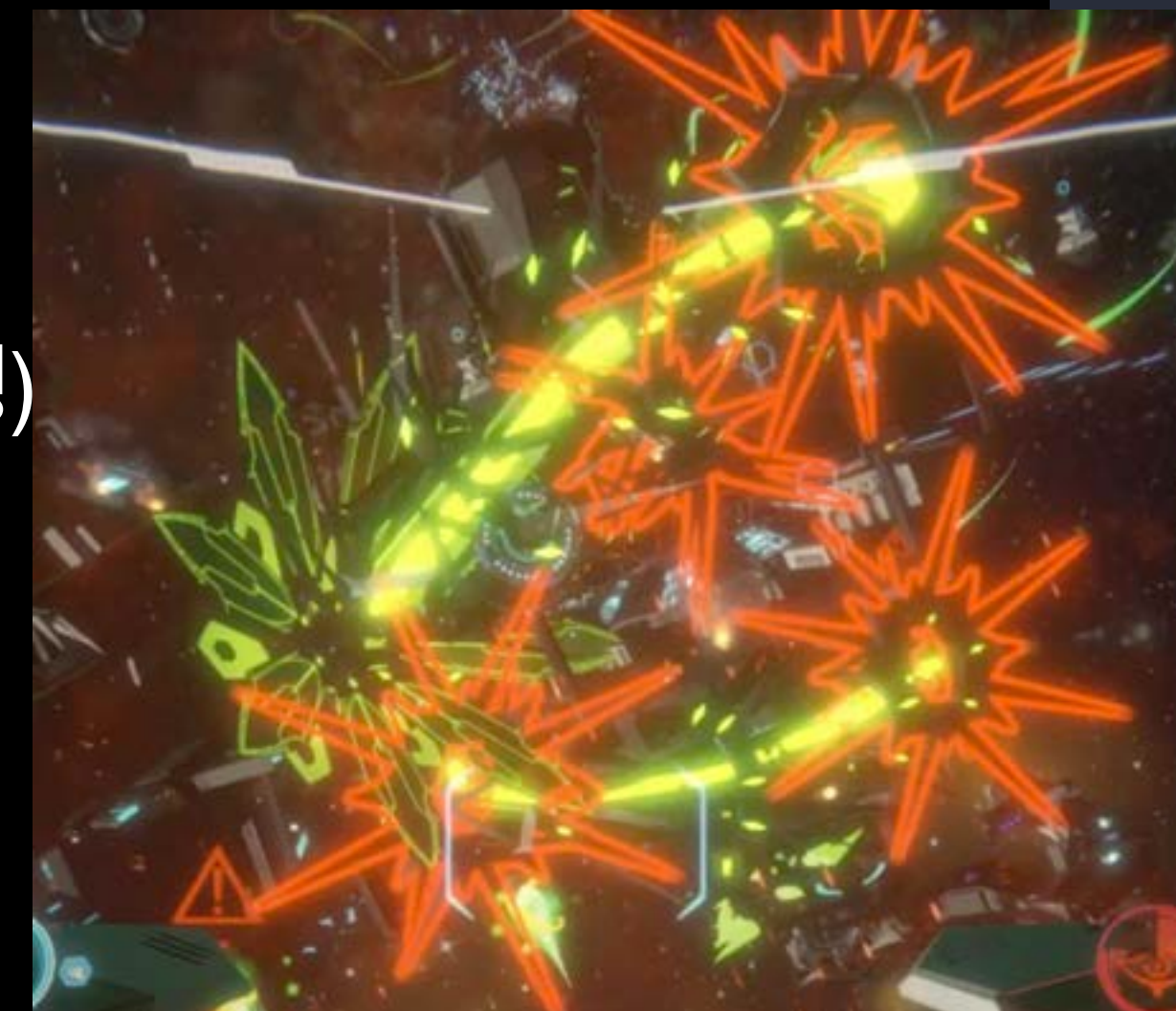
- ScriptableRendererFeature를 활용한 렌더 파이프라인 확장
- 마스크 기반 아웃라인 기법으로 성능 최적화
- 레이어 마스크 시스템으로 선택적 적용

## 핵심 구현 요소

- **OutlineRenderFeature**: 렌더 파이프라인 확장 설정
- **OutlinePass**: 2단계 렌더링(마스크 생성 → 아웃라인 합성)
- **Outline** 컴포넌트: 객체별 아웃라인 제어

## 기능 및 최적화

- 두께, 색상, 샘플링 정밀도 조절 가능
- 렌더링 레이어 마스크로 특정 객체만 처리
- CommandBuffer 풀링으로 메모리 부하 최소화



코드: <https://gist.github.com/Labae/713ea26afe6b1f052b274d5c089e6d48>

# Snake Game

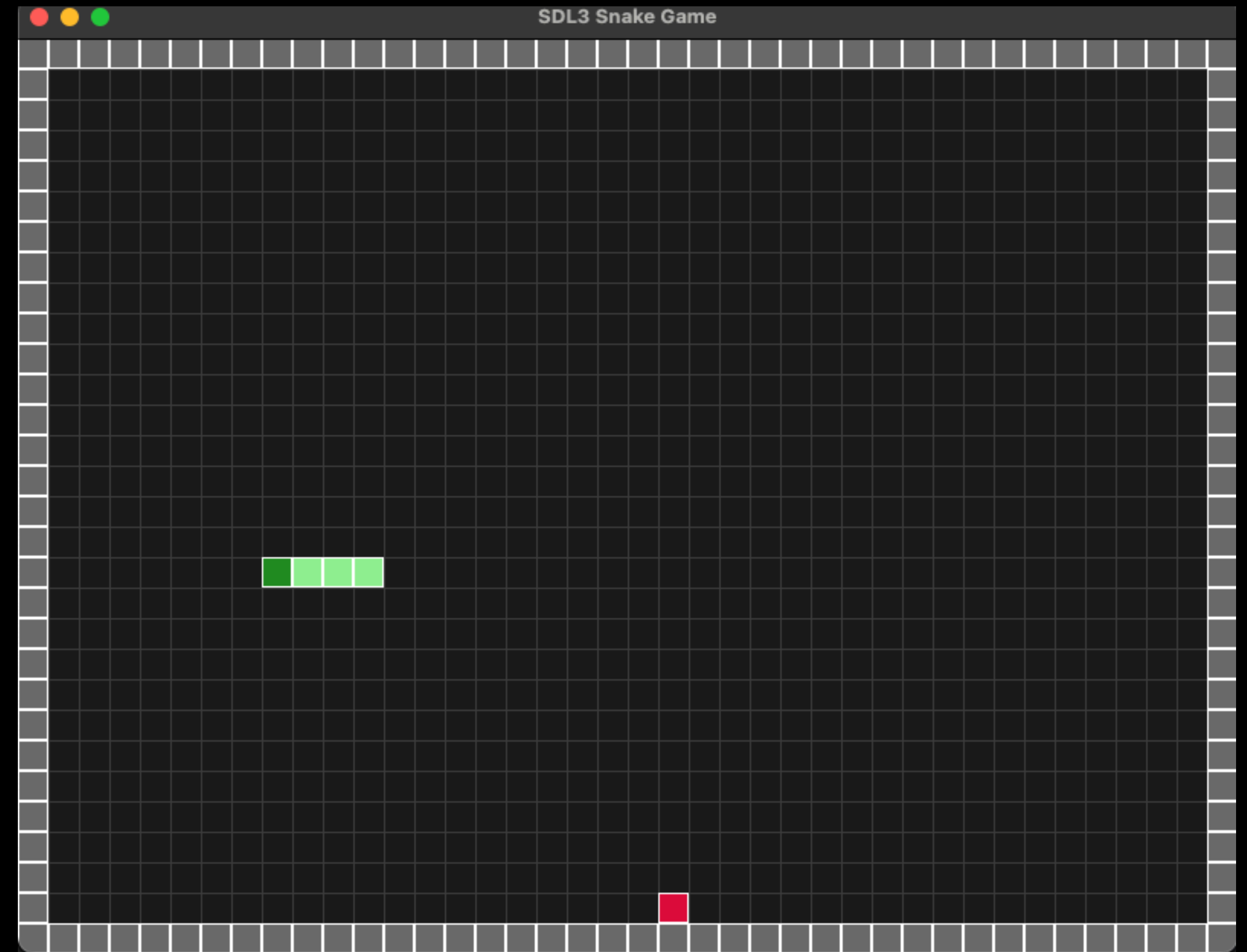
기간 | 2025.05.26 ~ 2025.06.07

인원 | 1명(프로그래밍1)

도구 | C++, SDL3, spdlog, entt

주요 역할:

- Snake 게임 개발
- ECS(entt) 활용



영상: [https://youtu.be/qV\\_ZRlorF80](https://youtu.be/qV_ZRlorF80)

코드: [https://github.com/Labae/SDL3\\_GameCollections](https://github.com/Labae/SDL3_GameCollections)



# 49 Defense

기간 | 2019.11.09 ~ 2019.12.23

인원 | 1명(프로그래밍1)

도구 | Unity / Mobile(Android)

주요 역할:

- CSV Parser
- A\* PathFinding



영상: <https://youtu.be/quYIRD8XWfw>

코드: <https://gist.github.com/Labae/c7f2933412206dbfadecc24bf93b153f>

# Phage

기간 | 2019.08.22 ~ 2019.10.08

인원 | 7명(기획 1, 아트 4, 프로그래밍2)

도구 | Unity / PC

주요 역할:

- Map Editor
- Tile 구현
- UI / Sound 구현



플레이 영상: [https://youtu.be/EHZ\\_LTrWGEO](https://youtu.be/EHZ_LTrWGEO)

맵 에디터 영상: <https://youtu.be/SzWRYQhAZ44>

# Bon Appetit

기간 | 2019.03.10 ~ 2019.06.16

인원 | 6명(기획 1, 아트 3, 프로그래밍2)

도구 | Unity / Mobile(Android)

주요 역할:

- Spine 설정
- 캐릭터 이동
- 상호작용 구현



영상: <https://youtu.be/WTAg9u2Xa8>



# Earth Dive

기간 | 2018.07.10 ~ 2018.10.30

인원 | 7명(기획 2, 아트 3, 프로그래밍2)

도구 | Unity / PC

주요 역할:

- 보스 AI
- 캐릭터 조작
- Json Parser



영상: <https://youtu.be/0WJZ5-IFpxg>

# Running Game

기간 | 2018.06.10 ~ 2018.06.30

인원 | 1명(프로그래밍1)

도구 | Unity / Mobile

주요 역할:

- 캐릭터 이동
- 저장 시스템



영상: <https://youtu.be/jSo2NT9ivYA>