

Portfolio : 드래곤 알 지키기

황민성

게임 소개

1. 게임 장르



모티브가 된 스타크래프트2 유즈맵



포트폴리오 "드래곤 알 지키기"

- 스타크래프트 2의 유즈맵 "랜덤 타워 디펜스"에 영감을 받아 만든 타워 디펜스 게임입니다.
- 평화로운 드래곤 숲에 침략한 인간들로부터 드래곤 알을 지키기 위해 몰려드는 적군들을 막아냅니다.

2. 기술 스택

- Unity 2022.3.30f1, Rider 2024.1.2

3. 개발 기간

- 2024.08.01 ~ 2024.09.09 (약 1개월)

4. 게임 영상 및 코드

- https://www.youtube.com/watch?v=ovLMa_hdL_Y
- <https://github.com/Sicheu/Dragon-Egg-Defense.git>

사용 기술

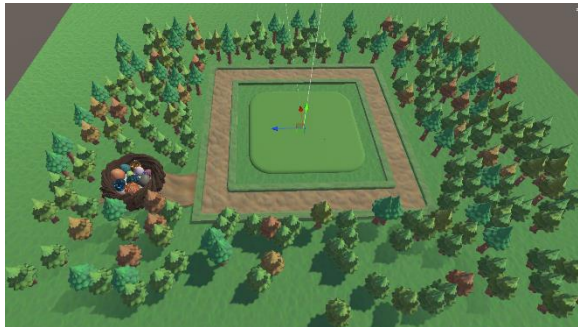
1. 디자인 패턴

싱글톤 (Singleton)	다양한 매니저 클래스들에 사용되었으며 게임 전역에서 상태, 데이터, 리소스 등을 하나의 인스턴스로 관리하고 접근성을 확보하였습니다.
유한 상태 기계 (FSM)	캐릭터들의 상태를 세분화하여 독립적으로 설계해 각 상태의 로직 충돌을 방지하고 특정 상태에서 발생하는 이벤트 처리를 분리하여 복잡한 전투 상황에서도 상태 간 캐릭터의 흐름을 명확히 유지하였습니다.
전략 (Strategy)	모든 캐릭터는 공통 기반 클래스를 상속받아 기본 전투 흐름을 공유하면서도 고유한 스킬, 이동, 공격 방식을 오버라이드하여 다양성을 확보하였습니다.
인터페이스 (Interface)	동일한 역할을 수행하는 다양한 클래스에 인터페이스를 구현함으로써, 클래스 간 결합도를 낮추고 기능을 행위 단위로 명확히 분리하였습니다. 이를 통해 각 컴포넌트가 어떤 기능을 제공하는지 명시적으로 정의할 수 있었으며, 새로운 캐릭터나 오브젝트가 추가되더라도 동일한 인터페이스를 통해 같은 기능을 일관되게 처리할 수 있어 유지보수성과 확장성이 크게 향상되었습니다.

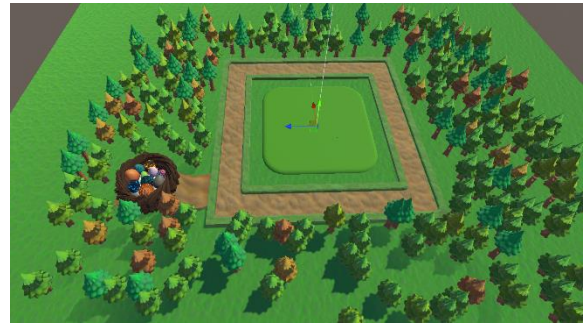
사용 기술

2. 응용 기술

- Post-process를 통해 시각적인 디자인을 향상시켰습니다.

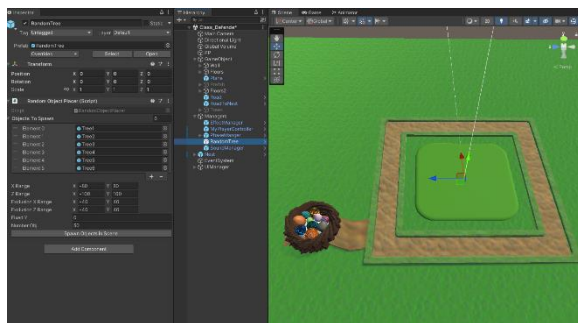


PP 사용 전의 모습

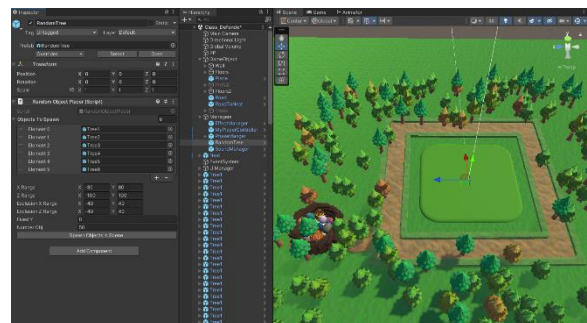


PP 사용 후의 모습

- 맵 디자인을 위해 사용자 지정 범위 내에 오브젝트 프리팹을 자동 배치하는 툴을 만들었습니다. 이를 위해 랜덤 배치 알고리즘을 개발하였고 커스텀 에디터를 통해 에디터를 확장하여 씬 내의 인스펙터로 간단하게 다룰 수 있게 하였습니다.



RandomTree 사용 전의 모습



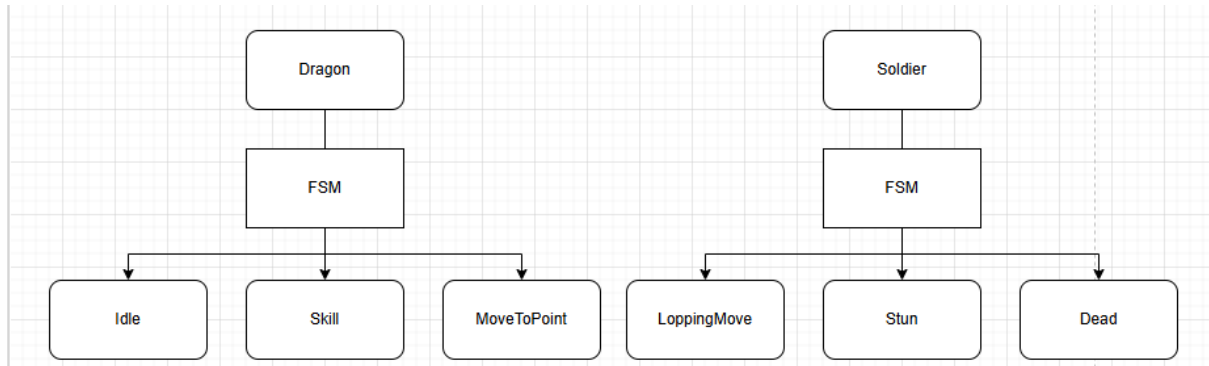
RandomTree 사용 후의 모습

- Ray를 활용하여 유닛을 클릭하거나 드래그 박스를 통해 선택할 수 있는 기능을 구현하였습니다.



기능 구현

1. 드래곤 및 적군의 AI



[FSM]

-모든 캐릭터들의 행동 기반이 되는 상태 기계로 enum을 통해 상태를 지정하고 모든 상태는 상태 진입 시, 상태 진행 중, 상태 퇴장 시 발생하는 함수가 존재하여 상태 변환 및 상태 별로 AI의 행동을 명확하게 함으로 기능을 구현하였습니다.

[Idle]

-드래곤의 대기 상태로 필드에 존재하는 적군들이 객체의 사거리 내로 들어오는지 감지하며 스킬의 쿨타임을 계산합니다.

[Skill]

-감지된 적군에게 객체 타입에 맞는 공격을 가합니다.

[Move To Point]

-마우스 입력에 의해 캐릭터 선택되었을 때 상태로 설정되며 입력에 따라 목적지로 이동합니다.

[Looping Move]

-씬에서 설정된 목적지로 이동하고, 도착했다면 다음 목적지로 이동하게하는 알고리즘으로 구성되어 있습니다.

[Stun]

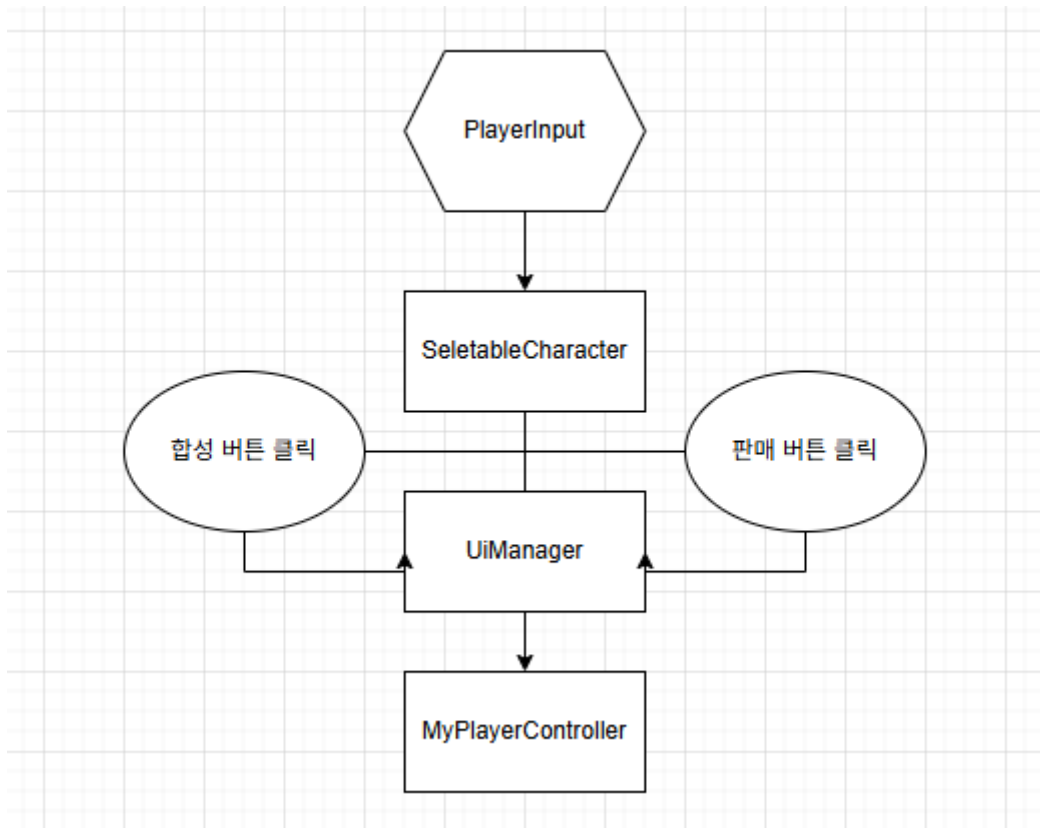
-드래곤의 스킬에 의해 기절했을 때 객체를 잠시 이동불가로 만드는 상태입니다.

[Dead]

-드래곤의 공격에 의해 체력이 0 이 되었을 때 진입하는 상태. 드래곤의 적군 감지 알고리즘에서 제외되며 Dead 애니메이션 재생이 모두 끝났다면 씬에서 사라집니다.

기능 구현

2. 유닛 선택 활성화와 이를 통한 합성 및 판매 등의 상점 시스템



[SeletableCharacter]

-유닛 선택 상태를 관리하고, 선택된 유닛을 리스트에 등록하여 선택되었음을 알립니다.

[UiManager]

-게임 플레이 중 UI에 표시되는 모든 인스펙터를 관리하고 있으며 이 정보를 모든 객체에게 알리며 이벤트를 발생시켜 커맨드 패턴에서의 커맨더와 유사한 역할을 합니다.

-합성 및 판매 버튼이 클릭되었다면 MyPlayerController 에 해당 이벤트가 실행되었음을 알립니다.

[MyPlayerController]

-유닛의 생성, 합성, 판매 등의 모든 기능을 담당하는 스크립터입니다.

-Ui로부터 합성 요청이 들어왔다면 선택된 유닛들의 합성 조건을 확인하는 알고리즘을 통해 상위 유닛을 생성합니다.

-Ui로부터 판매 요청이 들어왔다면 유닛을 해당 유닛의 타입을 확인하여 그에 맞는 골드를 획득하는 알고리즘을 실행시킵니다.