RTS류 게임 기술 소개서

\* 현재 제작 단계 중입니다.

1. **UI**

1) NGUI 사용

|  |  |
| --- | --- |
| 항 목 | 상 세 |
| 개 요 | 화면 해상도 대응을 용이하게 하기 위해, Scalying Style 설정 및  각 UI들의 Anchor를 관리 |
| 그림 1.1 Scalying Style 설정 | |
| 그림 1.2 Anchor를 관리 그림 1.3 Anchor를 관리 | |

2) Atlas 사용

|  |  |
| --- | --- |
| 항 목 | 상 세 |
| 개 요 | 드로우 콜 관리를 위해 Atlas로 이미지를 묶음 |
| 그림 1.4 Atlas | | |

3) Depth 관리

|  |  |
| --- | --- |
| 항 목 | 상 세 |
| 개 요 | 드로우 콜 관리를 위해 Texture 별로 Depth를 구분 |

4) Animation Curve 사용

|  |  |
| --- | --- |
| 항 목 | 상 세 |
| 개 요 | 애니메이션 속도의 설정에서 직관성을 높이기 위해 사용 |
| 그림 1.5 Animation Curve | |

5) 미니맵 구현

|  |  |
| --- | --- |
| 항 목 | 상 세 |
| 개 요 | - 전용 카메라와 RenderTexture로 구현  - 텍스쳐 전환으로 인한 드로우 콜에 대비해 Depth 설정  - 미니맵을 통한 이동 가능 |
| 그림 1.6 미니맵 그림 1.7 | |
| * 그림 1.7의 RenderTexture\_Minimap\_Depth\_700이 미니맵이며 아틀라스를 모두 그린 후 호출 | |

1. **전장의 안개**

|  |  |
| --- | --- |
| 항 목 | 상 세 |
| 개 요 | - 2개의 RenderTexture, 카메라를 사용  - 특정 Color값을 제거하는 방식으로 구현 |
| 그림 2.1 전장의 안개    그림 2.2 지울 Color가 표시된 Texture | |

1. **카메라 움직임**

|  |  |
| --- | --- |
| 항 목 | 상 세 |
| 일반 터치 움직임 구현 | - 터치(현재 마우스)가 움직인 거리만큼 카레라를 움직임 |
| 슬라이드 움직임 구현 | - 터치 종료 후 슬라이드 효과를 주어 카메라를 움직임 |

1. **애니메이션**

|  |  |
| --- | --- |
| 항 목 | 상 세 |
| 레거시 애니메이션 | - 몬스터의 애니메이션에 구현  - 일시 정지 효과 구현 |
| 메카님 애니메이션 | - 유닛에 메카님 애니메이션 구현  - 일시 정지 효과 구현 |
| 그림 4.1 레거시 애니메이션    그림 4.2 메카님 애니메이션 | |

1. **길찾기 구현**

|  |  |
| --- | --- |
| 항 목 | 상 세 |
| 개 요 | – Nav Mesh Agent와 지형 Bake로 구현 |
| 비정적 구조물(건물) 장애물화 | – Nav Mesh Obstacle로 비정적 구조물에 대응 |
| 그림 5.1 지형 Bake 그림 5.2 Nav Mesh Obstacle | |

1. **체력, 타이머, 타워 클래스 등을 스크립트로 구현**

1) 체력 클래스

|  |  |
| --- | --- |
| 항 목 | 상 세 |
| 개 요 | - 유닛, 건물에 포함될 체력 클래스 구현 |
| Has A 관계 | - 체력바 재사용성을 높이기 위해 체력바에서 체력을 독립 |
| 캡슐화 | - 변수의 직접 접근을 막음 |

2) 타이머

|  |  |
| --- | --- |
| 항 목 | 상 세 |
| 개 요 | - 체크할 시간 값만 대입하면 편리하게 사용할 수 있는 타이머 클래스 구현 |
| 캡슐화 | - 변수의 직접 접근을 막음 |

3) 타워

|  |  |
| --- | --- |
| 항 목 | 상 세 |
| 개 요 | - 적을 공격하는 건물인 타워의 움직임을 관리하는 클래스 |
| 포탄 관리 | - 포탄 재사용성을 높이기 위해 별도의 포탄 관리 클래스 구현  - 포탄의 움직임은 베지어 곡선으로 구현 |

6. **오브젝트 풀을 이용한 오브젝트 관리**

|  |  |
| --- | --- |
| 항 목 | 상 세 |
| 개 요 | - 건물, ParticleSystem, 유닛, 포탄 등을 풀로 관리 |
| 특 징 | - 각 오브젝트들의 특징에 맞게 List, Stack, 그리고 배열로 관리 |
| 그림 6.1 포탄 풀 그림 6.2 건물 풀 그림 6.3 ParticleSystem 풀 | |