**Kill The King API 문서**

문서 버전 1.0ver

작성자 : 창원대학교 컴퓨터공학과 손상훈

이메일 : [as\_hy@naver.com](mailto:as_hy@naver.com)

<들어가기 앞서>

해당 문서는 ‘Kill The king’의 전반적인 클래스의 동작 및 변수들에 대하여 서술하고 있습니다.

해당 클래스들의 수정 가능한 메서드나 멤버 변수들을 위주로 서술하고 있습니다.

해당 문서를 통하여 동작에 대한 이해를 돕거나 혹은 해당 동작들을 수정 및 보완하는데 도움을 주는 것을 목표로 작성되었습니다.

목차

[유닛/건물 스크립트 API 6](#_Toc103820814)

[objBasicFunction 6](#_Toc103820815)

[클래스 멤버 변수 6](#_Toc103820816)

[**클래스 메서드** 7](#_Toc103820817)

[unit 8](#_Toc103820818)

[클래스 멤버 변수 8](#_Toc103820819)

[클래스 메서드 11](#_Toc103820820)

[thiefScript 17](#_Toc103820821)

[클래스 멤버 변수 17](#_Toc103820822)

[클래스 메서드 17](#_Toc103820823)

[wildUnit 19](#_Toc103820824)

[클래스 멤버 변수 19](#_Toc103820825)

[클래스 메서드 19](#_Toc103820826)

[struct 21](#_Toc103820827)

[클래스 멤버 변수 21](#_Toc103820828)

[클래스 메서드 21](#_Toc103820829)

[HomeScript 23](#_Toc103820830)

[클래스 맴버 변수 23](#_Toc103820831)

[클래스 메서드 23](#_Toc103820832)

[PortalScript 24](#_Toc103820833)

[클래스 멤버변수 24](#_Toc103820834)

[클래스 메서드 24](#_Toc103820835)

[TableScript 25](#_Toc103820836)

[클래스 멤버 변수 25](#_Toc103820837)

[클래스 메서드 25](#_Toc103820838)

[IllusionTableScript 26](#_Toc103820839)

[클래스 멤버 변수 26](#_Toc103820840)

[클래스 메서드 26](#_Toc103820841)

[resource 27](#_Toc103820842)

[클래스 멤버 변수 27](#_Toc103820843)

[클래스 메서드 27](#_Toc103820844)

[IllusionScript 29](#_Toc103820845)

[클래스 멤버 변수 29](#_Toc103820846)

[클래스 메소드 29](#_Toc103820847)

[unitBuff 31](#_Toc103820848)

[클래스 멤버변수 31](#_Toc103820849)

[클래스 메서드 31](#_Toc103820850)

[Manager 스크립트 API 32](#_Toc103820851)

[manager 32](#_Toc103820852)

[게임 클라이언트 리소스 32](#_Toc103820853)

[UI 상호작용 33](#_Toc103820854)

[마우스 이벤트 핸들러 35](#_Toc103820855)

[키보드 이벤트 핸들러 39](#_Toc103820856)

[길 찾기 스크립트 API 40](#_Toc103820857)

[PathFinder 40](#_Toc103820858)

[클래스 멤버 변수 40](#_Toc103820859)

[클래스 메서드 40](#_Toc103820860)

[checkTile 42](#_Toc103820861)

[클래스 멤버 변수 42](#_Toc103820862)

[클래스 메서드 42](#_Toc103820863)

[Tile 43](#_Toc103820864)

[클래스 멤버 변수 43](#_Toc103820865)

[클래스 메서드 43](#_Toc103820866)

[네트워킹 스크립트 API 44](#_Toc103820867)

[GamePlayer 44](#_Toc103820868)

[클래스 멤버 변수 44](#_Toc103820869)

[클래스 메서드 45](#_Toc103820870)

[PingManager 47](#_Toc103820871)

[클래스 멤버 변수 47](#_Toc103820872)

[클래스 메서드 47](#_Toc103820873)

[UI 스크립트 API 49](#_Toc103820874)

[MainUI 49](#_Toc103820875)

[클래스 멤버 변수 49](#_Toc103820876)

[클래스 메서드 49](#_Toc103820877)

[GameRoomUI 51](#_Toc103820878)

[클래스 메서드 51](#_Toc103820879)

[UIController 53](#_Toc103820880)

[클래스 메서드 53](#_Toc103820881)

[기타 스크립트 API 54](#_Toc103820882)

[LoadingImageScript 54](#_Toc103820883)

# 유닛/건물 스크립트 API

## objBasicFunction

|  |
| --- |
| <요약>  개임 내 기본적인 오브젝트들이 가질 수 있는 상태를 다루기 위한 클래스입니다.  모든 상호작용 가능한 오브젝트들의 클래스는 해당 클래스를 상속받습니다. |

NetworkBehaviour

**->objBasicFucntion**

### 클래스 멤버 변수

#### 오브젝트 정보

|  |  |
| --- | --- |
| public String objName | 오브젝트의 이름입니다. |
| public float defense | 오브젝트의 방어력입니다. |
| public float MaxHP | 오브젝트의 최대 체력 값입니다. |
| public float hp | 오브젝트의 현재 체력 값입니다. |
| public int team | 오브젝트가 속한 팀입니다. |
| Public bool unitSelected | 오브젝트가 선택되었는지에 대한 값입니다.  해당 값을 통하여 오브젝트와 상호작용할 수 있는 상태인지 확인합니다. |

#### 오브젝트 리소스

|  |  |
| --- | --- |
| public GameObject[] selectEffectSet | 해당 유닛이 선택되었을 때 표시할 이펙트입니다.  기본값으로 3의 크기를 가진 배열이며 팀1, 팀2, 중립 순서대로 오브젝트를 할당합니다. |
| public GameObject selectEffect | 현재 유닛에게 생성된 이펙트입니다. |
| public Sprite unitSprite | 현재 유닛을 표시할 초상화입니다. |

### **클래스 메서드**

#### 선택 상태 관련 메서드

|  |  |
| --- | --- |
| public void setUnitSelected(bool selected) | 현재 유닛의 선택 상태를 매개변수의 값으로 변경합니다. |
| public void selectEffectOn() | select effect가 생성되지 않았다면 생성하고, 유닛의 위치로 선택 이펙트를 이동합니다. |
| public void selectEffectOff() | 유닛이 지니고 있는 select effect의 위치를 화면에 보이지 않는 아래쪽 좌표로 이동시킵니다.  \* 생성/삭제의 반복보다 생성된 이펙트를 화면 밖으로 밀어냄으로 오브젝트 관리에 대한 안정성을 증가시키기 위함 |

#### 유닛 상태 관련 메서드

|  |  |
| --- | --- |
| public bool isDead() | 유닛의 죽음을 확인하는 메서드입니다.  유닛의 죽음 이전에 필요한 전처리를 포함하고 있습니다.  해당 클래스에서는 오브젝트에 묶여 있는 select effect를 파괴합니다.  \* 이후 오브젝트에 바인딩 될 객체들이 늘어날 경우 본 메서드를 수정함으로 사망 시 핸들링해야 할 항목들을 쉽게 수정할 수 있게 해줍니다. |

## unit

|  |
| --- |
| <요약>  게임 내 유닛들에 대한 클래스입니다. 공격, 이동, 채취 등 공통적인 동작 구현 되어있습니다. |

NetworkBehaviour

->objBasicFucntion

**->unit**

### 클래스 멤버 변수

#### 유닛 정보

|  |  |
| --- | --- |
| public float moveSpeed | 유닛의 이동 속도입니다. |
| public float attackDamage | 유닛의 공격 데미지입니다. |
| public float attackSpeed | 유닛의 공격 주기입니다. |
| public float attackRange | 유닛의 공격 범위입니다. |
| public int farmmingAmount | 유닛이 자원을 한번 채취할 때의 얻을 수 있는 자원의 양입니다. |
| public float farmmingSpeed | 유닛이 자원을 채취할 때 걸리는 속도입니다. |
| public int unitType | 유닛의 타입입니다. 아래의 unitTypeSet에 정의된 값으로 결정합니다. |
| public float resource1Cost | 해당 유닛을 고용하기 위한 목재의 비용입니다. |
| public float resource2Cost | 해당 유닛을 고용하기 위한 광물의 비용입니다. |
| public float trackingRange | 유닛이 대상을 지정하지 않은 공격명령을 통하여 공격할 유닛을 탐색할 때의 탐지 범위입니다. |
| Protected enum unitTypeSet | 유닛의 타입을 구분하기 위한 열거형 입니다.  \* Melee = 0 : 근거리 유닛  \* Ranged = 1 : 원거리 유닛 |

#### 이펙트 및 사운드

|  |  |
| --- | --- |
| protected AudioSource attackAudioSource | 유닛이 공격 시 재생되는 소리입니다. |
| public GameObject attackEffect | 유닛이 공격 시 표시되는 이펙트입니다. |
| public GameObject deathEffect | 유닛이 사망 시 표시되는 이펙트입니다. |
| public GameObject progressBar | 유닛의 진행도를 타나내는 이펙트 입니다. |
| public GameObject farmingProgressBar | 유닛이 자원을 채취할 때 생성되는 이펙트 입니다. |

#### 유닛 물리 및 렌더링

|  |  |
| --- | --- |
| protected Rigidbody characterRigidbody | 유닛의 물리적 이동을 제어하는 리지드바디 컴포넌트를 참조하는 변수입니다. 회전 및 이동은 Y축에 영향을 받지 않고 XZ축으로만 이동을 하며 가속도 또한 받지 않는 상태입니다. |
| public Animator chacaterAnimator | 유닛이 가진 Animator를 참조하는 변수입니다. |
| Protected Animation anim | 유닛이 가진 Animation을 참조하는 변수입니다. |
| public float attackPoint | 유닛의 동작에서 공격 판정이 들어가는 시점을 나타낸 변수입니다. |

#### 스크립팅 변수

|  |  |
| --- | --- |
| public enum unitStateSet | 유닛이 가질 수 있는 행동들에 대한 정의를 포함하고 있는 열거형 입니다. 유닛에 정의된 상태들은 다음과 같습니다.  \* Stop  \* Move  \* Attack  \* Farmming  \* Tracking  \* Return  \* Death  \* StopSearch  \* MoveSearch |
| public int unitState | 현재 유닛의 행동 상태입니다.  열거형 unitStateSet을 이용합니다. |
| public int beforeBehavior | 유닛의 이전 행동 상태입니다.  특정 행동을 수행한 이후 원래 상태로 돌아가고자 할 때 사용합니다. |
| public Vector3 targetPosition | 현재 유닛이 대상으로 삼는 위치입니다.  이동과 같은 상태에서 사용됩니다. |
| public GameObject target | 현재 유닛이 대상으로 삼는 오브젝트입니다.  공격과 같은 상태에서 사용됩니다. |
| public bool commandable | 현재 유닛이 조작 가능한지에 대한 변수입니다. |
| public bool attackable | 현재 유닛이 공격 가능한지에 대한 변수입니다. |
| bool onChanneling | 현재 유닛이 집중 상태인지에 대한 변수입니다. |
| public bool targetable | 현재 유닛을 타겟으로 삼을 수 있는지에 대한 변수입니다. |
| public Stack<Tile> path | 유닛이 이동할 때의 경로를 담고있는 Stack입니다.  PathFinder로부터 받아옵니다. |
| Tile curTile | 현재 유닛이 위치하고 있는 타일 변수입니다.  유닛의 경로를 설정할 때 PathFinder에게 매개변수로 전달합니다. |
| float timer | 현재 유닛의 타이머입니다. |

\* 유닛 상태 \* 유닛 제어 \* 기타 스크립팅

#### 기타 변수

|  |  |
| --- | --- |
| Public Sprite[] workSprite | 현재 유닛이 가질 수 있는 동작에 대한 Spirte 이미지들입니다.  동작 UI에 표시되는 Sprite들입니다. |
| Float rotationCorrection | 각 모델의 회전을 통일시키기 위한 보정값입니다.  현재는 MKing에 적용되어있습니다. |

### 클래스 메서드

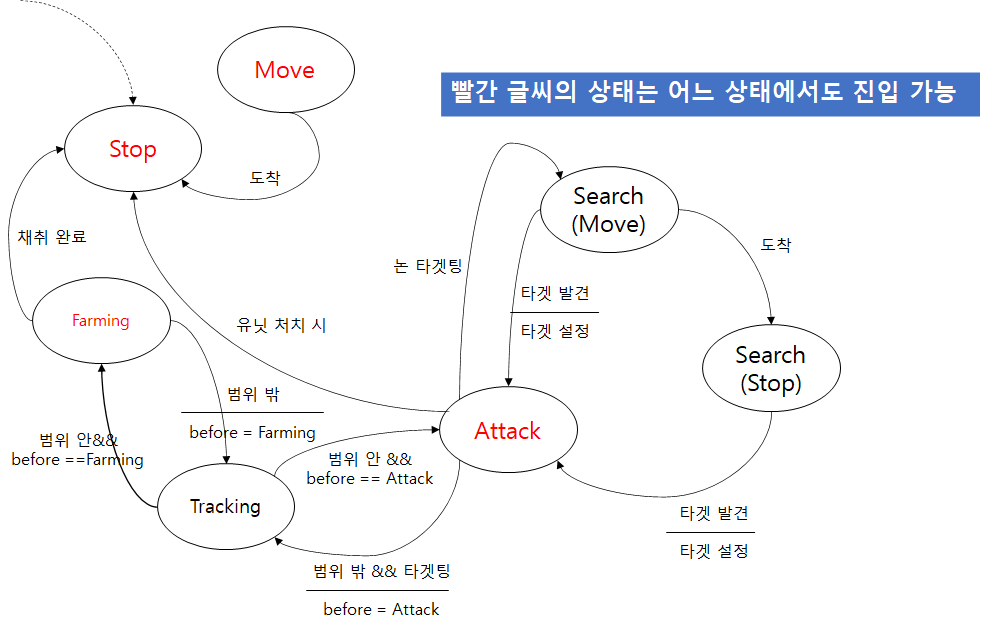
#### 유닛 행동

유닛의 행동은 Behavior로 구분되어 있습니다.

유닛이 가질 수 있는 행동은 unitState 변수로 제어됩니다.

또한 아래에 기술될 ‘set ~ Behavior’ 메서드는 오브젝트에서 해당 메서드를 호출하면 그 행동을 취하도록 작성되었습니다.

유닛은 다음과 같은 오토마타로 이루어져 있습니다.



빨간색으로 표시된 상태는 플레이어가 직접 명령을 함으로 해당 상태로 바로 변경 가능합니다.

또한 Tracking의 상태 전이는 진입하였을 때와 같은 색깔의 방향으로만 이동합니다.

|  |
| --- |
| public virtual void unitBehaviorUpdate() |

유닛들은 unitState 변수를 기준으로 각 행동을 취하는데, 행동에 대한 제어를 해당 메서드에서 실시합니다. 제어는 다음과 같이 이루어집니다.

- 기본적으로 unitState의 상태에 맞는 행동 메서드를 호출합니다.

- 유닛의 commandable 변수가 false 상태일 경우(명령 불가능 상태) 캐릭터의 속도를 0으로 만든 뒤 바로 반환하여 행동 메서드를 호출하지 않습니다.

너무 잦은 호출로 인한 클라이언트 및 네트워킹 성능 저하를 막기 위하여 해당 메서드는 unitBehaviorUpdater() 코루틴을 통하여 0.1초마다 호출됩니다.

|  |
| --- |
| protected void setAnimation() |

유닛의 행동에 맞게 유닛이 취하는 자세를 렌더링 해줘야 합니다. 본 메서드에서는 일반적으로 사용하는 Animator가 아닌 Animation으로 렌더링 하는 경우에 대한 제어 메서드입니다.

- 각 유닛의 unitState의 상태에 맞는 행동을 렌더링 합니다.

(다만 이미 해당 행동을 실행하고 있는 경우에는 실행하지 않습니다. : 유닛의 행동 첫 부분이 계속 반복되기 때문입니다.)

|  |
| --- |
| public void setStopBehavior() |

유닛을 stop상태로 만드는 메서드입니다. 해당 메서드에서는 다음과 같은 처리를 합니다.

- unitState 변수 제어

- 물리적 속도 및 target, path, onChanneling 변수의 초기화(null)

- 유닛의 렌더링 제어

|  |
| --- |
| public void setMoveBehavior() |

유닛을 move상태로 만드는 메서드입니다. 해당 메서드는 두가지 형태로 오버로딩되어 있습니다.

‘Vector3’의 위치값 혹은 ‘GameObject’의 타겟을 매개변수로 전달하여 호출할 수 있습니다.

\* public void setMoveBehavior(GameObject target\_)

- 유닛의 target을 매개변수 target\_으로 설정하고, target\_의 위치값을 baseMoveBehavior 메서드의 매개변수로 전달합니다.

\* public void setMoveBehavior(Vector3 targePosition\_)

- target을 null로 설정하고 매개변수 값을 baseMoveBehavior 메서드의 매개변수로 전달합니다.

|  |
| --- |
| public void baseMoveBehavior(Vector3 targetPosition\_) |

setMoveBehavior 메서드 내에서 호출되는 기본적인 물리 처리를 담당하는 메서드 입니다. 해당 메서드는 다음과 같은 처리를 합니다.

- 유닛의 targetPosition을 매개변수인 targetPosition\_으로 설정합니다.

- targetPosition으로의 path를 PathFinder로부터 받아옵니다.

(이미 path를 받아온 상태 (이동중인 상태) 일 경우 받아오지 않습니다.)

- 이후 path를 따라 이동하며 모든 path에 도달한 경우, 마지막으로 targetPosition으로 이동합니다.

( path는 tile 기반으로 이동됩니다. tile은 정확한 위치값을 취할수 없기에 마지막 tile로 도달 후 정확한 위치값으로 이동합니다.)

- 이동 중에는 setRotation 및 setUnitSpeed를 활용하여 유닛의 회전 및 속도를 조절합니다.

|  |
| --- |
| public void setAttackBehavior(GameObject target\_\_) |

매개변수로 전달받는 target\_을 공격하는 메서드입니다. 해당 메서드는 다음과 같은 처리를 합니다.

- 유닛의 target을 매개변수로 전달받은 target\_으로 설정합니다

- 유닛의 targetPosition을 target의 위치로 설정합니다.

- target이 공격 범위 밖에 있을 경우 beforeBehavior를 attack으로 설정한 뒤 setTrackingBehavior 메서드를 호출하여 target애게로 접근합니다.

- target이 공격 범위 안에 있을 경우 유닛을 멈추고 다음과 같이 공격 처리를 진행합니다.

1) 유닛을 정지, 상태를 Attack으로 설정합니다.

2) 유닛을 공격 모션으로 렌더링합니다.

3) attacking, setAttackDelay 코루틴을 활용하여 HP 처리 및 공격 딜레이를 관리합니다.

\* setAttackDelay Coroutine :attackSpeed시간 만큼 지난 뒤 attackable 변수를 true로 설정합니다.

\* attacking Coroutine : attackPoint 시간 만큼 지난 뒤 유닛의 공격력 만큼 HP 변경 처리합니다.

해당 HP처리는 attacked 메서드를 호출하는 것으로 처리됩니다

(네트워킹 스크립트 API 참고)

|  |
| --- |
| public void setFarmmingBehavior(GameObject target\_\_) |

매개변수로 전달받은 targetResource\_의 자원을 채취하는 메소드 입니다.

해당 자원이 재생성중일때는 해당 메소드는 동작하지 않습니다.

해당 메소드 또한 채취 가능한 범위 내에 도달하지 않을 경우 setTrackingBehavior를 호출하여 해당 위치까지 도달한 다음 farmingProgressBar를 생성하여 자원 채취를 시작합니다.

|  |
| --- |
| public void setTrackingBehavior(GameObject target\_\_) |

어떠한 행동을 할 때 해당 행동을 취할 수 있는 거리까지 다가가기 위하여 사용되는 함수입니다.

setTrackingBehavior는 특정 위치까지 이동한 다음 beforeBehavior 변수의 값으로 유닛의 상태를 돌려놓습니다.

따라서 해당 메서드를 활용하여 새로운 동작을 추가할 때는 해당 메서드를 호출하기 이전에 돌아가고자 하는 상태를 beforeBehavior 변수에 저장해야 합니다.

|  |
| --- |
| public void setSearchBehavior() |

해당 메서드는 두가지 형태로 오버로딩 되어있습니다.

- 매개변수로 타겟위치를 받아 탐색하는 형태 : 해당 위치로 이동하면서 타겟을 탐색합니다.

- 매개변수 없이 서치하는 형태 : 정지상태로 타겟을 탐색합니다.

해당 상태의 유닛은 targetSearching 메서드를 호출하여 탐색 범위 내에서 상대방의 유닛이 있는지 탐색하고 탐색된 적 유닛이 존재한다면 특정한 처리를 합니다.

탐색된 적에게 특정 처리를 하는 메서드는 아래에 기술된 targetSearching 메서드에서 처리를 합니다.

|  |
| --- |
| Void targetSearching() |

유닛 중심에서OverlapSphere를 생성하여 충돌되는 물체가 있는지 지속적으로 탐색합니다.

OverlapSphere의 크기가 유닛의 탐색 범위에 해당합니다.

또한 해당 메서드는 적 유닛을 탐색하고 탐색된 유닛에 대하여 setAttackBehavior를 호출하여 해당 적을 공격하도록 만들어졌습니다.

따라서 해당 부분을 수정한다면 탐색 이후 다른 행동을 취하도록 함수를 수정할 수 있습니다.

#### 물리 도구

|  |
| --- |
| public void setUnitSpeed(Vector3 targetPosition\_) |

유닛을 매개변수 'targetPosition\_’ 방향으로 이동시키기 위하여 Rigidbody의 속도 값을 셋팅합니다.

현재 유닛이 'targetPosition\_’으로의 벡터값을 계산한 뒤 normalized함수를 통하여 정규화 시킵니다. 이후 해당 값을 유닛의 이동속도 값인 moveSpeed를 곱하여 유닛 속도를 설정합니다.

|  |
| --- |
| protected void setRotation(Vector3 targetPosition\_) |

유닛의 회전을 매개변수 ‘targetPosition\_ ‘의 위치를 바라보는 방향으로 회전시킵니다

기본적인 유닛의 방향이 다른 유닛들과 다를 경우 rotationCorrection을 수정하여 회전을 수정할 수 있습니다.

#### 유틸리티 도구

|  |  |
| --- | --- |
| void onTimer() | 유닛의 타이머의 상태를 On으로 둡니다.  onChanneling 변수를 true를 설정, 집중 상태로 만듭니다.  timer 변수의 값을 0으로 초기화 합니다. \* timer는 updqte 함수 내에서 delta time을 활용하여 증가합니다. |
| void offTimer() | 유닛의 타이머 상태를 off로 둡니다.  onChanneling 변수를 false로 설정합니다. |
| protected bool isInRange  (float area, Vector3 myPos, Vector3 opponentPos) | 상대방 유닛의 위치가 나로부터 해당 범위 안에 있는지 확인하는 함수입니다.  공격 범위 내에 타겟이 존재하는지 검사할 때 사용합니다.  \* area : 범위 \* myPos : 기준점 \* opponentPos : 타겟의 위치 |

#### 유닛 수치 반환

|  |  |
| --- | --- |
| public float getAttackUpgradePoint() | 유닛의 현재 공격력을 반환하는 메소드입니다.  Manager로부터 현재 업그레이드 상태를 얻어오며, 유닛의 타입(원거리/근거리)에 따라 반환하는 값이 달라집니다. |
| public float getDefenseUpgradePoint() | 유닛의 현재 방어력을 반환하는 메소드입니다.  Manager로부터 현재 업그레이드 상태를 얻어오며, 유닛의 타입(원거리/근거리)에 따라 반환하는 값이 달라집니다. |

## thiefScript

|  |
| --- |
| <요약>  특수한 유닛인 thief의 동작을 서술한 클래스입니다.  thief는 다음과 같은 동작을 가지고 있습니다.  \*특정 범위(게임 내 짙은 녹색으로 이루어진 지역)에 들어간다면 적이 볼 수 없는 투명한 상태가 됩니다.  \*해당 유닛이 공격을 한번이라도 했다면 다시 투명한 상태가 되는 것은 불가능 합니다. |

NetworkBehaviour

-> objBasicFucntion

-> unit

->thiefScript

### 클래스 멤버 변수

|  |  |
| --- | --- |
| public bool enemyVisible | 적이 해당 유닛을 볼 수 있는지에 대한 변수입니다. |
| bool hitted | 유닛이 공격을 하였는지 |

### 클래스 메서드

|  |
| --- |
| void updateTrans(float uAlpha, Shader uShader) |

해당 유닛의 투명도와 쉐이더를 변경합니다.

Ualpha 값으로 투명도를 조정할 수 있고 uShader를 통하여 투명도를 조절할 수 있는 쉐이더로 현재 유닛의 쉐이더를 변경할 수 있습니다.

|  |
| --- |
| void updateTrans(float uAlpha, Shader uShader) |

해당 유닛의 위치에 따라 투명도를 조절합니다.

짙은 녹색의 구역에서는 상대방이 볼 수 없는 상태,

녹색의 구역에서는 상대방이 약간 투명한 상태지만 볼 수 있는 상태,

옅은 녹색의 구역에서는 상대방이 완전히 볼 수 있는 상태로 구현되어 있습니다.

해당 메서드를 수정하여 해당 유닛이 투명해지는 구역을 변경할 수 있습니다.

## wildUnit

|  |
| --- |
| <요약>  게임 내 중립 혹은 최종 목표인 Mking 유닛들에 대한 클래스입니다.  unit 클래스를 상속받으며 특정 구역 내에서 활동하는 특징을 추가적으로 가집니다. |

NetworkBehaviour

-> objBasicFucntion

-> unit

→wildUnit

### 클래스 멤버 변수

|  |  |
| --- | --- |
| public float unitArea | wildUnit가 스폰된 위치부터 해당 값만큼 떨어진다면 다시 돌아오는 동작을 한다. |
| Vector3 spawnPosition | 해당 유닛이 스폰된 위치 |
| Quaternion spawnRotation | 해당 유닛이 스폰된 뒤 바라보는 방향 |
| GameObject unitStruct | 해당 유닛이 가지는 건물 |

### 클래스 메서드

|  |
| --- |
| public void damaged(GameObject attacker) |

특정 유닛에게 공격받았을 때 그 유닛을 타겟으로 설정하는 메서드

해당 메서드를 통하여 wildUnit은 먼저 공격을 받지 않으면 공격하지 않도록 구현됩니다.

|  |
| --- |
| public override void unitBehaviorUpdate() |

unit클래스의 unitBehaviorUpdate 메서드를 오버라이딩 하여 구현되었습니다.

unit의 해당 메서드 이후 추가적으로 wildUnit이 자신의 범위 내에서 벗어났는지 확인을 하고, 그렇다면 원래 스폰된 위치로 돌아가는 동작을 수행합니다.

|  |
| --- |
| public void returnToArea() |

현재의 모든 타겟을 없애고 원래 유닛이 스폰된 위치로 돌아가도록 만듭니다.

|  |
| --- |
| protected override void Update() |

update문은 부모 클래스의 update를 호출하기 이전에 유닛의 죽음을 먼저 확인합니다.

사망한 유닛이 MKing일 경우 manager에게 Mking의 사망과 함께 게임의 끝을 알립니다.

다른 wildUnit일 경우 해당 유닛 및 포함된 건물을 파괴합니다.

## struct

|  |
| --- |
| <요약>  개임 내 건물들이 가지는 기본적인 동작 및 데이터들을 정의한 클래스입니다.  각 건물은 해당클래스를 상속받아 구현됩니다. |

NetworkBehaviour

->objBasicFucntion

**->Struct**

### 클래스 멤버 변수

건물 정보

|  |  |
| --- | --- |
| public List<workSet> workList | 해당 건물이 가질 수 있는 작업들에 대한 정보입니다. |
| publid Sprite[] workSprite | 해당 건물의 작업들에 대한 스프라이트 이미지입니다. |

#### 이너클래스 workSet

해당 이너클래스는 건물들의 작업들을 정의하고자 정의되었습니다.

다음과 같은 멤버 및 메서드를 가지고 있습니다.

|  |  |
| --- | --- |
| string work | 해당 건물의 작업 이름입니다. |
| float woodCost | 해당 작업에 필요한 목재의 양입니다. |
| float crystalCost | 해당 작업에 필요한 광물의 양입니다. |
| getWork()/getWoodCost()/getCrystalCost | 해당 클래스 각각의 멤버들을 반환합니다. |

### 클래스 메서드

세부 사항들은 각각의 건물들에서 구현이 되어 클래스 메서드는 따로 존재하지 않습니다.

다만 Update문에서 건물이 선택되지 않았을 때는 아무런 동작을 취하지 않는 기본적인 상태가 구현되어있습니다.

## HomeScript

|  |
| --- |
| <요약>  개임 내 건물들이 가지는 기본적인 동작 및 데이터들을 정의한 클래스입니다.  유닛의 스폰 및 건물 소환, 포탈을 활성화를 요청하는 역할을 수행합니다. |

NetworkBehaviour

->objBasicFucntion

->Struct

**->HomeScript**

### 클래스 맴버 변수

|  |  |
| --- | --- |
| Public GameObject[] spawnableUnitSet | 생성 가능한 유닛 목록입니다. |
| bool tableSpawn | 현재 table이 생성되었는지 확인하는 변수입니다. |
| Int activatedCrystalsNumber | 현재 활성화된 포탈의 크리스탈 개수입니다. |

### 클래스 메서드

|  |
| --- |
| bool payForUpgrade(int num\_) |

workList에 존재하는 work들의 자원 비용들을 비교하여 자원이 충분하다면 해당 자원을 소모하고 충분하지 않다면 prompt를 통하여 알림을 보냅니다.

|  |
| --- |
| void eventHandler() |

사용자의 키보드 입력을 처리하는 핸들러입니다.

KeyCode를 수정하여 해당 키입력의 처리를 변경할 수 있습니다.

## PortalScript

|  |
| --- |
| <요약>  Struct클래스를 상속받아 구현된 Portal의 동작을 서술한 클래스입니다.  유닛을 특정 위치로 이동시키는 역할을 수행합니다. |

NetworkBehaviour

->objBasicFucntion

->Struct

**->PortalScript**

### 클래스 멤버변수

|  |  |
| --- | --- |
| bool activate | 해당 포탈의 활성화 여부를 나타냅니다. |
| Vector3 warpPosition | 해당 포탈을 이용하여 이동할 위치를 나타냅니다. |
| public GameObject portalEffect | 유닛 이동 시 포탈에서 발생하는 이펙트입니다. |
| public GameObject unitEffect | 유닛 이동 시 유닛의 도착 짖머에서 발생하는 이펙트입니다. |

### 클래스 메서드

|  |
| --- |
| void OnTriggerStay(Collider other) |

해당 유닛이 포탈과 닿았을 경우 이동 처리입니다.

특정 유닛을 warpPosition으로 이동시키고, 유닛을 정지시킵니다.

## TableScript

|  |
| --- |
| <요약>  Struct 클래스를 상속받아 구현된 Table의 동작을 서술하는 클래스  유닛의 업그레이드 및 IllusionTable 생성을 요청하는 역할을 하는 건물입니다. |

NetworkBehaviour

->objBasicFucntion

->Struct

**->TableScript**

### 클래스 멤버 변수

|  |  |
| --- | --- |
| bool isTableSpawn | IllusionTable이 성생되었는지 확인하는 변수입니다 |

### 클래스 메서드

|  |
| --- |
| bool payForUpgrade(int num\_) |

workList에 존재하는 work들의 자원 비용들을 비교하여 자원이 충분하다면 해당 자원을 소모하고 충분하지 않다면 prompt를 통하여 알림을 보냅니다.

|  |
| --- |
| void eventhandler() |

각 키보드 버튼에 해당하는 동작들을 수행하는 키보드 이벤트 핸들러입니다.

KeyCode를 수정하여 해당 동작을 발생시키는 버튼을 변경할 수 있습니다.

## IllusionTableScript

|  |
| --- |
| <요약>  Struct클래스를 상속받아 구현된 IllusionTable의 동작을 서술한 클래스입니다.  유닛들에게 이롭거나 해로운 영역을 생성하도록 요청하는 기능을 수행합니다. |

NetworkBehaviour

->objBasicFucntion

->Struct

**->IllusionTableScript**

### 클래스 멤버 변수

|  |  |
| --- | --- |
| int targetIllusion | 현재 스폰하고자 하는 오브젝트의 번호입니다.  해당 값을 이용하여 스폰하고자 하는 오브젝트를 결정합니다. |

### 클래스 메서드

|  |
| --- |
| void eventHandler() |

각 키보드의 입력에 따라 스폰 준비를 처리합니다.

KeyCode의 값을 변경하여 스폰 준비에 필요한 키보드 입력을 변경할 수 있다.

아래의 readyToSpawn메서드를 활용합니다.

|  |
| --- |
| void readyToSpawn(int iluusionNumber) |

manager의 커서 상태를 변경하여 오브젝트를 스폰할 준비를 합니다.

또한 해당 건물의 targetIllusion을 매개변수의 상태로 변경합니다.

## resource

|  |
| --- |
| <요약>  게임 내 존재하는 자원의 동작을 서술한 클래스입니다.  자원량의 변화 및 자원의 재생성 등이 구현되어있습니다. |

NetworkBehaviour

->objBasicFucntion

->resource

### 클래스 멤버 변수

|  |  |
| --- | --- |
| public int maxResource | 자원의 최대 생성치입니다. |
| public int leftResource | 현재 남은 자원량입니다. |
| public int resourceType | 자원의 종류를 나타내는 변수입니다. |
| public float createTime | 자원의 재생성 시간입니다. |
| public bool onCreating | 자원이 재 생성중인지에 대한 변수입니다. |
| public GameObject createEffect | 자원이 재 생성될 때 나타나는 이펙트입니다. |

### 클래스 메서드

|  |
| --- |
| void setCreateRender() |

현재 오브젝트의 렌더러를 변경하고 반 투명하게 만듭니다.

|  |
| --- |
| void setNormalRender() |

현재 오브젝트의 렌더러를 변경하고 선명하게 만듭니다.

|  |
| --- |
| void createResource() |

현재 오브젝트의 상태를 재생성 상태로 만듭니다.

|  |
| --- |
| public int farmmed(int farmmingAmount) |

해당 자원이 채취 당할 때 호출됩니다.

해당 자원의 남은 자원량을 farmmingAmount만큼 감소시킵니다.

|  |
| --- |
| protected override void Update() |

update문에서는 두 가지 동작을 수행합니다.

- leftResource가 0 이하로 내려간다면 createResource 메서드를 호출하여 재생성 상태에 돌입합니다.

- 재 생성중일 때 재생성 시간이 지나면 자원량을 최대 생성량으로 증가시키고 보통 상태로 되돌립니다.

## IllusionScript

|  |
| --- |
| <요약>  IllusionTable에서 생성되는 영역 오브젝트들의 동작들을 정의한 클래스입니다.  해당 영역에 진입한 유닛들에게 버프 형태의 상태변화를 추가시킵니다. |

MonoBehaviour

->IllusionScript

### 클래스 멤버 변수

|  |  |
| --- | --- |
| public float destoryTime | 해당 영역 오브젝트가 사라질 때까지 걸리는 시간입니다. |
| const float effectCycle | 해당 영역 오브젝트에서 버프 형태의 상태변화가 추가되는 틱의 시간 간격입니다. |
| const float healMount | 체력을 회복하는 영역에서의 주기당 회복량입니다. |
| public int illusionType | 현재 영역 오브젝트의 종류를 나타내는 변수입니다. |

### 클래스 메소드

|  |
| --- |
| void OnTriggerStay(Collider other) |

현재 영역에 존재하는 충돌체들(other)에 대한 핸들링입니다.

illusionType에 따라 해당 유닛에게 디버프를 추가하거나, 혹은 해당 유닛의 HP를 변화시킵니다.

|  |
| --- |
| void healUnit(unit[] units) |

매개변수로 받은 유닛들에 대하여 체력회복을 시킵니다.

|  |
| --- |
| void setIllusion(int number\_, unit obj\_) |

매개변수의 obj\_ 오브젝트에게 number\_에 해당하는 버프를 추가합니다.

- 만약 해당 버프를 오브젝트가 이미 가지고 있다면 해당 효과의 남은 시간을 초기화합니다.

- 해당 버프가 없다면 유닛에게 새로운 buff컴포넌트를 추가합니다.

## unitBuff

|  |
| --- |
| <요약>  유닛들에게 이롭거나 해로운 효과를 주는 특정 버프들의 기능을 정의한 클래스 |

MonoBehaviour

->unitBuff

### 클래스 멤버변수

|  |  |
| --- | --- |
| unit targetUnit | 적용되는 타겟 유닛을 참조하는 변수입니다. |
| int buffNumber | 버프의 종류입니다. |
| float leftTime | 현재 버프가 사라질 때까지의 남은 시간입니다. |
| float originSpeed | 유닛의 이동속도를 원상복구하기 위한 유닛의 기존 속도값입니다. |

### 클래스 메서드

|  |
| --- |
| public void initUnitBuff(int buffNumber\_, float leftTime\_) |

처음 유닛에게 버프/디버프가 생성될 때 호출하는 메서드입니다.

매개변수로 전달받은 상태로 현재의 버프 상태 변수를 초기화합니다.

이후 setEffect메서드를 호출하여 상태를 적용시킵니다.

|  |
| --- |
| void setEffect() |

현재 클래스의 buffNumber값에 따라 targetUnit의 상태를 변경시킵니다.

|  |
| --- |
| void releaseBuff() |

targetUnit의 상태를 원래 상태로 복구합니다.

# Manager 스크립트 API

## manager

|  |
| --- |
| <요약>  개임 내 기본적인 오브젝트들이 가질 수 있는 상태를 다루기 위한 클래스  모든 상호작용 가능한 오브젝트들의 클래스는 해당 클래스를 상속받음. |

MonoBehaviour

**->manager**

### 게임 클라이언트 리소스

#### 클래스 멤버 변수

|  |  |
| --- | --- |
| public GameObject LoadImage | 로딩 시 나타나는 Image입니다. |
| public GameObject LoseImage | 게임 패배 시 표시되는 Image입니다. |
| public GameObject WinImage | 게임 승리 시 표시되는 Image입니다. |
| int screenHeight | 게임 화면의 높이입니다. 1080으로 셋팅 되어있습니다. |
| int screenWidth | 게임 화면의 높이입니다. 1920으로 셋팅 되어있습니다. |
| public Texture2D[] cursorSet | 게임 내에서 활용되는 커서입니다. |
| public GameObject MinimapCameraArea | 미니맵에서 현재 보고 있는 영역을 표시하는 오브젝트입니다. |

### UI 상호작용

|  |  |
| --- | --- |
| public Text resource1Text | 화면 우측 상단에서 가지고 있는 목재를 표시할 때 사용하는 텍스트입니다. |
| public Text resource2Text | 화면 우측 상단에서 가지고 있는 광물을 표시할 때 사용하는 텍스트입니다. |
| public GameObject[] selectListUI | selectUI에 다수 선택 시 사용되는 UI객체들입니다. |
| public GameObject[] singleSelectListUI | selectUI에 단독 선택 시 사용되는 UI객체들입니다. |
| public Sprite emptySprite; | UI에서 빈 메뉴를 나타내는 Sprite입니다. |
| public GameObject prompt | 게임 클라이언트가 보내는 메세지를 표시하는 오브젝트입니다. |

#### 클래스 멤버 변수

#### 클래스 메서드

|  |
| --- |
| Public void sendPrompt(string message) / public void setOffPrompt() |

게임 클라이언트의 prompt로 유저에게 메세지를 보낼 때 사용하는 메서드입니다. 메세지를 보낸 뒤 타이머를 실행시킵니다.

setOffPrompt는 텍스트를 지우고 타이머 상태를 false로 만듭니다.

update에서 메세지를 전송한 뒤 1.5초 뒤에 setOffPrompt를 호출하도록 되어있습니다.

|  |
| --- |
| void singleSelectListUiOn() / void singleSelectListUiOff() |

하나의 오브젝트가 선택되었을 때 정보를 표시해주는 UI를 켜고 끄는 메서드입니다.

표현할 수 있는 정보는 5개가 존재하며 해당 메서드를 수정하여 나타내는 값을 변경하거나 더 많은 정보를 나타낼 수 있습니다.

|  |
| --- |
| clearSelectList() |

현재 선택UI에 선택된 정보들을 모두 선택 해제하고 아무것도 선택되지 않은 상태로 만드는 메서드입니다.

|  |
| --- |
| updateMultiSelectedList() |

선택 UI를 갱신하는 메서드입니다.

자원, 유닛, 건물 등의 정보를 실시간으로 갱신하기 위하여 지속적으로 호출됩니다.

|  |
| --- |
| void updateRightMenu() |

현재 선택된 오브젝트의 종류에 따라 행동 메뉴의 표시를 설정합니다.

유닛 혹은 건물의 경우 각각 workSprite, workList를 가지고 있기 때문에 해당 오브젝트가 가진 정보를 활용하여 행동 메뉴를 표시합니다.

추가적인 행동이나 새로운 종류의 오브젝트들이 생성되면 해당 메서드를 수정함으로 행동 메뉴의 표시를 수정할 수 있습니다.

### 마우스 이벤트 핸들러

#### 드래그 이벤트

##### 클래스 멤버 변수

|  |  |
| --- | --- |
| bool dragFlag | 현재 마우스가 드래그 동작 중인지 판별하는 변수입니다. |
| Vector3 dragStartPos | 마우스 드래그 동작의 시작점입니다. |
| public GameObject dragPlane | 드래그에 사용되는 GameObject입니다. |
| public GameObject dragPlaneObj | 드래그를 위하여 생성되는 GameObject입니다. |

##### 클래스 메서드

|  |
| --- |
| public void updateDragSelect(List<GameObject> list |

drag된 유닛으로 선택 메뉴를 변경하는 메서드입니다.

선택된 유닛 유형에 맞게 select옵션을 모두 변경하고 select 메뉴를 업데이트합니다.

##### dragScript Class

|  |
| --- |
| <요약>  마우스 이벤트 중 하나인 drag로 생성되는 오브젝트의 기능을 정의한 클래스입니다.  다수의 유닛을 선택하는 등의 기능을 수행합니다. |

MonoBehaviour

**->dragScript**

###### 클래스 멤버 변수

|  |  |
| --- | --- |
| List<GameObject> unitList | 드래그 영역에 있는 오브젝트들을 각 종류에 따라 저장하는 리스트들입니다. |
| List<GameObject> structList |
| List<GameObject> resourceList |
| List<GameObject> wildUnitList |
| List<GameObject> list | 최종적으로 manager에게 반환 될 리스트 입니다. |

###### 클래스 메서드

|  |
| --- |
| bool isDragable(Collider col) |

드래그 영역에 존재하는 오브젝트들이 드래그로 선택될 수 있는 유닛인지를 판별하는 메서드입니다.

tag를 통하여 판별합니다.

|  |
| --- |
| void addObj(Collider other) |

오브젝트를 종류에 따라 각 리스트에 맞게 추가하는 함수입니다.

드래그 영역에 해당 오브젝트가 진입할 때 호출됩니다.

|  |
| --- |
| void removeObj(Collider other) |

오브젝트를 종류에 따라 각 리스트에서 제거하는 함수입니다.

드래그 영역에서 해당 오브젝트가 벗어날 때 호출됩니다.

|  |
| --- |
| private void OnTriggerEnter(Collider other) |

드래그 가능한 오브젝트가 드래그 영역으로 진입할 때 해당 오브젝트를 addObj를 통하여 리스트에 추가시킵니다.

|  |
| --- |
| private void OnTriggerExit(Collider other) |

드래그 가능한 오브젝트가 드래그 영역에서 벗어날 때 해당 오브젝트를 removeObj를 통하여 리스트에서 지웁니다.

|  |
| --- |
| void setList() |

드래그의 결과로 반환될 리스트를 결정하는 메서드입니다.

if 분기문으로 반환될 리스트를 결정하며, 위쪽에 선언된 분기부터 반환될 우선순위를 결정합니다.

현재는 다음과 같은 우선순위로 리스트를 반환합니다.

\* unitList -> structList -> resourceList ->wildUnitList

#### 마우스 상호작용

##### 클래스 메서드

|  |
| --- |
| void moveCam() |

마우스가 화면 모서리쪽에 위치해있을 때 카메라의 위치를 이동시키는 메서드입니다.

8방향의 이동을 지원하며 해당 메서드의 camSpeed 변수를 변경하여 카메라의 이동 속도를 조절할 수 있습니다.

또한 area값을 통하여 화면 모서리에 얼마나 근접해야 화면이동이 일어나는지 조절할 수 있습니다.

|  |
| --- |
| IEumerator mouiseController() |

마우스를 통하여 이루어지는 게임 오브젝트들과의 상호작용을 서술한 메서드입니다.

코루틴으로 호출되며 매끄러운 게임 진행을 위하여 호출 주기는 0.1초로 설정되어 있습니다.

해당 메서드를 수정하여 새로운 마우스 상호작용을 추가하거나 기존의 상호작용을 변경할 수 있습니다.

현재 구현되어 있는 상호작용은 다음과 같습니다.

**\* 드래그**

- 왼쪽 마우스 버튼을 누른 상태에서 커서에 유닛이 없다면 드래그 모드로 진입합니다.

- 드래그 모드에서 MouseButtonUp 상태가 된다면 드래그를 종료합니다.

**\* 유닛 선택**

- 쉬프트와 왼쪽 클릭을 동시에 누르면 유닛을 개별적으로 선택/해제합니다.

- 왼쪽 클릭 시 마우스 커서에 오브젝트가 존재한다면 해당 오브젝트를 선택합니다.

**\* 명령 처리**

- 마우스 커서가 특정 명령을 대기하고 있는 상태라면 마우스 왼쪽 버튼을 눌러 해당하는 위치에 명령을 실행합니다.

- 마우스 커서가 명령을 대기하고 있는 상태에서 우측 버튼을 눌러 명령 대기를 취소할 수 있습니다.

- 보통의 커서 상태에서 유닛을 선택한 뒤 오른쪽 버튼을 누르면 해당 위치로의 이동 명령을 수행합니다.

- 미니맵 영역에서 마우스 왼쪽 버튼을 누른다면 해당 미니맵 위치로 카메라를 이동합니다.

상호작용을 버튼의 종류로 나열하면 다음과 같습니다.

|  |  |  |
| --- | --- | --- |
| **상태** | **제어** | **비고** |
| 왼쪽 버튼 Down + 마우스 커서에 오브젝트가 없을 때 | 드래그 진입 |  |
| Shift 버튼 + 왼쪽 버튼 Up | 개별 선택/해제 |  |
| 왼쪽 버튼 Up + 드래그 모드 | 드래그 탈출 |  |
| 왼쪽버튼 Up | 카메라 이동 | 미니맵 위에 커서가 위치 |
| 명령 처리 | 마우스 커서가 해당 명령을 대기하는 상태 |
| 단일 유닛 선택 | 마우스 커서가 기본상태 |
| 오른쪽 버튼 Down | 이동 명령 |  |
| 명령 취소 | 마우스 커서가 특정 명령을 대기하는 상태 |

\* 마우스 왼쪽 버튼 상호작용 \* 마우스 오른쪽 버튼 상호작용

### 키보드 이벤트 핸들러

#### 키보드 상호작용

|  |
| --- |
| IEumerator keyboardController() |

키보드를 통하여 게임 클라이언트와 상호작용하는 동작을 구현한 메서드입니다.

코루틴으로 호출되며 매끄러운 게임 진행을 위하여 호출 주기는 0.1초로 설정되어 있습니다.

KeyCode 값을 변경하여 각 동작을 수행하는 조건을 변경할 수 있습니다.

구현된 상호작용은 다음과 같습니다.

\* ESC 버튼

현재 선택된 모든 유닛들을 선택 해제시킵니다.

\* S버튼

- 유닛이 선택된 상태라면 선택된 유닛들의 상태를 모두 정지상태로 변경합니다.

\* A버튼

- 유닛이 선택된 상태라면 마우스 커서를 공격 명령 상태로 변경하며 매니저의 상태를 공격 명령 대기로 변경합니다.

\* G버튼

- 유닛이 선택된 상태라면 마우스 커서를 채취 명령 상태로 변경하며 매니저의 상태를 채취 명령 대기로 변경합니다.

\* F10버튼

- 게임 클라이언트 메뉴 창을 활성화시킵니다.

- 해당 상태에서 E 버튼을 누른다면 게임을 종료합니다.

# 길 찾기 스크립트 API

## PathFinder

|  |
| --- |
| <요약>  tile을 노드로 사용하여 경로를 찾아 반환하는 A\*알고리즘이 구현된 클래스입니다. |

Monobehaviour

-> PathFinder

### 클래스 멤버 변수

|  |  |
| --- | --- |
| public GameObject[] tileSet | tile 오브젝트들을 참조하는 변수입니다. |
| Int[, ] tileMatrix | 각 tile간의 비용이 저장 되어있는 변수입니다. |
| Bool[] moveable | tile이 이동 가능한지 여부에 대한 변수입니다. |
| Dictionary<int, Tile> openList | A\* 알고리즘의 openList입니다. |
| Dictionary<int, Tile> closeList | A\* 알고리즘의 closeList입니다. |

### 클래스 메서드

|  |
| --- |
| private void Awake() |

PathFinder는 tile 오브젝트를 생성하고 tileSet에 추가합니다.

또한 tileMatrix에 각 tile간 가중치를 초기화합니다.

8방향으로의 이동이 가능하도록 설계되었으며, 상하좌우는 10의 가중치, 대각선으로는 14의 가중치를 가지고있습니다.

|  |
| --- |
| publid Stack<Tile> PathFinding(Vector3 startPos, Vector3 endPos) |

현재 대상이 위치한 위치 값과 이동하기 원하는 위치 값을 매개변수로 받습니다.

두 위치에 해당되는 tile을 시작점과 도착지점으로 선정한 뒤 A\*알고리즘을 통하여 경로를 탐색합니다.

경로 탐색은 최대 5000회까지 가능하도록 설정 되어있습니다.

반환값은 Stack으로 반환되어 pop을 통하여 역 정렬 과정을 거치지 않고 바로 이동할 수 있도록 하였습니다.

|  |
| --- |
| void checkTile(int tileNumber\_, Tile par\_, int cost\_, Vector3 endPos\_) |

A\* 알고리즘 중 이미 OpneList에 같은 타일 번호를 가진 타일이 존재하는지 확인하는 메서드입니다.

기존 타일까지의 비용이 더 낫다면 아무것도 하지 않고 해당 없다면 OpenList에 새로운 타일을 추가합니다.

|  |
| --- |
| public void setMoveable(int tileNumber\_, bool moveable\_) |

매개변수로 전달받은 번호의 타일의 이동 가능 여부를 매개변수로 전달받은 moveable\_값으로 변경시킵니다.

|  |
| --- |
| public bool isMoveable(float tileX\_, float tileZ) |

위치값을 매개변수로 받아 해당 위치 타일의 현재 이동 가능 여부를 반환합니다.

|  |
| --- |
| Stack<Tile> makeReturnPath(Tile startTile\_, Tile lastTile\_) |

현재 타일로부터 부모를 계속 따라가 경로로써 반환될 Stack을 생성하고 반환합니다.

## checkTile

|  |
| --- |
| <요약>  게임 내 오브젝트로 존재하는 Tile입니다.  타일 생성 시 초기화를 통하여 이동 가능한 지형인지 판별하는 역할을 수행합니다. |

Monobehaviour

-> checkTile

### 클래스 멤버 변수

|  |  |
| --- | --- |
| float xPos | 타일의 x좌표입니다. |
| float zPos | 타일의 z좌표입니다. |
| public int tileNumber | 타일의 번호입니다. |

### 클래스 메서드

|  |
| --- |
| public void tileInit() |

OverlapBox를 활용하여 현재 주변에 Collider가 존재하는지 판별합니다.

ground 혹은 fog 태그가 붙어있지 않은 물체(이동 판정을 무시할 수 없는 물체)가 존재한다면 PathFinder에게 해당 tile의 이동 여부를 불가능으로 알려줍니다.

## Tile

|  |
| --- |
| <요약>  PathFinder에서 노드로써 활용되는 Tile을 정의한 클래스입니다.  PathFinder에서 각 상황에 맞게끔 생성되어 해당하는 G, H값과 parent Tile을 가집니다. |

Monobehaviour

-> Tile

### 클래스 멤버 변수

|  |  |
| --- | --- |
| float xPos | 타일의 x좌표입니다. |
| float zPos | 타일의 z좌표입니다. |
| float G | 타일이 현재 가지고 있는 G값입니다. |
| flaot H | 타일이 현재 가지고 있는 H값입니다. |
| Tile parent | 현재 타일의 부모 Tile입니다. |
| public int tileNumber | 현재 타일의 번호입니다. |

### 클래스 메서드

|  |
| --- |
| public void initTile(Tile parent\_, int cost\_, Vector3 EndPos\_) |

현재 타일을 초기화 하는 메서드입니다.

parent를 매개변수로 전달받은 부모 타일 값으로 설정한 뒤 G값과 H값을 초기화 합니다.

H값은 현재 위치와 목적지까지의 거리의 값으로 초기화 됩니다.

|  |
| --- |
| public void setMaxTile() |

현재 타일의 H값과 G값을 999값으로 초기화합니다.

PathFinder의 A\*알고리즘에서 사용되며, openList에서 최저 비용을 찾기 위하여 큰 값을 가진 노드 하나를 초기화할 때 사용됩니다.

# 네트워킹 스크립트 API

## GamePlayer

|  |
| --- |
| <요약>  게임 Scene의 관리 및 유닛 스폰을 담당하는 클래스입니다.  해당 클래스에 스폰될 유닛들의 정보를 정의하여 스폰 할 유닛을 등록할 수 있습니다. |

NetworkBehaviour

**-> GamePlayer**

### 클래스 멤버 변수

|  |  |
| --- | --- |
| Vector3[] struct1SpawnPositions | 플레이어 1의 등록된 건물의 스폰 위치입니다. |
| Vector3[] struct2SpawnPosition | 플레이어2의 등록된 건물 스폰 위치입니다. |
| Quaternion[] rotations1 | 플레이어 1의 등록된 건물의 회전값입니다. |
| Quaternion[] rotations2 | 플레이어2의 등록된 건물의 회전값입니다. |
| public NetworkIdentity networkID | 게임에 접속한 플레이어의 NetworkIdentity값입니다.  각 플레이어의 Connection을 가져올 때 사용됩니다. |
| public GameObject[] structSet | 네트워킹 대상이 되는 오브젝트들을 모은 변수입니다. 해당 변수에 각 종류에 맞는 오브젝트를 할당하여 스폰 대상으로 등록할 수 있습니다. |
| public GameObject[] unitSet |
| public GameObject[] wildSet |
| public GameObject[] resourceSet |
| public GameObject[] wildUnitStructureSet |
| public GameObject effectSet |

### 클래스 메서드

|  |
| --- |
| public void portalActive(int team) |

매개변수로 받은 team의 포탈을 활성화시키는 메서드입니다.

PortalScript의 메서드를 활용하여 클라이언트 차원에서 활성화하고 상대방 클라이언트에 활성화됨을 알립니다.

|  |
| --- |
| public void spawnUnit(int unitID, bool isStruct) |

외부에서 해당 메서드를 호출하여 유닛 스폰을 요청할 수 있습니다.

매개변수의 unitID값은 등록된 objectSet 배열의 해당 유닛의 인덱스이며, bool isStruct값으로 스폰하고자 하는 오브젝트가 유닛인지 건물인지 선택할 수 있습니다.

이후 스폰된 오브젝트의 Team을 스폰한 플레이어의 Team으로 초기화 시킵니다.

|  |
| --- |
| public void spawnUnit(int unitID, bool isStruct, Vector3 pos\_) |

SpawnUnit 메서드의 오버로딩된 형태입니다.

위치를 정하여 스폰을 하는 경우 해당 메서드를 호출하여 스폰 요청을 할 수 있습니다.

|  |
| --- |
| public void animatorUpdate(GameObject unit, int unitState) |

Animator를 사용하는 유닛을 위한 메서드입니다.

매개변수 unit의 Animator 상태를 unitState로 변경시키는 메서드입니다.

|  |
| --- |
| public void spawnResource(int resourceID) |

서버에서 호출되어 자원들을 생성하는 메서드입니다.

매개변수에 따라 해당하는 광물과 목재, 포탈을 스폰합니다.

|  |
| --- |
| public void spawnWild(int resourceID) |

서버에서 호출되어 자원들을 생성하는 메서드입니다.

매개변수에 따라 해당하는 Wild Unit을 스폰합니다.

|  |
| --- |
| public void spwanEffect(int effectID\_, Vector3 pos\_) |

매개변수로 받은 pos\_ 위치에 effectID에 해당되는 이펙트를 스폰합니다.

|  |
| --- |
| public void updateUpgrade(float MA, float MD, float RA, float RD) |

현재 업그레이드 상태를 매개변수 값으로 상대방에게 업데이트 하는 메서드입니다.

해당 함수를 호출하여 상대방에게 현재의 자신 업그레이드 값을 전달할 수 있습니다.

|  |
| --- |
| public override void OnStopClient() |

상대방의 게임 클라이언트가 접속 정지될 때 호출되는 메서드입니다.

게임이 끝나지 않은 상태에서 클라이언트 간 연결이 끊길 경우 게임 Scene에서 탈출하고 메인 메뉴로 돌아오도록 되어있습니다.

## PingManager

|  |
| --- |
| <요약>  네트워크 오브젝트들의 SendInterval을 네트워크 환경에 맞게끔 조절하는 동작을 수행하는 클래스입니다.  RTT를 기반으로 동작하며 해당 값을 통하여 네트워크 전송 주기를 조절합니다. |

NetworkBehaviour

**-> PingManager**

### 클래스 멤버 변수

|  |  |
| --- | --- |
| public List<GameObject> registedObjects | 등록된 오브젝트들의 리스트입니다. |
| public const double goalInterval | 목표 전송 주기입니다. |
| public double managedInterval | 현재의 전송 주기입니다. |
| private double lastRtt | 마지막으로 측정한 RTT 값입니다. |
| private int current | 전송 주기를 조절하는 현재 가중치 값입니다. |
| private float updateInterval | 전송 주기를 업데이트하는 주기입니다. |

### 클래스 메서드

|  |
| --- |
| void managingSendInterval() |

RTT 값을 기반으로 오브젝트의 Send Interval을 설정합니다.

lastRtt/2 값을 기반으로 전송 주기를 증가 혹은 감소시킵니다.

전송 주기는 goalInterval 이하로 떨어지지 않습니다.

변동 값은 2의 지수승으로 변동되며 현재 지수의 값은 current 변수의 값을 사용합니다.

|  |
| --- |
| void setSendInterval() |

등록된 오브젝트들의 sendInterval을 managedInterval 값으로 변경합니다.

|  |
| --- |
| public void registerObject(GameObject obj) |

매개변수로 전달받은 오브젝트를 registedObjects 리스트에 추가하여 관리 대상으로 등록합니다.

|  |
| --- |
| public void unregisterObject(GameObject obj) |

매개변수로 전달받은 오브젝트를 registedObjects 리스트에서 제거하여 관리 대상에서 제외합니다.

# UI 스크립트 API

## MainUI

|  |
| --- |
| <요약>  게임 시작 시 맨 처음으로 보게 되는 메인 화면을 구성하는 UI 및 해당하는 UI에 대한 이벤트를 정의한 클래스입니다.  메인 화면은 OffLine으로 동작합니다. |

MonoBehaviour

-> GameRoomUI

### 클래스 멤버 변수

|  |  |
| --- | --- |
| Input Field NameField | 입장 시 표시되는 이름을 입력하는 InputField입니다. |
| Text PromptText | 메인 화면의 Prompt로 표시되는 Text입니다. |
| float disconnectTime | 응답을 기다리는 시간입니다. 해당 시간까지 응답이 없으면 연결을 해제하고 프롬프트에서 응답 없음을 알립니다. |
| public Text SettingPrompt | IP Setting창에서 출력되는 Prompt Text입니다. |

### 클래스 메서드

|  |
| --- |
| public void createRoom() |

호스트 모드로 시작하고자 할 때 호출되는 메서드입니다.

CreateRoom 버튼 이벤트로 지정 되어있습니다.

플레이어의 이름을 지정하였고 현재 IP로 호스트를 동작 중이지 않은 상태면 새로운 GameRoom을 생성합니다.

만약 이미 IP를 사용 중이라면Prompt를 통하여 메시지를 전달합니다.

|  |
| --- |
| public void enterRoom() |

클라이언트 모드로 시작하고자 할 때 호출되는 함수입니다.

enter 버튼 이벤트로 지정되어 있습니다.

플레이어 이름이 지정된 상태이면 연결을 요청합니다.

만약 disconnectTime만큼이 지난 상태에서 연결이 만들어지지 않으면 연결을 종료하고 Prompt를 통하여 메시지를 전달합니다.

|  |
| --- |
| public void exitRoom() |

게임을 종료하고자 할 때 호출되는 함수입니다.

Exit버튼 이벤트로 지정되어 있습니다.

해당 함수를 호출하면 프로그램을 종료합니다.

만약 Unity 에디터에서 해당 함수가 호출되면 실행모드를 종료하도록 되어있습니다.

## GameRoomUI

|  |
| --- |
| <요약>  각 플레이어가 입장하여 게임을 시작하기 전 단계에 해당하는 Game Room에서 이루어지는 UI의 동작들을 정의한 클래스입니다. |

MonoBehaviour

-> GameRoomUI

### 클래스 메서드

|  |
| --- |
| public void OnReadyButton() |

클라이언트 플레이어가 버튼을 누를 때 발생하는 이벤트입니다.

서버 플레이어 에게 현재 준비 완료됨을 알립니다.

|  |
| --- |
| public void OnStartButton() |

서버 플레이어가 게임 시작 버튼을 누를 때 발생하는 이벤트입니다.

클라이언트 플레이어가 준비가 완료되었을 때 게임 시작 버튼이 활성화되어 버튼을 누를 수 있습니다.

|  |
| --- |
| public void colorChange() |

Ready 버튼에 이벤트로 할당 되어있는 메서드입니다.

서버의 화면에서 클라이언트 플레이어의 이름이 파란색으로 변하며 준비 완료됨을 시각적으로 알려줍니다.

|  |
| --- |
| public void OnExitButton() |

Exit버튼에 할당 되어있는 이벤트 메서드이며 호스트와 클라이언트의 동작이 구현되어 있습니다.

호스트의 경우 호스트 모드를 종료하고 해당 방이 파괴됩니다.

클라이언트의 경우 호스트의 화면에서 클라이언트의 이름을 지우고 클라이언트 모드를 종료하여 메인 화면으로 돌아갑니다.

## UIController

|  |
| --- |
| <요약>  UI의 기본적인 동작 외에 추가적인 기능들을 구현한 클래스입니다. |

MonoBehaviour

-> GameRoomUI

### 클래스 메서드

|  |
| --- |
| void setPortraitColor(GameObject UIObj, GameObject targetObj) |

유닛의 HP 비율에 따른 초상화 색깔 변화를 UI에 반영하는 메서드입니다.

현재 유닛 HP 상태에 따라 매개변수로 받은 UIObj의 Image의 색을 변경합니다.

초상화의 색이 변화하는 분기점은 다음과 같습니다.

* ( 100% / 75% / 50% / 25% )

|  |
| --- |
| public void OnMultiSelectUI() |

Select Menu에서 다수의 유닛이 선택되었을 때 특정 유닛의 초상화를 클릭함으로 해당 유닛을 개별 선택하도록 만드는 메서드입니다.

manager의 selectedList의 해당 UI의 번호 인덱스에 있는 유닛으로 개별 선택하도록 합니다.

# 기타 스크립트 API

## LoadingImageScript

|  |
| --- |
| <설명>  게임Scene에서 각 클라이언트가 동일한 시간에 게임에 진입할 수 있도록 Loading Image를 제어하는 동작을 정의한 클래스입니다. |

MonoBehaviour

-> LoadingImageScript

클래스 멤버 변수

|  |  |
| --- | --- |
| bool isSetup | 현재 게임 클라이언트가 준비되었는지 여부에 대한 변수입니다. |
| String startTime | 게임의 시작시간을 저장하는 문자열입니다. |

클래스 메서드

|  |
| --- |
| public void setup(string time\_) |

매개변수로 전달받은 시간으로 시작 시간을 셋팅합니다.

|  |
| --- |
| void Update() |

현재 시간을 체크합니다.

만약 지정된 시작 시간이 되었다면 이벤트 시스템을 활성화하고 loadingImage를 제거합니다.