

AnimalPark

<the 동물 사냥꾼>

< 목차 >

1. 게임의 배경
2. 게임의 구성
3. 게임의 진행
4. 코드구성

AnimalPark

<the 동물 사냥꾼>

1. 게임의 배경

유전자 변형을 당한 동물들이 주민들을 위협하며 자신들의 영역을 넓히고 있습니다.

삶의 터전에서 고통받던 주민들은 전설적인 사냥꾼을 고용하게 되고,
사냥꾼은 위협적인 능력을 가지게 된 동물들을 사냥하러 그들의 서식지로 출발합니다.

AnimalPark <the 동물 사냥꾼>

2. 게임의 구성

동물 List

- Pudu(사슴)
- Sheep(양)
- Squid(오징어)
- Flower(식인꽃)
- Taipan(뱀)
- Mole(두더지)

Stage List

- Nature
- Island
- Desert

블록 List



AnimalPark <the 동물 사냥꾼>

2. 게임의 구성

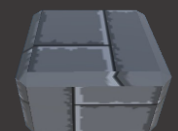
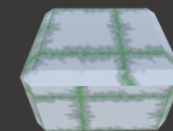
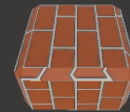
동물 List

- Sparrow(참새)
- Muskrat(쥐)
- Herring(물고기)
- Penguin(펭귄)
- Colobus(원숭이)
- Gecko(도마뱀)

Stage List

- City
- Winter
- Space

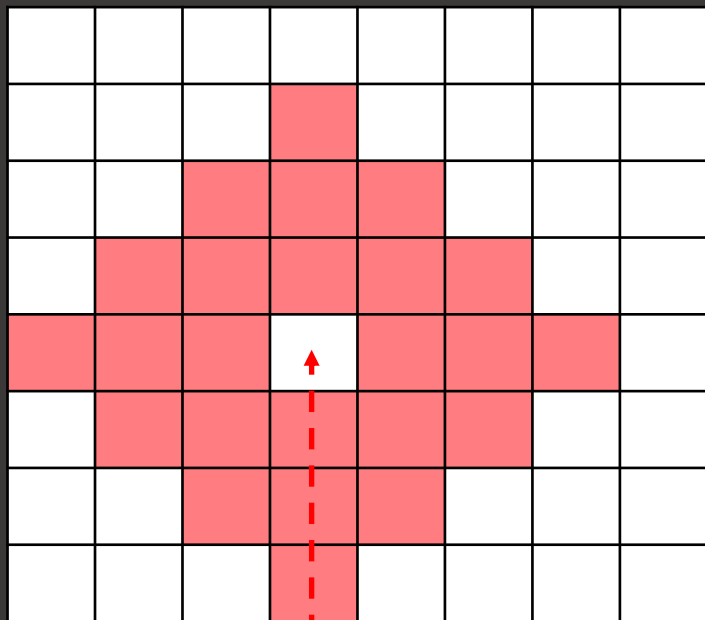
블록 List



AnimalPark <the 동물 사냥꾼>

2. 게임의 구성

[이동범위]

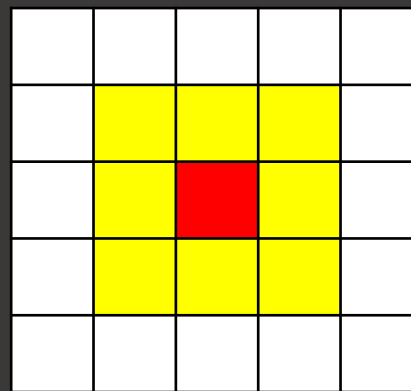


Hunter

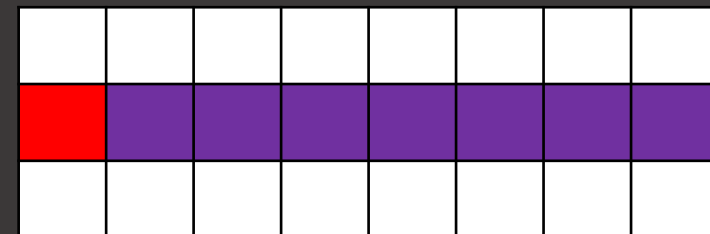
[공격범위]

Hunter

[근거리]

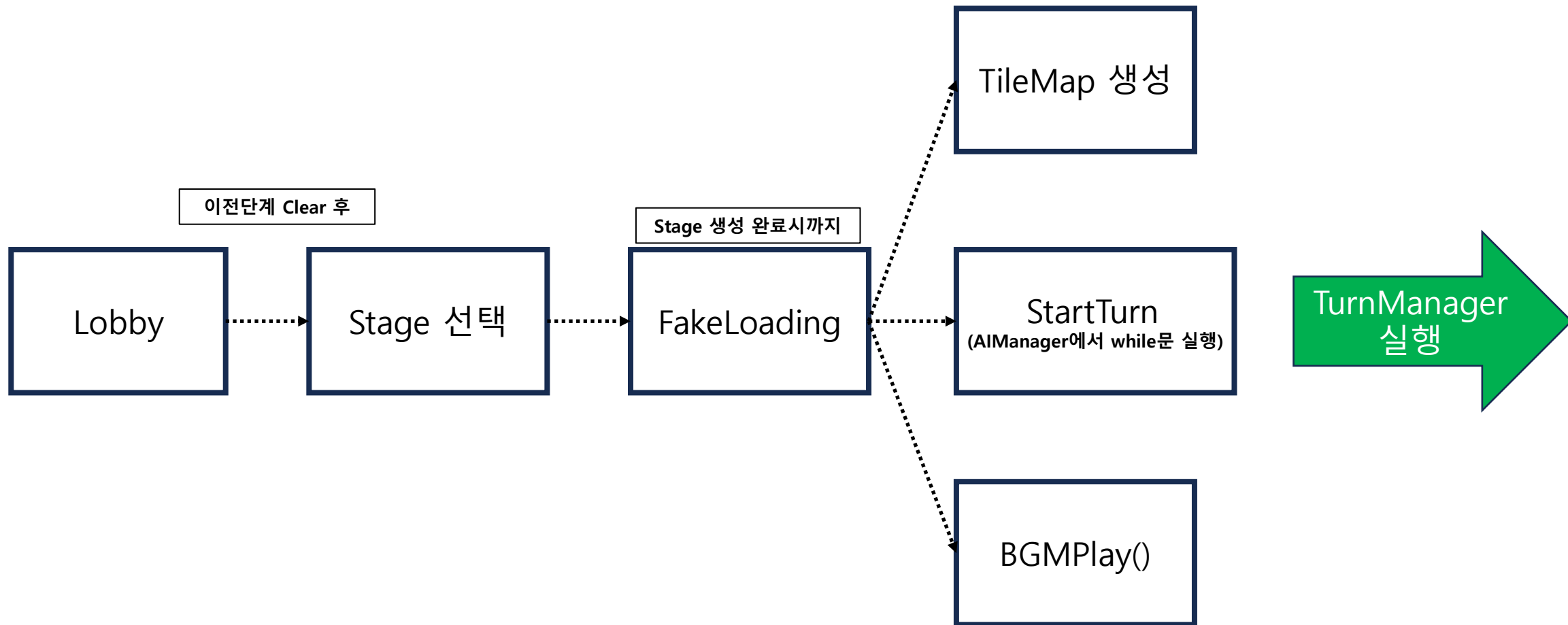


[원거리]



AnimalPark <the 동물 사냥꾼>

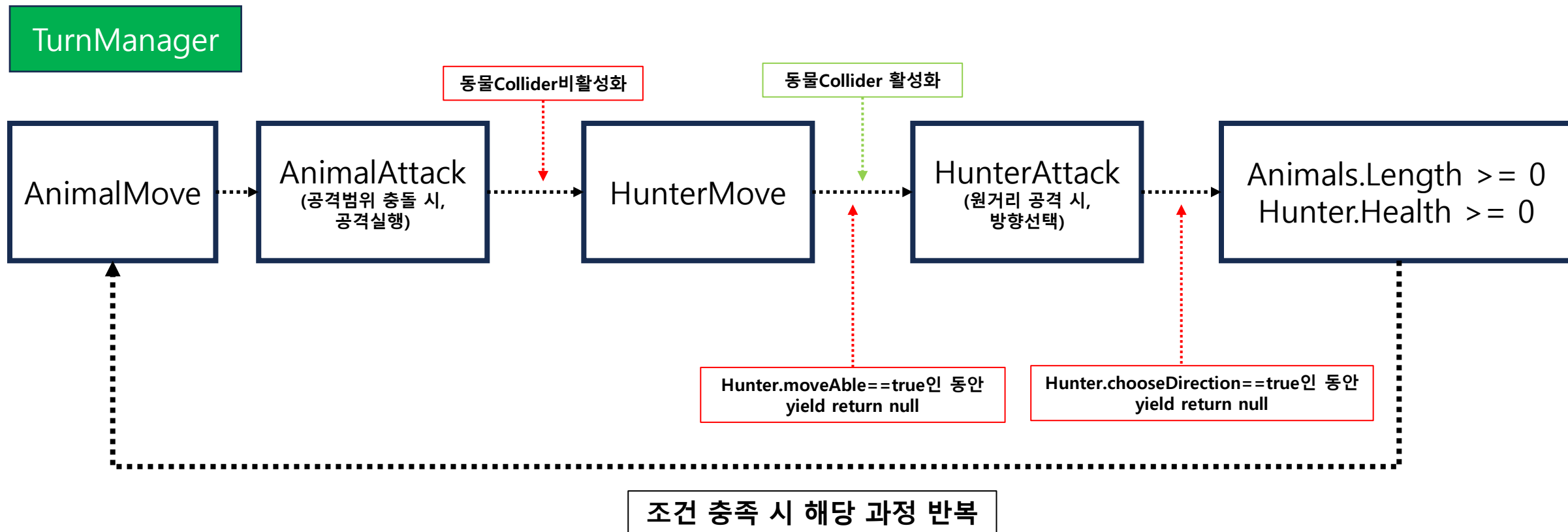
3. 게임의 진행



AnimalPark

<the 동물 사냥꾼>

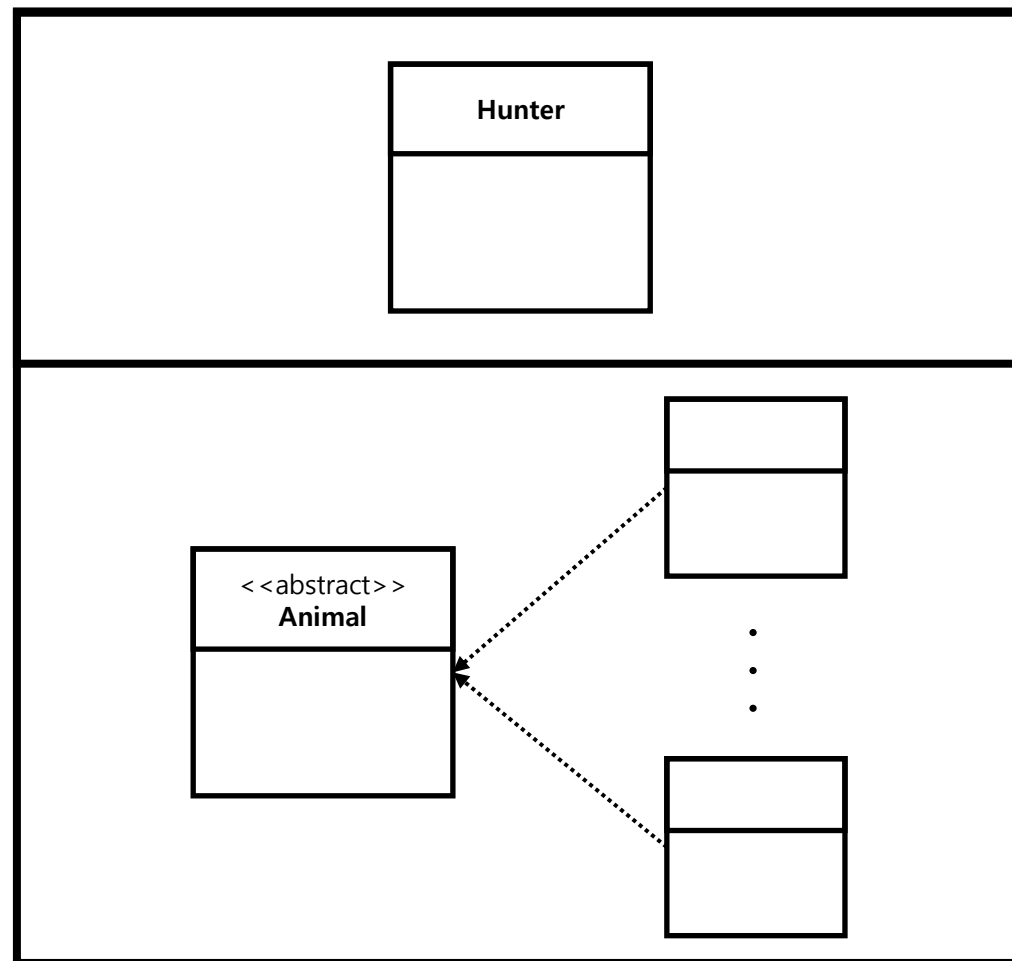
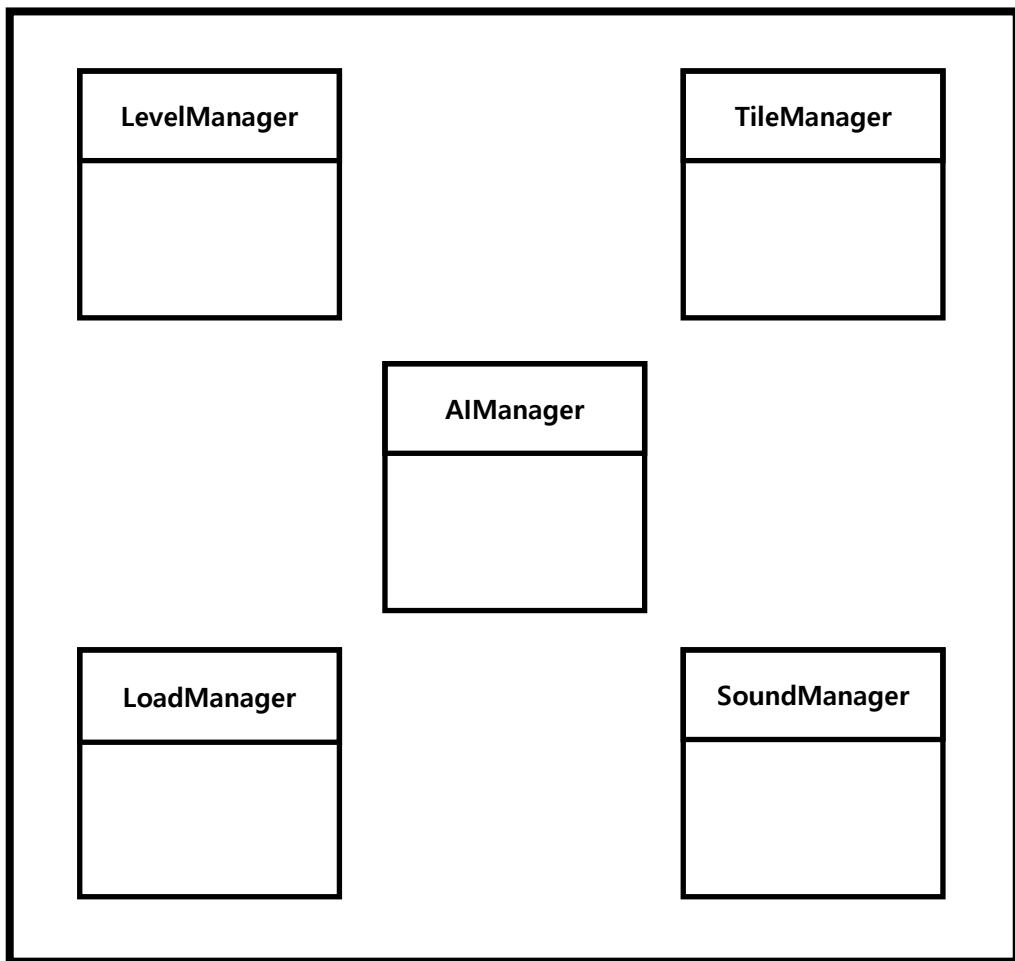
3. 게임의 진행



AnimalPark

<the 동물 사냥꾼>

4. 코드 구성



AnimalPark

<the 동물 사냥꾼>

LoadManager

- ...

- + LoadScene(string sceneName) : void
- + <<coroutine>>LoadNextScene(string next) : IEnumerator
- + CreateTileMap() : TileManager
- + StartTurn() : AIManager
- + BGMPlay() : SoundManager

TileManager

- Blocks : GameObject[]
- SceneNumber : Dictionary<string , int>
- SceneName : string[]

- + <<callback>> OnSceneLoaded
(scene : Scene , mode : LoadSceneMode) : void

- + CreateTileMap() : void

AIManager

- Animals : GameObejct[]
- TileMap : int[,]

- + StartTurn() : void
- + <<coroutine>> ActiveAiManager() : IEnumerator
- + UpdateAnimalList() : void
- + <<coroutine>> TurnManager() : IEnumerator
- + AnimalMove() : void
- + AnimalAttack() : void
- + HunterMove() : void
- + HunterAttack() : void

LevelManager

- sceneName : String
- MaxLevel : bool
- Level : bool[]

- + ClickMap(Button button) : void
- + LinkMaps() : void
- + LevelUp() : void
- + SetSceneName() : void
- + GetSceneName() : string

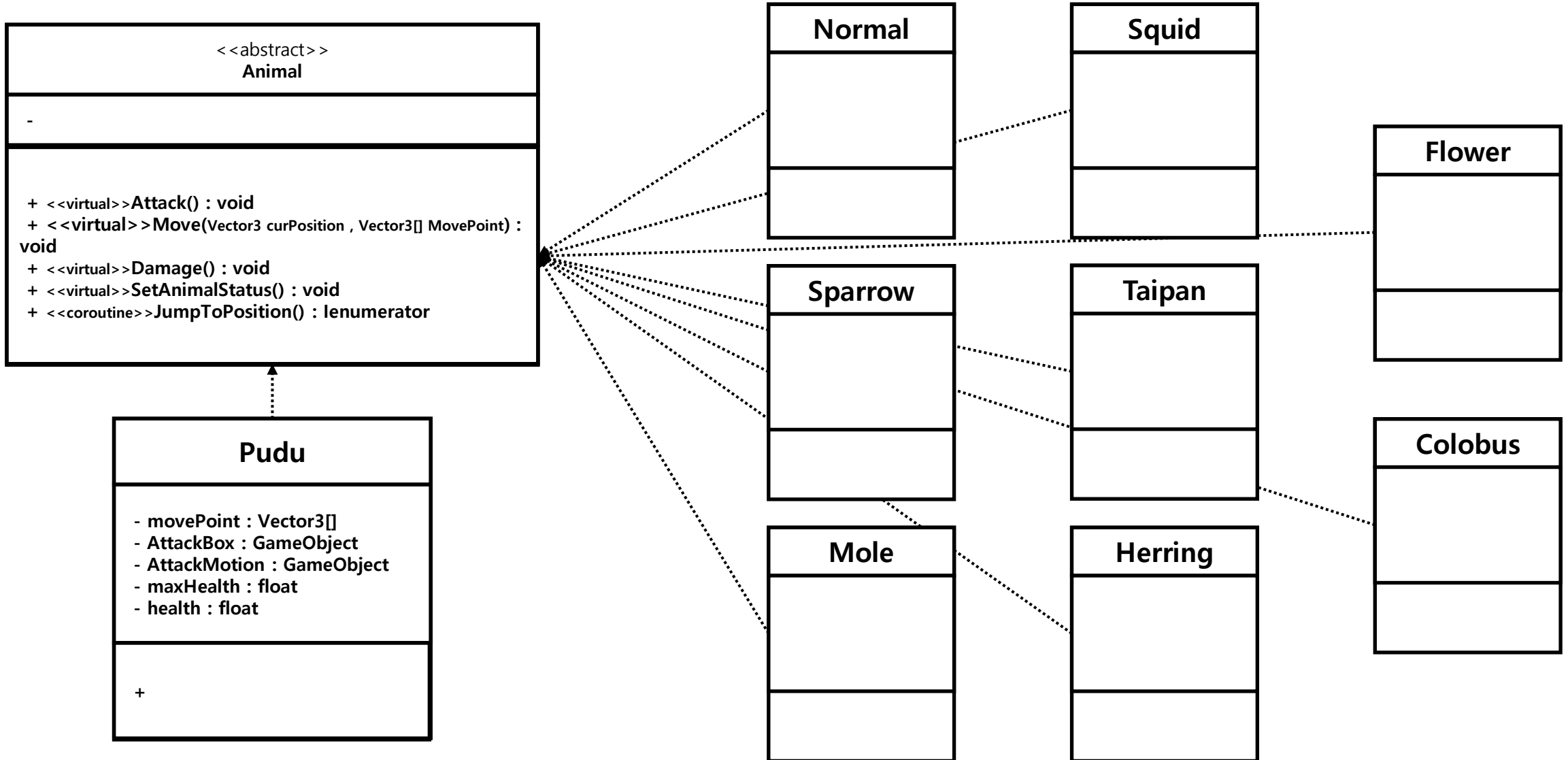
SoundManager

- bgm : AudioClip[]
- audioClips : AudioClip[]
- backgroundAudioSource : AudioSource
- effectAudioSource : AudioSource

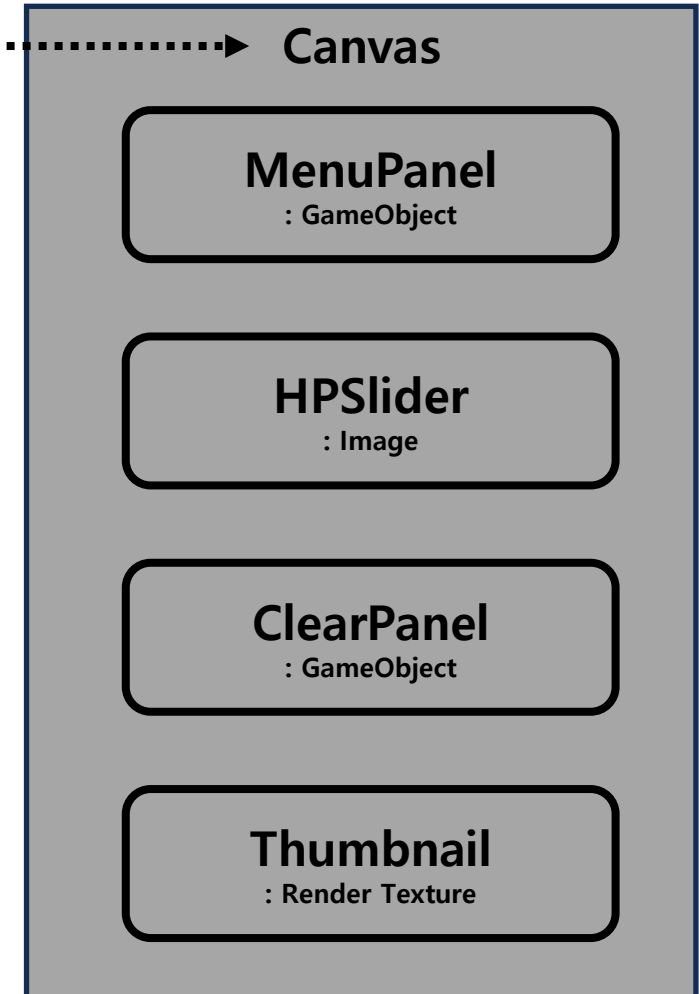
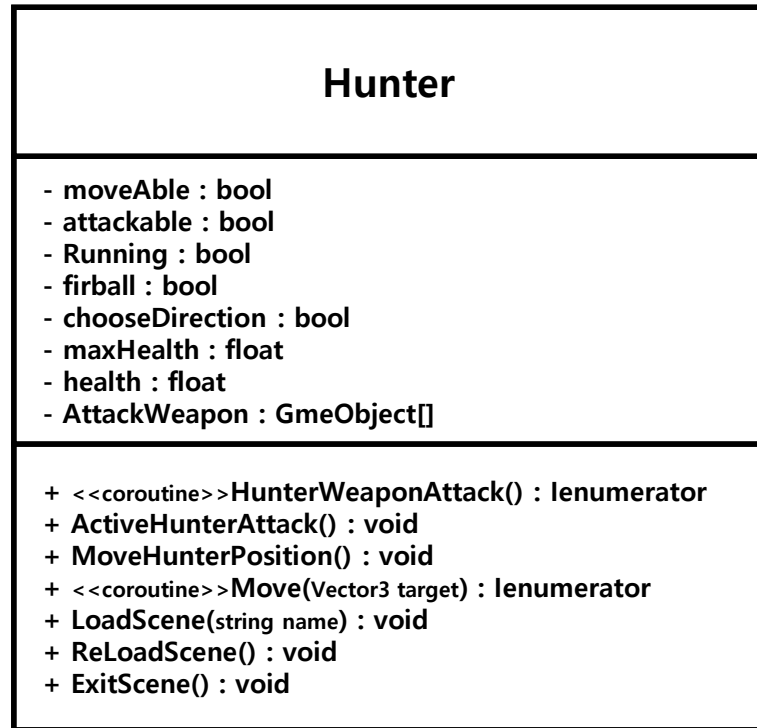
- + BGMPlay() : void
- + SoundPlay(string name) : void

AnimalPark

<the 동물 사냥꾼>



AnimalPark <the 동물 사냥꾼>



AnimalPark

<the 동물 사냥꾼>

5. 핵심 코드 _AIManager

참조 1개
private IEnumerator TurnManager()
{

while (Hunter.Health>=0)

{
for (int i = 0; i < 8; i++)

{
for (int j = 0; j < 8; j++)

{
TileMap[i, j] = 0;
}

}
yield return null;

AnimalMove();

yield return new WaitForSeconds(1.0f);

AnimalAttack();

yield return new WaitForSeconds(1.0f);

if (CheckAnimalAttack()){yield return new WaitForSeconds(3.0f);}

else{ yield return new WaitForSeconds(1.5f); }

AnimalUnAttackBox();

HunterMove();

while (Hunter.Moveable) { yield return null;}

hunter.Attack();

while (Hunter.Attackable) { yield return null; }

if (Animals.Length <= 0 || Hunter.Health<=0)

{
break;
}

}

동물의 움직임 관리는 애니메이션
시간을 고려해서 시간을
상이하게 설정합니다.

헌터의 위치선택을 방해하지 않기 위해 동물들의
collider를 비활성화 시켜놓는 함수입니다.

헌터가 가지고 있는
AttackAble 과 moveAble을 이용하여
게임의 진행을 판단합니다.

AnimalPark <the 동물 사냥꾼>

5. 핵심 코드 _LoadManager

참조 1개

```
IEnumerator LoadNextScene(string next)
{
    float loadTime = 2f;
    float elapsedTime = 0f;

    float changeInterval = 0.1f; //로딩 시 이미지 전환 속도
    float nextChangeTime = changeInterval;

    AsyncOperation asyncLoad = SceneManager.LoadSceneAsync(next);
    asyncLoad.allowSceneActivation = false; .....
    soundManager.MoveStage(); // 로딩 BGM 설정

    // 로딩 시 사용할 움직이는 이미지 설정
    while (elapsedTime < loadTime) .....
    {
        yield return null;
        elapsedTime += Time.deltaTime;

        MoveDucks((int)(elapsedTime / changeInterval) % loadingImages.Length);
        nextChangeTime += changeInterval;

        loadingBar.value = Mathf.Lerp(loadingBar.value, elapsedTime / loadTime, Time.deltaTime);
    }
}
```

씬 전환을 지연시키고, 조건이 만족되면 씬을 활성화합니다.

로딩이 진행되는 동안 움직이는 오리 이미지를 구현할 while문

AnimalPark <the 동물 사냥꾼>

5. 핵심 코드 _LoadManager

```
while (!asyncLoad.isDone)
{
    yield return null;

    // 진행률 0.9 미만일 때
    if (asyncLoad.progress < 0.9f)
    {
        loadingBar.value = asyncLoad.progress;
    }
    else
    {
        // 마지막 로딩 완료
        loadingBar.value = 1f;

        // 씬 전환 허용
        asyncLoad.allowSceneActivation = true;
        aiManager.ActiveHintPanel();
        yield return new WaitForSeconds(0.5f);
        tileManager.CreateTileMap();
        aiManager.StartTurn();
        loadingBar.value = 0;
        loadingCanvas.SetActive(false);
        soundManager.BGMPlay();
    }
}
```



로딩이 끝나고 나면 기본 세팅을 진행합니다.

1. 힌트 패널 활성화
2. Stage에 맞는 블록타일로 맵 생성
3. AIManager의 게임진행함수 실행
4. Stage에 맞는 BGM Play

AnimalPark

<the 동물 사냥꾼>

END

블로그 주소 : <https://baegwoo.github.io>

깃 허브 주소 : <https://github.com/BaeGWoo>