클러스터노이드 개발 리뷰

김용욱

들어가기 전에

대본과 발표자료는 카페에 올리겠습니다.

잘 된 점

빠른 개발 시작

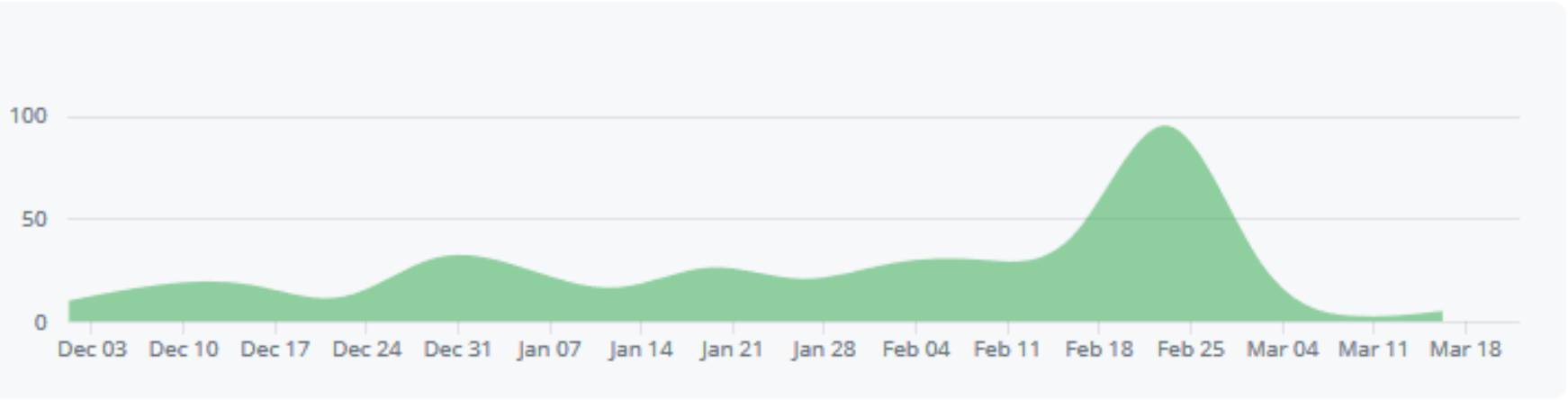
Initial commit
[redacted] committed on 4 Dec 2017

클러스터노이드 첫 모임 (2017. 12. 3)

Dec 3, 2017 – Mar 24, 2018

Contributions: **Commits** ▾

Contributions to master, excluding merge commits



빠른 기획 확정

게임 컨셉

복셀 그래픽의 2D 게임. (점프없음)
탑다운 슈터.
뉴클리어쓰론 + 체력대신에 복제인간들이 나온다.
이동과 조준이 별개.
빠르게 나올 수 있는 게임. <<

개발시작 - 12월 중순 ~ 말 사이에 시작해야 (x) 학교다니는 멤버가 한명뿐이다.
그래도 12월 중순까지는 초안이 나와줘야.
중간발표 - 1월 중순 / 1월말
목표 - 넥프 (2월말) -> 3월말~4월초 스팀

외부 에셋의 사용




UNITY TECHNOLOGIES

Post Processing Stack

★★★★★ 198개 유저리뷰

Note: version 2.0 of the post-processing stack is curre

The new Unity post-processing stack is an über effect tl
advantages :

- Effects are always configured in the correct order.
- It allows combination of many effects into a single pas
- It features an asset-based configuration system for ea
- All effects are grouped together in the UI for a better u

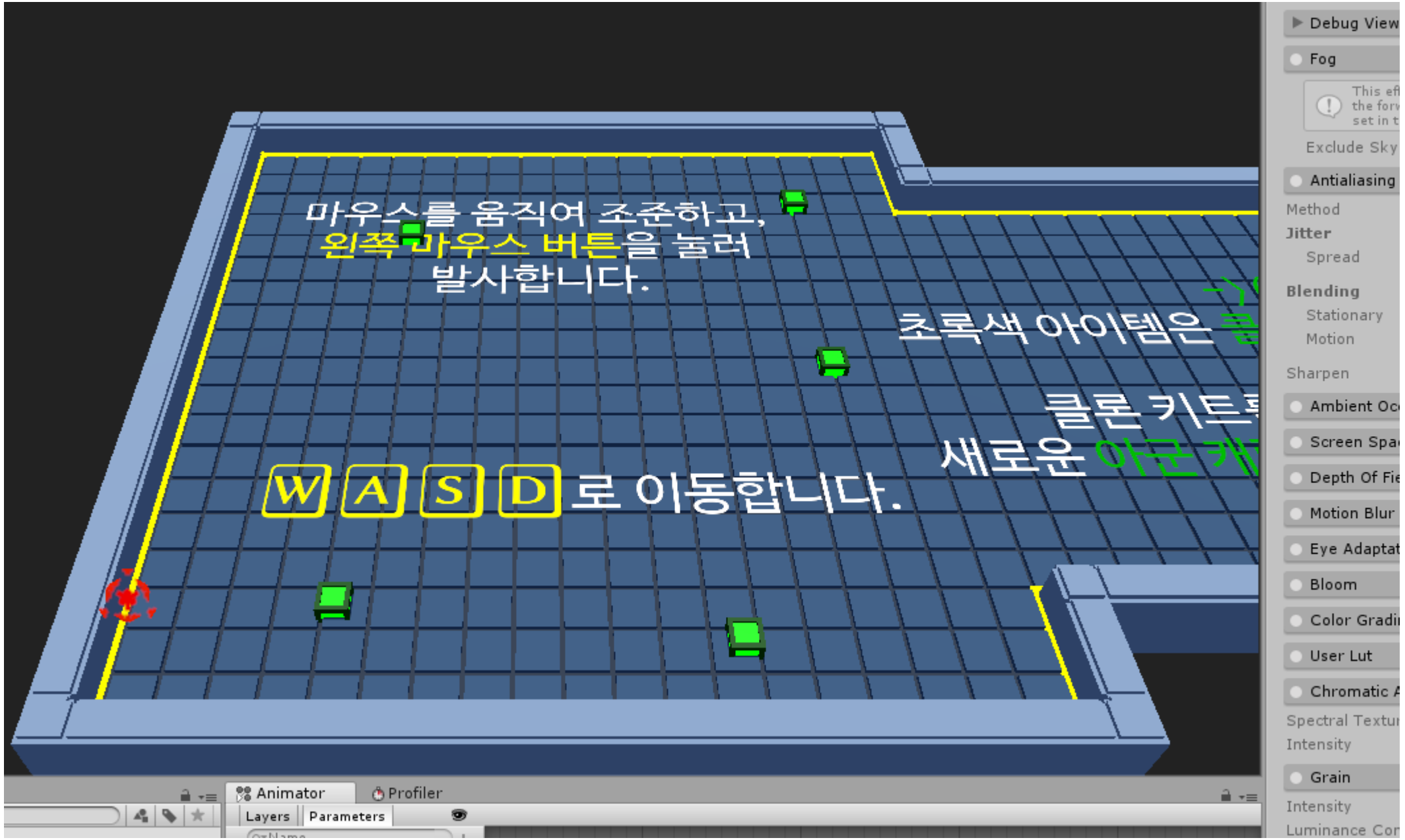
It comes with the following effects :

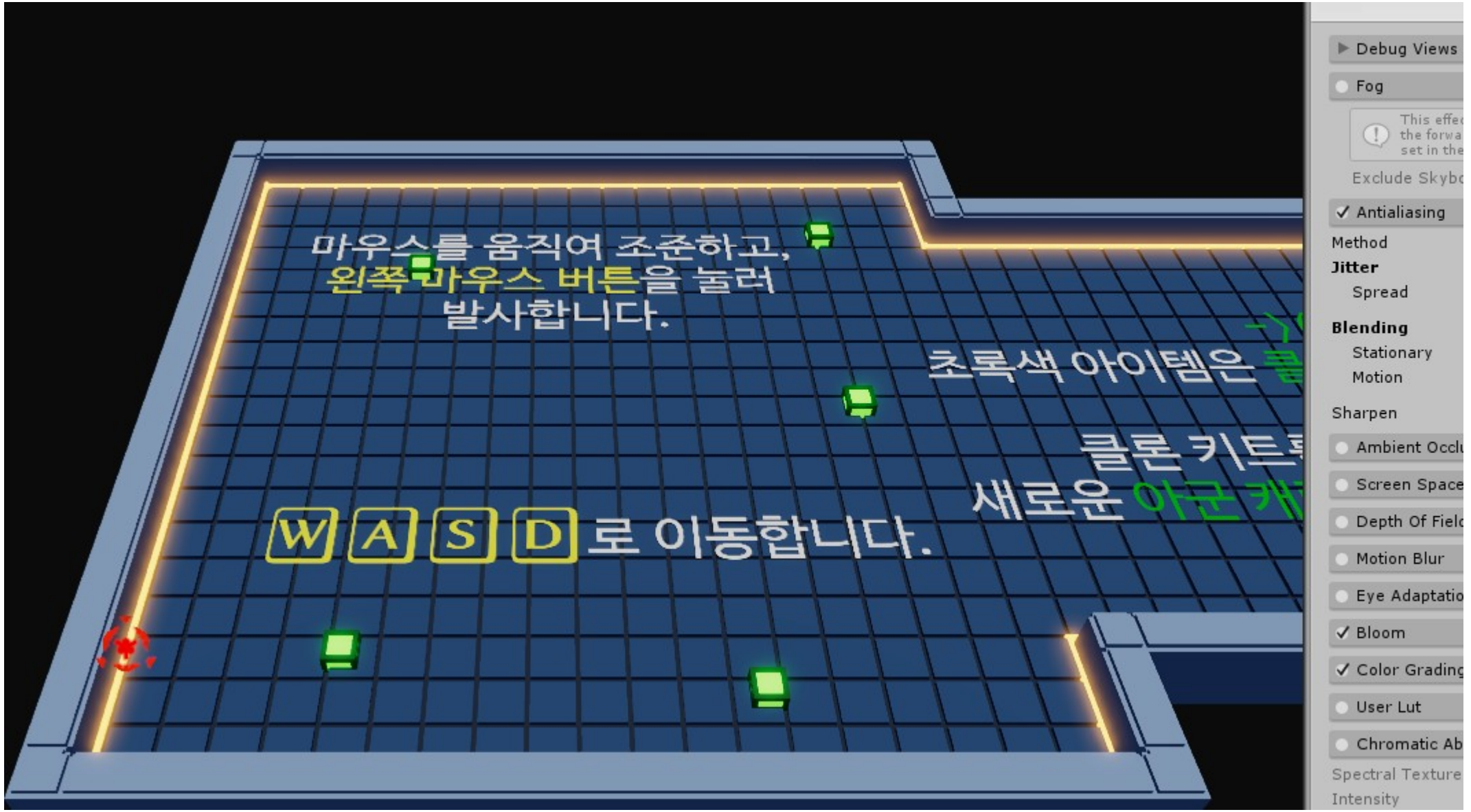
- Antialiasing (FXAA, Temporal AA)
- Ambient Occlusion

더 보기

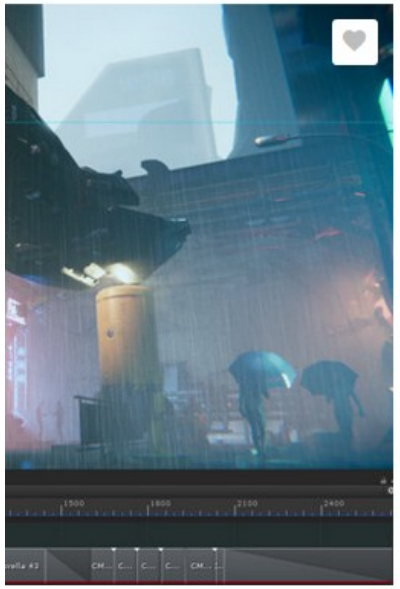
공유하기 목록에 추가하기

패키지 구성	31.1 MB
출시	최신 버전 1.0.4
지원되는 유니티 버전	5.5.0 버전 이상


마우스를 움직여 조준하고,
왼쪽마우스 버튼을 눌러
발사합니다.
초록색 아이템은
클론 키트
새로운
W A S D 로 이동합니다.
Debug View
Fog
Exclude Sky
Antialiasing
Method
Jitter
Spread
Blending
Stationary
Motion
Sharpen
Ambient Oc
Screen Spa
Depth Of Fie
Motion Blur
Eye Adaptat
Bloom
Color Gradi
User Lut
Chromatic A
Spectral Textu
Intensity
Grain
Intensity
Luminance Cor
Animator
Profiler
Layers
Parameters


마우스를 움직여 조준하고,
왼쪽마우스 버튼을 눌러
발사합니다.
초록색 아이템은
클론 키트
새로운 아군
W A S D 로 이동합니다.
Debug Views
Fog
This effe
the forwa
set in the
Exclude Skybo
Antialiasing
Method
Jitter
Spread
Blending
Stationary
Motion
Sharpen
Ambient Occlu
Screen Space
Depth Of Field
Motion Blur
Eye Adaptatio
Bloom
Color Grading
User Lut
Chromatic Ab
Spectral Texture
Intensity

새로운 에셋의 사용



UNITY TECHNOLOGIES

Cinemachine

★★★★★ 68개 유저리뷰

Cinemachine is unified procedural camera system for in-game cameras, cinematics and cutscenes, f
eSports solutions.

If you have a camera in your project, you'll benefit from Cinemachine being your camera system.

Cinemachine includes these components:

Procedural composition cinematically tracks and composes whatever target you define, be it an obje
operator which procedurally films the actions based on your direction of where you want it on screen

Follow modes mount cameras to objects with real-time offset tuning and per-axis dampening contro
options for following based on world angle, position delta and more.

더 보기

6.1 MB

최신 버전 2.1.10

2017.1.1 버전 이상

공유하기 목록에 추가하기

컴포넌트 분할

오브젝트의 기본 컴포넌트 및 rigidbody, 콜라이더

체력 및 타격을 받는 오브젝트용 스크립트

아이템 드랍 스크립트

적 AI 스크립트

“로봇”을 위한 스크립트

샷건 무기를 쏘는 오브젝트를 위한 스크립트

움직이는 적을 위한 스크립트

소리가 나는 오브젝트를 위한 스크립트

3D 모델링+애니메이션

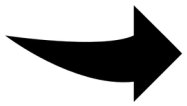
- 모델링과 애니메이션이 있어 든든했습니다.

발표 자료의 퀄리티

- 발표자료는 게임에 대한 첫인상을 좌우!
- 더 이상의 자세한 설명은 생략한다

이미 만든 스크립트의 재사용

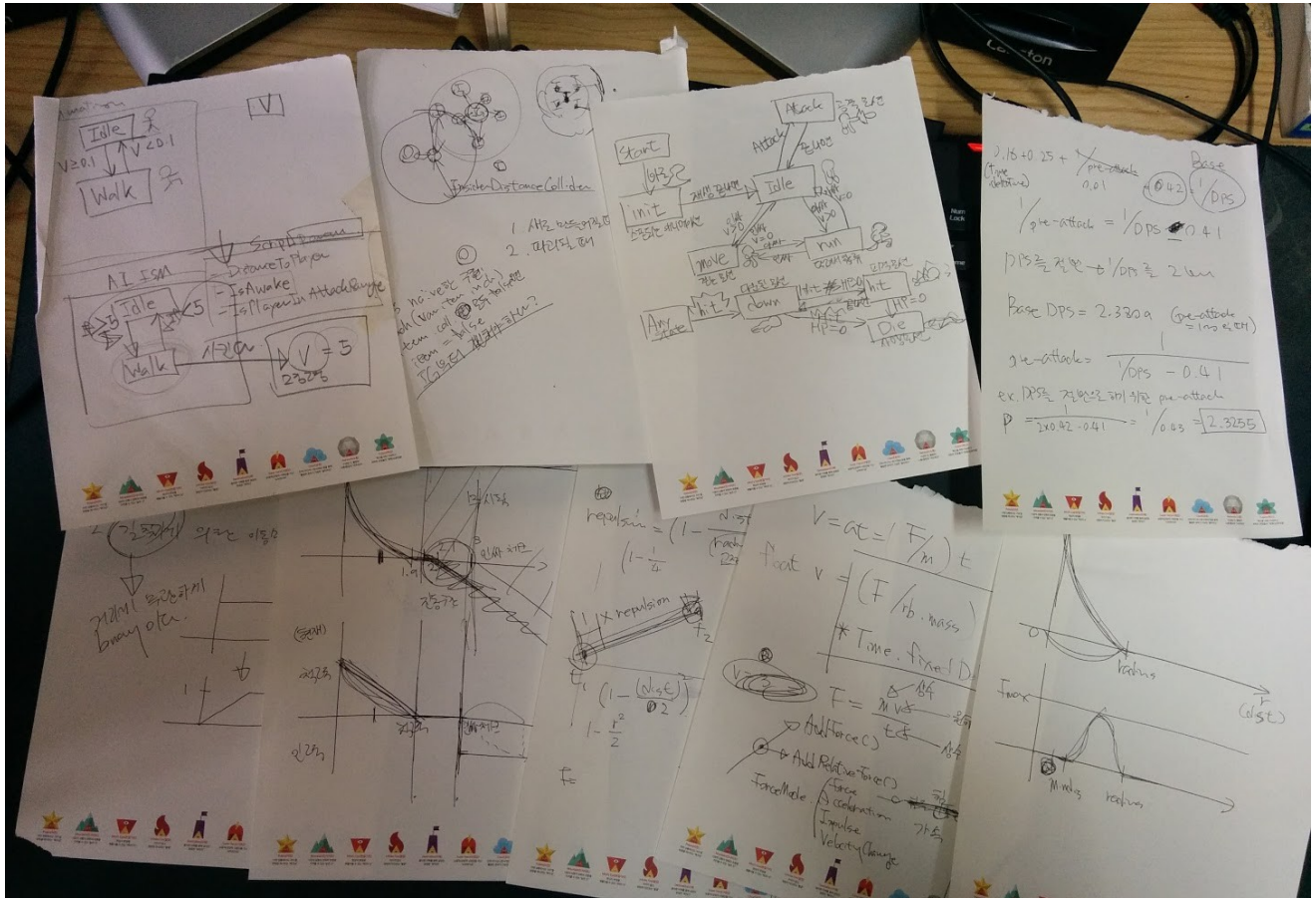
```
 using System.Collections.Generic;
using System.Linq;
using UnityEngine;

public enum SoundType { ChangeLight, PassGreen, PassYellow, PassYellowWithSunglass, PassRedWithPolice,
    BurningIntro, BurningStatus, BurningChangeLight, ItemEquip, ItemSpeed, ItemBurning }
public enum MusicType { Main, Ingame, GameOver }

[System.Serializable]
public class SoundDic{
    public SoundType type;
    public AudioClip clip;
}
public class SoundManager : MonoBehaviour{
    static SoundManager instance;
    static Queue<SoundPlayer> spPool;
    static SoundPlayer musicPlayer;
    static SoundPlayer burningPlayer;

    public GameObject standardSoundPlayer;
    public SoundDic[] soundDictionary;
    public AudioClip[] musicDictionary;
    static SoundPlayer GetSoundPlayer(){
        if (spPool.Count > 0) {
            SoundPlayer sp = spPool.Dequeue();
            sp.gameObject.SetActive(true);
            return sp;
        } else {
            GameObject go = Instantiate(instance.standardSoundPlayer, instance.gameObject.transform);
            return go.GetComponent<SoundPlayer>();
        }
    }
```



```
using System.Collections.Generic;
using System.Linq;
using UnityEditor;
using UnityEngine;

public enum SoundType { Player_Clone, Player_Footstep, Player_Dash, Player_Hit,
                        Weapon_Single_Fire, Weapon_Single_Hit, Weapon_Shotgun_Fire, Weapon_Shotgun_Hit,
                        Enemy_ShotgunRobot_Fire, Enemy_Burster_Fire, Enemy_Turret_Aim, Enemy_Turret_Fire,
                        Object_Spawner_Enable, Object_Spawner_Disable,
                        UI_Button_Hover, UI_Button_Click, UI_Button_Start,
                        Object_Door_Destruct_Start, Object_Door_Destruct_Finish, Object_CloneKitBox_Destruct,
                        Enemy_Death
                        }
public enum MusicType { Main, Ingame, GameOver }

[System.Serializable]
public class SoundDic{

    public SoundType type;
    public AudioClip clip;
    [Range(0, 256)] public int priority = 128;
    [Range(0, 1)] public float volume = 1;
    [Range(-3, 3)] public float pitch = 1;

}
/// <summary>
/// from code of SMZ
/// </summary>
[CustomEditor(typeof(SoundManager))]
[CanEditMultipleObjects]
public class SoundManager : Singleton<SoundManager>{

    public static float soundVolume = 1;
    public static float musicVolume = 1;

    static SoundPlayer musicPlayer;
    static SoundPlayer soundPlayer;

    public SoundDic[] soundDictionary;
    public AudioClip[] musicDictionary;

    public static AudioClip GetAudioClip(SoundType type)
    {
        return instance.soundDictionary.ToList().Find(dic => dic.type.Equals(type)).clip;
    }
```

계획을 하고 코딩을 하자

- 먼저 어떻게 코딩을 할 지 계획을 세우면, 나중에 삽질을 덜 해도 됩니다.
- 자료구조, 알고리즘 등



먼저 만들고,
그 뒤
최적화하자

- 계획만 하다 보면 구현을 못 한다
- 일단 만들고 최적화를 해도 늦지 않다
- 사실 최적화를 안 해도 된다

잘 안 된 점

기획/ 프로그래밍/ 모델링 사이의 지연

- 의견교환이 느렸다
- 더 빠른 의견교환/rapid prototyping의 필요성
- 동아리 프로젝트라서 한계도 있는 듯

(거의)온라인 only 회의

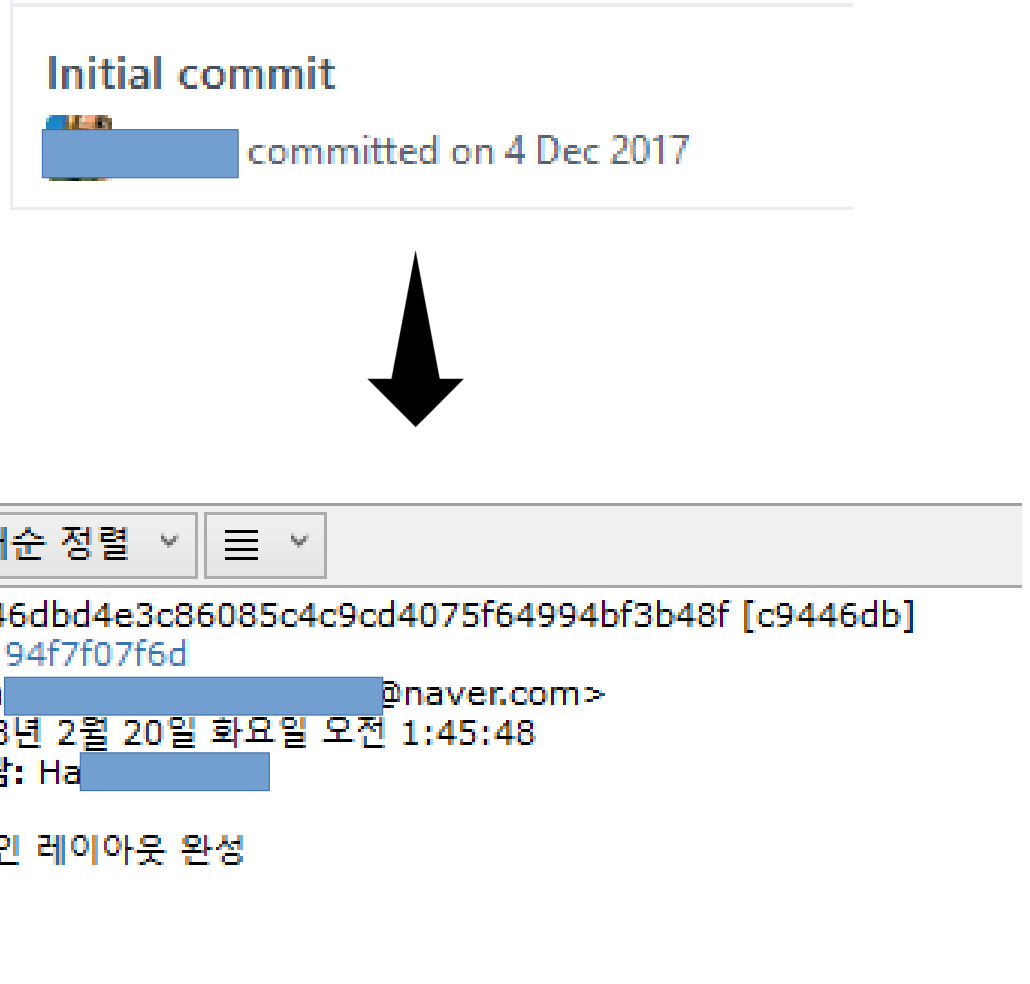
회의가 온라인으로만 진행되다보니 참여도가 떨어졌다

말이 잘 나오지 않는다

피드백이 느리다

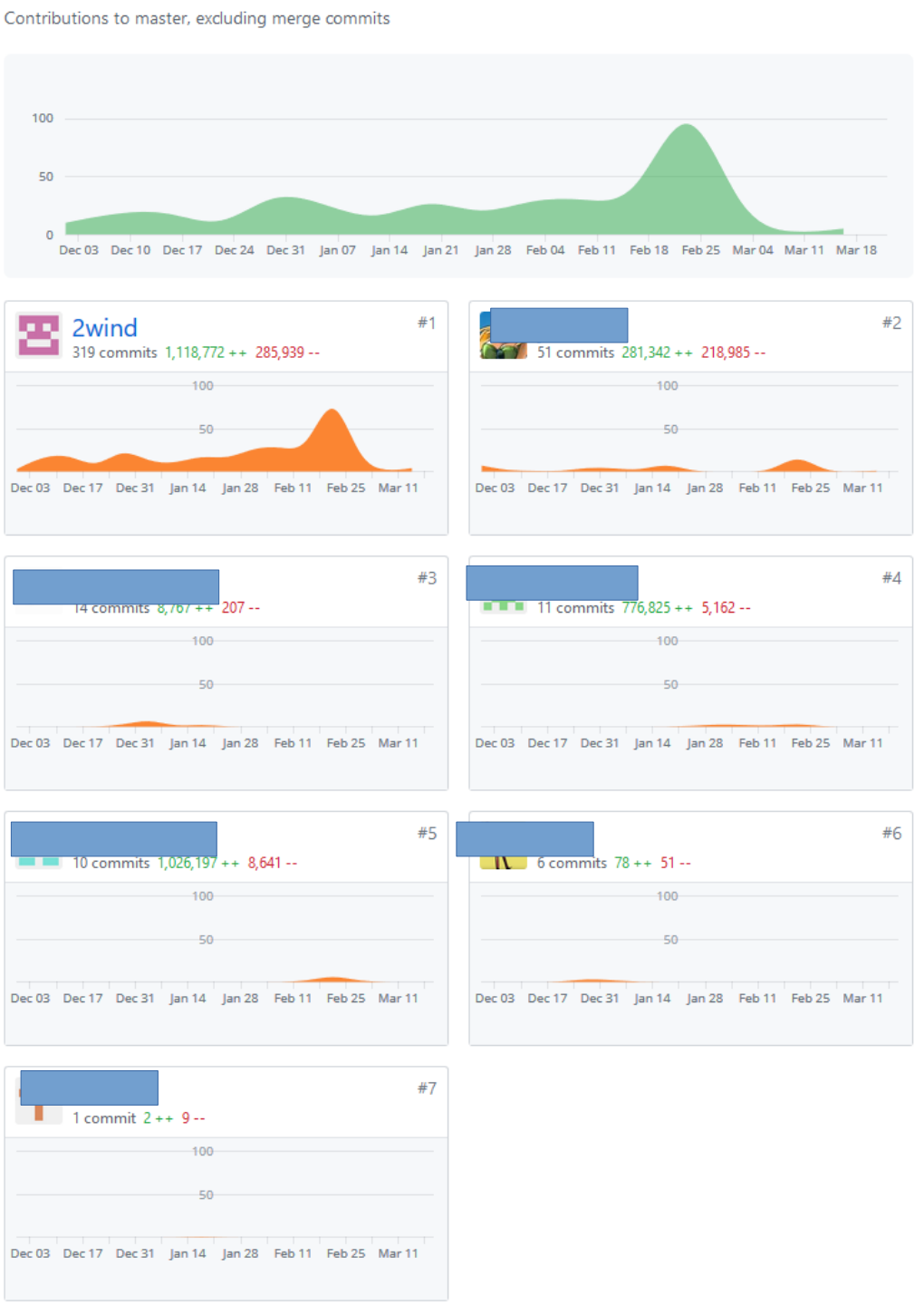
회의는 오프로 하고 기록을 동시에 남기는 것이 좋다

실제 맵 구현이 늦어짐


Initial commit
committed on 4 Dec 2017
파일 상태순 정렬
커밋: c9446dbd4e3c86085c4c9cd4075f64994bf3b48f [c9446db]
상위 항목: 94f7f07f6d
작성자: Ha @naver.com>
날짜: 2018년 2월 20일 화요일 오전 1:45:48
커밋한 사람: Ha
map 디자인 레이아웃 완성

프로그래밍 인원 부족

- 구인을 처음부터 잘 했어야 하나…
- 어느 정도는 어쩔 수 없는 것 같습니다.



코드 리딩의 문제

```
IEnumerator Start()
{
    if (map == null) yield break;
    resultPixels = new Color[map.width * map.height];
    while (target == null) yield return null;
    while (true)
    {
        var targetPosition = WorldToPixelSpace(target.position);
        yield return
            StartCoroutine(DistanceMapCalculator.CalculateFlowMap(map, null, targetPosition, resultPixels));
        calculated = true;
    }
}

Vector2Int WorldToPixelSpace(Vector2 targetPosition)
{
    var center = new Vector2(map.width / 2f, map.height / 2f);
    return new Vector2Int(Mathf.FloorToInt(targetPosition.x + center.x),
        Mathf.RoundToInt(targetPosition.y + center.y));
}

// 일부 벽을 외부로 판정하는 버그가 있으므로, 특히 Update()에서 돌리지 말 것.
public static bool IsInMap(Vector2 position)
{
    var pos = instance.transform.InverseTransformPoint(position);
    var pixelPos = instance.WorldToPixelSpace(pos);
    var map = instance.map;

    if (map.GetPixel(pixelPos.x, pixelPos.y).Equals(Color.black))
    {
        return false;
    }

    return pixelPos.x >= 0 && pixelPos.x < map.width
```

```
public static IEnumerator CalculateFlowMap(Texture2D target, Texture2D render, Vector2Int start,
    Color[] returnPixels)
{
      var watch = new Stopwatch();
      watch.Start();
    if (prevTarget == null || prevTarget != target)
    {
        pixels = target.GetPixels();
        result = new float[pixels.Length];
        resultPixels = new Color[pixels.Length];
    }
    prevTarget = target;
    var width = target.width;
    var height = target.height;
    var distTask = Task.Run(() => CalculateDistanceMapAsync(pixels, width, height, start, result));
    var flowTask = distTask.ContinueWith(dmap => CalculateFlowMapAsync(width, height, dmap.Result, resultPixels
    while (!flowTask.IsCompleted) yield return null;
      Debug.Log(watch.Elapsed);
    for (var i = 0; i < returnPixels.Length; i++) returnPixels[i] = flowTask.Result[i];
    if (render == null) yield break;
    render.SetPixels(flowTask.Result);
    render.Apply();
}

static float[] CalculateDistanceMapAsync(Color[] map, int width, int height, Vector2Int start, float[] resultA
{
    Func<Vector2Int, int> V2I = v => Vec2Index(width, height, v);
    var distMap = resultArray;
    for (var i = 0; i < distMap.Length; i++) distMap[i] = 0;
    var startPoints =
        new List<Vector2Int> {start, start + Vector2Int.right, start + Vector2Int.up, start + Vector2Int.one};
    foreach (var v in startPoints) distMap[V2I(v)] = 1;
    return Calculate(V2I, distMap, startPoints, new List<Vector2Int>(), map);
}

static Color[] CalculateFlowMapAsync(int width, int height, float[] distanceMap, Color[] resultArray)
{
    for (var x = 0; x < width; x++)
    {
        for (var y = 0; y < height; y++)
        {
            SetFlowMapPoint(x, y, width, height, distanceMap, resultArray);
        }
    }
    return resultArray;
```

작업 트래킹 도구의 미비함

일정관리 트렐로 사용하자 -> ㅇㅋ 혼자 써서 유명무실해짐

문서저장 -> 구글드라이브 정리가 안되었음

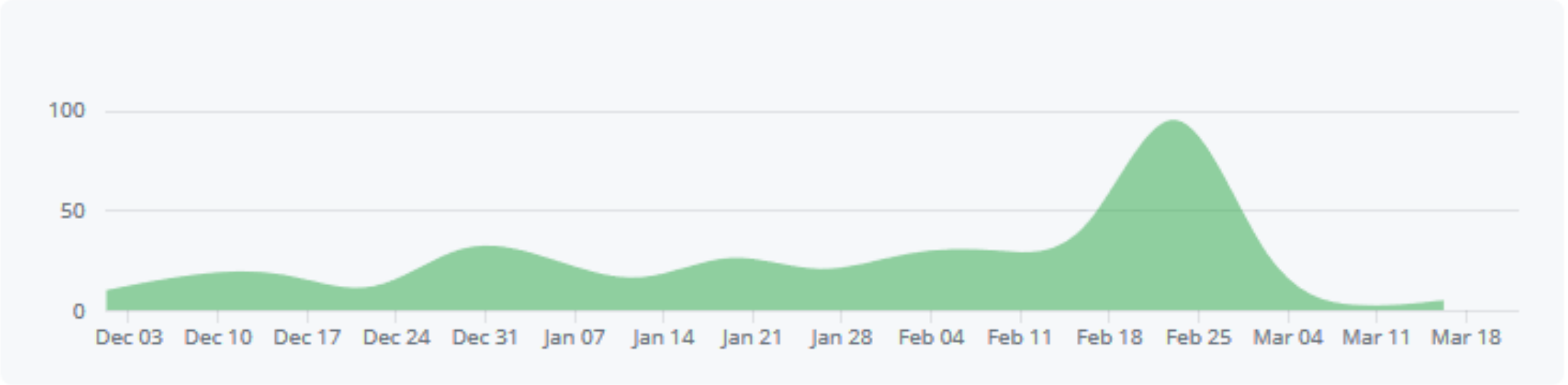
메신저 -> 카카오톡 카톡으로 회의하면 안됨

소스코드 -> 깃헙 잘됨

Dec 3, 2017 – Mar 24, 2018

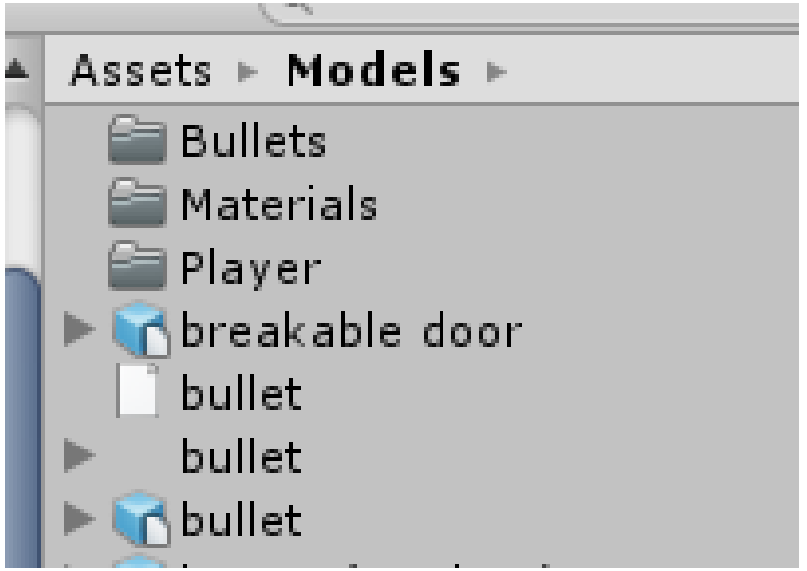
Contributions: **Commits**

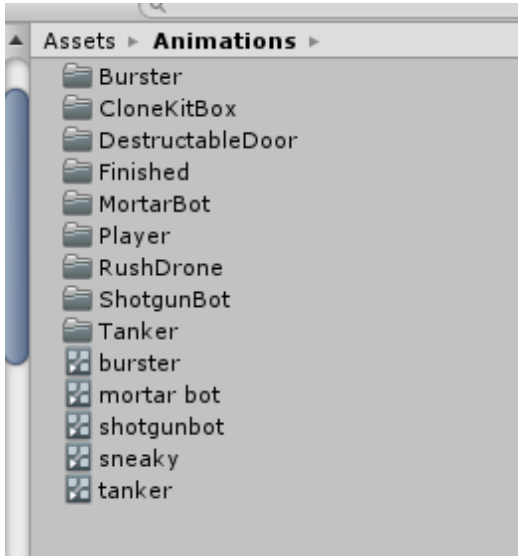
Contributions to master, excluding merge commits

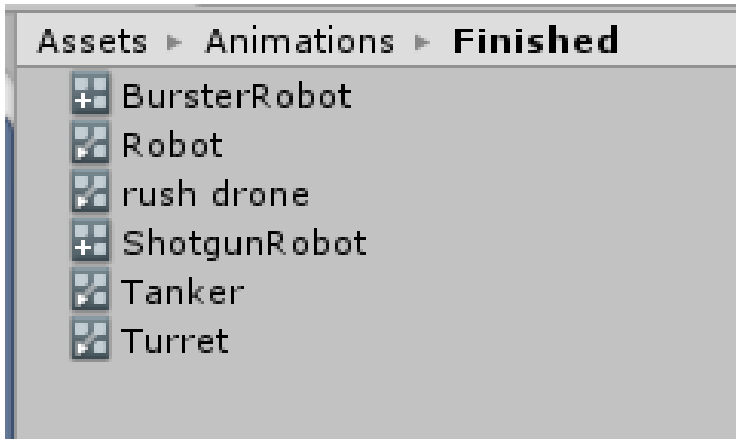


일의 배분을 고르게 하자

바이너리 파일 관리의 문제

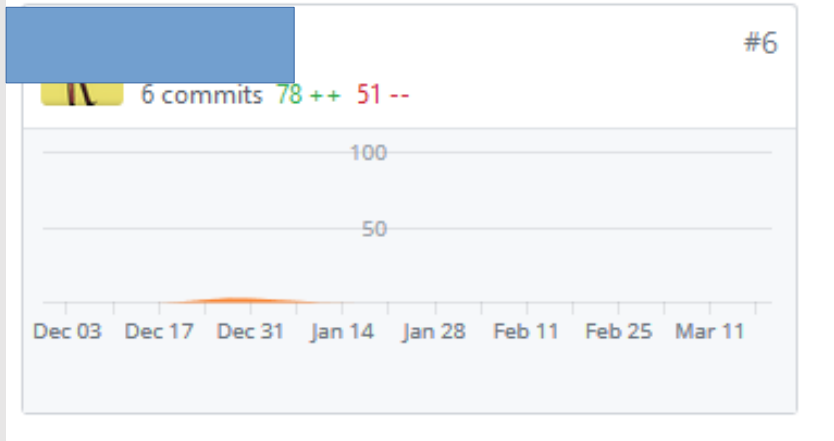






사라진 사람들

- 과연 무슨 문제가 있었던 것일까요?



코드 퀄리티

```
// Use this for initialization
void Awake()
{
    Clean();
    //Initialize();
    // Set up references.
    // anim = GetComponent<Animator>();
}

void Start()
{
    StartCoroutine(DistanceWorkCoroutine());
}

public void Clean()
{
    CharacterPool.ClearPool();
    characters.Clear();
    charPairs = new HashSet<Tuple<Character, Character>>();
    distanceWorker = new CharacterDistanceWorker();
    distancePairs = new Dictionary<Character, List<Character>>(
    //var targetGO = new GameObject("PathFinder Target");
    //target = targetGO.transform;
    var targetGO = transform.Find("PathFinder Target");
    target = targetGO.transform;
    target.SetParent(transform);
    groupCenter = this;
    xyPlane = new Plane(Vector3.forward, Vector3.zero);
    emittingCount = 0;
}
```

```
public void RemoveCharacter(Character character, bool silently = false)
{
    if (characters.Count > 1 && leader.Equals(character))
    {
        characters.Remove(character);
        ResetCenterOfGravityCharacter();
    }
    else
    {
        characters.Remove(character);
    }
    charPairs.RemoveWhere(p => p.Item1 == character || p.Item2 == character);
    character.GetComponent<SoundPlayer>().SetPlayable(!silently);
    character.SendMessage("KillCharacter");
    if (!characters.Any() && SceneLoader.instance.isLoadedSceneInGame)
    {
        //GAMEOVER 게임오버 처리
        input = Vector2.zero;
        GetComponent<AudioSource>().Stop();
        if (!ScoreBoard.instance.isMapCleared)
        {
            ScoreBoard.instance.StopTracking();
            GameManager.GetComponent<GameOverPanel>().SetGameOverPanel(true);
        }
    }
}
```

```
public void Initialize()
{
    Vector2 startPosition = Vector2.zero;
    try
    {
        startPosition = GameObject.Find("StartPosition").transform.position;
        groupCenter.transform.position = startPosition;
        GameManager.GetComponent<SceneLoader>().mainVCamera.transform.position =
            (Vector3) startPosition + new Vector3(0, -20, -20);
    }
    catch (Exception)
    {
        if (Debug.isDebugBuild)
            Debug.LogError("Start Position not found!");
        throw;
    }
    if (!characters.Any())
    {
        leader = AddCharacter();
        leader.isInsider = true;
    }
    foreach (var item in characters)
    {
        Vector2 diff = item.transform.position - transform.position;
        var tempPosition = startPosition + diff;
        while (!PathFinder.IsInMap(tempPosition))
        {
            tempPosition = tempPosition + PathFinder.GetAcceleration(tempPosition);
        }
        item.transform.position = tempPosition;
    }
    distanceWorker = new CharacterDistanceWorker();
    distancePairs = new Dictionary<Character, List<Character>>();
    transform.position = startPosition;
    StopCoroutine(nameof(DoInsiderCheck));
    StartCoroutine(nameof(DoInsiderCheck));
    PathFinder.instance.target = target;
    ScoreBoard.instance.StartNewTracking();
}
```

총평

- 많은 것을 배울 수 있었던 프로젝트
- 새로운 요소를 여럿 도입했던 프로젝트
- 팀원이 더 있었다면 좋았을 프로젝트
- 팀원간의 소통이 더 필요했던 프로젝트
- 그래도 재미있게 일 할 수 있었던 프로젝트