

# Portfolio

Oh HyunJong

オ ヒャンジョン

吳晥鍾

오현종

Email : ohj106992@gmail.com

Mobile : +82 10 3345 2041



# Portfolio

## 01 自己紹介

- ・ 本人紹介
- ・ 教育機関
- ・ 保持技術

## 02 使用技術

- ・ プロジェクトの説明
- ・ 企画理由
- ・ 技術説明

## 03 プレイ映像

- ・ YouTube

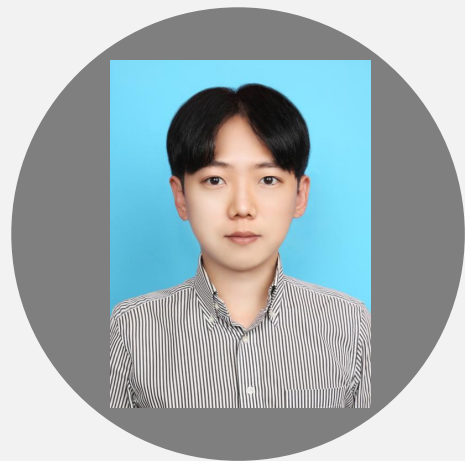
# 01 自己紹介

---

履歴書の内容以外の簡単な自己紹介をさせていただきます。

自己紹介

無限に挑戦する



Oh Hyun Jong

1996. 02. 11

+82 10 3345 2041

ohj106992@gmail.com

Career

大学卒業 2023. 02. 15

Education

大学 電気工学科  
ゲームアカデミー Unity

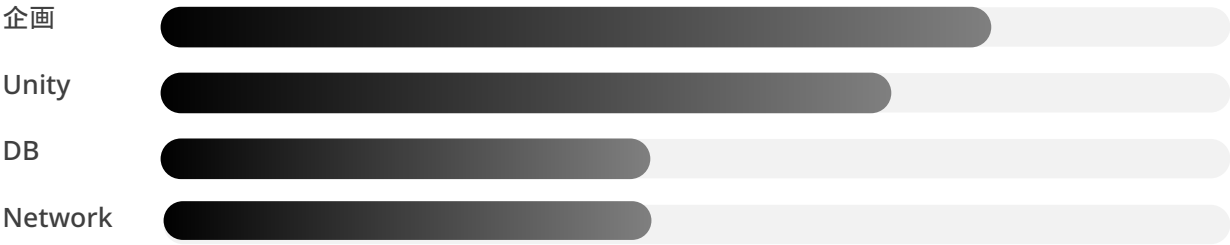
Other Experience

兵役 海軍兵長満期全域  
日本居住経験 ウォーキングホリデー(大阪)

Awards

模範賞 アカデミー修了の時

Skills & Ability





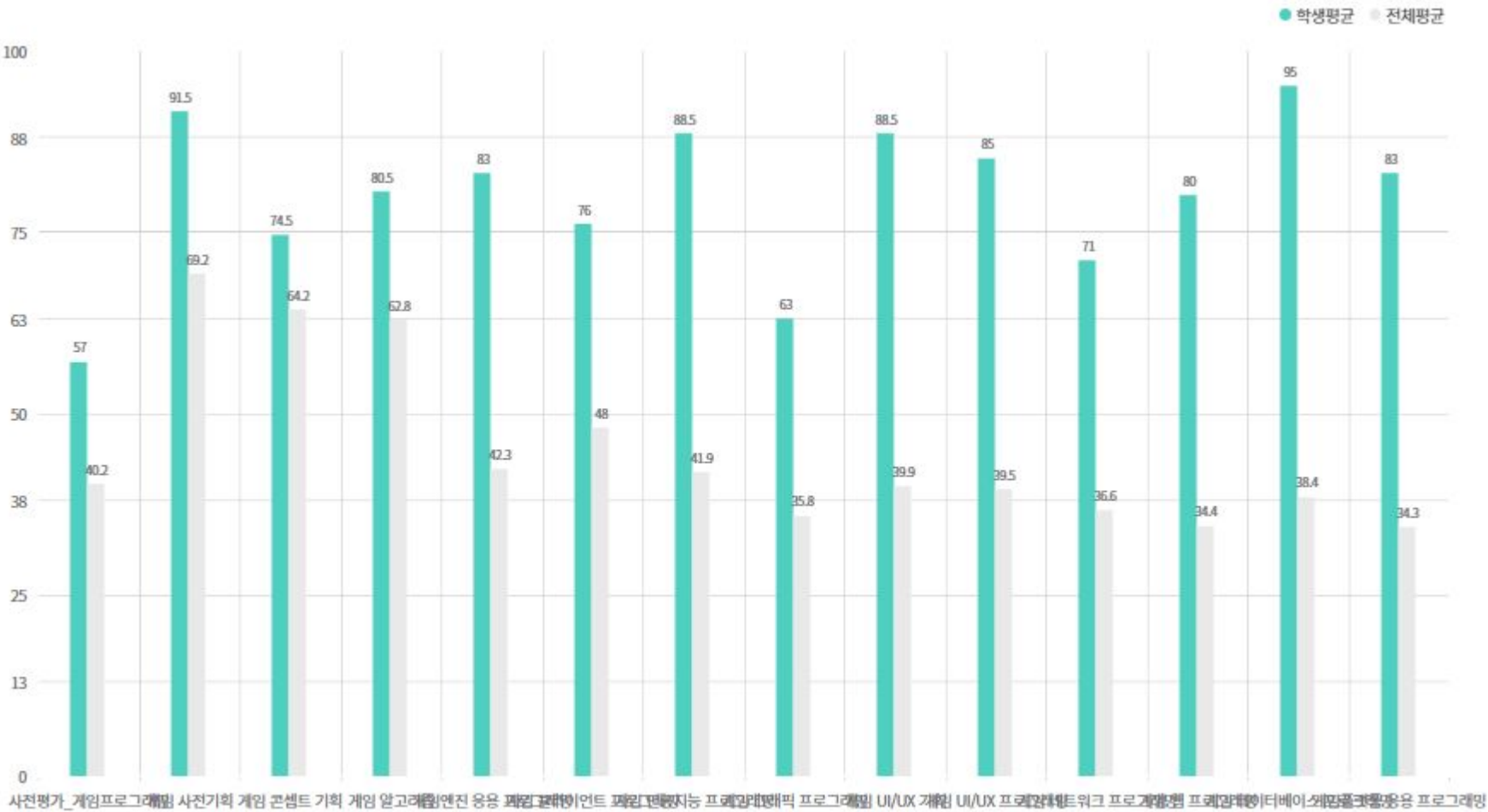
# アテンツゲームアカデミー

国費支援アカデミーとして、現業従事者の教育により、6ヶ月間のゲーム企画からポートフォリオ制作に必要な様々な技術の基礎を学ぶことができました。

번호	능력단위명	평가자	평가일	본평가점수	석차	최종점수
1	사전평가_게임프로그래밍		2022.07.25	57	7/32	57
2	게임 사전기획	이규훈	2022.07.29	91.5	2/32	91.5
3	게임 콘셉트 기획	이규훈	2022.08.04	74.5	16/32	74.5
4	게임 알고리즘	이규훈	2022.08.31	80.5	14/32	80.5
5	게임엔진 응용 프로그래밍	이규훈	2022.09.30	83	5/32	83
6	게임 클라이언트 프로그래밍	이규훈	2022.10.07	76	8/32	76
7	게임 인공지능 프로그래밍	이규훈	2022.10.14	88.5	1/32	88.5
8	게임 그래픽 프로그래밍	이규훈	2022.11.04	63	11/32	63
9	게임 UI/UX 기획	이규훈	2022.11.09	88.5	2/32	88.5
10	게임 UI/UX 프로그래밍	이규훈	2022.11.21	85	3/32	85
11	게임 네트워크 프로그래밍	이규훈	2022.12.12	71	13/32	71
12	게임 웹 프로그래밍	이규훈	2022.12.21	80	3/32	80
13	게임 데이터베이스 프로그래밍	이규훈	2023.01.02	95	1/32	95
14	게임플랫폼 응용 프로그래밍	이규훈	2023.01.19	83	2/32	83

National Competency Standards基盤の企画、アルゴリズム、UI/UX、ネットワーク、データベースなどの教科目ごとの単位評価を通じて本人技量を測定して進めてきました。

# アテンツゲームアカデミー



グラフは、評価ごとの本人スコアを緑色で、学生全体のスコア平均をグレーで表します。  
学生全体の平均を上回る良い成績をもとに模範賞を受賞し、6ヶ月間の教育期間を修了しました。

# 保有技術の簡単な説明

保有技術	技術名	保有能力技術
	Git	githubを使ってプロジェクトを進めた経験があります。
	DB	MySQLを通じてクエリステートメントとview、stored procedureを使ってみた経験があります。
	Network	.NETベースのシンプルなC / Sモデルを設計できます。(エコ通信)
	C	C言語ベースの資料構造を勉強しました。
	C++	単純な与えられた問題のコードを生成することができ、STLを理解しています。
	C#	Unityを主力で勉強したので、C #をよく使ってきました。
	Unity	現在、3ヶ月間ポートフォリオ用デモゲームを制作中です。

# 02 使用技術

---

ポートフォリオの作成に使用された技術の説明があります。



# 事前調査

調査は22年7月の企画段階の教育を受けた際に実施したため、現在とは違いがあります。

モバイルゲーム当日の売上ランキング (韓国Google Playベース) / 調査日時:22.07.25				
ランク	ゲーム名	ディストリビューター	ジャンル	発売日
1	ウマ娘 Pretty Derby	カカオゲームズ	シミュレーション	22.06.20
2	リネージュ M	NCソフト	MMORPG	17.06.21
3	リネージュ W	NCソフト	MMORPG	21.11.04
4	オーディン	カカオゲームズ	MMORPG	21.06.29
5	リネージュ 2M	NCソフト	MMORPG	19.11.27
6	DNF モバイル	ネクソン	ベルトスクロールアクション	22.03.24
7	ヒーローズテイルズ	37ゲームズ	MMORPG	21.07.01
8	ミルM	ウィメイド	MMORPG	22.06.23
9	リネージュ 2 レボリューション	ネットマーブル	MMORPG	16.12.14
10	ディアブロ イモータル	ブリザード	アクションRPG	22.06.01

売上高の大多数がRPG系のゲームであることを確認することができ、過去のオンラインゲーム「リネージュ」のIPを利用したゲームがなんと4つもランクであることが確認できます。

## 企画モチーフ

---

2007年から2009年までサービスしていた TheosオンラインというMMORPGをモチーフにして企画しました。



**タイトル：テオスオンライン (THEOS online)**

**プラットフォーム：Online(PC)**

**ジャンル：MMORPG**

**サービス/ランチング：2007.02(商用化)～2009.03(サービス終了)**

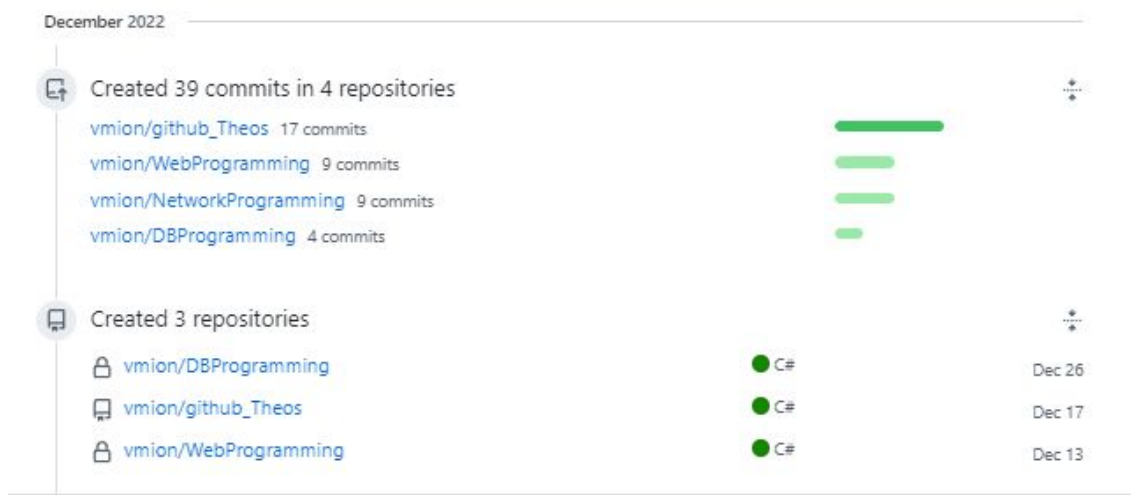
**ディストリビューター：CJネットマーブル**

最近の新作ゲームの大半は、従来のIPを使って作る方法を採用しています。

サービスされていたときに楽しくプレイしたこともありましたが、ギリシャローマ神話というフレンドリーなIPを使ったという点で大きな意味があると考え、これを利用して製作してみました。

# 開発過程

アカデミーで勉強しながら 22年12月に開発を始めました。



開発人員 : 1人

開発環境 : Unity

開発期間 : 3ヶ月



# 技術説明

---

ポートフォリオに実装されている、  
または実装予定の技術リスト

1. ロードシーン、タイトルメニューの実装
2. テーブル(csv,json,xml)ベースパーサーの実装
3. テーブルベースの自動配置
4. Mecanim状態クラスの使用
5. インベントリシステムの実装
6. ミニマップシステム
7. キャラクターやモンスターのスキル実装
8. オブジェクトプーリングの実装
9. 三人称カメラの実装
10. クエストシステムの実装
11. Colliderを使用していない衝突処理
12. Nav Meshの使用
13. Post Processingの使用
14. 自動コライダーボックス保存コンポーネント製作
15. マルチマップベースの有限状態マシンの実装
16. 自動狩りの実現
17. 自動パッチ実装
18. GpuInstancing機能を使用したインスタンスの作成
19. 頂点カラーとチャンネルマップを利用した地形シェーダ製作

# 技術説明

ポートフォリオに実装されている、  
または実装予定の技術リスト

## 2. テーブル(csv,json,xml)ベースパーサーの実装

## 3. テーブルベースの自動配置

	A	B	C	D	E	F	G	H	I	J	K	L
1	index	name	parent	posx	posy	posz	rotx	roty	rotz	scalex	scaley	scalez
2	1	CenterTree	-1	0	0	0	270	217,1209	0	1	1	0,5
3	2	tree	-1	90,41689	7,315983	0	273,6899	90,0753	65,00021	1,993218	1,998521	1,002075
4	3	tree (1)	-1	57,57848	49,84505	0,558004	273,6899	90,07544	92,13269	1,993218	1,998521	1,002075
5	4	tree (2)	-1	14,66867	71,34441	1,199994	270	-2,41E-06	0	2	2	1
6	5	tree (3)	-1	-14,7381	93,60145	0,299993	270	-2,41E-06	0	2	2	1
7	6	tree (4)	-1	-58,6121	64,92937	0,299995	270	-2,41E-06	0	1,399999	2	0,7
8	7	tree (5)	-1	-90,4734	22,83308	0,499999	270	90	0	2	2	0,7
9	8	tree (6)	-1	-112,589	-6,38685	0,500002	270	90	0	1,399999	2	0,7
10	9	tree (7)	-1	-67,52	-40,5	0,6	270	90	0	1,399999	2	0,7
11	10	tree (8)	-1	2,329372	-102,871	0,300012	270	90	0	1,399999	2	0,7
12	11	mushroom	-1	58,57413	-47,6617	-0,0904	-2,45E-06	153,196	-2,93E-06	16,84141	8,420711	16,84141
13	12	flower2	-1	-46,1765	-70,3224	0,790005	270,6667	246,9993	94,35676	81,57638	568,3495	1,000068
14	13	flower2 (1)	-1	-28,8446	-80,4304	0,680003	270,6667	246,9993	94,35676	81,57638	568,3495	1,000068
15	14	tree (9)	-1	22,84253	-90,7809	0	270	245	0	6	6	3
16	15	tree (10)	-1	17,22	-125,83	0	270	245	0	6	6	3
17	16	tree (11)	-1	88,2	-30	-0,47	273,6899	90,0753	65,00021	1,993218	1,998521	1,002075
18	17	tree (12)	-1	82,4	30,9	0,47	273,6899	90,07544	92,13269	1,993218	1,998521	1,002075
19	18	tree (13)	-1	42,7	50,1	1,2	270	-2,41E-06	0	2	2	1
20	19	tree (14)	-1	-35,66	72,37	0,3	270	-2,41E-06	0	1,399999	2	0,7
21	20	tree (15)	-1	13,3	72,4	0,3	270	-2,41E-06	0	2	2	1
22	21	tree (16)	-1	-72,83	38,91	0,5	270	90	0	2	2	0,7
23	22	tree (17)	-1	-84,6	-27,6	0,5	270	90	0	1,399999	2	0,7
24	23	tree (18)	-1	-39,5	-61,7	0,49	270	90	0	1,399999	2	0,7
25	24	tree (19)	-1	37,7	-97,9	0,3	270	89,99989	0	1,399999	2	0,7
26	25	flower2 (2)	-1	-23,2	-88,7	0,62	270,6667	246,9993	94,35676	81,57638	568,3495	1,000068
27	26	flower2 (3)	-1	-62,04	-57,64	0,89	270,6667	246,9993	94,35676	81,57638	568,3495	1,000068
28	27	flower2 (4)	-1	-105,1	-4,29	1,29	270,6667	246,9993	94,35676	81,57638	568,3495	1,000068
29	28	tree (20)	-1	-3,74	85,28	0,3	270	-2,41E-06	0	2	2	1
30	29	tree (21)	-1	26	62,78	0,3	270	-2,41E-06	0	2	2	1
31	30	tree (22)	-1	57,98	-77,89	0,3	270	89,99989	0	1,399999	2	0,7

[MenuItem("MapData")] を使用して地形オブジェクトを csv ファイルとして保存し、その後 csv ファイルの内容に基づいて地形オブジェクトをロードして特定のシーンに配置します。

添付スクリーンショット以外の部分はプレイ映像でご確認ください。

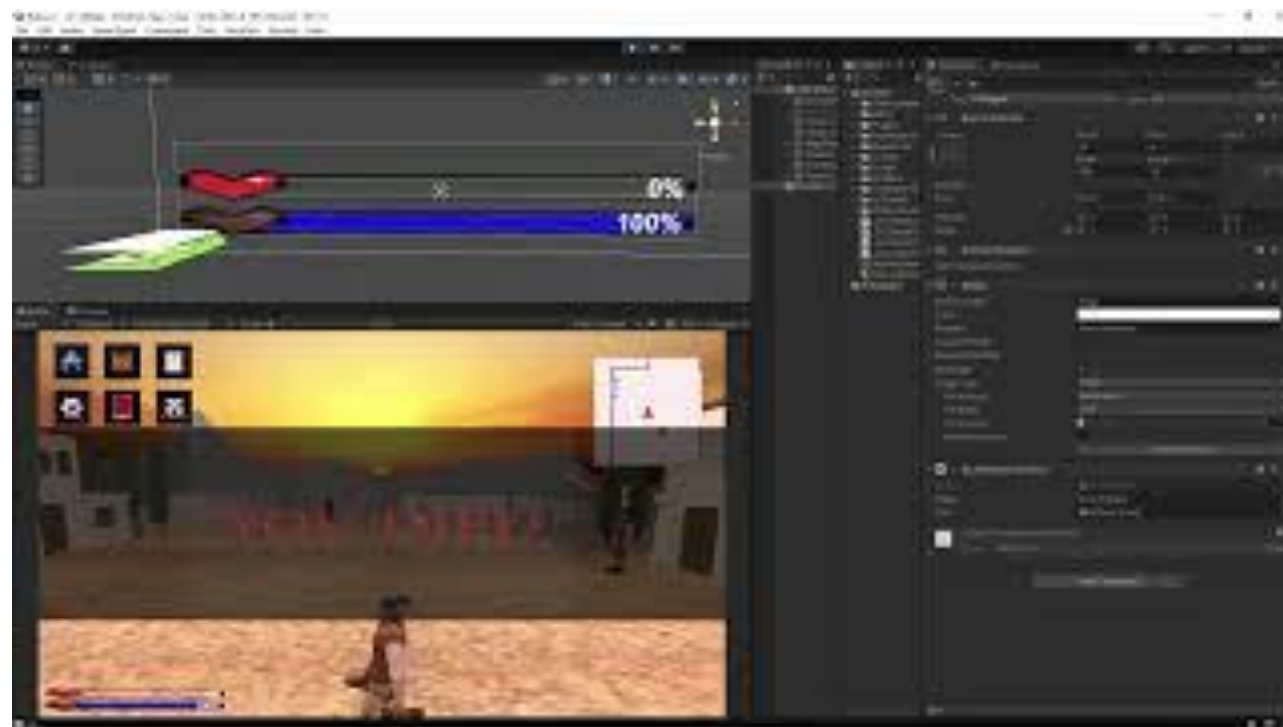
# 技術説明

---

ポートフォリオに実装されている、  
または実装予定の技術リスト

## 4. Mecanim状態クラスの使用

<https://youtu.be/WnoLccJ4wKM>



HPをfilled方式で実装し、これが0になった時に死亡する Mecanimアニメーションが再生されます。

添付スクリーンショット以外の部分はプレイ映像でご確認ください。



# 技術説明

ポートフォリオに実装されている、  
または実装予定の技術リスト

## 8. オブジェクトプーリングの実装

## 11. Colliderを使用していない衝突処理

```
@Unity 메시지 |참조 0개
private void OnTriggerEnter(Collider other)
{
    if (other.gameObject.tag == "weapon")
    {
        hp -= damage;
        if (hp <= 0)
        {
            CharInventory.AddCharItem(item);
            Vector3 particlePos = new Vector3(0, 1, 0);
            ParticleSystem cloneParticle = Instantiate(particle, transform.position + particlePos,
                transform.rotation);

            Destroy(cloneParticle.gameObject, cloneParticle.main.duration);
            gameObject.SetActive(false);
        }
    }
}
```

左側はモンスターをオブジェクトプールしたコードで、右側はコライダーの boundsをチェックしてマップ間の移動を行ったコードです。

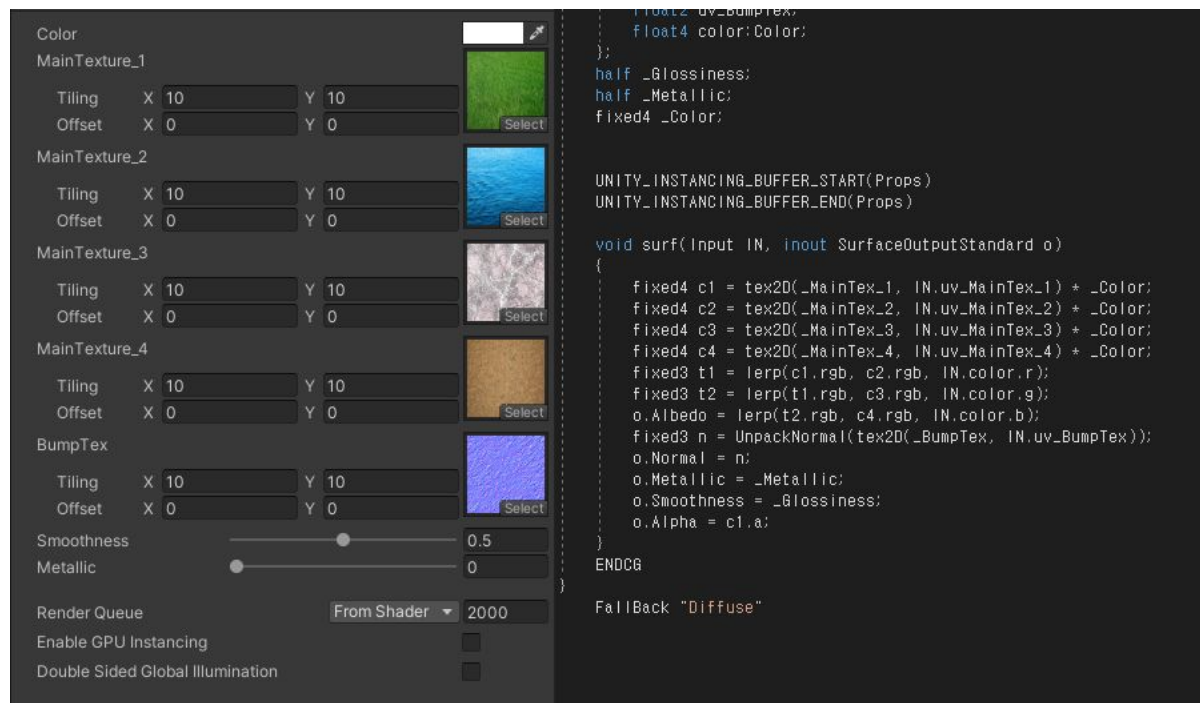
```
참조 2개
public bool COLLISIONCHECK { get; set; }
@Unity 메시지 |참조 0개
void Start()
{
    ObjCollider = GetComponent<Collider>();
    Player = Character_Manager.playerCollider;
    COLLISIONCHECK = true;
}
@Unity 메시지 |참조 0개
void Update()
{
    if (Player.gameObject.transform.GetComponentInParent<Char_ani>().enabled == true)
    {
        if (COLLISIONCHECK)
        {
            if (ObjCollider.bounds.Intersects(Player.bounds))
            {
                caution.SetActive(true);
            }
            else
            {
                caution.SetActive(false);
            }
        }
    }
}
```

添付スクリーンショット以外の部分はプレイ映像でご確認ください。

# 技術説明

ポートフォリオに実装されている、  
または実装予定の技術リスト

## 19. 頂点カラーとチャンネルマップを利用した地形シェーダ製作



Multi Vertex ShaderとPolyBrushのRGB値を使用して、Planeに  
さまざまなテクスチャを一度に適用しました。

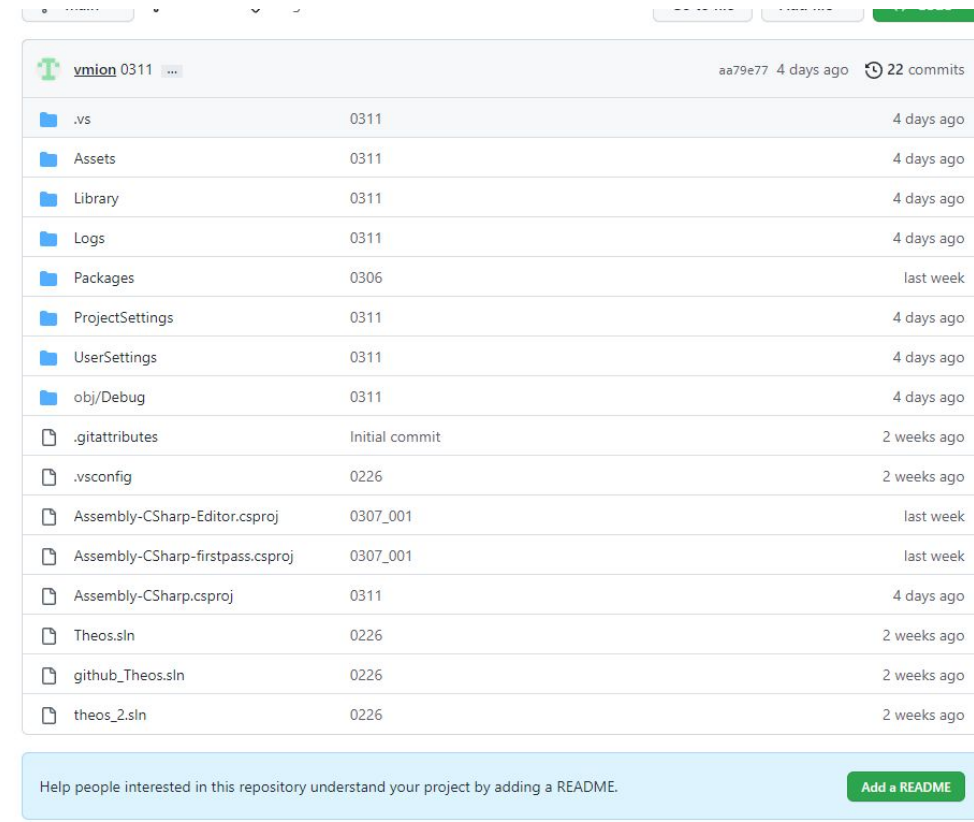
添付スクリーンショット以外の部分はプレイ映像でご確認ください。

# ソースコード

## Githubリンク

Githubで作成し、詳細なソースコードへのリンクを残します。

[https://github.com/vmion/theos\\_2](https://github.com/vmion/theos_2)



他にもアカデミーで勉強した内容をテーマ別にgithubにアップロードしておき、簡単なモザックゲーム2つのコードまた、「UnityStudy」というレポジトリにコードが保存されています。

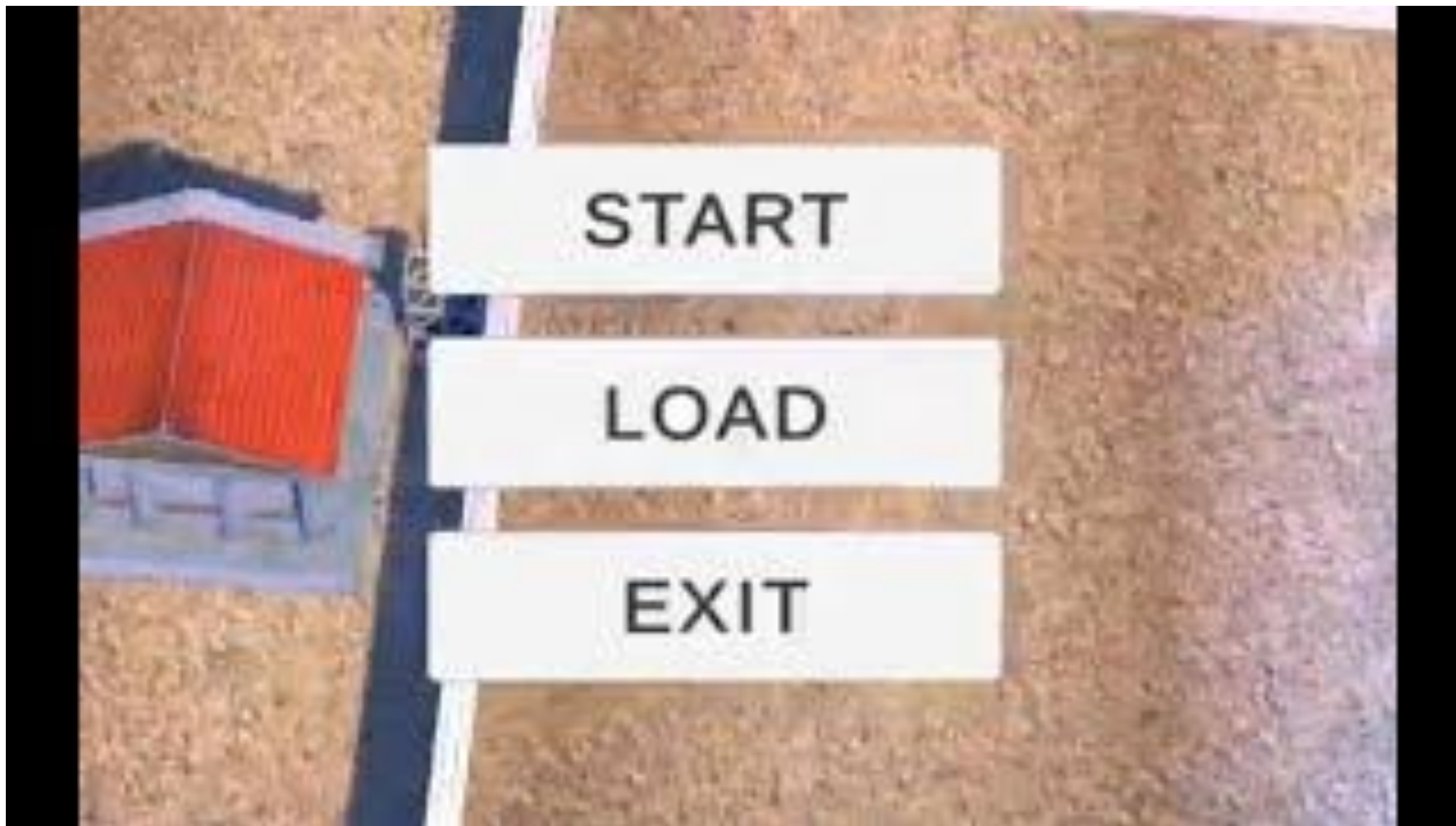
# 03 プレイ映像

---

まだ補完する部分が多いですが、制作したゲームの映像を添付します。

# プレイ映像

---





以上で終了させていただきます。  
ありがとうございます。