



PORTFOLIO

廣川 鳳介

目次

自己紹介	1 P
ツール作品「OraraEngine」	2 ~ 9 P
ゲーム作品4作品	10~14P
ゲーム大賞応募作品	11P
学内コンテスト作品	12P
スマートフォン個人制作作品	13P
プチコン応募作品	14P

自己紹介

— 1 P —



ひろかわ ほうすけ
名前：廣川 鳳介

生年月日：2002年12月26日

学校名：HAL東京

出身地：静岡県

趣味：カラオケ、WEB小説

資格・受賞歴

- ・ 基本情報技術者
- ・ J検 情報システム試験 プログラミングスキル
- ・ 情報システム試験 プログラマ認定
- ・ J検 情報システム試験 基本スキル
- ・ CGエンジニア検定ベーシック

自分が使用できるソフトや言語など

言語

- ・ 使用歴 1年：C
- ・ 使用歴 3年：C++
- ・ 使用歴 2年：C#

エンジン

- ・ 使用歴 2年：Unity
- ・ 使用歴 1年：UnrealEngine4

自己PR

私の長所は一つの目標に向かって最後まで取り組むことができることです。任せられた仕事は自分できうる限りの力をもって取り組み、わからなければ調べたり、聞いたりして実装していきます。学校でゲーム大賞に向けてのチーム制作の際にあまりうまくチーム運営ができておらず、プログラマーが8人中私を含め2人だけという状況になってしまい、チーム全体のモチベーションが低下していました。そんな中でも私は残ったメンバーと制作に取り組み、レベルデザインや素材の発注、背景オブジェクトの設置そして内部のプログラミングをおこないました。そして最終的には完成まで作り上げることができコンテストでは一次審査を通過することができました。この経験は入社後にも役立つと考えています。この長所を生かし、与えられた業務は責任をもち最大限の努力で取り組む最後まで作り切ります。

作品Googleドライブ

https://drive.google.com/drive/folders/11O1aK1JnY-Nazwq15dvGv60boJZhQDB3?usp=drive_link

Githubアカウント名
hodukr

ツール作品

—2P—

「OraraEngine」



作品紹介

就職作品「OraraEngine」

制作人数 2人

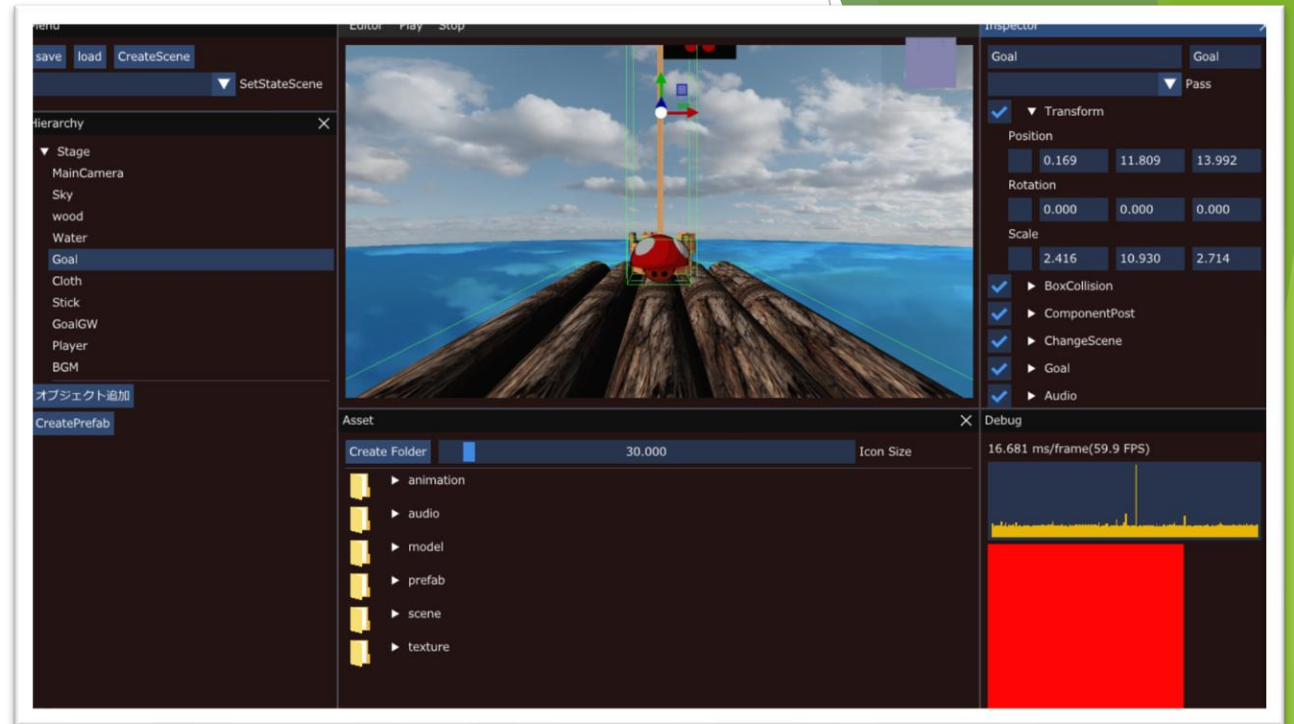
制作日数 約4カ月

制作環境 DirectX11

使用ソフトや言語

- VisualStudio2022
- DearImgui
- ImGuizumo
- Cereal
- GitHub
- Sourcetree
- C++

—3P—



技術研究を目的にUnityを参考に制作を行いました。二人での制作で私は主にデータ保存とGUI出力の制作を行いました。また、パーティクルやPlayerなどの一部コンポーネントの制作や当たり判定の処理も一部制作を行いました。

GitHubURL

<https://github.com/hodukr/OraraEngine>

紹介動画

https://drive.google.com/file/d/1TSSNNYAqRJgSokX7r3nrWy4nsuFBv-LL/view?usp=drive_link

作品紹介「OraraEngine」

—4P—


何故作り始めたか。

→入学してからゲームを制作する上でツール制作の方にも興味を持ったからです。

元々は三人で制作を行っていましたがマージを行う上でコンフリクトが発生したり、描画順番制御がうまくいかなかったり、データ読込の整合性が取れずにエラーが発生し、制作の進行が行えなくなってしまいました。その間にインターンシップ授業が始まり完全に制作が止まってしまいました....



そしてインターンシップが終わり制作を再開しようとなったのですが続行不可能なほどのエラーが出てしまいプロジェクトを最初から作り直すことにしました。



タスク	担当者	ステータス	開始日	終了日
Inputの充実	廣川	完了	2023/00/00	2023/00/00
コリジョンクラス	佐藤	完了	2023/00/01	2023/00/01
コリジョンジェネレーター	佐藤	完了	2023/00/02	2023/00/02
スキニング	未定	完了	2023/00/03	2023/00/03
文字出力				2023/00/04
デバッグ表示				2023/00/05
デバッグ表示				2023/00/06
デバッグ表示				2023/00/07
描画				
スカイ				
scene				
シーン遷移				
レンダリング				
デバッグ				2023/00/16
サウンド				2023/00/17
デバッグでcollisionの可視化		未着手	2023/00/18	2023/00/18
マウスでオブジェクト選択	未定	未着手	2023/00/19	2023/00/19
prjectの表示	金丸	進行中	2023/00/20	2023/00/20
デバッグでcomponentのAddComponent	廣川	完了	2023/00/22	2023/00/22
デバッグでGameObjectのAddGameObject	廣川	完了	2023/00/23	2023/00/23

作品紹介「OraraEngine」

—5P—

再始動する上でマージ時のコンフリクトの対策として、タスク管理の方法を今までよりも細かくし、ソースファイルやタスクが被らないようにしたり、Sourcetreeを使用してバージョン管理をやすくしました。

ソースファイル管理

タスク	担当者	ステータス	開始日予定日	終了予定日	開始日	終了日	※ある程度作られた状態から始めた
Assetフォルダの実装	金丸	完了	11/2/2023	11/9/2023	11/2/2023	11/9/2023	
コリジョンクラス	金丸	完了	11/9/2023	11/30/2023	11/9/2023	11/30/2023	
波の動き	金丸	完了	12/1/2023	12/2/2023	12/1/2023	12/3/2023	
水面シェーダー	金丸	完了	12/2/2023	12/5/2023	12/4/2023	12/5/2023	
ポストエフェクト	金丸	完了	12/5/2023	12/8/2023	12/6/2023	12/8/2023	
布の動き	金丸	完了	12/9/2023	12/13/2023	12/8/2023	12/11/2023	
環境マッピング	金丸	完了	12/14/2023	12/15/2023	12/14/2023	12/15/2023	
レンダリングパスを使いやすくする	金丸	完了	12/16/2023	12/19/2023	12/16/2023	12/19/2023	
ImGuiの実装	廣川	完了	11/2/2023	11/5/2023	11/2/2023	11/5/2023	
セーブ/ロード機能の実装	廣川	完了	11/7/2023	11/24/2023	11/7/2023	11/24/2023	
Transformコンポーネント化	廣川	完了	11/27/2023	11/27/2023	11/27/2023	11/27/2023	
MashGui対応	廣川	完了	11/27/2023	12/1/2023	11/27/2023	12/1/2023	
パーティクルシステム	廣川	完了	12/11/2023	12/14/2023	12/11/2023	12/14/2023	
MashGui対応	廣川	完了	11/14/2023	11/15/2023	11/14/2023	11/15/2023	
手動マージ(色々変えたため)	金丸	完了	12/21/2023	12/22/2023	12/21/2023	12/22/2023	
コンポーネント間の情報のやり取り	廣川	完了	12/22/2023	10/15/2023	12/22/2023	10/15/2023	
パーティクルアップパーキャメル直す	廣川	完了	12/23/2023	12/23/2023	12/23/2023	12/23/2023	
ノードエディター	廣川	未着手	12/23/2023	12/23/2023	12/23/2023	12/23/2023	
2DMeshコンポーネント作成	廣川	完了	1/2/2024	1/5/2024	2/21/2024	2/21/2024	
シーンごとにレンダリングパスの設定	金丸	完了	12/23/2023	12/25/2023	12/23/2023	12/25/2023	
ソフトシャドウ作成	金丸	完了	12/28/2023	12/28/2023	12/28/2023	12/28/2023	
フォルダを改良	金丸	完了	1/16/2024	1/16/2024	1/16/2024	1/16/2024	

Component			Debug			pass			Manager		
audio	未使用	11/7/2023	accessor_older	未使用	1/23/2024	environmentMapping	未使用	1/2/2024	collisionManager	未使用	12/5/2023
boxCollision	未使用	12/5/2023	dragFile	未使用	1/11/2024	pass	未使用	2/23/2024	guiManager	未使用	1/14/2024
camera	未使用	12/21/2023	hierarchy	未使用	12/5/2023	postPass	未使用	1/2/2024	manager	未使用	12/23/2023
cloth	未使用	12/22/2023	inspector	未使用	2/11/2024	depthShadow	未使用	1/2/2024	shaderManager	未使用	2/23/2024
collisionShape	未使用	12/5/2023	menu	未使用	12/5/2023	createTexture	未使用	2/23/2024	textureManager	未使用	12/5/2023
commonComponent	未使用	12/21/2023	nodeEditor	未使用	12/22/2023				gameManager	未使用	1/15/2024
component	未使用	2/11/2024	reflection	未使用	2/11/2024						
inputSystem	未使用	12/21/2023	GuiWindowBase	未使用	12/14/2023						
material	未使用	12/21/2023	gameManagerGui	未使用	12/23/2023						
mesh	未使用	12/21/2023	gui_common	未使用	12/14/2023						
plane	未使用	2/18/2024	sceneWindow	未使用	1/23/2024						
particle	未使用	1/15/2024	sceneCamera	未使用	1/29/2024						
shadow	未使用	11/7/2023	debug	未使用	2/18/2024						
sphereCollision	未使用	12/5/2023									
sprite	未使用	11/7/2023									
transform	未使用	1/14/2024									
Uitext	未使用	12/21/2023									
waterSurface	未使用	12/22/2023									

タスク管理

作品紹介「OraraEngine」

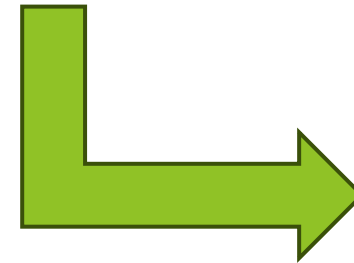
—6P—

Inspectorの作成・効率化

選択されたゲームオブジェクト内の情報を表示します。

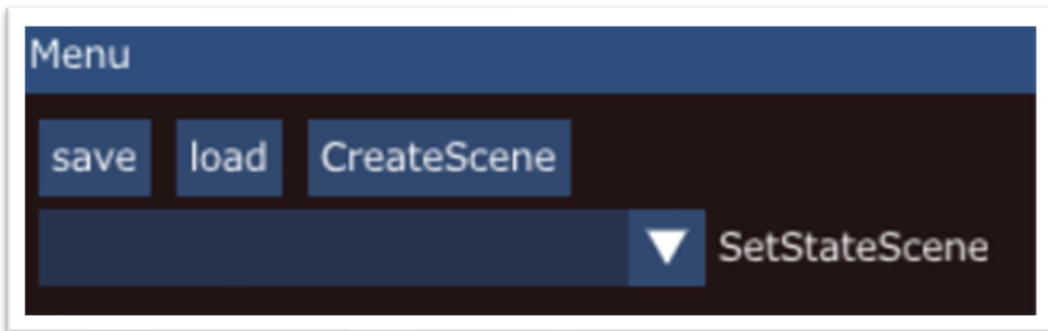
特に力を入れたのは各コンポーネント内の変数をInspector上に描画するかの指定方法です。最終的にはComponentの基底クラスにDrawInspector関数を作りその中に表示したい変数と表示名、表示方法を指定することでInspectorに表示することができます。こうすることでコンポーネント内だけで行えるので、多人数の制作がしやすく、可読性も上がり作業の効率化も上昇しました。またよく使用する変数の型は指定する必要がなくなるようにしました。そして変数が変更されたタイミングでの処理も行うことができます。

```
void PraticleSystem::DrawInspector()
{
    SET_NAME_DATE("StartColor", m_Emitter.StartColor);
    SET_NAME_DATE("StartRotation", m_Emitter.StartRotation);
    SET_NAME_DATE("StartLifetime", m_Emitter.StartLifetime);
    SET_NAME_DATE("EmissionRate", m_Emitter.EmissionRate);
    SET_NAME_DATE("Speed", (m_Emitter.Speed));
    SET_NAME_DATE("MaxParticles", (m_Emitter.MaxParticles));
    SET_NAME_DATE("AddStateVelocity", (m_Emitter.AddStateVelocity));
    SET_NAME_DATE("AddVelocity", (m_Emitter.AddVelocity));
    SET_NAME_DATE("GradationColor", (m_Emitter.GradationColor[0]));
    SET_NAME_DATE("GradationColor1", (m_Emitter.GradationColor[1]));
    SET_NAME_DATE("GradationColor2", (m_Emitter.GradationColor[2]));
    SET_NAME_DATE("IsGradation", (m_Emitter.IsGradation));
    SET_NAME_DATE("Fluctuation", (m_Emitter.Fluctuation));
    SET_NEXT_FOLDER("asset%%texture%%", ".");
    if (SET_NAME_DATE_STATE("TexName", (m_Emitter.TexName), CASTOMDRAWSTATE_STRING_FOLDER))
    {
        string name = "asset%%texture%%" + m_Emitter.TexName;
        m_TexNum = TextureManager::LoadTexture(name.c_str());
    }
    SET_NAME_DATE("GradationLength", (m_Emitter.GradationLength));
    SET_NAME_DATE("Fade", (m_Emitter.IsFade));
    SET_NAME_DATE("StartFadeTime", (m_Emitter.StartFadeTime));
    SET_NAME_DATE("declineFade", (m_Emitter.DeclineFade));
}
```



作品紹介「OraraEngine」

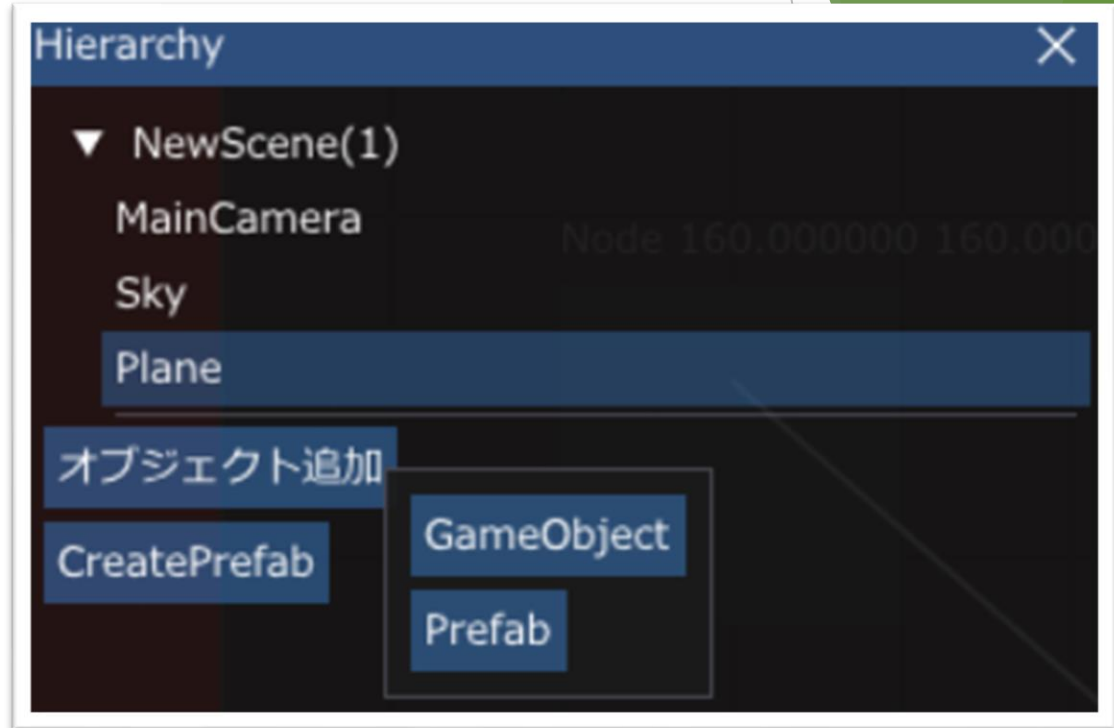
Sceneのセーブ・ロード



外部ライブラリのCerealを使用してセーブ・ロードを行いました。Scene内のGameObject・Componentの情報を一つのファイルにまとめて保存することで可読性の向上・Scene読込時に一つのファイルを読み込みだけで行えるようになりました。またJsonファイルに書き込むことでデバッグがしやすく行えるようにしました。

—7P—

Prefabの実装



GameObjectをPrefab化することができるようにしました。

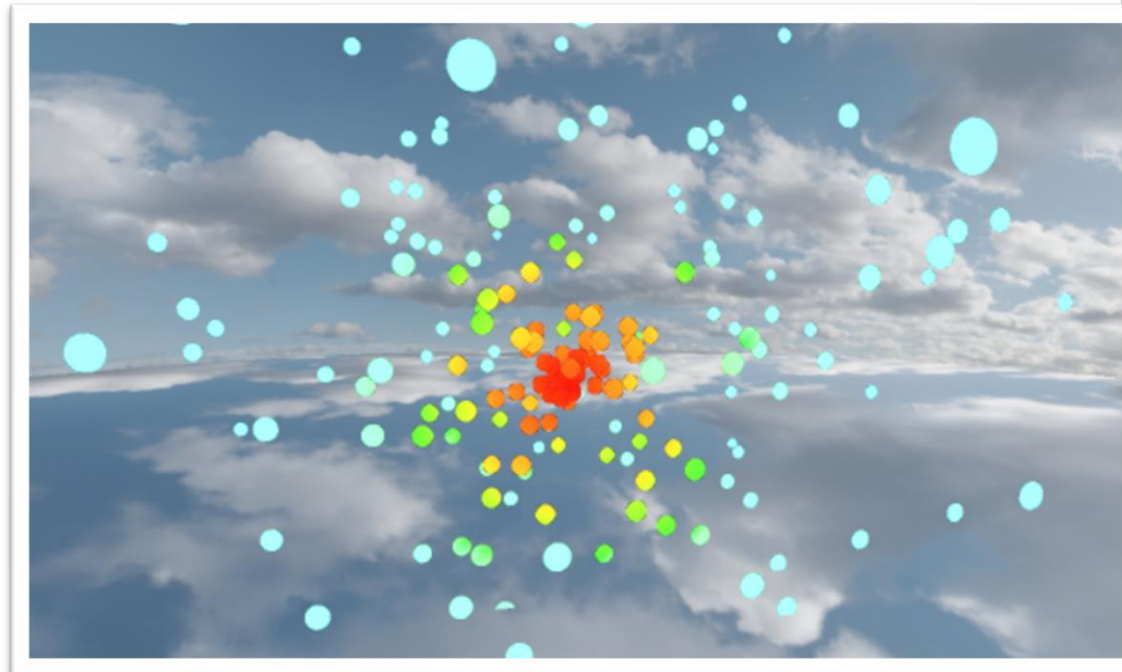
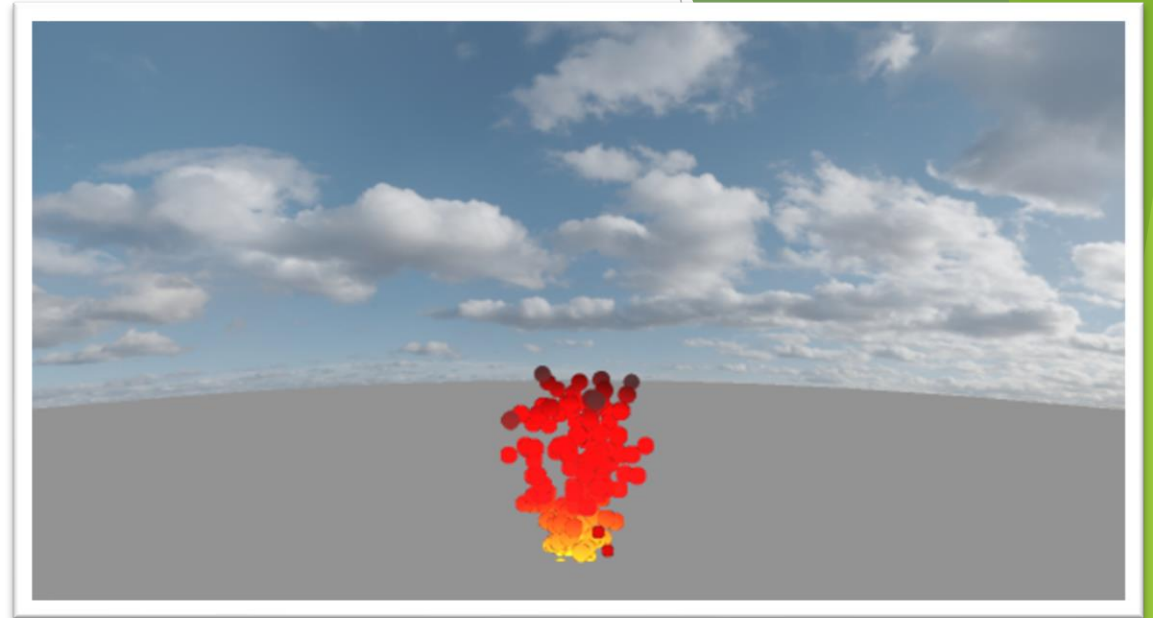
Prefab化させたいゲームオブジェクトを指定してCreatePrefabボタンを押すことでGameObject情報をファイルに保存することができ、複製を行うことができます。これを実装したことで作業効率が大幅に上昇しました。

作品紹介「OraraEngine」

Particleの作成

—8P—

ビルボードで画像を描画し様々なエフェクトが作成できるParticleを作成しました。色や速度、透明度、初速度、描画画像など様々なステータスを変更できるように作成しました。



作品介绍「OraraEngine」

—9P—

OraraEngineを使用して制作したゲーム

作品名「キノコの海上冒険」

制作人数

二人

使用エンジン

OraraEngine

開発期間

二週間

ジャンル

3Dアクションゲーム

ゲーム制作を行うことができるかを確認するために制作を行いました。制作をするうえで私が特にこだわった点はプレイヤーの手触りがよくなるようにこだわりジャンプ時の音だったり死亡時のエフェクトやカメラにシェイプなどを入れ、手触りを意識して制作しました。また、ゴール時の演出にもこだわりました。



ゲーム作品

—10P—



作品紹介

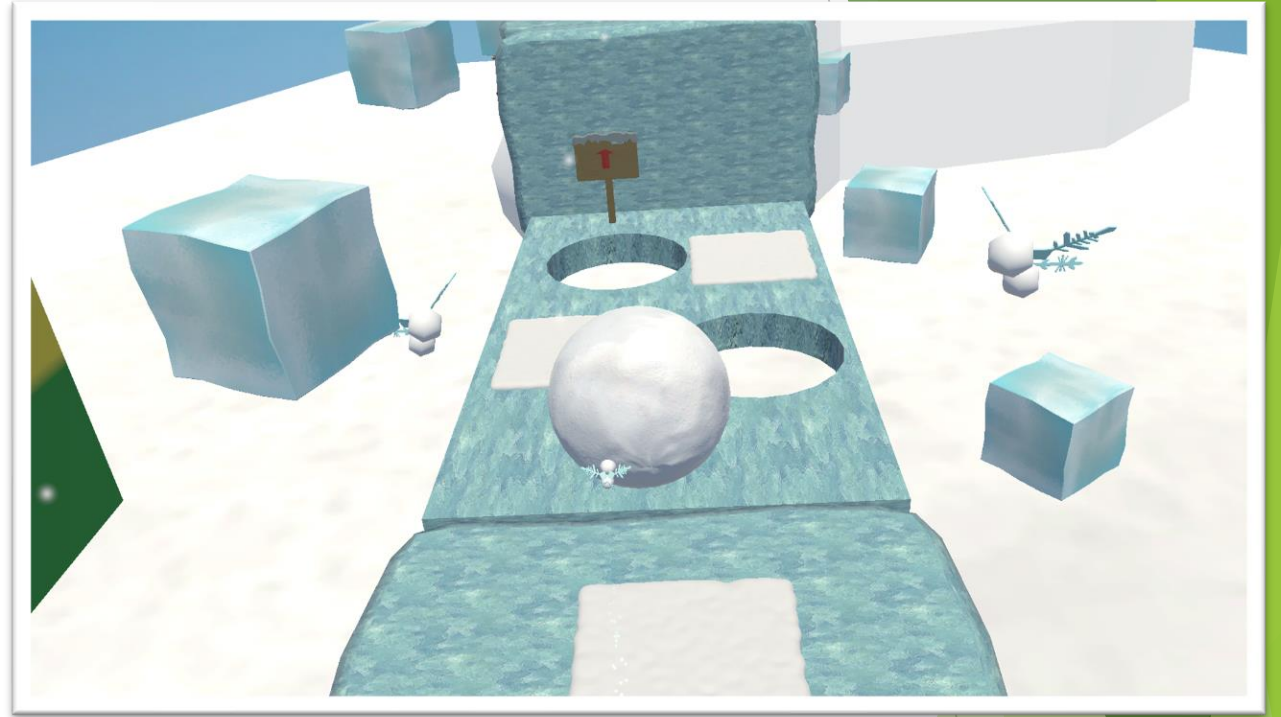
ゲーム大賞作品

「ティルトワールド」

制作人数 11人
制作日数 約4カ月
使用エンジン Unity
ジャンル 3Dアクション
操作方法 コントローラ

ゲーム大賞に向けて制作した作品で360度すべての方向にステージを傾けてゴールを目指すゲームになっています。私はメインのステージ操作の部分やギミックの作成、ステージ選択などのアウトゲーム部分、音付け、背景設置、レベルデザイン、UI選択部分、などを担当しました。チーム制作では最長のゲーム作品になりました。

—11P—



ソースファイル

<https://drive.google.com/drive/folders/19BhbCzF1j7NfNKGccQrD-MmNe9ep484n?usp=sharing>

紹介PV

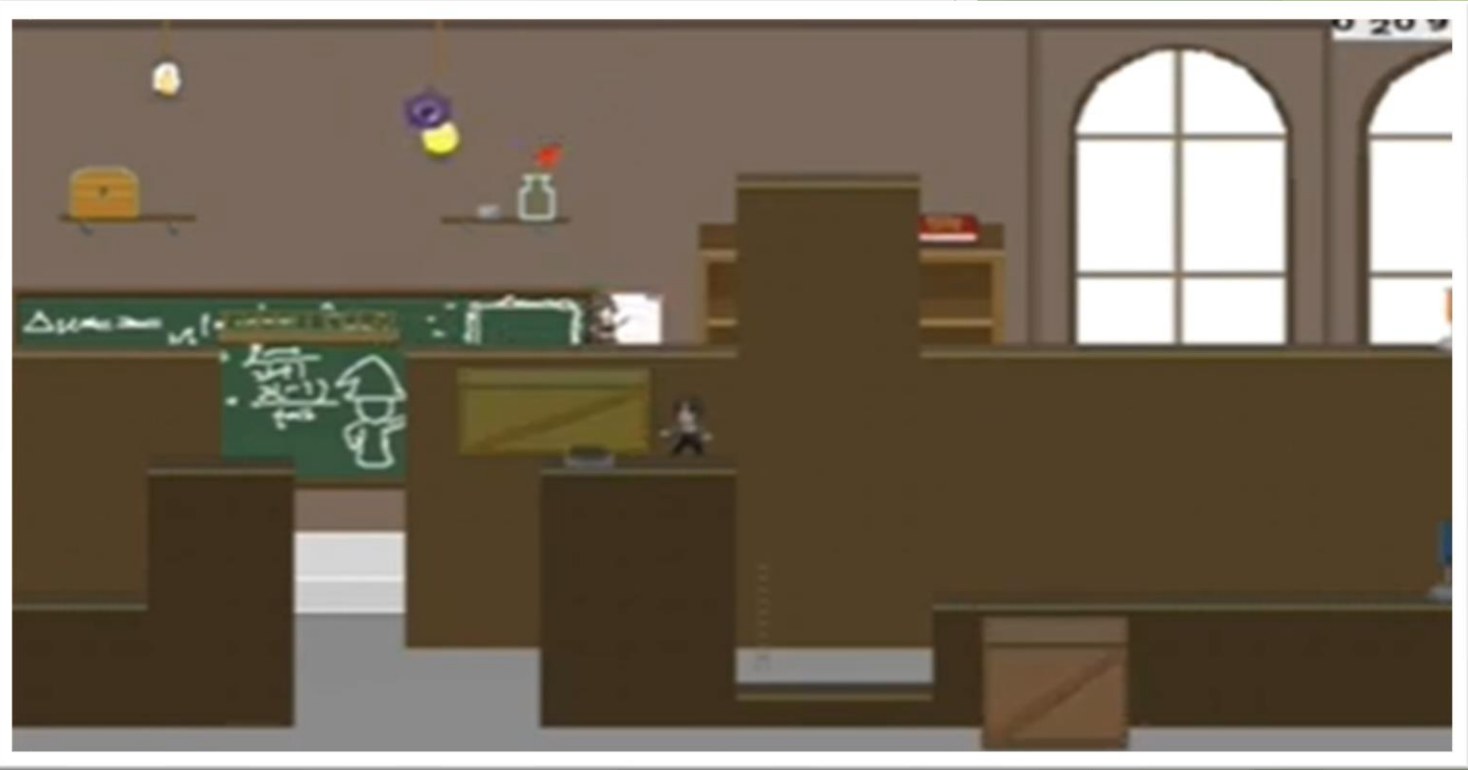
https://drive.google.com/file/d/1bytaWcsXB1rluMH9lH-445S4lpxCRxz-/view?usp=drive_link

作品紹介

—12P—

学内コンテスト作品 「バーズツインズ」

制作人数	11人
制作日数	約3カ月
制作環境	DirectX11
ジャンル	2Dパズルアクション
操作方法	コントローラ



学内コンテストに向けて制作した初めてのチーム制作作品です。手前と奥側に操作キャラクターがおり魔法を使って物を手前と奥に交互に送り謎を解き明かしていくゲームです。
私はプログラム全般を制作しました。

紹介PV

https://drive.google.com/file/d/1hhOJamRrkjJqap1gloM8secWo_aq5Ew_4/view?usp=drive_link

作品紹介

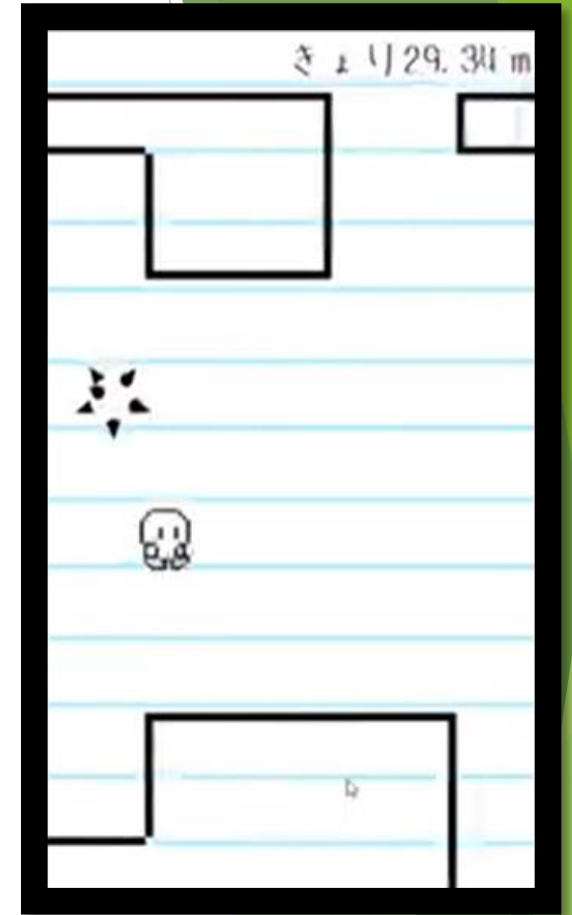
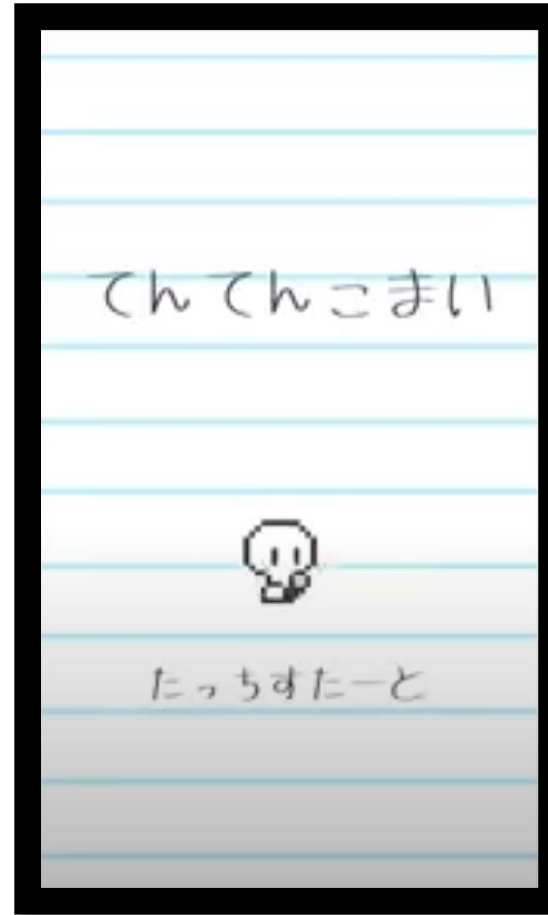
個人制作作品

「てんてこまい」

制作人数	1人
制作日数	約3カ月
制作環境	Unity
ジャンル	横スクロールランゲーム
操作方法	タッチ操作

スマートフォン向けに作成したゲーム作品でタップをすることで上下の重力を入れ替え、障害物や段差を乗り越えながら、できるだけ前に進むゲームになっています。特に世界観に力を入れた作品になっていてノートの中の世界をイメージして制作を行いました。

—13P—



プレイ動画

https://drive.google.com/file/d/1bFawIB6JBmA-st2yiqrDdU844dMx2dCN/view?usp=drive_link

作品紹介

個人制作作品

「ミミクソホリダー」

制作人数	1人
制作日数	1 週間
制作環境	Unreal Engine
ジャンル	3Dシュミレーション
操作方法	キーボード

プチコンに向けて制作した作品です。耳の中に耳かきを入れて耳垢を取り除くシュミレーションゲームになっています。途中ブロックや動く壁などをよけながら耳垢を取り除くゲーム性になっています。

—14P—



プレイ動画

https://drive.google.com/file/d/1x1nN4Wbj6571yhdFLTJeSUxEiE9GU65v/view?usp=drive_link

以上にです。

—15P—

ありがとう
ございました！