

プロフィール

氏名：

魏 文俊 (ガイ マンチュン)

使用言語：

C、C++、C#、HLSL、Python、PHP、SQL、HTML、JavaScript、CSS、Objective-C

使用ツール：

Visual Studio、Maya、Unity 6、Unreal Engine 5、Xcode

受賞：

Microsoft Summer Coding Camp 2012	Best Design App
学内進級制作展(HAL EVENT WEEK)1年次	銀賞
HAL3校合同学内コンテスト(Switch2Dゲーム)	銅賞
HAL3校合同学内コンテスト(Unity TPS)	技術賞
HAL3校合同学内コンテスト(DirectX)	技術力賞
学内進級制作展(HAL EVENT WEEK)3年次	技術賞

香港出身です。

元Webエンジニア。現在は HAL東京に在学しています。

ギター、ベース、ピアノ趣味程度に弾けます。



目次

主要作品

- ◆ MG2 個人制作 DirectX C++
- ◆ 銀河激流ミルキーウェイブ チーム制作 DirectX C++ Maya
- ◆ きゅい〜ん ちりくま チーム制作 Unity C#
- ◆ カラダ振る〜お前もギタリストにしてやろうか〜 個人制作 C言語
- ◆ 爆走ランニングフィンガー 個人制作 OpenGL C++
- ◆ SATELLITE MISSION 個人制作 Unity C#

DCCツール プラグイン

- ◆ CSV・FBX出力 Mayaプラグイン 個人制作 Python Maya

技術検証

- ◆ 3Dアニメーション遷移テスト 個人制作 DirectX C++ Maya

短期制作

- ◆ ボールをとりもどせ!! 個人制作 DirectX C++ Maya
- ◆ 地雷ゲーム 個人制作 C言語
- ◆ パズルゲーム 個人制作 DirectX C++
- ◆ SOUMEN SLIDER チーム制作 DirectX C++

CG演習・アセット制作

- ◆ 『銀河激流ミルキーウェイブ』アセット チーム制作 Maya
- ◆ くまっちの部屋 個人制作 Maya

職務経歴

主要作品

MG2 DirectX C++ 個人制作



開発環境／プラットフォーム

Windows、DirectX 11、C++、HLSL、Visual Studio 2022

制作期間

2025年12月1日～2026年2月24日（フェーズ1、約295時間）

受賞

HAL EVENT WEEK 技術賞（校内コンテスト）

ソース／実行ファイル／ライセンス情報

<https://github.com/NgaiManChun/MG2>

デモ動画

<https://youtu.be/ZyIKfmWsgh0>

◆ 概要

【1000体以上のスキニングキャラクターが個別行動！？】本作品は、大量のキャラクターを同時に扱うことを目的として制作しているゲームフレームワークです。連続メモリによる高速処理設計、GPUメッシュアニメーション、GPUカリングなどを用い、「大量表示」と「個別制御」の両立とパフォーマンス向上を重視した作品になってます。

◆ 技術的工夫

大量処理に適したデータ構造設計

コンポーネント型単位で更新処理をまとめ、オーバーヘッドの削減とキャッシュ効率を意識した設計を採用しています。データは連続メモリ上に集約し、大量オブジェクト更新時のパフォーマンス向上を図っています。

GPUデータ転送最適化

描画に必要なデータを一箇所に集約し、一括でGPUへ転送することによって転送回数の削減を実現しています。それと同時に、データへのアクセスはハンドルを通じて参照する構成を採用していて、大量に描画されつつも、キャラクターを一体一体個別管理可能な設計になります。



描画時の処理負荷を軽減しています。

データの前処理

モデルやアニメーションデータは、実行時の読み込み処理や変換コストを抑えるため、独自フォーマットへ事前変換しています。

OBB当たり判定

当たり判定はGameObjectに付与するコンポーネントとして実装しており、Transform情報に基づいて形状が決定されます。これにより、スケールや回転の変更がそのまま当たり判定へ反映されます。

◆ こだわり（ゲーム表現の工夫）

移動慣性表現

反対方向への入力時に自然な回旋と加速が発生するよう調整し、機械的にならない動きを意識しています。

遅延追従カメラ

プレイヤーの動きに対してわずかな遅れを持たせることで、視覚的な重量感と滑らかさを表現しています。



アウトライン表現

キャラクターの視認性向上を目的としてアウトライン描画を実装しています。アニメ的なルックを意識し、色味は映画『サマーウォーズ』を参考に赤系で調整してみました。



シャドーマップ

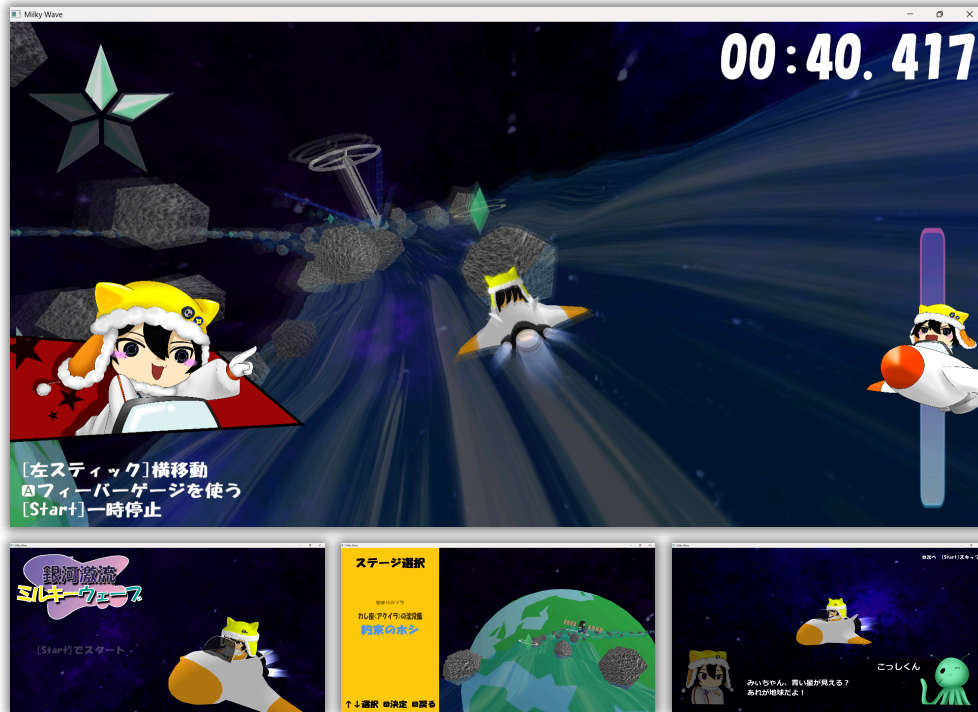
カリングやLODを活用し、必要な範囲に限定することで、動的オブジェクトに対応しつつ、パフォーマンスとのバランスを意識しています。

アタッチメント構造

ボーンアニメーションに加え、武器を柔軟に装着できる構造としています。



銀河激流ミルキーウェーブ DirectX C++ Maya チーム制作



開発環境／プラットフォーム

Windows、DirectX 11、C++、HLSL、Visual Studio 2022、Maya 2023

制作期間

2024年11月1日～2025年2月28日（約410時間）

チーム構成

プランナー3名、プログラマ6名、デザイナー3名

担当

プログラマ統括、フレームワーク作成、プレイヤー制御、各種演出、アセット制作補助

ソース／実行ファイル／ライセンス情報

<https://github.com/NgaiManChun/MilkyWave>

デモ動画

https://youtu.be/ln_VBHvnFA8

◆ 概要

「天の川の川下り」をコンセプトとしたタイムアタック型3Dレースゲームです。ねじれた「天の川」を滑走し、障害物を回避しながらベストタイムを目指します。アイテムを収集することでブースターを発動でき、加速と障害物の破壊による爽快感を味わえます。波打つコースに合わせた船体の傾きや浮き沈みも取り入れ、スピード感と臨場感を表現しています。

プランナー向けのコースの仕組みと作り方の説明：

[Google Driveスライド](#)

◆ 技術的工夫

3D向けに拡張した自作フレームワーク

これまでのプロトタイプ開発で構築した自作フレームワークを3Dゲーム向けに再設計。描画・入力・リソース管理を整理し、デモシーンやサンプルを用意することでチーム全体で利用可能な基盤として提供しました。

[source/hew_year2_base](#) 


ゲームデータ前処理CLIツール

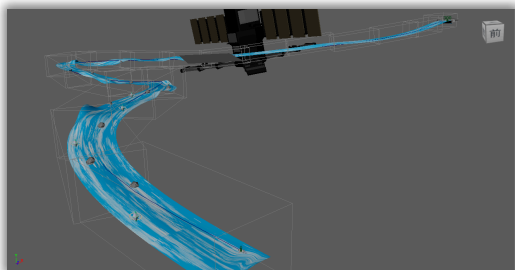
FBXから以下3種類の独自データへ分離変換：

.mgm (モデル)

.mga (アニメーション)

.mgi (インスタンス配置・コリジョン)

実行時負荷を抑えるため、オフライン前処理で不要情報を削減。シンプルで再利用性の高いデータ構造とすることで、メモリ使用量を抑えながら高速な読み込みを試みました。 [source/fbx/main.cpp](#) 




ワークフローの構築

デザイナーとプランナーは慣れ親しんだMaya上でモデルとコースを設計し、ツールでモデル・コリジョン設定・オブジェクトの配置情報などのゲームデータに変換しています。加えて、各種パラメータはCSVで管理しており、非エンジニアでも柔軟な調整が可能にしています。


◆ こだわり (ゲーム表現の工夫)

浮き沈み描写と重力方向が変化

コースに沿って重力の導線を設定し、位置によって重力方向が変化します。波によって船は細かく傾けながら浮力でバウンドさせたり、船の自然な浮き沈みを演出しています。 [source/player.cpp\(line 116-154\)](#) 



3Dと2Dを連動させたUI演出

3D空間で取得したアイテムが、そのまま2Dのゲージへ吸収される演出を実装しました。取得結果を視覚的にUIへ連動させることで、ゲーム世界とUIの一体感を強め、プレイヤーが状態変化を直感的に把握できるようにしています。 [source/player.cpp\(line 94-105, line 321-341\)](#) 



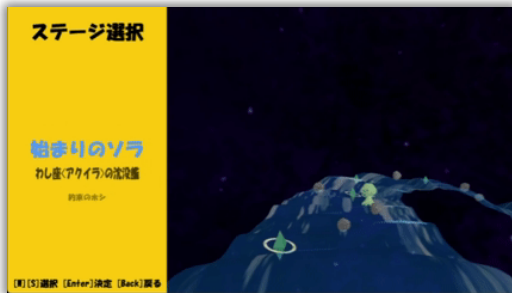
加速時の視覚効果

カメラを引きつつ多重レンダリングによるブラーを付加。視覚的变化でスピード感を伝えていきます。

[source/mainGameScene.cpp\(line 841-879\)](#) 

細部のリアクション設計

隕石衝突時の速度・角度に応じた吹き飛び挙動 水面接触時の星型エフェクト
加速時のキャラカットイン



非同期処理によるステージプレビュー

マルチスレッド処理を用いて、ステージ選択中のステージプレビューを非同期で読み込み。ユーザー体験を損なわない設計を意識しています。

[source/stageSelectScene.cpp\(line 511-536\)](#)

入力管理の抽象化とジャイロ操作

キーボード、XInput、DualSenseに対応。ゲーム実行中いつでもデバイスを差し抜きし、自動切替できる設計を実装。操作ガイド表示も動的に変更されます。さらに、DualSenseではHIDレポートに解析によるジャイロ操作の対応を実装しています。

[source/input.cpp](#)

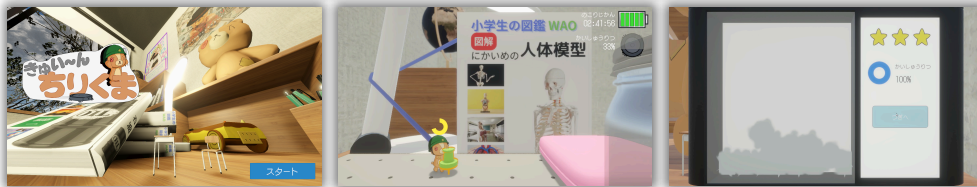


リザルト・タイトル演出の強化

ランク表示の光沢エフェクトや、背景宇宙の映り込み演出などを実装。メインゲーム以外も含めた体験設計を意識しました。

[source/resultScene.cpp](#)





開発環境／プラットフォーム

Windows、Unity、C#、Visual Studio 2022

制作期間

2025年6月4日～2025年7月18日（約130時間）

チーム構成

プランナー2名、プログラマ4名、デザイナー3名

担当

企画、掃除ロボットの制御、ロープギミック、UIプログラム、リザルト

ソース／実行ファイル／ライセンス情報

<https://github.com/NgaiManChun/Chirikuma>

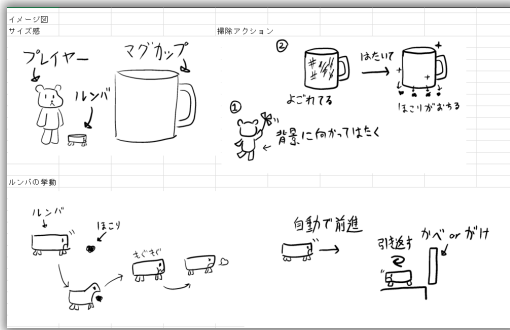
デモ動画

<https://youtu.be/ut-ACU0Dlw8>

◆ 概要

ぬいぐるみのクマを操作し、自律行動する掃除ロボットを誘導しながらステージ上のほこりを回収していくアクションパズルゲームです。プレイヤーは汚れを叩いてほこりを落とし、掃除ロボットが回収できる環境を整えながら、制限時間内にチャージスポットへ導くことを目指します。チーム制作の途中で企画の見直しを経て完成に至り、仕様の再調整を行いながら開発を進めました。

◆ 立案・こだわり



自律行動する掃除ロボットとプレイヤーの協調設計

掃除ロボットは自律的に移動してほこりを回収し、プレイヤーは先回りして道を開き、汚れを叩いてほこりを落とす役割を担います。両者の役割分担によって進行する、間接的な操作を軸としたゲーム構造を設計しました。



エリア単位の固定カメラによる空間設計と視認性の両立

横スクロールを基調としつつ、エリアによって奥行方向（Z軸）に進行するステージ構造を採用しています。進行方向の変化に対応するためエリアごとの固定カメラを導入し、さらにプレイヤーと掃除ロボットの両方を適切なサイズで視認できるよう、エリア単位でカメラ位置とズーム倍率を調整しました。

◆ プログラム担当箇所



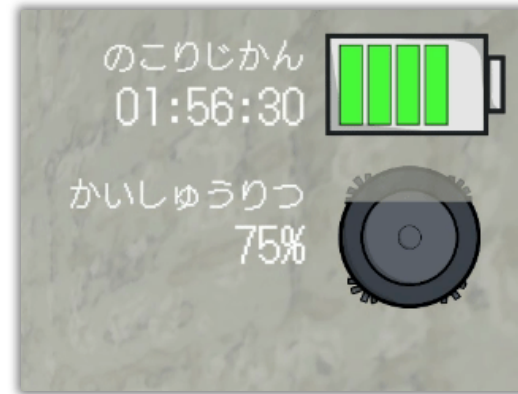
掃除ロボット制御

自動移動、方向転換、ほこり回収、状態管理など、掃除ロボットの挙動全般を実装しました。 source/Assets/010_Scripts/Rumba.cs [🔗](#)



ロープギミック

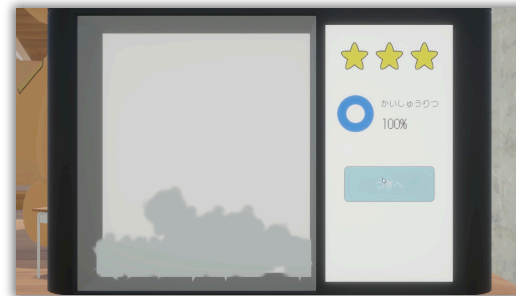
ステージ内のロープオブジェクトの挙動とインタラクション処理を実装しました。 source/Assets/010_Scripts/Rope.cs [🔗](#)



UIプログラム

ゲーム中のUI表示および状態に応じた更新処理を実装しました。

source/Assets/010_Scripts/PlayerCanvas.cs 



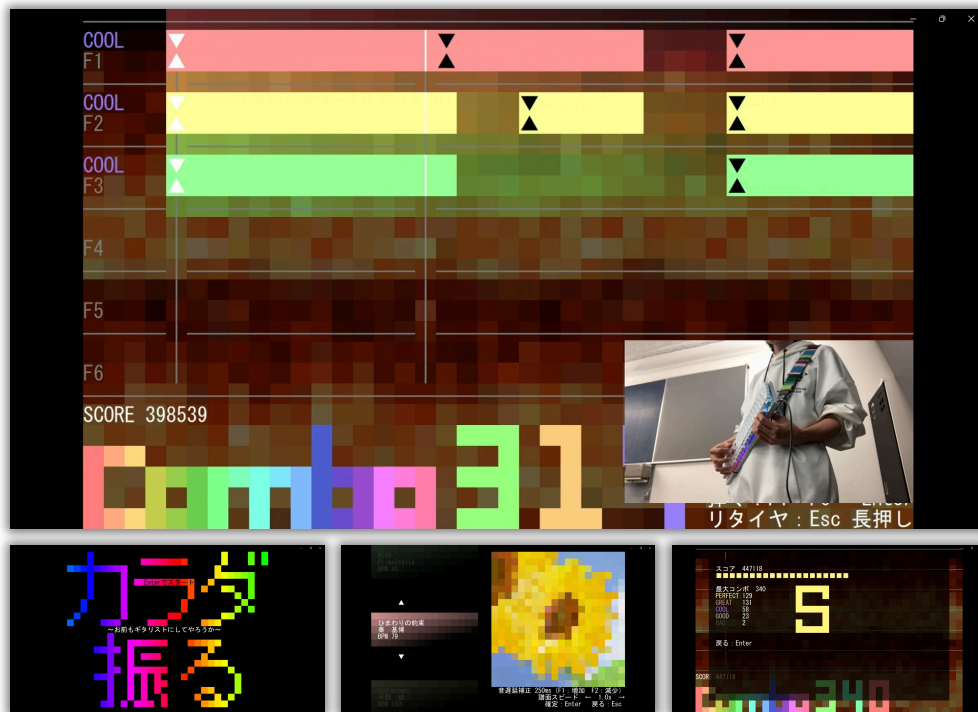
リザルト画面

回収率やクリア時間に応じた評価表示など、リザルト画面の表示処理を実装し

ました。 source/Assets/010_Scripts/Goal.cs 



カラダ振る～お前もギタリストにしてやろうか～ C言語 個人制作



開発環境／プラットフォーム

Windows (コンソール)、C、Visual Studio 2022

制作期間

2024年1月15日～2024年1月31日 (約60時間)

受賞

HAL EVENT WEEK 銀賞 (校内コンテスト)

ソース／実行ファイル／ライセンス情報

<https://github.com/NgaiManChun/Karadafuru>

※楽曲はYoutubeにのみ使用しており、著作権者はYouTubeでのコンテンツの使用を許可しています。ソース及び実行ファイルに含まれる音源ファイルはすべてダミーです。

デモ動画

<https://youtu.be/VMUpB2KFo5g>

◆ 概要

キーボードをギターに見立てて遊ぶ、リズムアクションゲームです。F1～F6キーを押さえながら、タイミングよくEnterキーでピッキングすることでノーツを消していきます。タイミングに応じてスコアが加算され、さらに「サスティン」や「ハンマリング・プリング」など、実際のギター奏法も再現したシステムが特徴です。「カラフル」というテーマのもと、ビジュアルや演出、そしてギターを鳴らし「カラダ振る」という解釈で展開しました。

◆ こだわり・チャレンジ

実在の奏法を再現した入力システム

ギターのピッキング、ハンマリング、プリングを模したキー入力を導入。音楽ゲームとしての独自性と操作の面白さを図りました。

本物のギター譜面を参考に譜面を設計

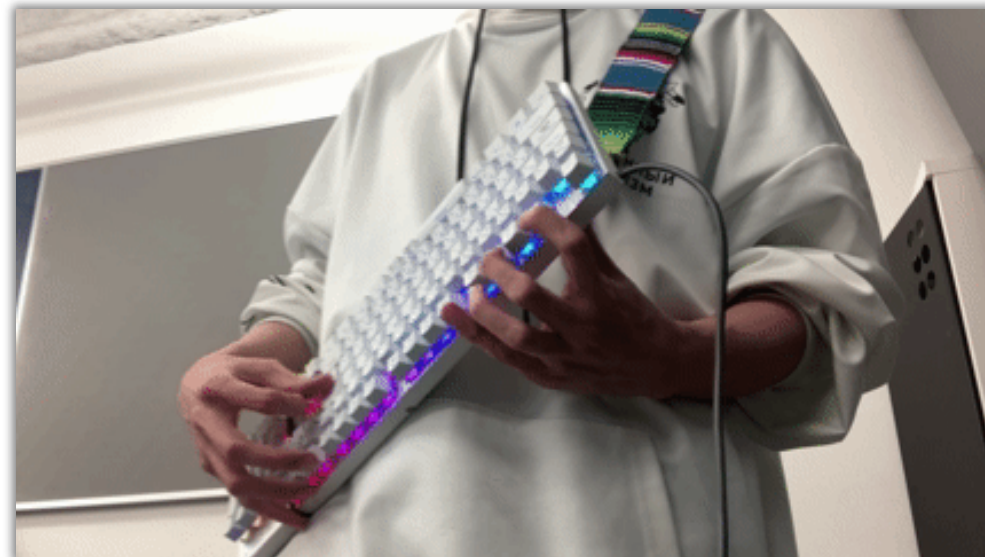
実際の演奏に近い感覚を意識して譜面を作成。ミュージック学科の先輩にプレイさせたら、いきなり私よりも高得点という実績もありました。なお、譜面データは外部のCSVファイルに格納しています。

BMPファイルからの色情報読み込みによるカラフルな演出

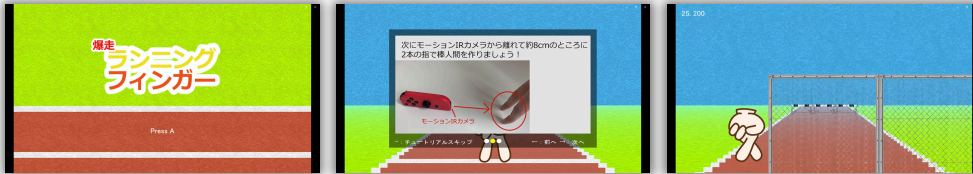
画面上のドット絵風演出は、画像データから色情報を読み取って文字の背景色に反映しています。アルファ合成を意識したエフェクトも実装しています。

ちらつきを抑えるバッファ描画処理

全画面分の文字情報を事前にバッファで構築し、一括描画することで、コンソール環境でもスムーズな画面更新を実現しています。



爆走ランニングフィンガー OpenGL C++ 個人制作



開発環境／プラットフォーム

Nintendo Switch、OpenGL、C++、Visual Studio 2022

制作期間

2024年7月21日～2024年9月6日（約160時間）

受賞

Switch 2Dゲーム 学内コンテスト 銅賞（校内コンテスト）

ソース／実行ファイル

守秘義務により公開出来ません

デモ動画

<https://youtu.be/usSkmJEs9w4>

◆ 概要

『爆走ランニングフィンガー』は、指人形を机の上で走らせるという身近な遊びを、Nintendo Switchで再現したミニゲームです。Joy-ConのIRカメラを使って、実際の指の動きでキャラクターを操作する仕組みになっています。

初めてのNintendo Switch向けに開発した作品で、ステージは擬似3Dを使って奥行きを表現しています。IRカメラを活用した遊び方が特徴です。

◆ こだわり・チャレンジ

IRカメラによる直感的な操作

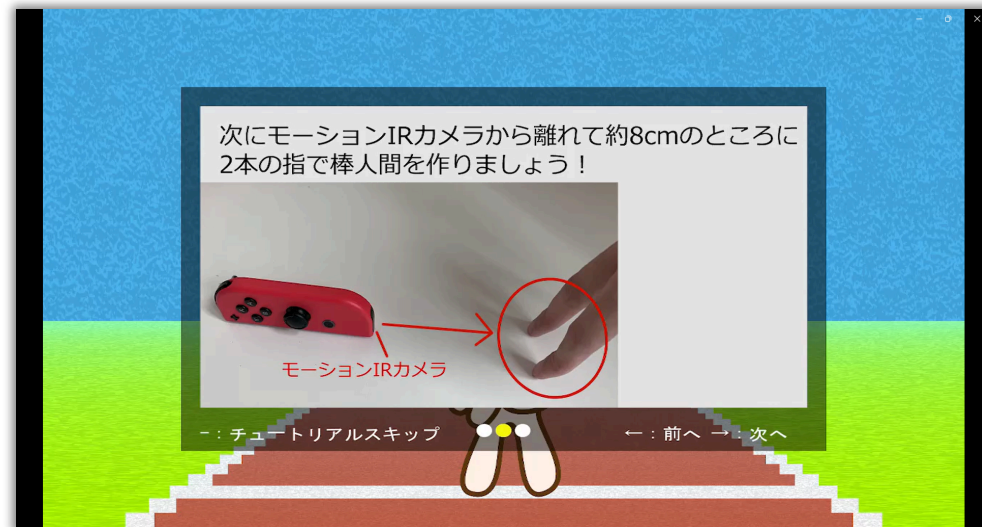
Joy-ConのIRカメラに2本の指を映し、指の動きに合わせてキャラクターが前進・左右移動・ジャンプする操作を実現しました。特別な装置を使わず、素手で遊べます。

SwitchとPC両対応を意識した構成

Switchでの開発は学校設備に依存するため、多くの学生が夏休みにも通学して作業していました。そこで私はアダプターパターンを用い、Switch（OpenGL）とPC（DirectX）それぞれの環境で、ゲームロジック部分を変えずに動作する構造を設計し、自宅での開発を中心に進められるように工夫しました。

擬似3Dの実装

もともとは横スクロール型の構成を想定していましたが、実際に指人形で遊ぶ際は「前に進む」動きのほうが自然で操作しやすいと感じたため、ゲーム画面もそれに合わせて奥行き方向のステージに変更しました。レギュレーション上は2Dゲームに限られており、当時の授業でも3Dゲーム制作はまだ扱われていなかったため、擬似3Dの表現を試みました。



SATELLITE MISSION Unity C# 個人制作



開発環境／プラットフォーム

Windows、Unity、C#、Visual Studio 2022

制作期間

2023年12月17日～2024年1月12日（約90時間）

受賞

オリジナルTPS学内コンペ 技術賞（校内コンテスト）

ソース／実行ファイル／ライセンス情報

https://drive.google.com/drive/folders/157H_fz9dh3hqUr8_ZoMH...

デモ動画

https://youtu.be/kQ2IMom_x7c

◆ 概要

入学から3か月目に制作した、はじめてのUnity製3Dアクションシューティングゲームです。プレイヤーは空を飛べるキャラクターを操作し、味方拠点を守りながら、敵ユニットと敵拠点を殲滅することが目的としたゲームです。ミサイルとマシンガンを使い分けて戦うTPSスタイルで、拠点防衛と撃破を両立させる戦術性が特徴です。

◆ こだわり・チャレンジ

プレイヤーを中心に飛び回る「サテライトライフル」の演出

不規則に飛び交う複数の軌道を重ね、SFアニメに登場する精神感應兵器のような動きを再現しました。本作の象徴的なギミックです。

戦術を持つ敵ルーティン

敵は4つの行動パターン（自陣防衛・プレイヤー攻撃・レーダー攻撃・距離優先）を20秒ごとに見直し、残存数に応じて役割を自動分担します。単なる動くのではない緊張感ある戦闘を実現しています。

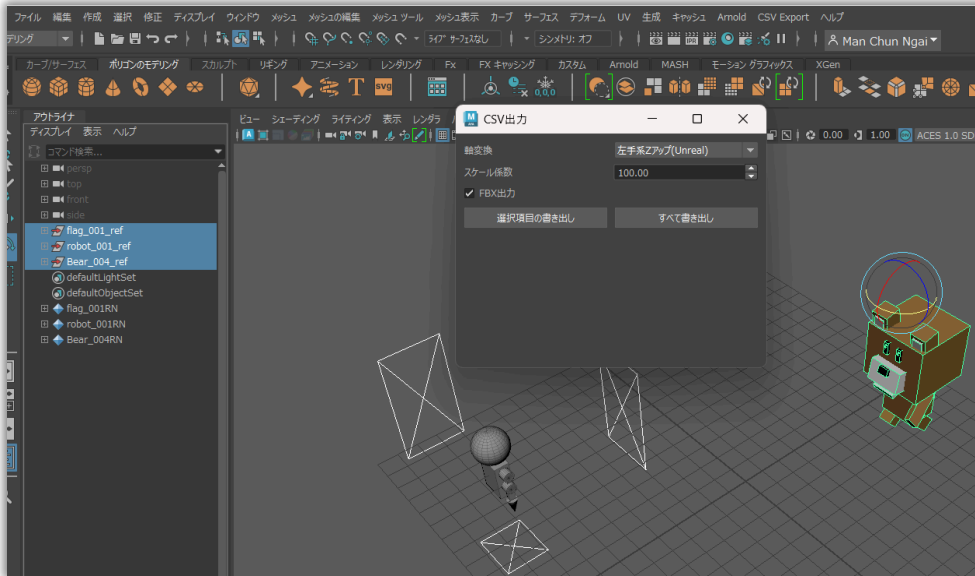
UI・演出のディテール再現

ホログラム風パネルUIを採用し、SFらしいビジュアルを表現。撃破時の連続爆発、ダメージ時のUI振動・点滅、敵との距離表示、画面外の敵を示すインジケーターなど、既存のゲームを参考に細かい演出を多数実装しました。



DCCツール プラグイン

CSV・FBX出力 Mayaプラグイン Python Maya 個人制作



開発環境／プラットフォーム

Python、VSCode、Maya 2023

制作期間

2025年10月20日～2025年10月24日（約30時間）

ソース／実行ファイル

<https://github.com/NgaiManChun/MayaCSVExport>

◆ 概要

Mayaシーンに配置されたリファレンス情報を読み取り、CSV形式で書き出すためのプラグインです。必要に応じて、同じフォルダへFBXも同時にエクスポートできます。大量の参照データを扱う場面で、外部ツールとの連携をスムーズにすることを目的に制作しました。

◆ こだわり・チャレンジ

同じリファレンスを複製しただけのものやインスタンスは、重複出力を避けて効率的にエクスポートできるようにしました。座標系変換やスケール指定、選択項目だけの書き出し、シーン全体の一括出力など、運用でよく使われるオプションにも対応しています。また、設定したオプション内容をコンフィグとして保存し、次回以降は同じ条件で素早く出力できる仕組みも整えています。



技術検証

3Dアニメーション遷移テスト DirectX C++ Maya 個人制作



開発環境／プラットフォーム

Windows、DirectX 11、C++、Visual Studio 2022、Maya 2023

制作期間

2025年3月26日

ソース／実行ファイル

<https://github.com/NgaiManChun/AnimationTransition>

デモ動画

<https://youtu.be/O4Xq-fOzjE4>

◆ 概要

複数の3Dアニメーションを、状態に応じて自然に遷移させるためのテスト。
停止、歩行、攻撃の3状態間をスムーズに補間させる挙動を検証した。

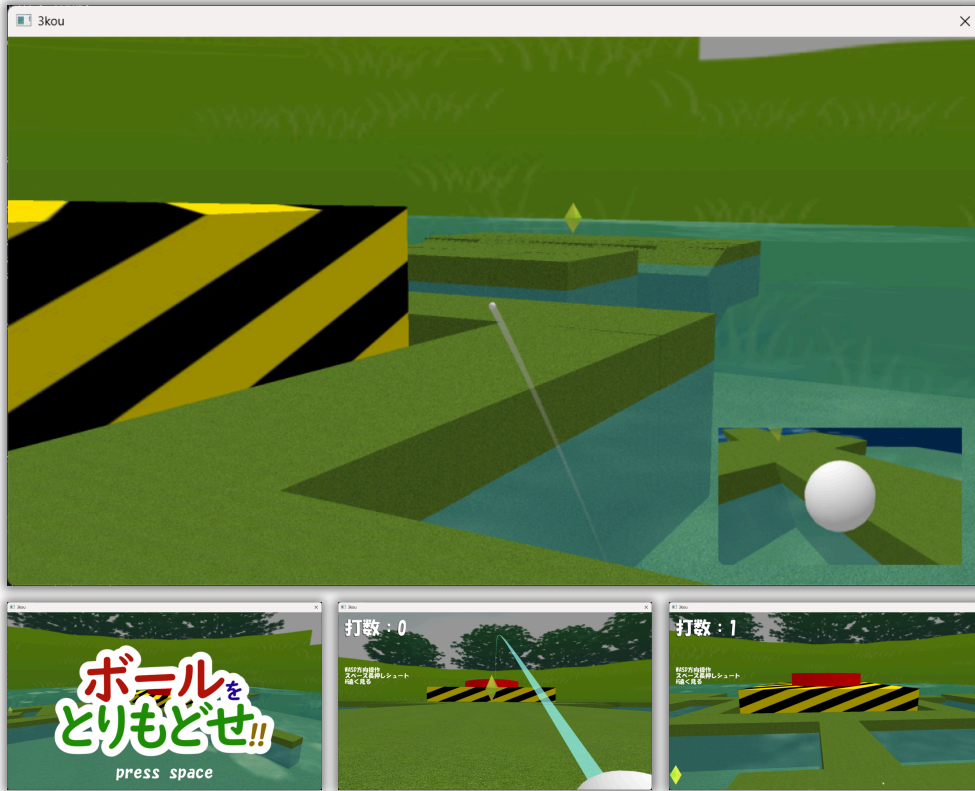
◆ こだわり・チャレンジ

アニメーション同士を自然につなげるため、線形補間によるスムーズな遷移制御を行い、状態の切り替え時に違和感が出ないように調整しています。また、どの状態でも独立して進行し続けるまばたきアニメーションも組み込んでいます。加えて、キャラクターの手に持ったアイテムを追従させ、スラッシュ軌道のエフェクトを連動させる処理も実装しています。



短期制作

ボールをとりもどせ!! DirectX C++ Maya 個人制作



開発環境／プラットフォーム

Windows、DirectX 11、C++、Visual Studio 2022、Maya 2023

制作期間

2025年1月10日～2025年1月13日（約32時間）

受賞

D X オリジナルゲーム学内コンペ 技術力賞（校内コンテスト）

ソース／実行ファイル

<https://github.com/NgaiManChun/Golf> [🔗](#)

デモ動画

<https://youtu.be/EHuAcm5lv1Y> [🔗](#)

◆ 概要

本作は、フィールド上に配置されたアイテムを集め、最終的に中央の巨大なボタンを押すことでクリアとなるミニゲームです。もともとは複数ステージ構成のゴルフゲームを目指していましたが、制作時間の見積もりが甘く、開発途中で方向転換を決断。結果的に、1ステージ完結型かつ派手な演出で締めくくる構成に落ち着きました。

制作時間の制約から仕様を絞った分、印象に残る演出や操作感を重視して設計しました。

◆ こだわり・チャレンジ

短期間でも印象に残る演出設計

当初の計画から方向転換したミニゲーム形式において、1ステージでも遊びごたえと達成感が得られるよう、ゲーム終盤には必殺技演出を導入。短期間でも「記憶に残る体験」を目指しました。

視認性への配慮：半透明化と軌道予測

カメラとプレイヤーの間にオブジェクトが入った際、それらを半透明化する処理を実装。加えて、弾道の予測線を描画することで、操作性を向上させています。

物理表現の自力実装

ステージにはわずかですが斜面を含む構造があり、当たり判定には変形や回転を考慮した処理が必要でした。当時授業ではAABBまでしか扱われていませんでしたが、必要性を感じ、自力で拡張しました。

技術検証のプロトタイプとしての位置づけ

『銀河激流ミルキーウェーブ』で本格的に使用した自作フレームワークのプロトタイプにあたる作品であり、「オリジナル3Dアセット変換CLIツール」や「開発と運用の効率化」といった機能をこの時点で試行していました。



地雷ゲーム C言語 個人制作



開発環境／プラットフォーム

Windows (コンソール)、C++、Visual Studio 2022

制作期間

2023年11月24日～2023年11月26日 (約18時間)

ソース／実行ファイル

<https://github.com/NgaiManChun/Landmine> [🔗](#)

デモ動画

<https://youtu.be/q2VGZxDSw2A> [🔗](#)

◆ 概要

入学2か月目に、C言語の授業の課題「動きのあるコンソールゲーム」として、『マインスイーパ』を模倣して制作しました。

◆ こだわり・チャレンジ

空白マスの自動展開を再帰関数で実装。限られたコンソール環境でも遊びやすくするため、現在選択中のマスの色を表示するなど、視認性にも配慮しました。本物の『マインスイーパ』の代用品を目指して制作していました。



パズルゲーム DirectX C++ 個人制作



開発環境／プラットフォーム

Windows、DirectX 11、C++、Visual Studio 2022

制作期間

2024年9月4日～2024年9月13日（約54時間）

ソース／実行ファイル

<https://github.com/NgaiManChun/PuzzleGame> [🔗](#)

デモ動画

<https://youtu.be/LsuBDuJ15w4> [🔗](#)

◆ 概要

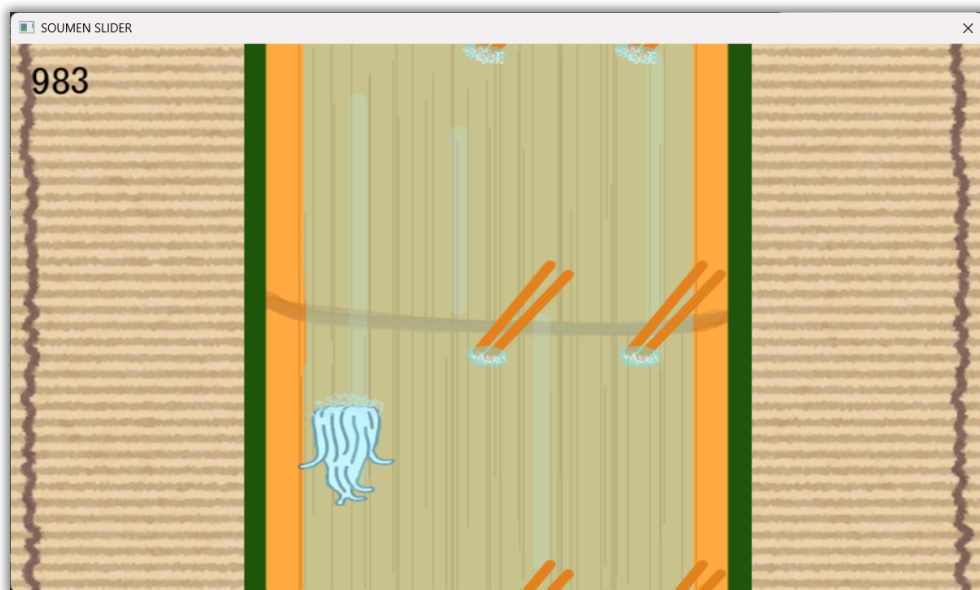
授業課題として制作した落ち物パズルゲーム。『コラムス』に似せた、色が揃うとブロックが消えるDirectXの2Dゲームです。

◆ こだわり・チャレンジ

スコア加算時の演出や背景の変化など、プレイ中の視覚的な気持ちよさを意識して作りました。特に、シーン遷移時には暗転を使わずにフェードイン・フェードアウトを同時に行う処理を実装し、自然な切り替えにこだわっています。



SOUMEN SLIDER DirectX C++ チーム制作



開発環境／プラットフォーム

Windows、DirectX 11、C++、Visual Studio 2022

制作期間

2024年8月30日

チーム構成

プログラマ2名

ソース／実行ファイル

<https://github.com/NgaiManChun/SoumenSlider>

デモ動画

<https://youtu.be/OigaxxVG-No>

◆ 概要

ハッカソンで1日で制作した、お箸を避けながら流しそうめんとして長く流れ続けることを目指すミニゲームです。

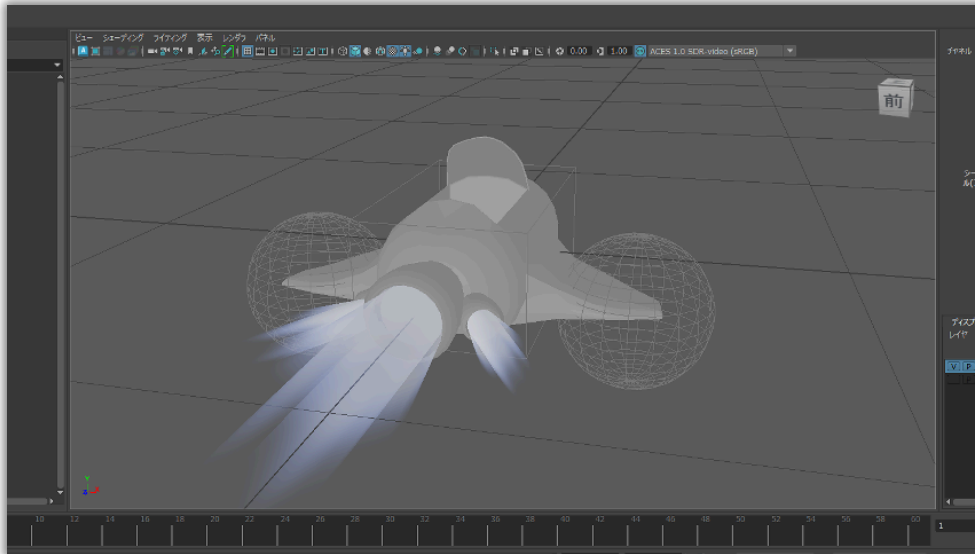
◆ こだわり・チャレンジ

シンプルなルールとテンポ感を重視して設計しました。

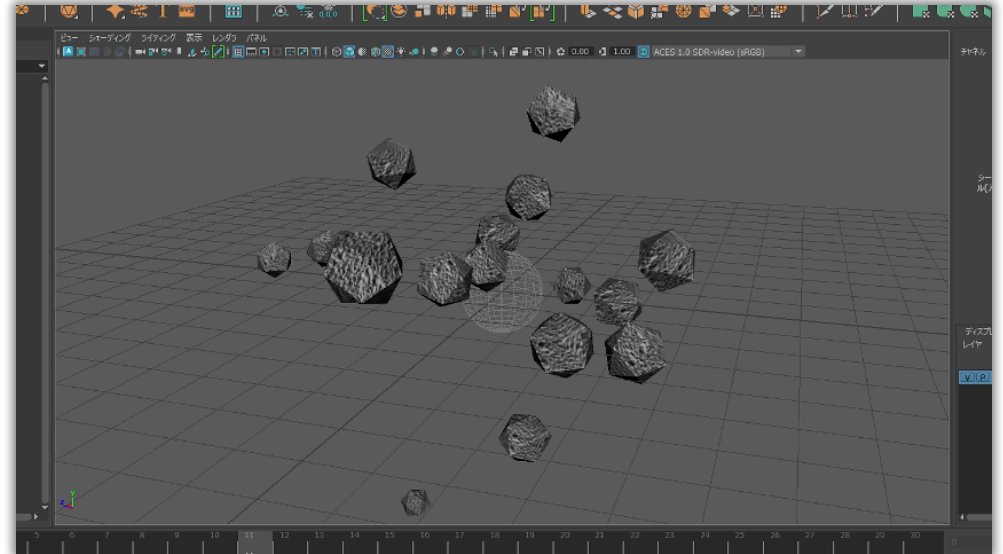


CG・アセット

『銀河激流ミルキーウェイ』アセット チーム制作



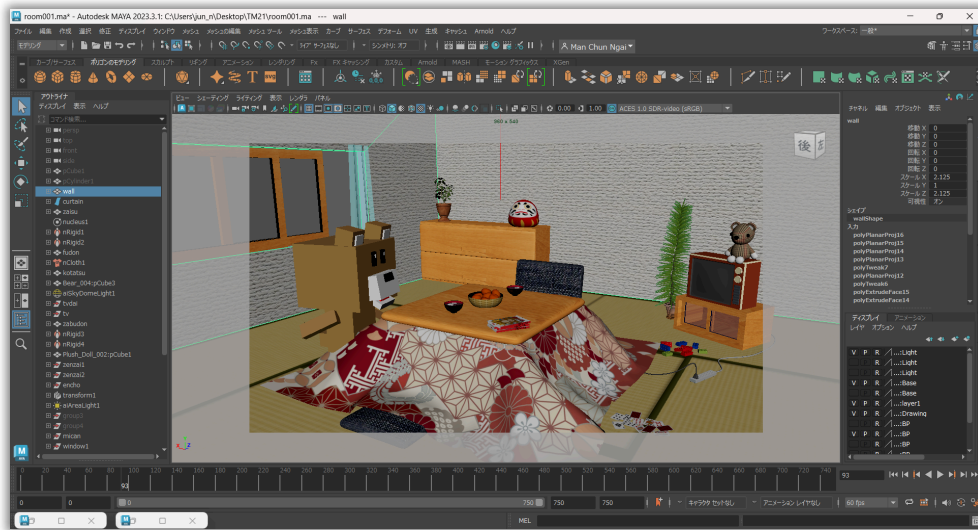
『銀河激流ミルキーウェイ』で使われているブースターのエフェクト



障害物が破壊される時のアニメーション



くまっちの部屋 個人制作



Maya+Arnoldレンダリング、制作時間：約18時間



職務経歴

Wai On Services Limited (Junior Programmer)

期間： 2013年6月 ～ 2014年8月

業務内容： Webアプリケーションの開発・改修を担当

担当プロジェクト： 中国工商銀行（香港）
○ ポータルサイトリニューアル
○ 文書ファイルのリアルタイム合成機能の開発
香港鉄道（MTR）
○ ポータルサイトリニューアル
○ 社内コミュニティフォーラムの改修
など

Re. Group Limited (Web Developer)

期間： 2015年3月 ～ 2023年9月

業務内容： Webアプリケーションの設計および開発を担当

担当プロジェクト： Universal Music Hong Kong
○ 公式サイトリニューアル
○ 公式通販サイトの開発
The Hood Production
○ 公式通販サイト開発
○ スマートフォンアクセサリーのカスタム
注文機能の開発
など

