

# ポートフォリオ

ASOポップカルチャー専門学校  
ゲーム・CG・アニメ科 ゲームコース

筒井 杏二

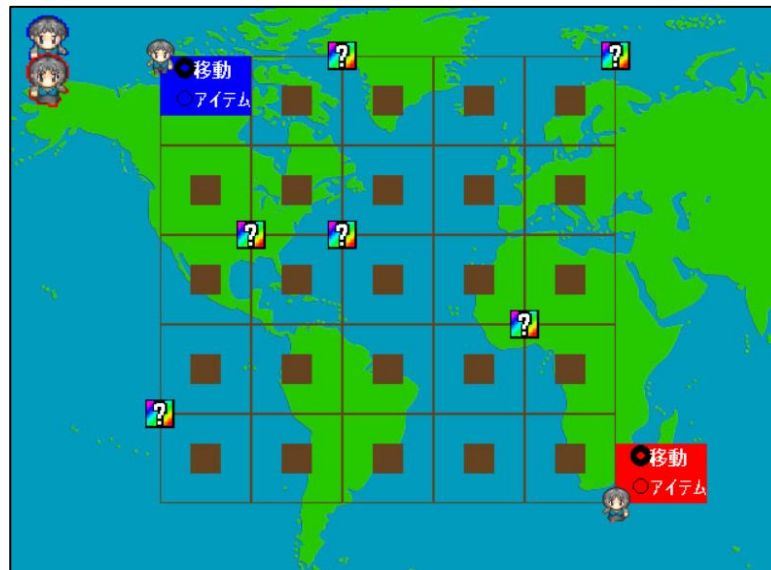
## 自主制作作品

学内コンテストに向けて仕様書から個人で制作した作品です。

25マスの中に自分のお宝が眠っておりアイテムを使用し、得られるヒントを使ってお宝を探し出すゲームです。  
宝の位置を予想したらそこを囲んで宝を発見します。  
1人用モードと2人用モードがあります。

2人用モードでは一つの画面にしか情報を出せないのが、アイテム使用時の効果を表すのが大変でした。

1人用モードではランダムで敵が出現しプレイヤーを妨害してきます。ターンがかかるほど敵は増えていきます。



## 制作時期

2年後期

# 使用言語

# C++言語

# 使用ライブラリ

# DXライブラリ

## 制 作 人 数

1人

# シーンの切り替え

## 色の乗算

シーンの切り替えに独特な形でシーンの切り替えを行います。  
Aのスクリーンに先に描画しておいて、Bのスクリーンにはフェイド用の形を描画します。それを合成し乗算することで独特な形でのフェイドインアウトを実装しました。

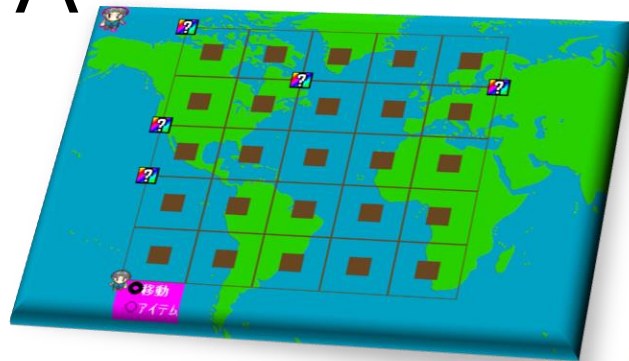
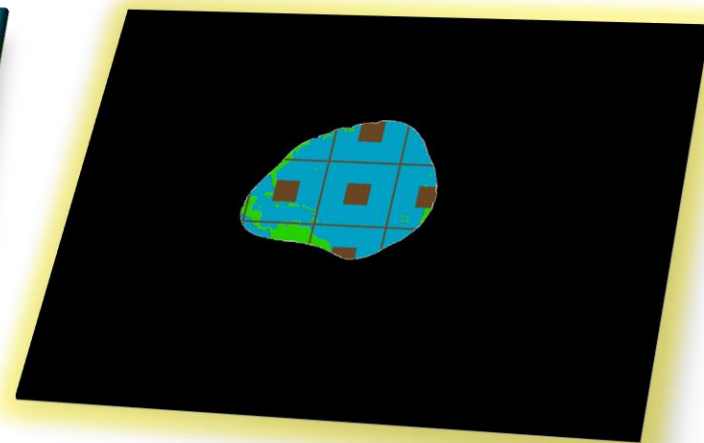
```
//描画処理でもらうハンドルの入手
SetDrawBlendMode(DX_BLENDMODE_NOBLEND, 255);

screen_ = uniqueScene_>Draw();
if (screen_ == -1)
{
    break;
}

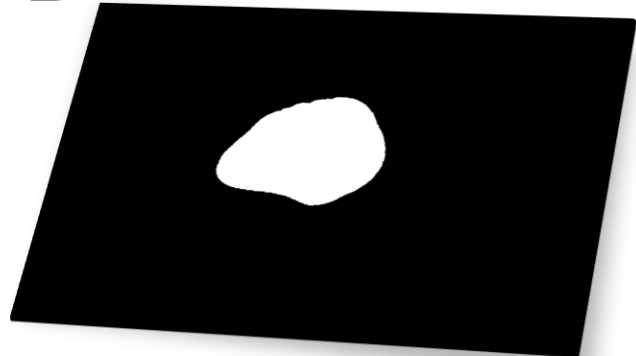
//ゲームの描画シーン
SetDrawScreen(DX_SCREEN_BACK);
ClsDrawScreen();

if (fadeOut_ == true || fadeIn_ == true)
{
    //フェイド処理 & 描画
    FadeCommand(static_cast<SCENE_ID>(nextScene_), delta);
    SetDrawBlendMode(DX_BLENDMODE_MULA, 255);
}
DrawGraph(0, 0, screen_, 1);
```

A

 $A \times B$ 

B



白の部分にだけ描画される

# プレイヤーの強調

## 色の加算

プレイヤーの後ろのレイヤーに影を描画し  
SetDrawBright関数で影の色値を更新しそれを、プレイヤーに加算することでプレイヤーやタイトルが発光するような演出を作りました。

```
if (over == false)
{
    //影描画
    DrawGraph(
        drawPos_[0].x - PLAYER_HAFE_SIZE.x,
        drawPos_[0].y - PLAYER_HAFE_SIZE.y,
        image_["shadow"][num], true);
}
else
{
    //影描画
    DrawGraph(
        drawPos_[0].x - PLAYER_HAFE_SIZE.x - size,
        drawPos_[0].y - PLAYER_HAFE_SIZE.y - size,
        image_["shadow"][num], true);
}
SetDrawBright(255, 255, 255);
SetDrawBlendMode(DX_BLENDMODE_ADD, 255);
if (over == false)
{
    //プレイヤー描画
    DrawGraph(
        drawPos_[0].x - PLAYER_HAFE_SIZE.x,
        drawPos_[0].y - PLAYER_HAFE_SIZE.y,
        image_["normal"][num], true);
}
else
{
    //プレイヤー描画
    DrawGraph(
        drawPos_[0].x - PLAYER_HAFE_SIZE.x - size,
        drawPos_[0].y - PLAYER_HAFE_SIZE.y - size,
        image_["normal"][num], true);
}
SetDrawBlendMode(DX_BLENDMODE_NOBLEND, 255);
```



# Color3

## 色の計算と取得

GetColor関数で取得した変数をRGBの3つに分けてくれます。そこから加算、減算、乗算をしてくれます。

GetColor関数とユーザの直接入力の色コードが異なるので別々に計算するようにしています。

SetDrawBright関数を使用して、デフォルトに設定されているプレイヤーのカラーに合わせて数字のカラーも変化します。一枚の画像でいろんな色の数字を作ることができます。

```
const unsigned int GetColorCode(void)
{
    return GetColor(red, green, blue);
}

void Color3Code(unsigned int color) //GetColorをColor3に変換
{
    red = color / 0x010000 % 0x000100;
    green = color / 0x000100 % 0x000100;
    blue = color % 0x000100;
}

void Color3Code(int color) //色コードをColor3に変換
{
    red = color / 0x010000;
    green = color / 0x000100 % 0x000100;
    blue = color % 0x000100;
}
```

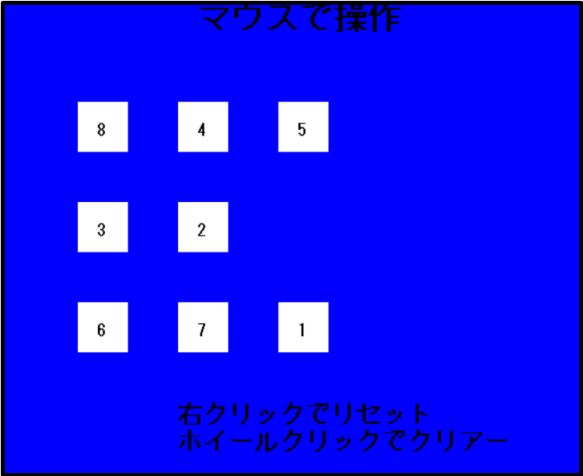
```
SetDrawBright(color .red, color .green, color .blue);
DrawRotaGraph(
    drawPos_[0].x + 24, ※プレイヤーのカラー
    drawPos_[0].y - 30,
    0.5, 0.0,
    image_["count"][player_.move],
    true);
SetDrawBright(255, 255, 255);
```

0123456789





スライドパズル

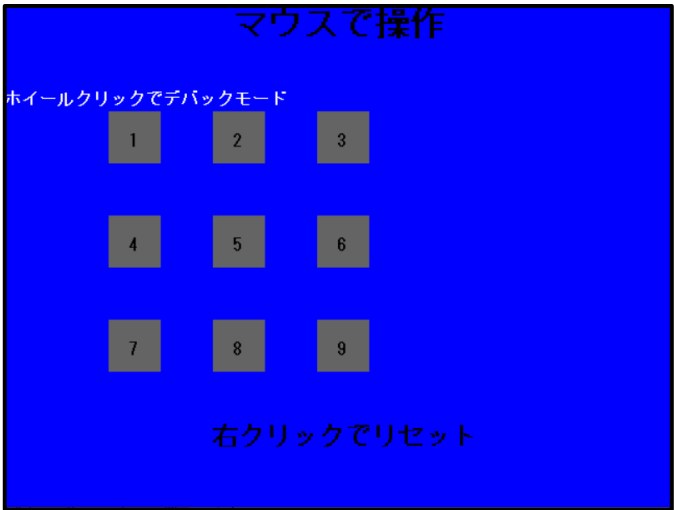


DXライブラリの、ランダム関数を使用し数字の位置がランダムになります。積んだと起用にリセットボタンもあります。カーソルを合わせた時のUIなどを工夫しました。

制 作 時 期	1年前期
使 用 言 語	C言語
使 用 ラ イ ブ ラ リ	DXライブラリ
制 作 人 数	1人

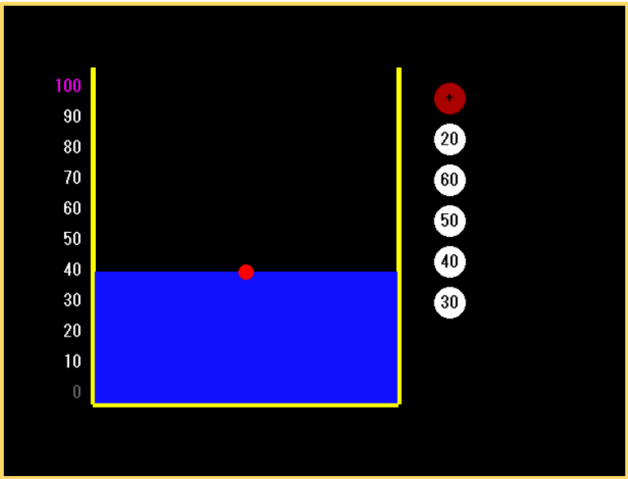
ライトブロック

制 作 時 期	1年前期
使 用 言 語	C++言語
使 用 ラ イ ブ ラ リ	DXライブラリ
制 作 人 数	1人



スライドパズルと同じ感じで制作しました。クリックしたの場所をSwicth文によって処理を分けています。それぞれクラスをインスタンスし、それぞれに属性を持たせています。

バケツゲーム

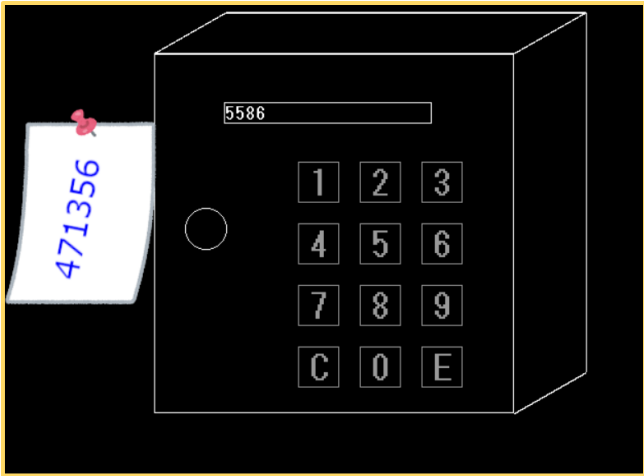


バケツに入っている水を入れたり、減らしたりして、目標の値にして赤いボールを手に入れるゲームです。ボタンを押したら一気に水が変わるのではなくゆっくりと水の量が変化します。

制 作 時 期	1年後期
使 用 言 語	C++言語
使 用 ラ イ ブ ラ リ	DXライブラリ
制 作 人 数	1人

金庫

制 作 時 期	2年前期
使 用 言 語	C++言語
使 用 ラ イ ブ ラ リ	DXライブラリ
制 作 人 数	1人



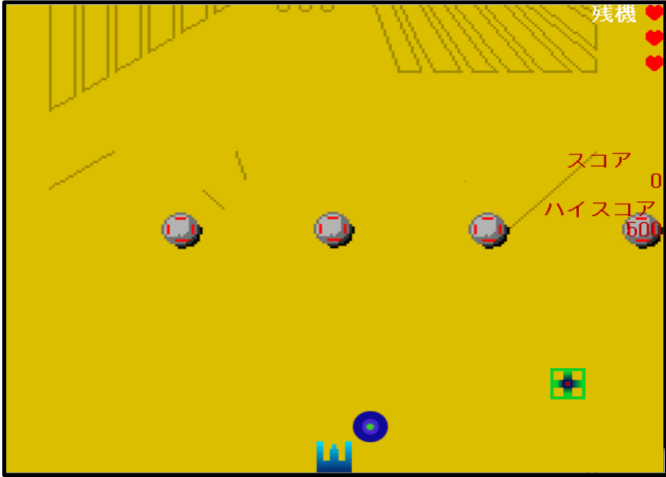
入力した文字列と設定された文字列が一致していたら金庫を開くことができます。  
Std::vectorでサイズと文字が一致しているかどうか確認します。20桁以上は入力できません。



# 授業作品



# Asoシューティング



C言語の基礎を学習しながら、ゲームを制作しました、画像のロード、描画、当たり判定、背景のスクロール、if文・else文などを学習しました。後期でC++言語化させたときは爆発のエフェクトとBossを実装しました。

制作時期	1年前期と1年後期
使用言語	C++言語
使用ライブラリ	DXライブラリ
制作人数	1人

# Asoインベーター

制作時期	1年前期
使用言語	C++言語
使用ライブラリ	DXライブラリ
制作人数	1人



C++言語を使用しクラスのインスタンス化、カプセル化、継承、隠ぺいとfor分のループ、switch文などを学習しました。キャラクターの選択、タイトルシーンとゲームシーンによって処理が異なるのが特徴です。

# AsoulQuest



クラスのオーバーロード、タイトルシーン→ゲームシーン→リザルトシーンのループ、画像を分担してマップを生成、スコアの保存とロード、シーンが変わるときにフェードインとフェードアウトを学びました。  
地下のステージ、ショップ、平原の3種類があります。

制作時期 1年後期

使用言語 C++言語

使用ライブラリ DXライブラリ

制作人数 1人

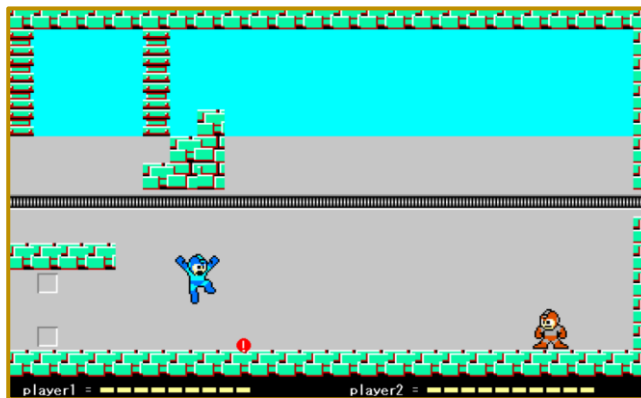
# AsoRockman

制作時期 1年後期

使用言語 C++言語

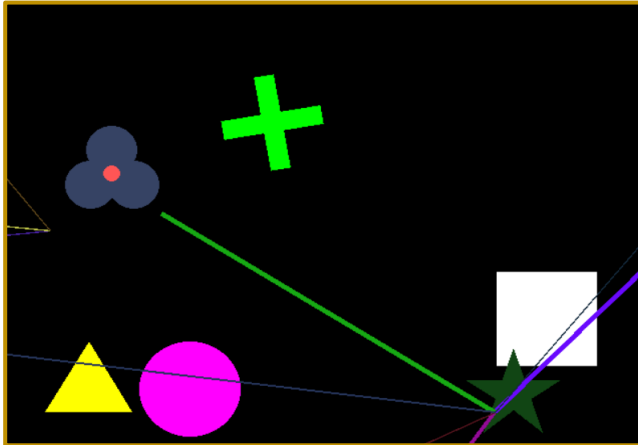
使用ライブラリ DXライブラリ

制作人数 1人



シーンごとの親のクラスをインスタンスしてシーンそれぞれがほかのシーンから受け取った情報をもとにシーンを作成していきます。  
対戦型のアクションゲームです。ライバルに9回弾を当てたら勝ちです。ステージにはいろんな仕掛けが用意されています。

## AsoShape



ユニークポインタを使ったプログラムです。生成処理、破棄処理、それぞれに違う処理を実装しました。様々な図形が飛び回り、図形同士がぶつкаると分裂や加速、原則などが起こります。万華鏡をイメージしてカラフルに仕上げました。

制作時期 2年前期

使用言語 C++言語

使用ライブラリ DXライブラリ

制作人数 1人

## AsoImage

制作時期 2年前期

使用言語 C++言語

使用ライブラリ DXライブラリ

制作人数 1人



画像の変形、環境変数、`std::vector`、`std::list`を学習しました。矢印がsinカーブを描いてめぐる動きします。  
`DrawRotaGraph`の派生についても学習しました。

## Asoシューティング2



敵の位置を確認しホーミングする弾を発射できます。敵はいろんなパターンの弾幕を撃ってきます。弾幕ゲームです。時間がたつと上からそれぞれの特徴を持つレーザーを撃ってきます。

制作時期 2年前期

使用言語 C++言語

使用ライブラリ DXライブラリ

制作人数 1人

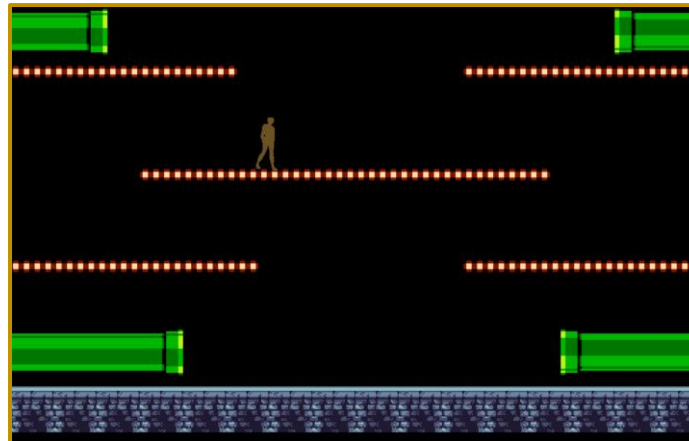
## AsoAdventure

制作時期 2年前期

使用言語 C++言語

使用ライブラリ DXライブラリ

制作人数 1人



TMXファイルのロードし、処理や画像のロード、クラスのシングルトンや、std::map、std::pairを学習しました。アクション系のゲームです。コマンドで技が使えます。



## AsoCarryLog



カプセルの当たり判定・カプセルの回転を利用して丸太を使って滝登りをして頂上にあるゴールを目指すゲームです。  
下まで行きすぎたら落ちたり、傾けすぎたり、すると爆発して落下してしまうところを工夫しました。

制作時期 2年前期

使用言語 C++言語

使用ライブラリ DXライブラリ

制作人数 1人

## ゲームからの脱出

制作時期 2年前期

使用言語 C++言語

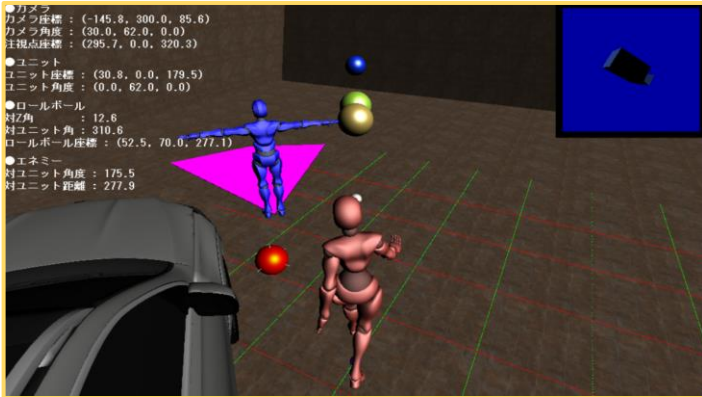
使用ライブラリ DXライブラリ

制作人数 3人



学内コンテスト2021(夏)に提出した謎解きアクションゲームです。  
謎を解いて出口を目指します。  
初めてのチーム制作で自分は仕掛けの部分を担当しました。今まで作ったスライドパズルやライトゲームなどを実装し、新しく、バケツゲーム、金庫も制作しました。

## 3DViewer



初めての3D作品です。3Dの基本とカメラ操作、モデルのロード、アニメーション描画、3Dの計算と当たり判定を使って、制作しました。移動して青いキャラクターの視線に入るとキャラクターが変色します。プレイヤーは弾を撃つこともできます。

制作時期 2年後期

使用言語 C++言語

使用ライブラリ DXライブラリ

制作人数 1人

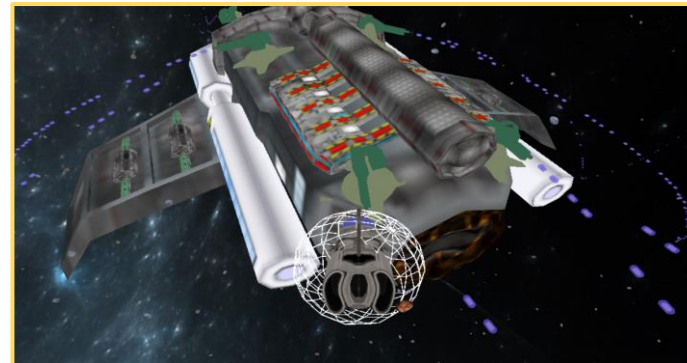
## Asoシューティング3D

制作時期 2年後期

使用言語 C++言語

使用ライブラリ DXライブラリ

制作人数 1人



3Dのシューティングゲームです。当たり判定をより正確にし、モデルの回転に合わせて、別のモデルが同期して一緒に回転します。自分の球が当たったら、敵が爆発してどんどん消えていきます。タイトル画面も2Dの文字が表れてよりゲームらしい感じになりました。



## RayTracing



古典的レイトレーシングを利用したプログラムです。床の画像をプレートにして無限に同じ画像が表示されます。スペキュラーやディフューザーなどの反射を利用した細かい計算を使っています。画像のドット値を入手し床に描画しています。

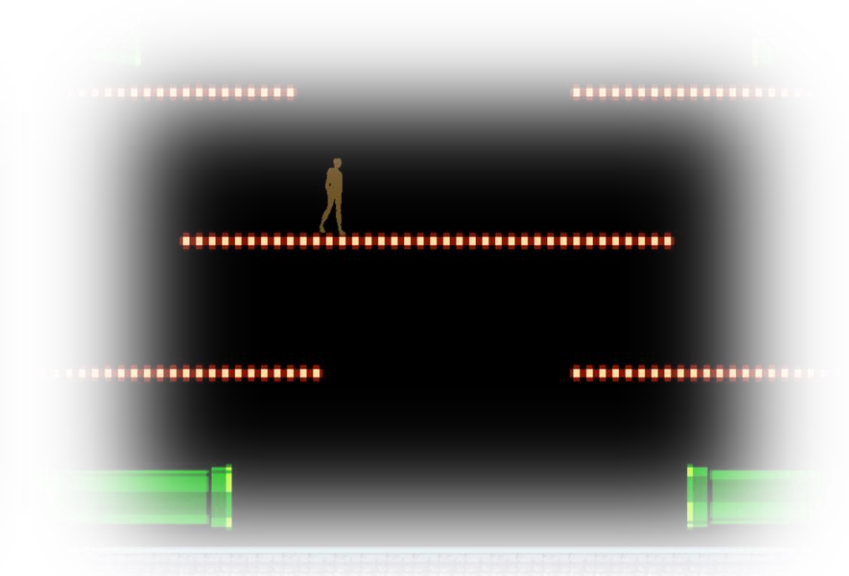
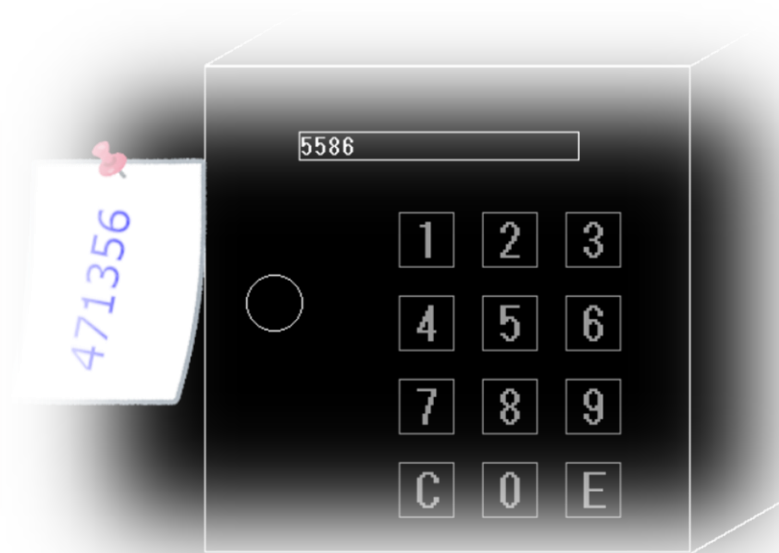
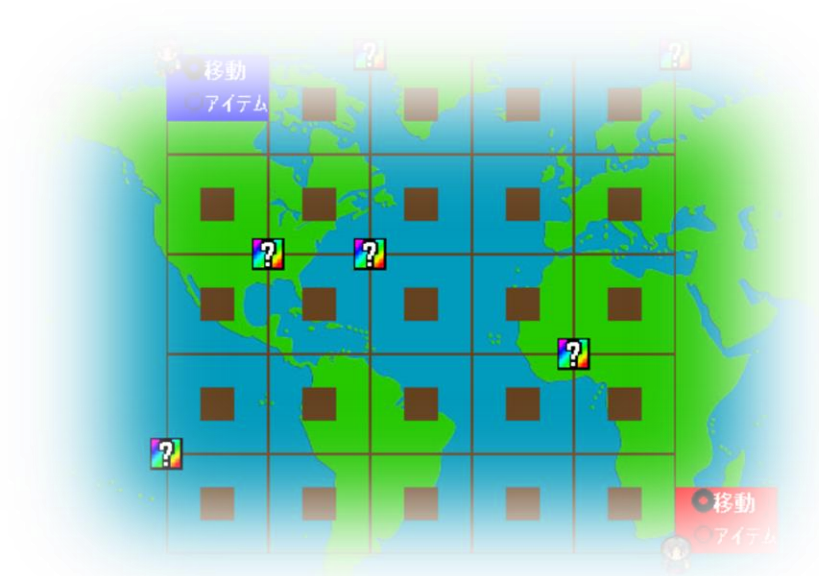
制 作 時 期	2年後期
使 用 言 語	C++言語
使 用 ラ イ ブ ラ リ	DXライブラリ
制 作 人 数	1人

## 3 DAction

制 作 時 期	2年後期
使 用 言 語	C++言語
使 用 ラ イ ブ ラ リ	DXライブラリ
制 作 人 数	1人



マリオギャラクシーのようなゲームです。それぞれの重力が異なる惑星が存在します。当たり判定やエフェクトの実装をしました。傾斜で止まると自動で滑っていくようにプログラムしています。3Dをより理解するためにタンクモードというものも実装しました。



ありがとうございました！

