

```

#include "RankingInputScene.h"
#include "../Utility/InputControl.h"
#include "DxLib.h"

RankingInputScene::RankingInputScene() : background_image(NULL),
ranking(nullptr), score(0),
name_num(0), cursor_x(0),
cursor_y(0)
{
    memset(name, NULL, (sizeof(char) * 15));
}

RankingInputScene::~~RankingInputScene()
{
}

// 初期化处理
void RankingInputScene::Initialize()
{
    // 画像の読み込み
    background_image = LoadGraph("Resource/images/Ranking.bmp");

    // エラーチェック
    if (background_image == -1)
    {
        throw ("Resource/images/Ranking.bmpがありません\n");
    }

    // メモリの動的確保
    ranking = new RankingData;
    ranking->Initialize();

    // リザルトデータを取得する
    FILE* fp = nullptr;
    // ファイルオープン
    errno_t result = fopen_s(&fp, "Resource/dat/result_data.csv", "r");

    // エラーチェック

```

```
    if (result != 0)
    {
        throw ("Resource/dat/result_data.csvが読み込めません\n");
    }

    // 結果を読み込む
    fscanf_s(fp, "%6d,\n", &score);

    // ファイルクローズ
    fclose(fp);
}
```

// 更新処理

```
eSceneType RankingInputScene::Update()
{
    bool is_change = false;

    // 名前入力処理
    is_change = InputName();

    // シーン変更は可能か？
    if (is_change)
    {
        // ランキング表示に遷移
        return eSceneType::E_RANKING_DISP;
    }
    else
    {
        return GetNowScene();
    }
}
```

// 描画処理

```
void RankingInputScene::Draw() const
{
    // 背景画像の描画
    DrawGraph(0, 0, background_image, TRUE);

    // 名前入力指示文字列の描画
```

```

DrawString(150, 100, "ランキングに登録します", 0xFFFFFFFF);
DrawFormatString(100, 220, GetColor(255, 255, 255), ">%s", name);

// 選択用文字を描画
const int font_size = 25;
for (int i = 0; i < 26; i++)
{
    int x = (i % 13) * font_size + 15;
    int y = (i / 13) * font_size + 300;
    DrawFormatString(x, y, GetColor(255, 255, 255), "%-3c", 'a' + i);
    y = ((i / 13) + 2) * font_size + 300;
    DrawFormatString(x, y, GetColor(255, 255, 255), "%-3c", 'A' + i);
}
DrawString(40, 405, "決定", GetColor(255, 255, 255));
DrawString(40 + font_size * 2, 405, "消す", GetColor(255, 255, 255));

// 選択文字をフォーカスする
if (cursor_y < 4)
{
    int x = cursor_x * font_size + 10;
    int y = cursor_y * font_size + 295;
    DrawBox(x, y, x + font_size, y + font_size, GetColor(255, 255, 255),
FALSE);
}
else
{
    if (cursor_x == 0)
    {
        DrawBox(35, 400, 35 + font_size * 2, 400 + font_size,
GetColor(255, 255, 255), FALSE);
    }
    else
    {
        DrawBox(0, 0, font_size, font_size, GetColor(255, 255, 255),
FALSE);
    }
}
}
}

```

```
// 終了時処理
void RankingInputScene::Finalize()
{
    // ランキングにデータを格納
    ranking->SetRankingData(score, name);

    // 読み込んだ画像を削除
    DeleteGraph(background_image);

    // 動的メモリの解放
    delete ranking;
}

// 現在のシーン情報を取得
eSceneType RankingInputScene::GetNowScene() const
{
    return eSceneType::E_RANKING_INPUT;
}

// 名前入力処理
bool RankingInputScene::InputName()
{
    // カーソル操作処理
    if (InputControl::GetButtonDown(XINPUT_BUTTON_DPAD_LEFT))
    {
        if (cursor_x > 0)
        {
            cursor_x--;
        }
        else
        {
            cursor_x = 12;
        }
    }
    if (InputControl::GetButtonDown(XINPUT_BUTTON_DPAD_RIGHT))
    {
        if (cursor_x < 12)
        {

```

```

        cursor_x++;
    }
    else
    {
        cursor_x = 0;
    }
}
if (InputControl::GetButtonDown(XINPUT_BUTTON_DPAD_UP))
{
    if (cursor_y > 0)
    {
        cursor_y--;
    }
}
if (InputControl::GetButtonDown(XINPUT_BUTTON_DPAD_DOWN))
{
    if (cursor_y < 4)
    {
        cursor_y++;
        if (cursor_y == 4)
        {
            cursor_x = 0;
        }
    }
}

// カーソル位置の文字を決定する
if (InputControl::GetButtonDown(XINPUT_BUTTON_B))
{
    if (cursor_y < 2)
    {
        name[name_num++] = 'a' + cursor_x + (cursor_y * 13);
        if (name_num == 14)
        {
            cursor_x = 0;
            cursor_y = 4;
        }
    }
    else if (cursor_y < 4)

```

```

{
    name[name_num++] = 'A' + cursor_x + ((cursor_y - 2) * 13);
    if (name_num == 14)
    {
        cursor_x = 0;
        cursor_y = 4;
    }
}
else
{
    if (cursor_x == 0)
    {
        name[name_num] = '\0';
        return true;
    }
    else
    {
        name[name_num--] = NULL;
    }
}
}

return false;
}

```