

PORTFOLIO

西口 煙大

CONTENTS

1. プロフィール

2. 就職作品プロジェクト

3. 個人制作実績

4. さいごに

1. プロフィール

其の一、自己紹介

- |- プロフィール
- |- スキル

其の一、自己紹介



氏名 西口煌大 Nishiguchi Kouta

生年月日 2004年11月19日生まれ
20歳

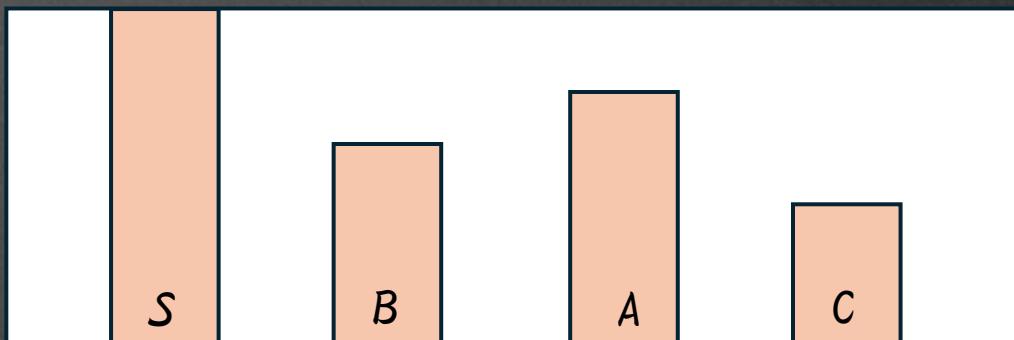
趣味 カラオケ・卓球

Sales Point

「言語化・概念理解に
神経を注ぐプログラマー」

スキル

最高 S



2. 就職作品プロジェクト

其の一、全体像

|- 概要

其の二、こだわりたい所

|- 可読・保守性

|- トーンレンダリング

|- 入力

其の一、概要

ゲーム概要

タイトル : 隕石の止め方(仮)

操作 : コントローラー

ゲームシステム : コマンドゲーム

ゲーム画面イメージ

落ちる隕石を
コマンド入力で何かするゲーム

未決定



其の二、こだわりたい所

①可読性・保守性の確保

可読性

- ・命名規則
- ・最低限のコメント

保守性

- ・private指定子
- ・constの使用 など

コールバック関数を使用する。

#defineの削除を徹底

```
// 指定ループ目で更新するための名前空間
namespace Counter
{
    int Increment(int counter); // > > > > >
    bool isLimitOver(int &counter, int counterLimmit);
    int AnimationLayer(int, int); // > > > > >
    int AnimationLayer(int, int, PLAYER_STATE); // > > > > // Player用オーバーロード

    // 上記四つを使ったAnimation用関数
    void Animate(int& in_Counter, const int in_CounterLimit, int& in_NumU, const int in_NuMLimit);
    // 一定周期の更新に対して行う処理をコールバックする
    void UpdateConstant(int& in_Counter, const int in_CounterLimit, Callback in_Func);
}
```

Callback in_Func;

Defineで容易に作成できたマクロを多用することで
起きていたデータサイズの膨張を廃止。
再利用性を意識した関数の作成をおこなう。
保守性・可読性の両立を図る。

其の二、こだわりたい所

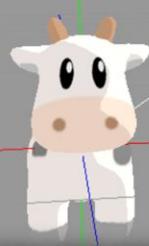
②トゥーンレンダリング

現在学習中!!

授業用プロジェクトの実行結果

光源

根本的な部分から理解し



使用しているランプテクスチャ

1番の

アピール

ポイントにする！

・モデル

- 頂点

- 法線

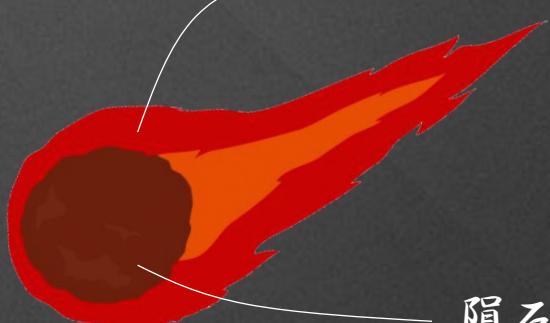
- ボーン(親子関係)

・セルシェーディング

・輪郭線

目標：違和感を生まないトゥーン表現

まとうプラズマ



隕石の輪郭



識別しやすく

前のページへ

次のページへ

其の二、こだわりたい所

②トゥーンレンダリング



参照URL:<https://www.youtube.com/watch?v=0DuLzP0H4oI>

落ちる隕石のデサチュレーション演出を実装したい。
アニメの力がたまつた瞬間に画面を一瞬モノクロに切り替える演出

力がたまる瞬間を可視化することによって印象に残る作品にする。

鉛筆で描いたような輪郭線を実現する

其の二、こだわりたい所

③入力

作るゲームは面白く!!

ゲームシステム:コマンド

P.6 参照

表現にコストを注力するため

ゲームシステムはシンプルにする

入力にこだわる!

- ・1入力毎にSEを入れる(ド→レ→ミ..)
- ・人気ゲームのコマンドをオマージュ

3. 個人製作実績

其の一、棒人間の乱

- 概要
- 特徴
- こだわり

其の二、棒人間のDAN

- 概要
- こだわり

其の一、棒人間の乱



製作期間

2023年/11月～ 2024年/02月

ジャンル

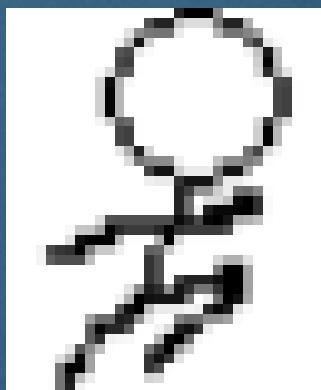
ランアクション

使用言語 / 開発環境

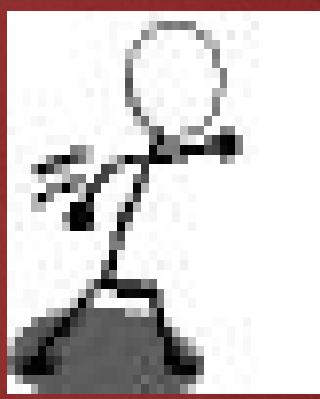
C / コンソール

Tips : 走る(RUN)と掛けけてます！

1年生の **初制作！**

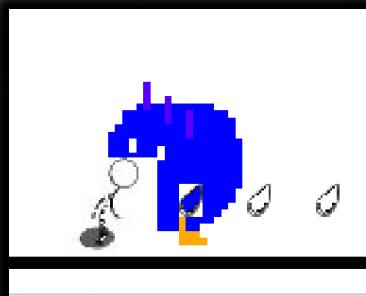


「ジャンプ」
SPACE



「パンチ」
ENTER

で遊べます



「敵に当たる」



「穴に落ちる」と

GAME OVER

前のページへ

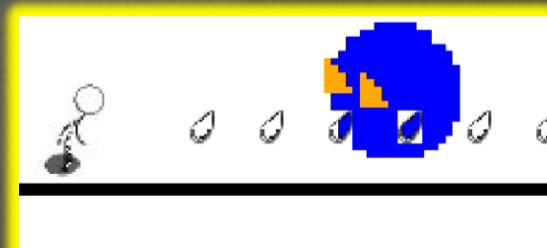
次のページへ

特徴 「死体 ~~跳~~ 跳び！」

パンチで “敵をぶつ飛ばす！” ことができます



しばらく進むと倒れた敵が“倒れたまま”現れて..



もう一回 ぶつ飛ばす



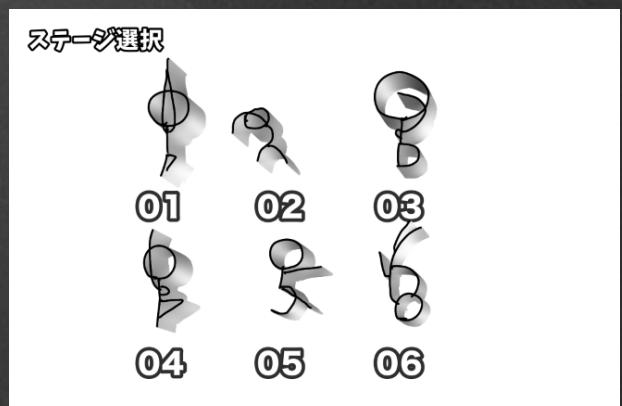


棒人間の乱 制作時のこだわり

① イラストは
すべて自作！



体で数字を体現した
可愛げのある棒人間





棒人間の乱 制作時のこだわり

② ぶつ飛ばされる敵が
シユールに見えるようにした

Enemy構造体

```

102 struct ENEMY {
103     ANI picture;
104     unsigned char data[12];
105     void Init()
106     {
107         this->picture.ani[0] = this->picture.one;
108         this->picture.ani[1] = this->picture.two;
109         this->picture.ani[2] = this->picture.three;
110     }
111     void Draw()
112     {
113         DrawBmp(xp: this->picture.pos.x, yp: this->picture.pos.y, this->picture.ani[this->data[this->picture.cnt]], tr: true);
114     }
115 };

```

Enemyのアニメーション構造体

```

68 struct ANI {
69     POINT pos;
70     Bmp* one; // BMP画像・1～3枚
71     Bmp* two;
72     Bmp* three;
73     Bmp* ani[3]; // 上記3つのBmp*を格納する配列
74     unsigned char cnt = 0; // カウンター変数
75 };

```

Enemy宣言

```

94 ENEMY blue = { .picture: { .pos:
95     .one: LoadBmp("pic/enemy2_1.bmp"),
96     .two: LoadBmp("pic/enemy2_2.bmp"),
97     .three: LoadBmp("pic/enemy2_3.bmp"),
98     .ani[0]: blue.picture.one, .ani[1]: blue.picture.two, .ani[2]: blue.picture.three,
99     .data: { 0,1,1,1,1,1,1,1,1,1,1,2 } },
100 };

```

コマの速さを調整



前のページへ

次のページへ

其の二、棒人間のDAN



棒人間のDAN!

Tips : 弾(だん)と掛けてます！

製作期間

2025年/2月、6月

ジャンル

約2ヶ月

2Dアクション

使用言語 / 開発環境

C++ / DirectX

2年生制作課題

条件 : std::vector<>を使用

Player構造体

```
28  class Player : public ... {  
29  private:  
30      PLAYER_STATE m_State;           // 状態  
31      vector<Bullet*> m_MyBullets;    // 所持する弾  
32  };  
33  
```

基本操作



「移動」

← A D →



「弾を打つ」

ENTER



「ジャンプ」

SPACE

前のページへ

- 16 -

次のページへ



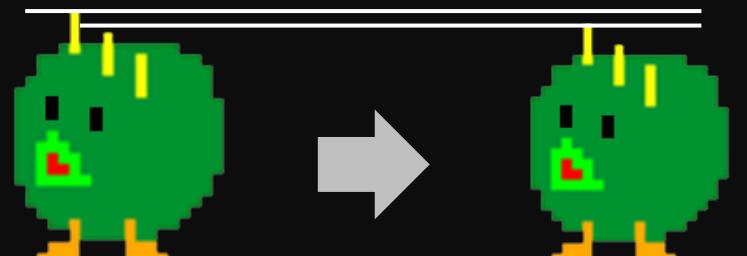
棒人間のDAN 制作時のこだわり

①「弾を打ち込む爽快感」



ひたすら敵に打ちまくりたくなる
を目指して制作

弾が当たっても
敵が小さくなるだけで地味だけど...



SEで「当てた手応え」に
注力



音声は
動画で

当て続ける事で
楽しくなるように



棒人間のDAN 制作時のこだわり

② 必ず2回はプレイさせる



初見プレイでは
絶対できない要素を
リザルトに挿入して
プレイ意欲を増加!!

プレイ回数 増加!! 詳しくは動画を参照



弾を撃ち込む回数 増加!!



爽快感 増加!!

ご覧いただき
ありがとうございました



西口 煌大

