ポートフォリオ

芦　恒一郎

1. アクションゲームの概要

　このゲームは大学1年及び2年でそれぞれ半年程度かけて作成したアクションゲームに対し、ステージ作成ツールの運用も兼ねて大学4年の時にステージを一新し、新たなシステムをいくつか加えたアクションゲームです。

制作の経緯としては、2Dアクションは自分の好きなジャンルであり、サークルのゼミで基礎を学んだのもあったため大学1年の時に制作しました。大学4年での制作は後述するステージ作成ツールを改良したのでそれに対応したゲームを作るために制作しました。

1. ルール説明

　横スクロールアクションで、ゴールを目指して進んでいきます。

　プレイヤーはジャンプと攻撃ボタンがあり、攻撃ボタンを押して敵を斬ることで倒せます。トゲなど一部の敵は倒すことができません。

　HPが0になったり穴に落ちたりするとゲームオーバーです。

1. ゲーム実行

実行するときはゲーム実行用ファイル内のAction2018.exeを実行してください。ハイスコアはsavedataファイル内のscore.datを削除することによりデータを初期化することができます。

1. 操作方法

ゲームパッドのボタン表記はロジクールのゲームパッドに対応しています。その場合、裏面のスイッチはDにしてください。

・操作方法

キーボード

　上下左右　　　移動・項目の選択

Zキー　　　　ジャンプ・決定

Xキー　　　　攻撃・キャンセル

SPACEキー　　ポーズ

ゲームパッド

上下左右　　 移動・項目の選択

AボタンまたはBボタン　 ジャンプ・決定

XボタンまたはYボタン　 攻撃・キャンセル

STARTボタン　ポーズ

1. 開発環境

* プログラム部分

　言語：C++

　開発環境：Microsoft Visual C++ 2015 Community

　使用ライブラリ：DXライブラリ

* 画像作成

　高機能ドット絵エディタ EDGE

　AzPainter2

* エフェクト作成

　2Dゲームエフェクト作成ツールProminence 1.6.2

* 音楽作成

　GaregeBand(音楽、効果音)

　sfxr - sound effect generator(効果音)

1. 工夫した点
2. ステージの構成

　ステージ制作ツールを使うことにより、実際のゲームプレイ画面を見ながら制作できるので配置に拘ったステージを制作しました。この作品ではステージ制作ツールを使うことをメインにしたのでボスは用意していないですが、マンネリ化しないようプレイしている人から見てインパクトがあるようなステージ構成を重視して制作しました。

　音楽・背景に関しても多くの種類を用意することでマンネリ化しないようにしました。

1. ステージの読み込み

　ステージはソースコード中に記載せず、テキストファイルから読み込んでいます。これにより、マップファイルを書き換えるだけでソースコードに手を加えずにステージ構成を変更することができます。

　マップファイルにはブロックや敵の配置だけでなく、敵の向きや速度といった情報まで持たせてマップファイル内でステージ情報を完結させています。

1. ステージの難易度分け

　大学祭には初心者から長時間やりこむ方まで来場するため、難易度を分けています。

　EasyやNormalの序盤は操作に慣れていない人でもクリアしやすいようにしています。Easyはライフの数自体を多くすることでミスを許容しやすくする調整も行っています。

Hardはステージを短くしてリトライ性を上げた一方で難易度はかなり高くしてあります。

1. 加速度の調整

　操作しやすさを重視し、移動開始時やブレーキをかける時は速度の変化量を多くし、移動中は加速が緩やかに上がるようにしています。

1. アニメーション

　敵を倒した時の星が弾けるエフェクトなど、各種アニメーションをファイルから読み込み表示できるようにしています。プレイヤーがダメージを受けた時やゲームオーバー時のエフェクトは透過度を変更することで表現しています。

1. メニュー画面

　メニュー画面の動作についても自作しています。ポーズ画面からのステージ選択や、その状態からキャンセルを押した場合など複雑な画面遷移でも正常に動作するようにしました。また、フェードアウト・フェードインや音楽の停止・再開なども実装し、自然な動作を行うようにしています。

1. 現在の自分が見たソースコードの改善すべき点
2. ブロックや敵の情報を管理する配列の処理効率が悪すぎる

現在の実装ではブロックも敵もマップファイルから読み込んだ時の位置を２次元配列の行と列として持っています。そのため、マップが広くなるとその大きさだけ配列を取ってしまうため、非常に無駄が多くなってしまいます。

このままだとステージ構成によっては計算速度に影響が出てしまう可能性があります。

ブロックや敵はマップの大きさ分の配列を使用するのではなく、１次元配列に前詰めで必要な分だけ使用するべきです。

大学１年の時に書いたこの仕様はプログラム全体に存在しています。修正するのは非常に手間がかかるため残ってしまっています。

1. 敵の種類の初期化の記述が冗長

敵の動きの種類は横移動、回転、追尾など数種類にまとめられるので動きの種類を一つ変数に持ち、種類毎に初期化処理の関数も作成したほうが分かりやすそうです。

1. 関数化されていない処理が多い
2. 変数名が何を指しているか分かりづらいものが多い
3. 構造体やクラスを使用していない関数・変数が多い
4. 定数宣言されていない定数がある
5. 敵が弾を撃つ時の処理が関数化されていないため読みづらい
6. for文にかっこが使用されていない箇所が存在する