**艦隊司令長官**

大原情報ビジネス専門学校

丁　俊宇

**・実行方法**

WindowsでProjectWarshipbattle.exeを開けばゲームを起動できます。

操作はすべてマウスでできますが、テストのために、一部の操作はキーボードでできます。

WキーとSキー：出力増大／減少

AキーとDキー：舵の方向変更

Sキー：射撃

0キー：デバッグモードを開く

9キー：操作している船にダメージを与える(沈む演出テスト時に使った)

他の操作はすべてゲーム内のチュートリアルに掲載しています。

**・制作期間**

2019.1.22-2019.5.31 合計四ヵ月

**・制作環境**

Windows　PC、Visual Studio 2017

**・制作意図**

C++を利用してゲーム内部の処理の流れを研究します。その上にＡＩと物理演算を実現します。プログラムのほうは、過去の考え方を捨てて、完全オブジェクト指向に変更するようにします。それで、メモリ関係とプログラムの効率の向上とコードの再利用性を中心にプログラムを組立ちました。

**・担当箇所**

ゲームプランナー、ＵＩデザイナー、ゲームプログラマー

**・プログラムを作成する上での苦労した箇所**

-ゼロからゲームを組み立つ：

　専門学校に入った一年の中にプロトタイプも含めて二タイトルしかなりませんでした。前のゲームを作る時にオブジェクト指向の知識はまだ学んでいませんでしたため、拡張性がすごく低いでした。

　今回のゲームを作る前に、大体な動きを想定して、ピラミッドと同じように上から下まで作って、具体的な処理は全部下の部分から処理するようにしました。DXライブラリを使いましたが、データの関係や保存方式はすべて自分で決めないといけません。この部分は全部一人でできました。

-完全オブジェクト指向：

　完全オブジェクト指向に目指したが、一部分の全体関数（特に計算用関数）と定数を指定しました。しかし、それ以外の物はすべてオブジェクト化にして、各クラスの間にデータを渡しながら処理を行います。それに、全体変数の使用もありませんでした。

-物理演算と３Ｄ演算：

　船の移動と弾の飛行をリアル世界に似ているようにしたいため、自分はリアルの事実に基づいて物理計算式を決めて、物理演算を実現しました。船を移動する時にエンジンから出力をもらって、その力と重さから加速度を計算します。弾が飛ぶ時にも空気抵抗と重力の影響を受けます。実際に弾の射程を決まるのはいつ海に落ちることとなりました。

　最初に３Ｄ演算を使って弾の飛び方を再現したいと思います。それに、もし３Ｄモデルを手に入れば描画方式を変更して、ゲームを３Ｄ化することもできます。

**-ロックシステムとＡＩ（注意されてほしい）**

　戦闘中に砲塔と船の向きを調整することが難しくなりと思い、元の仕様の上にロックシステムを作りました。平面の調整は簡単でしたが、砲塔の垂直角度を調整することが難しいでした。物理演算に使われた計算方式でしたが、シミュレーション系の計算が多くため、射程を測る方法は一回弾を撃つしかありませんでした。しかし、船の数が増えていくと処理の数がかなり多くなると考えました。そこで二分探索と先処理この二つの手を使いました。実装する時にも影響が出ませんでした。

　実際の海戦を研究した後、プレーヤーそれぞれ船を操作するより、プレーヤーはフラグシープとして動くほうがリアルと思いました。そこで初めてＡＩを実装しましょうと決めました。

　今のＡＩ業界ではPythonを使って人工知能を実現することもできましたが、ゲームの中では普通イベント制とラーニングを使えばうまく行けると考えています。自分のゲーム中にもイベントに基づいてＡＩを実現しました。リーダーＡＩと一般ＡＩで敵と友軍の他の船を操作するようにしました。

**・別ファイル**

-使ったライブラリ

　ＤＸライブラリを使いました。

<https://dxlib.xsrv.jp/>

-船画像：

[https://ja.wikipedia.org/wiki/](https://ja.wikipedia.org/wiki/%E3%83%A1%E3%82%A4%E3%83%B3%E3%83%9A%E3%83%BC%E3%82%B8)

-音声データ

<https://dova-s.jp/>

<https://soundeffect-lab.info/sound/machine/>

<https://soundeffect-lab.info/sound/machine/machine2.html>