

# Fasted commander

## ドキュメント



新潟高度情報専門学校  
ゲーム開発専門家  
ゲームクリエイターコース 3 年

三膳 祐太  
平成 26 年 9 月 24 日

## 目次

・ 製作意図.....	3
・ コンセプト.....	3
・ アピールポイント.....	3
・ 開発環境.....	4
・ ゲーム内容.....	5
・ ルール説明.....	6
・ 考察　まとめ.....	9
・ プログラム一覧.....	9

## 製作意図

今まで教わった技術を使い、3D のゲームを作成し完成させること。

また東京ゲームショウへの出展及び、日本ゲーム大賞応募作品に応募することを決め製作を行いました。

ブロックで地形を作り、その地形を生かしてゲームを作りたいと思い、また当時レースゲームのプログラムを作っており、それを流用して、このゲームの製作を行いました。

## コンセプト

- ・今までに教わったことや自分で調べたことを生かし 1 つのゲームを作り上げる

今まで教わってきたことを生かし 3D のゲームを完成させることを目標に製作に入りました。

- ・常に新鮮な感覚でプレイできるマップとルール

ゲームでマップを走る途中動的にマップの地形を変化させ、ただ周回するのではなく、その時・その周回でコースの内容が変化するようにし、常に新しい間隔でプレイができるようなルールを目指して設計しました。

- ・多人数でワイワイと騒ぎながらできるような設計

多人数プレイを想定して設計し、友達同士でワイワイできるようなゲームを目指しました。

## アピールポイント

- ・1 人で設計から開発までを行いました。

今回、1 人で開発を行うことになり、設計から開発までをすべて 1 人で行いました。

行列の計算やカメラの角度など 3D の基本的な設計を自分で組みながら覚え、3D の基礎的な操作を行えるようになり自分のスキルアップに繋がりました。

また、1 人で開発を行いスケジュール管理にも気を使い、意欲的に開発に専念することができました。

- ・ビューポートを分割し 4 人対戦を実現

今回多人数プレイを想定し、ビューポートを分割して、コントローラーを 4 つ対応させ、4 人用の分割画面を実装しました。

# 開発環境

使用 OS

Microsoft Windows7 64bit

使用アプリケーション 開発言語 ライブラリ

- ・ Microsoft Visual Studio 2013
- ・ C++
- ・ Microsoft DirectX 9.0c SDK (June 2010)

製作期間

- ・ 2014 年 4 月～6 月

製作メンバー

- ・ プログラマー 1 人
- ・ デザイナー 1 人

自分の担当箇所

- ・ ゲームの仕様・設計
- ・ プログラム
- ・ 画面レイアウト

サウンド

BGM

DOVA-SYNDROME <http://dova-s.jp/>

SE

ザ・マッチメイカーズ 2nd <http://osabisi.sakura.ne.jp/m2/>

## ゲーム内容

アメリカのある地域に、戦車が好きで戦車を購入し、自分好みに改造を施し、街中を走ったりをしている 4 人の戦車マニアがいました。

4 人は今回のレース用に改造し、戦車を使い誰の戦車が一番早いかを決定するため、レースを行うことにしました。色々なコースを回りながら総合順位を競い総合で勝った者は **FastestCommander** を名乗れるようになり 4 人の中で一番早いと名乗ることができるようになります。

## ルール説明

4 人のプレイヤーは **FastestCommander** 用に特別改造した戦車を使い、マップを誰よりも早くゴールし、総合順位で 1 位になることが目標です。

4 つの戦車には特別な改造がしており、装甲など重いものを全部外し、ジャンプが出来るようになっています。

またジャンプを行うとその勢いで主砲から特別性の弾が発射され、他の戦車や壁などに当たっても何も起こりませんが、地面に落ちるとその地面に特別なオイルを地面に張り付けます。オイルが付けられた地面は色が変わり、オイルを付けたプレイヤーはその地面に触れると加速することができます。 ですが、他のプレイヤーのオイルに触れてしまうと戦車はそのオイルとの相性が悪くスリップして減速してしまいます。

プレイヤーは自分に有利な地形を作りつつ相手より早く進むことが重要になります。

## 操作説明

- ・タイトルでの操作



スタート画面はコントローラー1しか反応しません。

コントローラー1の人はAボタンを押すとプレイヤーの人数選択画面に移行し、プレイヤーの人数を選択します。

## ゲーム画面での操作

左スティック … 右に入れると右旋回

左に入れると左旋回を行います。

旋回直後はあまり曲がりませんが、ずっと入れ続けることに徐々に曲がりやすくなります。

右スティック … タレット(砲塔)を旋回させます。

右に入れると右旋回、左に入れると左旋回を行います。

旋回をやめると元の位置に戻ろうとします。

RT トリガー … アクセルボタン

押し込みの強さにより加速の数値が変わります。

LT トリガー … ブレーキボタン

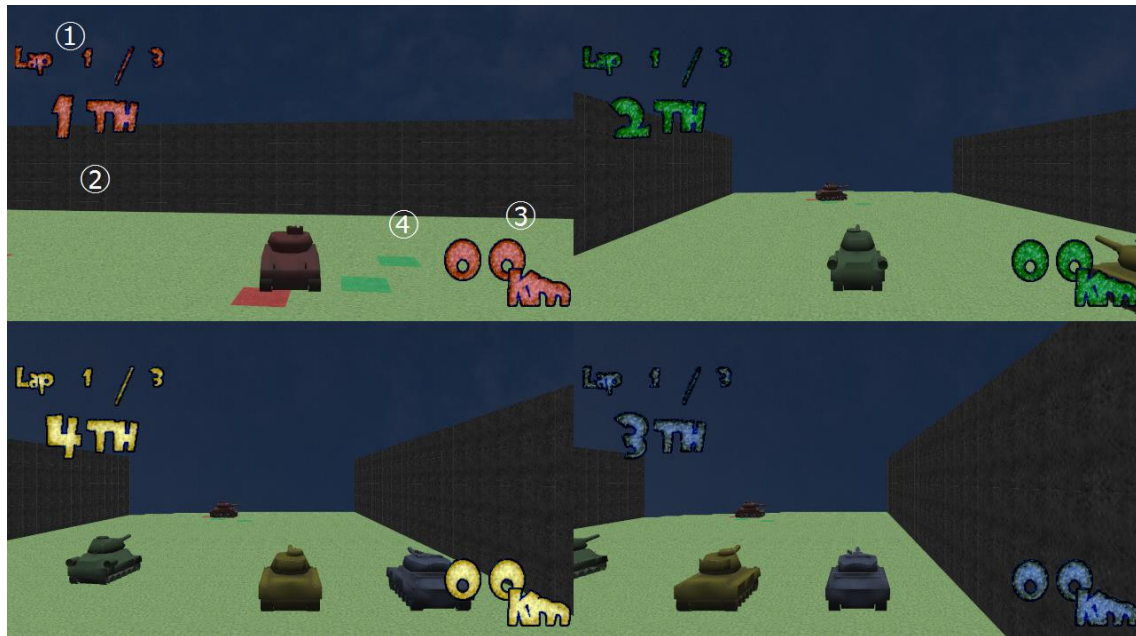
押し込みの強さにより減速の数値が変わります。

A ボタン … ジャンプボタン

戦車がジャンプを行い、地面の地形効果や段差などを飛び越えることができます。

またジャンプを行った時に戦車は弾を発射することができ、弾が地面に触れると自分の色のオイルを地面にまき散らします。

## ゲーム中の説明



プレイヤーは規定周回を誰よりも早く走るのが目的です。

### ① 現在のラップ数

マップの規定ラップ数と現在のラップ数が表示されます。

### ② 現在のマップ上でのランキング

現在のマップでのランキングが表示されます。

### ③ 現在の速度

自分の現在のスピードが表示されます。

### ④ 色の付いた地形

自分または他のプレイヤーが付けた地面です。

地面には

基本の地面と

色のついた地面が存在します。

基本の地面には何も効果が無いです。

ですが戦車の弾が触れると色の付いた地面に変化します。



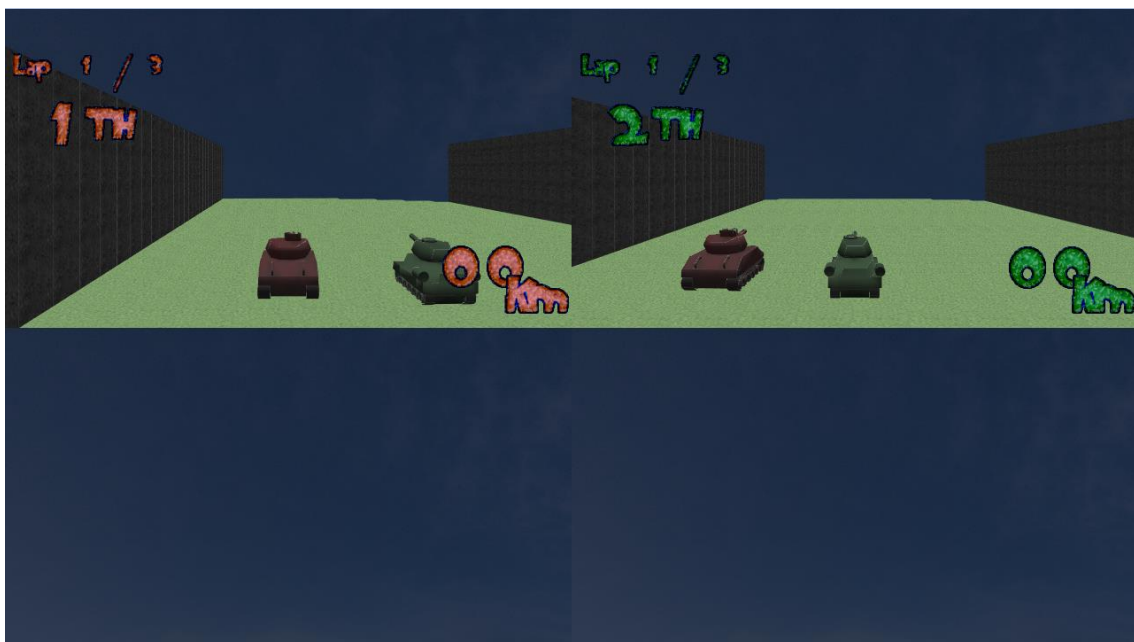
赤・緑・黄・緑の地面には  
それぞれの戦車の弾から出たオイルがついています。

自分の色のオイルを踏むと滑りがよくなり通常よりも一時的に加速を行うことができます。  
ですが自分以外のオイルを踏むと戦車との相性が良くなく、滑ってしまいその場に乗っている間最高速が落ちてしまいます。また乗っている間はジャンプを行うことが出来なくなってしまうです。

黒い地面は元々と汚れている地面でどの戦車に乗ってもスリップを起こしてしまいます。

マップはそれぞれ

草原



最初のマップ  
曲がり角が少なく道も広いので速度を出しやすい

## 雪原



## 2つ目のマップ

地形が狭く、曲がり角が多い

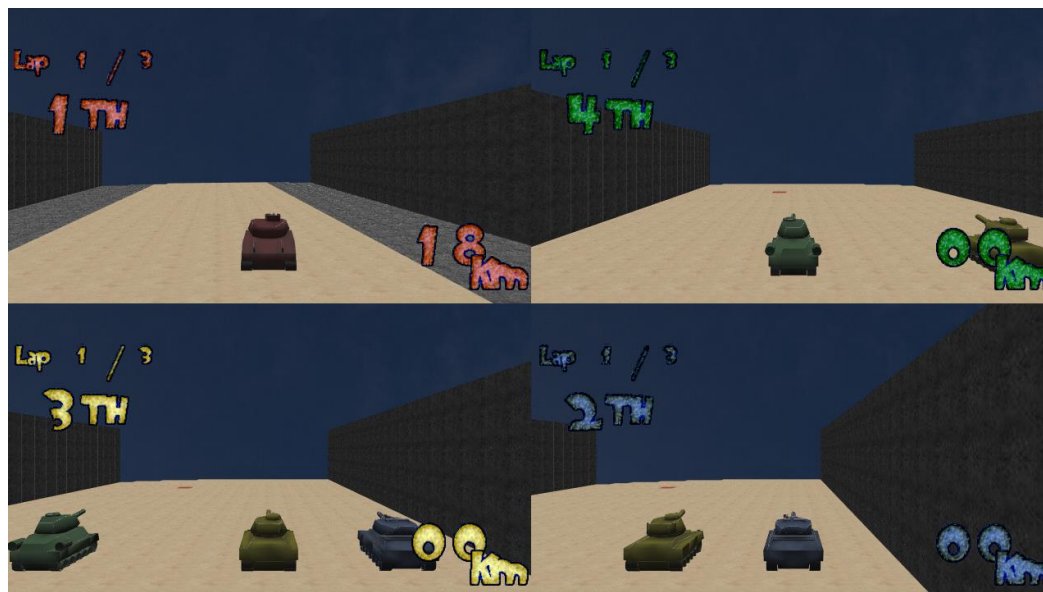
## 山脈



## 3つ目のステージ

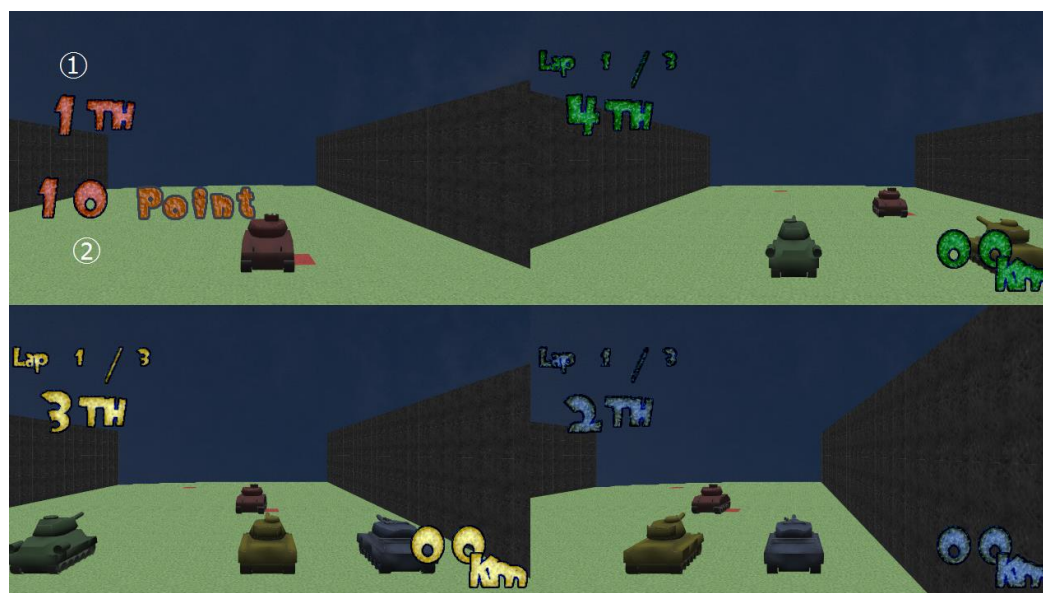
段差が多く弾を狙ったところに撃ちづらい

砂浜



黒い地面が多く道が広いので迷いやすい  
この4つのマップを走り、順位を競います

周回後のリザルト画面



周回を終えるとその合計スコアが表示されます。

① プレイヤーのリザルトでの順位を表示します。

② プレイヤーの現在の合計値スコアを表示します。

全員が周回を終えてスコアを確認したら次のマップへ移行します。

4つのマップを回り最終的に一番スコアが高かったプレイヤーが勝者となります。

## 考察

自分は今まで 2D のゲームしか作っておらず、3D 対して難しいと思ってしまい中々手を付けられませんでした。ですが今後 OpenGL などの 3D ゲームを開発したいと思い。まずは DirectX からやることにしました。初めての 3D ゲーム製作を 1 人で体験し、スケジュールの管理が甘く、細かい所まで修正を出来ずに中々完成度を上げることが出来ませんでした。

製作中効率化を考え、新しい仕様などを試行錯誤しながら実装し、3D の基礎的な設計を行うことができ、自分のスキルアップを意識して行えました。

まだ自分はプログラミング技術が低く、新しい技術などに対して消極的になっていますが、より派手で見て楽しいゲームを作れるようになりたいと考えています。

## 自分の作成したプログラム

### • bullet クラス

プレイヤーの発射する弾の描画や管理を行うクラスです。

### • player クラス

Character クラスを継承し、プレイヤーごとの管理を行います

### • playerAI クラス

ゲームでプレイヤーが動かした情報を txt に書き出し読み込むことでリプレイとして動かします。

### • character\_base クラス

キャラクターの共通操作をこちらで定義し、これを継承した player クラスで細かい個別の仕様を行います。

### • fieldobject クラス

ブロックの管理と描画を行います。

### • field クラス

マップ用 csv を読み込みリストに書き出しプレイヤーの周りだけ当たり判定の管理などを行います。

•multi\_game クラス

game\_base クラスを継承し、多人数用のゲーム画面を表示し、管理します。

•solo\_game クラス

game\_base クラスを継承し、1 人用のゲーム画面を表示し、管理します。

• game\_base クラス

ゲームでの共通の動作をこちらで定義します。

•progress クラス

関数ポインタを使い、状態遷移を管理するクラスです。

• title クラス

タイトル画面を表示・管理するクラスです

• flow\_base クラス

状態遷移はこのクラスを継承し、状態遷移内で関数ポインタとして扱うことで遷移の効率化を行いました。

• checkpoint クラス

チェックポイントの管理 現在のランキングなどを表示するクラスです

• speed クラス

プレイヤーのスピードを表示・管理するクラスです。

• time クラス

ラップタイムやカウントダウンなどを行うクラスです。

• layout\_base クラス

スプライト描画は主にこのクラスを継承して行います。

• camera クラス

カメラと光源を管理するクラスです。

•collision

レイを使い当たっているかの判定を行うクラスです。

- **common** クラス

ゲーム共通で使う情報などを管理しています。

- **object\_base** クラス

フィールドのブロックの元となるクラスで、これを継承して設計を行いました。