

情報コミュニケーション創成 レポート

クラス B4 (グループ D)

REN JIMIN - 2600130451-7

提出日:2015 年 7 月 20 日

目 次

1	目的	2
2	プログラムの機能	2
3	外部仕様	2
3.1	起動方法	2
3.2	操作方法	2
3.3	画面遷移	2
4	内部仕様	3
4.1	クラス図	3
4.2	各クラスの説明	3
5	実行例	4
5.1	ディスクトップ	4
5.2	プログラム起動した直後	5
5.3	ゲーム画面	6
5.4	ゲームオーバー	6
6	工夫した点と考察	8
6.1	工夫した点	8
6.1.1	プログラム上	8
6.1.2	機能上	8
6.1.3	チームワーク	8
6.2	感想	8

1 目的

今回のプロジェクトでは限られているシステム内のソフトウェアを用い、オリジナルなオブジェクトを作成する目標であるので、プロトプトで実行するプログラムより、ブラウザを使い、グラフィックスユーザインターフェイスを実装したゲームの方は迫力的であると考えられる。今回はこのようにブラウザで実行するゲームを開発した。

2 プログラムの機能

AirForce は、簡単な 8bit 風のシューティングゲームである。キーボードで戦機をコントロールし、弾を射撃し、敵を倒すことができる。

3 外部仕様

3.1 起動方法

コンピューターを立ち上げると、ディスクトップに大きな AirForce のアイコンがある。このアイコンをダブルクリックすると、ブラウザが起動し、プログラムが自動的に実行する。

3.2 操作方法

プログラムが実行すると、左側はゲーム画面が表示し、右側はプレイヤーのヘルスポイント、点数を表示する。ゲームが自動的に始まる。キーボードの WSAD キーを使うことにより、上下左右へ移動することができる。J キーを押すと、弾を打つことができる。

3.3 画面遷移

ゲームが始まった約 30 秒後、ボスが現す。ボスを倒せば、ゲームが終了し、Mission Accomplished という勝利画面を表示する。

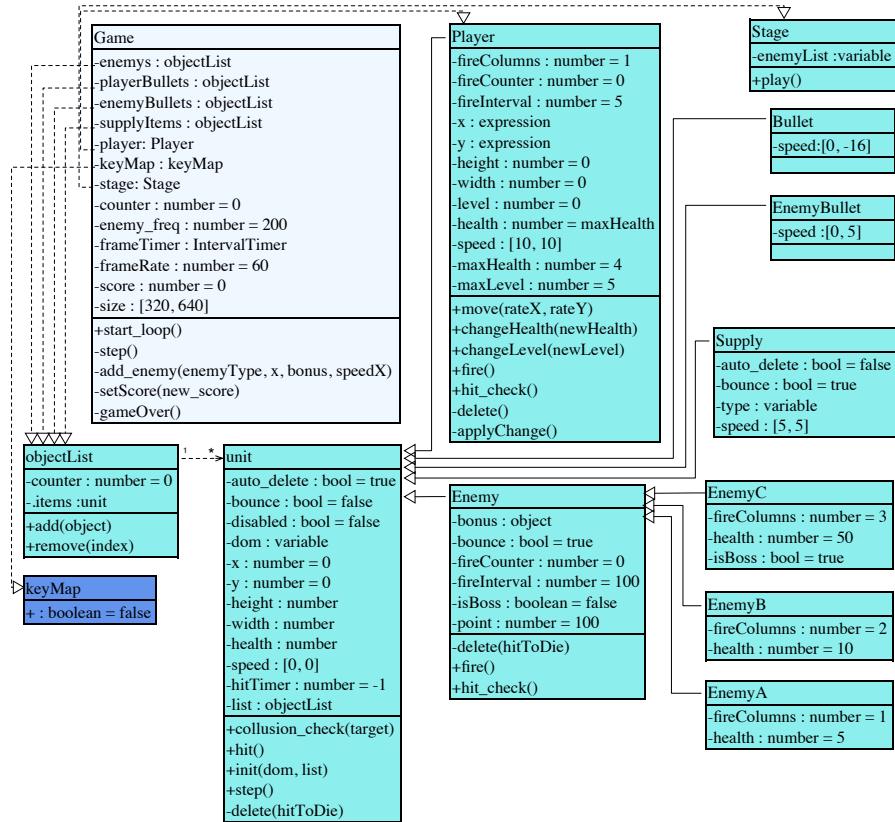
もしボスを倒す前にヘルスポイントがゼロに減ると、ゲームオーバーになり、Game Over の画面が表示する。

ゲームが終わったら、もう一度ゲームをプレイする場合、F5 を押し、ブラウザがリフレッシュすることにより、ゲームがリスタートする。

4 内部仕様

4.1 クラス図

このクラス図はチームメンバーと共同に作ったものである。



4.2 各クラスの説明

- Game

Game クラスはゲーム全般の管理の役割を担当する。具体的に、コントロールの入力、ゲームの表現、オブジェクトの管理、ステージの管理、スコアの計算を行っている。

- unit

unit クラスは、ゲームにある全てのオブジェクトの抽象クラスである。unit クラスはオブジェクトとしての基本機能を提供する。具体的には、オブジェクトの速度、位置、ヘルスポイント、状態、DOM オブジェクト、フレームごとに行動などを実装した。

- Player

Player クラスは、unit クラスを拡張し、プレーヤーの機能を実装したものである。武器レベルはこのクラスに実装した。

- Enemy

Enemy クラスは、全ての敵の抽象クラスである。Enemy クラスは unit クラスを拡張し、敵の機能を実装したものである。

- EnemyA, EnemyB, EnemyC

EnemyA、EnemyB、EnemyC クラスは 3 種類の敵のクラスである。三つとも Enemy クラスを拡張したものであり、それぞれ異なるヘルスポイント、標準移動速度などを設置した。

- Bullet, EnemyBullet

Bullet, EnemyBullet クラスは unit クラスを拡張し、弾の機能を実装したものである。

- ObjectList

ObjectList は特殊な配列として実装したクラスである。配列に新しいエレメントを入れる時、そのエレメントに自動的に ID を割り合い、そのエレメントを簡単に削除できるようにした。

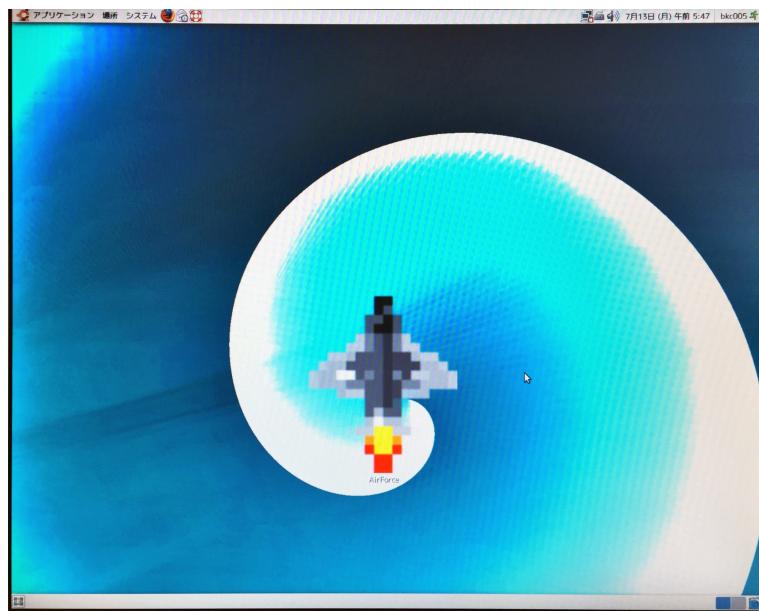
- Stage

Stage はゲームのステージのマネージャークラスである。ステージの敵を管理する。

5 実行例

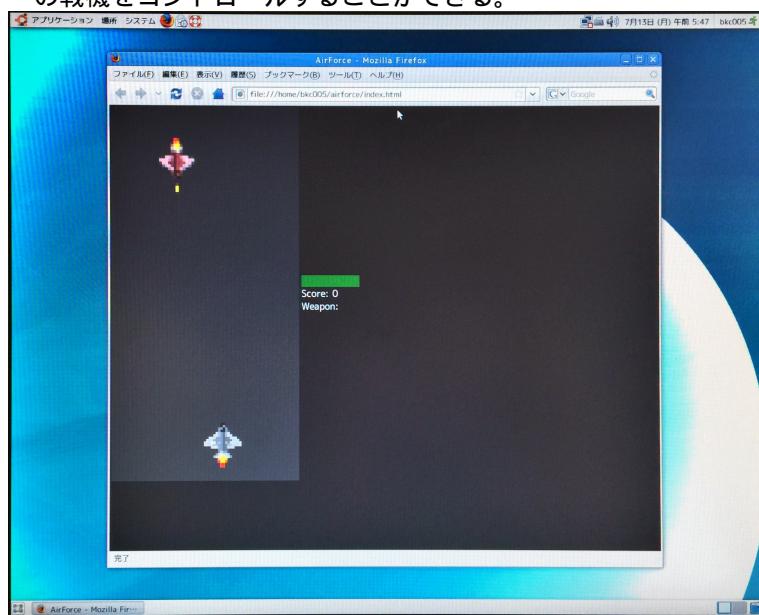
5.1 ディスクトップ

ディスクトップにゲームのアイコンをクリックすると、ブラウザーが立ち上げ、ゲームが始まる。



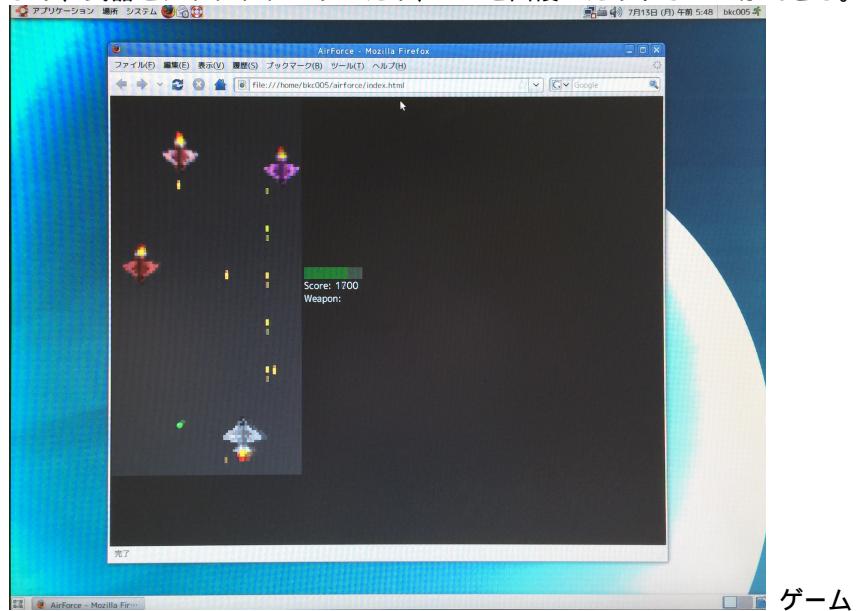
5.2 プログラム起動した直後

プログラムが起動したら、ゲームが自動的に始まる。キーボードでプレイヤーの戦機をコントロールすることができる。



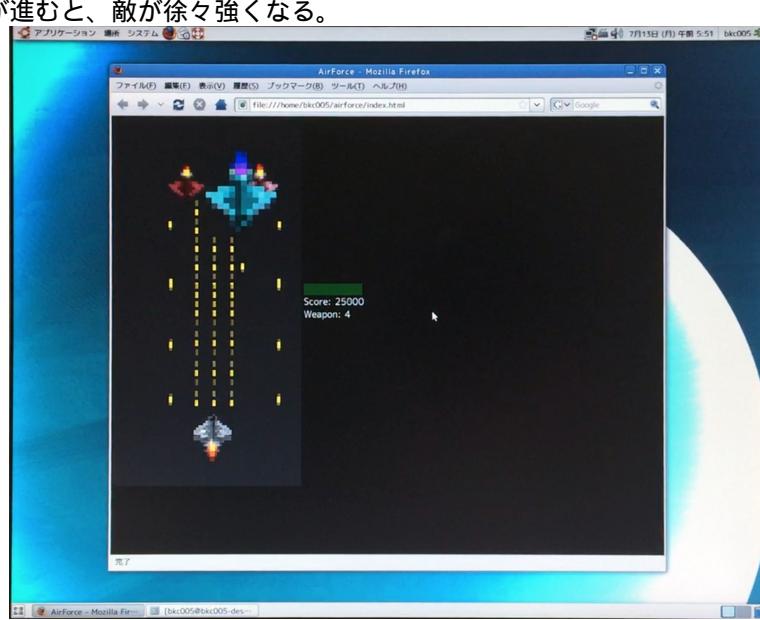
5.3 ゲーム画面

敵を倒せば、緑の円のラプライが出る場合がある。このサプライをとることにより、武器をアップグレードしたり、HPを回復したりすることができる。



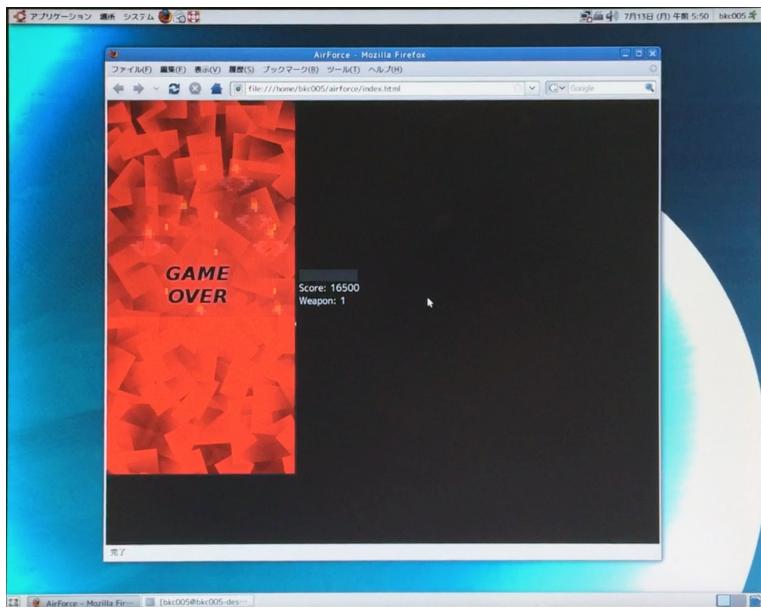
ゲーム

が進むと、敵が徐々強くなる。

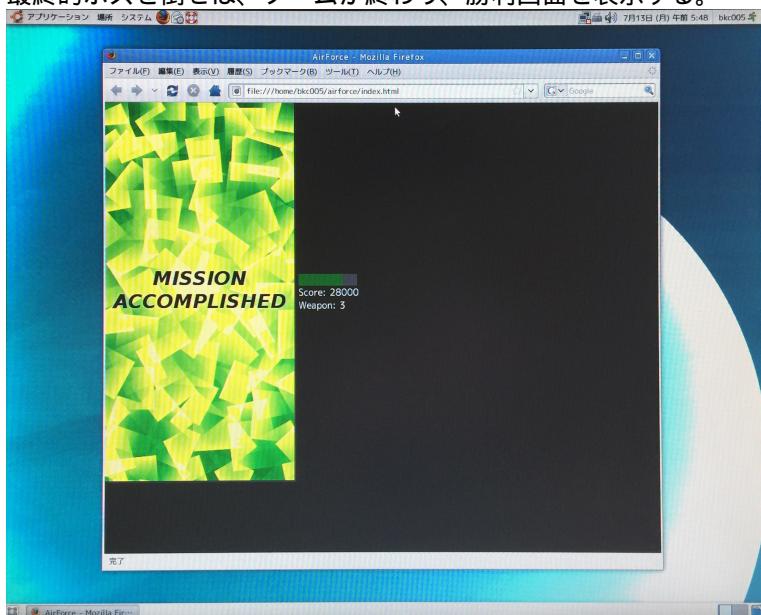


5.4 ゲームオーバー

プレーヤーが倒れば、ゲームオーバーとなり、ゲームオーバーの画面を表示する。



最終的ボスを倒せば、ゲームが終わり、勝利画面を表示する。



6 工夫した点と考察

6.1 工夫した点

6.1.1 プログラム上

- JavaScript のプロトタイプチェンによるオブジェクト指向プログラミングで実装することにより、プログラムの各モジュールを独立させた。
- 共有するメソッドをプロトタイププロパティーにアサインすることにより、良い性能を実現できた。
- オブジェクトの共通部分を継承ツリーとして作成したため、将来新しい敵、武器などを拡張しようとすると簡単にできる。

6.1.2 機能上

- GIMP を使いオリジナルな画像素材を幾つか作成した。
- プレーヤーの武器の 5 段階のレベルを用意した。
- 敵を 3 種類を作り、それぞれ異なるヘルス、武器、速度などを設定した。

6.1.3 チームワーク

- GitHub を利用し、うまくチーム単位でのプログラミングができた。

6.2 感想

今回のプロジェクトでは、かなり古いバージョンの開発環境を利用し、開発を行った。特にブラウザが古いため、デバッグ用のツールが一切搭載していなかった。また、最近の新しい技術（HTML5, CSS3）なども対応していなかった。それゆえ、自分のデバイスで実行できても、実験環境で動作できるかどうか分からなかった、もしくは、プログラムを入れたらちゃんと動作しない場合、コンソルなどの開発者ツールがないため、どこに間違えたかわかりにくかった。だが、最終的に設計の通りに実験環境でもうまく動作したのは良かったと思われる。将来古いシステムでの開発はやはり自分のマシンに同じ環境を仮想マシンとして作っておいた方がいいと思われる。

参考文献

- [1] The jQuery Foundation , jQuery API Documentation, <http://api.jquery.com> , 2015 年 7 月.