

1 このゲームの内容と流れ

本ゲームは独自の戦車を改造し、自分だけの軍隊で敵を駆逐して遊ぶゲームである。まず、戦車の選択画面から説明する。プレイヤーはまず、設定されたコスト以内で自分のガレージから戦車を選ぶ。コストの総合計は現在は100と設定してある。このコスト以内でなければ戦車を選択することが出来ない。例えば、コスト60の戦車を2台選択することは出来ないが、コスト50の戦車を2台選択することは出来る。コスト50の戦車とコスト20の戦車が2台選択された場合コストが10だけ余る。この余った分は自軍のライフに加算される。自軍のライフはでフォルトでは100に設定してあるが、例えば、コスト10だけ余った場合、自軍のライフは110となる。

次に戦闘画面であるが、戦闘は将棋のようなタクティカルシミュレーションを想定している。コストの低い戦車は足回り、射程距離ともに劣るが、コストの高い戦車は足回りも射程距離も強い。自分のターンでは2回の行動が許される。戦車2台を移動させてもいいが、2台を攻撃にまわしても良い。または、移動1回に攻撃1回といった組み合わせも可能である。戦車が砲撃により破壊された場合、撃破された戦車の2倍のコストがライフから失われる。つまり、コスト50の戦車が撃破された場合、ライフが100減ることになる。戦車が全滅するか、ライフがなくなったらゲームセットである。

自軍の戦車は黄色、敵の戦車は赤で表示してある。青は池を表しており移動できない。緑は草原なので移動できる。操作の仕方は戦車を一回クリックすると移動フェーズに移行し、二回クリックすると攻撃フェーズに移行する。なお、この操作は取り消せないようになっている。移動可能範囲、攻撃可能範囲はマスで灰色で表示する。撃破された戦車は残骸になり、そのマスに残り続ける。つまり、青で表示されそのマスはもうゲーム中に使えない。

現在実装されているのはこの部分までである。

なお、絵素材はOP画面しかないので戦闘画面などの迫力は残念ながらない。また、敵の戦闘に関するアルゴリズムであるが、乱数を用いて動かしているので非常にいまいちである。以上で述べた内容は実際ユーザーにプレイしてもらい、そのフィードバックでゲームバランスを調整する予定である。



図 1: OP 画面の様子

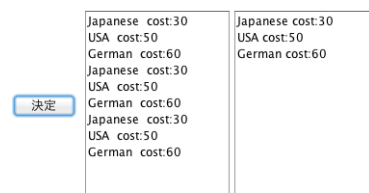


図 2: 戦車選択画面の様子

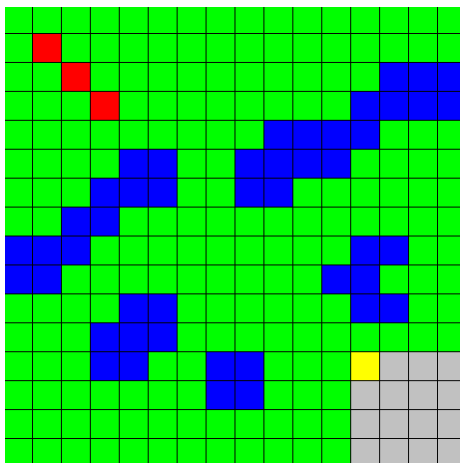


図 3: 戦車配置画面の様子

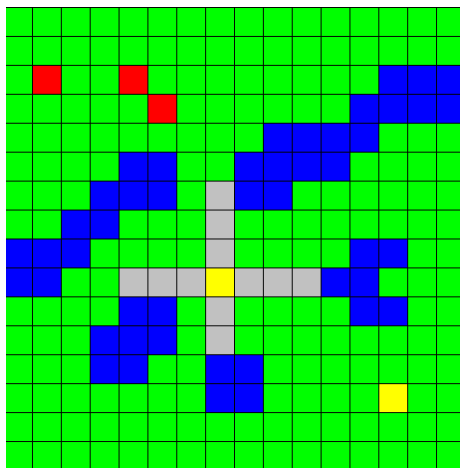


図 4: 戦車の移動可能範囲

2 ディレクトリの構成

ディレクトリは CombatPanzer の下に配置してある。この下にパッケージが複数あり、そのパッケージより深いディレクトリは無い。bin には.class ファイルが入れてある src にはソースコードが入っている。src の下には、Game/,MainApp/,Object/,AbstractPanzers/,Algorithm/, ConcreatPanzers/のディレクトリが入っており、同時にオープニングのイメージ画像も入っている。MainApp/にはこのアプリケーションの枠、すなわちフレームを記述する MainFrame.javaが入っている。Game/にはフレーム上で入れ替え出来るパネルが多数入っている。基本的にこれらのパネルを差し替えるようなイベントをボタンに持たせている。Object/にはゲームで使われる実態物、例えば、戦車、マップ、プレイヤー、敵などがクラスに分けて入っている。AbstractPanzers/,ConcreatPanzers/には戦車をオブジェクトを簡単に組み立てるためのクラスが入っている。Algorithm/クラスでは敵の AI に関するクラスが入っているが敵の具体的な AI は結局出来てないが、ランダムで動く AI は入っているので一応正常に動作する。

3 アプリ作成に際し心がけたこと

アプリ作成で心がけていたことは、大きく3つある。まず一つ目は、フルスクラッチで作ること。外部のライブラリ (Unity など) を使わず、全て java と Swing だけで作る事を心がけた。必要なゲームフレームワークは独自で設計した。二つ目はなるべく再利用可能なクラスを設計すること。そのため、デザインパターンを使っている部分が多くなっている。戦車はそのうち種類が増えるとコーディングが面倒なので、AbstractFactory で大量生産できるようにしてある。敵の AI も現段階でランダムな動きしかしていないが、将来的には Algorithm のディレクトリの下のクラスを入れ替えるだけでいいようにしてある。(Strategy) 一つしかないとわかっているものは Singleton で設計してある。これらのクラス構成が正解かどうかは今見てみるとよく分からないがコードを見返すのにそれほど苦労しなかったのも、そこまで変な分け方じゃないと思われる。三つ目はコードを書くときはキャメル記法とインデントを徹底しているつもりである。インデントの深さは tab 一つ分である。

4 他人との共同作業

絵だけ発注して、コードは全て自作。

5 実行ファイル名

CombatPanzer!.jar をダブルクリックすれば実行できる。

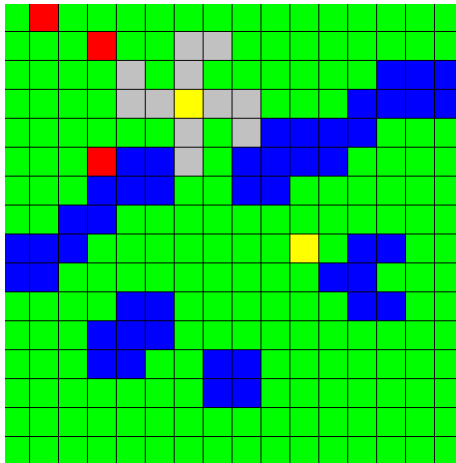


図 5: 戦車の攻撃範囲の選択