

オブジェクト指向を用いたゲームプログラミングの検討

小林 和輝[†] 田那部 洋平[‡] 小泉 寿男[†]

東京電機大学理工学部情報システム工学科[†]

東京電機大学理工学研究科情報システム工学専攻[‡]

1. はじめに

現在、ゲーム機の性能向上などに伴いゲームに求められる品質も向上している。しかしそれに反して開発期間は短期化が求められており、開発期間短縮のためにプログラムの再利用が注目されている。

しかしゲームと一括りにしても、そこにはプレイヤーが敵を撃ち落とすスコアを競うシューティングゲーム、立ちはだかる敵や障害物乗り越えゴールを目指すアクションゲーム、実際のスポーツをゲームにして CPU や人と対戦するスポーツゲームなど様々な種類のゲームが存在する。さらに同じジャンルのゲームでも、例えばスポーツゲームでは野球ゲームとサッカーゲームのようにシステムが異なるゲームも多く存在する。

ジャンルの異なるゲームでは当然のことながら、同じジャンルのゲームでもシステムの確立したシリーズ化されているゲームでない限り、開発を行なうゲームのプログラムの大部分を再利用で補うことはほぼ不可能である。本研究ではゲームプログラムの開発の現状から考えられる問題を解決するために、オブジェクト指向をゲームプログラムの開発に適用して、ある 1 つのジャンルにおけるプログラムの再利用に特化したプログラムの作成を行なう。

2. オブジェクト指向のフレームワーク

近年、ゲームのプログラムに C++ が使われるようになるなど、ゲームの開発の主にプログラミング部分にオブジェクト指向が適用されるようになった。オブジェクト指向は主に『オブジェクト指向分析』、『オブジェクト指向設計』、『オブジェクト指向プログラミング』の 3 つに区別される。オブジェクト指向をゲーム開発に適用した流れを図 1 に示す。

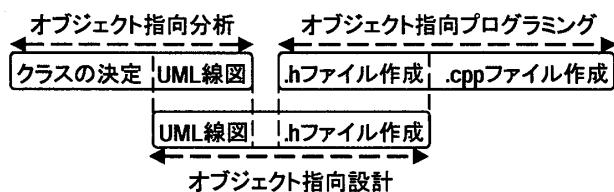


図 1 オブジェクト指向のゲーム開発への適用例

このオブジェクト指向設計をゲームプログラムの開発に適用した例として、Pop フレームワークというものが存在する。これは Microsoft Visual Studio で生成した「ドキュメントビュー」フレームワークを基に構築されており、この基層フレームワークを MFC フレームワークと呼ぶ。図 2 にフレームワークの構築を示す。

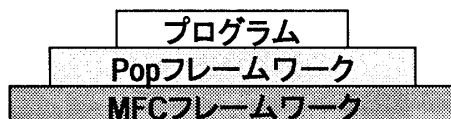


図 2 フレームワークの構築

この Pop フレームワークを用いることで、コンピュータゲームやその他のシミュレーションに適したプログラムを簡単に構築することができ、2D、3D のどちらのグラフィックスでもゲームを作成することができる。また、Pop フレームワークにはゲームのサンプルプログラムがあり、そのジャンルもシューティングゲーム、アクションゲーム、スポーツゲーム、パズルゲームと様々なものが存在する。

3. オブジェクト指向を用いたゲームプログラミング方式

Pop フレームワークの中心となる主要クラスとして、cCritic というクラスが存在する。cCritic クラスを継承して作成できるものとしてプレイヤーキャラクター、敵キャラクター、弾、設備、ゲーム世界を見るためのカメラなどが挙げられる。つまり、cCritic クラスを継承することでゲームに必要なほとんどのものを作成することができるといえる。しかし、cCritic クラスの問題点として、継承を行なうことで様々なジャンルのゲームを作成できる為に 1 つのジャンルのゲームプログラムだけを見ると、使用しない処理や継承先のサブクラスでよく定義される処理が存在することが考えられる。

そこで、本研究ではこの cCritic クラスを用いて、シューティングゲームならシューティング用の、アクションゲームならアクション用のというようなジャンルごとのプログラム作成に特化したスーパークラスの作成を行なう。

まず、どのジャンルのプログラム作成に特化したスーパークラスを作成するかを決定した後に、そのジャンルのゲームプログラムで使用していない処理を cCritic クラスから取り除く。次にいくつかのゲームプログラムにある cCritic クラスをスーパークラスとして作成されたサブクラスを調べ、共通した処理がある場合はその処理をスーパークラスで定義する。さらに、筆者らが必要と考える処理をスーパークラスに新しく追加する。これを行なうことでジャンルごとのプログラム作成に特化したスーパークラスを作成することができる。

本研究ではこの手法を用いて最初にシューティングゲームの処理に特化したスーパークラスを作成した。シューティングゲームは Pop フレームワークに存在するサンプルプログラムの種類が最も多いゲームのジャンルであり、使用しない処理や追加すべき処理を決定しやすいこ

Examination of game programming that uses object-oriented
[†]Kazuki Kobayashi, Hisao Koizumi; Department of Science and Engineering, Tokyo Denki University
[‡]Yohei Tanabe; Department of Computers and Systems Engineering, Graduate school of Science and Engineering, Tokyo Denki University

と、そしてゲームプログラムが最も作成しやすいジャンルであるため、シューティングゲームを選択した。

(1) オブジェクト指向分析

まずオブジェクト指向分析の段階において、使用するクラスやメソッド等の決定をする。そのために以下の要件からクラスやメソッド等の候補となる名詞や行為を見つける作業を行なった。

・要件

動くクリッターを持つシューティングゲーム用のスーパークラス。クリッターは多角形、ビットマップ、ビットマップのアニメーションループとして描画される。クリッターは周囲の世界を基に自分自身を更新する。クリッターはマウスやキーボードの制御を監視する。クリッターは弾を撃てる。

・名詞

ゲーム、クリッター、多角形、ビットマップ、アニメーションループ、世界、マウス・キーボード制御、弾

・行為

動き、描画、更新、監視、射撃

(2) オブジェクト指向設計

次にオブジェクト指向設計の段階において UML 線図を作成し、使用しない処理や追加すべき処理を決定した。例を用いて図 3 にその流れを示す。

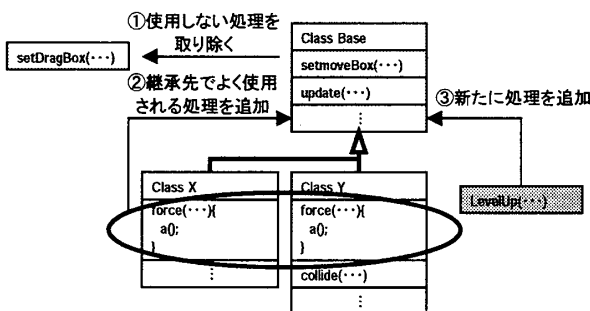


図 3 UML 線図と本手法の流れ

(3) オブジェクト指向プログラミング

最後にオブジェクト指向プログラミングの段階においてヘッダーファイルやソースファイルの作成を行なった。以下に cCritic クラスから取り除いた処理について示す。

・ setDragBox 関数

ゲーム中のオブジェクトをドラッグしたい場合に使用する。この関数でゲームウィンドウ内の範囲を指定し、その範囲内でオブジェクトをクリック & ドラッグすることができる。

しかし、シューティングゲームにおいてオブジェクトをドラッグする必要はほとんどないと考えたため、この関数を取り除いた。

以下にサンプルプログラムから cCritic クラスに追加した処理について示す。

・ damage 関数

プレイヤーキャラクターや敵キャラクターがダメージを受けた際の処理が記述されている。

この関数は cCritic クラスにある damage 関数をオーバーライドして使用されていたが、シューティ

ングゲームのサンプルプログラムで記述されている処理がほとんど同じだったため、それらをまとめて cCritic クラスで定義することにした。

これに加えて筆者らが必要と考える処理を cCritic クラスに追加する。

このように cCritic クラスの使用しない処理を取り除き、新たな処理を追加することでシューティングゲームのプログラム作成に特化したスーパークラスの作成を行なった。

4. 評価

本研究での評価は作成したスーパークラスを Pop フレームワークに適用して行なった。まずは作成したスーパークラスを継承してゲームプログラムを独自に作成し、それが動作するかについて検証を行なった。

独自に作成したゲームプログラムが動作することを確認し次第、Pop フレームワークに元からあるサンプルのゲームプログラムにスーパークラス作成の際に生じた変更点を修正するだけで動作するかについて検証を行なった。

作成したスーパークラスを用いてゲームプログラムを実行した結果を図 4 に示す。

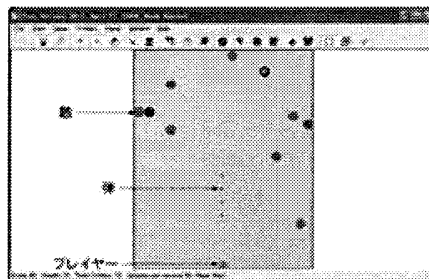


図 4 自作プログラム実行結果

このように動作することが確認できた。そして、Pop フレームワークに元からあるサンプルのゲームプログラムも同様に動作することが確認できた。また、作成したスーパークラスを Pop フレームワークに適用し、サンプルのゲームプログラムの変更点を修正したことで、プログラムの記述量を削減できることを確認した。

ある 1 つのジャンルにおいて、cCritic クラスから使用しない処理を取り除き、サブクラスから処理を追加することで、そのジャンルのプログラム作成に特化したスーパークラスを作成した。

5. まとめ

ゲームプログラムへの効果的なオブジェクト指向の適用方法を提案した。提案した方法を実際に適用し、それに対して検討を行なった。

今後の課題として、さらに多くのゲームジャンルでのプログラム作成に特化したスーパークラスを作成する。本研究の提案方法を用いることでシューティング以外のゲームジャンルでも同様にスーパークラスを作成することができる。

6. 参考文献

- [1] Rudy Rucker, "ソフトウェア工学とコンピュータゲーム", 株式会社ボーナデジタル, 2004 年
- [2] 川野 洋, "シューティングゲームの敵機攻撃弾発射アルゴリズムに関する考察", 情報処理学会 研究報告, No.70, pp61-68(2006)