

# 初心者向けオブジェクト指向プログラミング学習ゲームの開発

筒井 優<sup>†</sup>

岩澤 京子<sup>‡</sup>

拓殖大学工学部情報工学科<sup>†</sup>

拓殖大学工学部情報工学科<sup>‡</sup>

## 1. 背景と目的

現在のプログラミング技術の中で、オブジェクト指向は様々な言語に使われており、大変重要な考え方である。しかし、私の身の回りでオブジェクト指向を理解している人は少なく、多くの人にとってプログラムを学習する際の壁となっているように感じた。しかし、自分がそうであったように、一度理解さえしてしまえば、プログラムについての世界が広がり、プログラミング技術のさらなる成長につながると考えた。

本研究の目的は、学生などのプログラミング初心者が、オブジェクト指向の基本的な要素についてイメージを掴み、理解できるよう支援するシステムを実現することである。

## 2. 研究の方針

本システムは、次のような特徴を持っている。

### (1) ゲームの要素を取り入れる

難しそうであるという先入観を取り払い、気軽に始めることができる。また、操作した結果を簡単に確認でき、間違えても容易にやり直すことができる。このような理由によって、ゲームという手段を選択した。

### (2) ロボットをモチーフとする

ロボットは、オブジェクトという抽象的なイメージよりも具体的で、想像しやすくなるであろうと考えた。そして、設計図や組み立て、機能の搭載などの要素が、オブジェクトの喩えとして適切であると考えた。

### (3) 拡張を容易にする

問題を追加する際、追加、変更が容易となるよう、外部のテキストファイルで問題を記述するようにした。

## 3. ゲームの仕様とオブジェクト指向

### 3.1 ゲームの仕様

このゲームは、ユーザが作成した設計図を元に、キャラクターの動作が変わる、オリジナル

Development of the game to learn object oriented programming for beginners.

<sup>†</sup>Tsutsui, Masaru, Dept. of Computer Science, Takushoku University.

<sup>‡</sup>Iwasawa, Kyoko, Dept. of Computer Science, Takushoku University.

のパズルゲームである。ゲームの目的は、コースをクリアすることである。1つのコースは2〜3つのレベルからなり、それらをすべて順番にクリアすることでコースクリアとなる。レベルをクリアするために、ユーザは次の操作を行う。

- (1) 適切な機能を持ったロボットの設計図を、図1のように作成する。
- (2) 設計図を元に、ロボットを研究所で生成する。
- (3) 図2のように、ロボットに命令を送り、ゴールまで導く。



図1. ロボット設計図作成画面



図2. ロボット操作画面

### 3.2 オブジェクト指向との対応

このゲームを遊ぶ事で、ユーザがオブジェクト指向について、下の表1のように、具体的に体験するよう意図している。

表1. ゲーム上の要素とオブジェクト指向の対応

ゲーム上の要素	オブジェクト指向	体験
設計図とロボットの関係	クラスとインスタンスの関係	設計図からロボットをコンストラクトする
個々のロボットが持つ属性値と機能	カプセル化	属性値と機能を設定しクラスを設計する
個々のロボットへの命令	メッセージパッシング	ロボットに命令を出しメソッドを呼び出す
設計図の再利用	クラスの継承	既存の設計図を継承し、差分プログラミングを行う

設計図とロボットの関係は、クラスとインス

タンスの関係が対応する。設計図を元にロボットを生成する体験を通して、クラスからインスタンスを生成する事を体験するように意図している。

ロボットが属性値と機能をまとめて持っている事は、カプセル化に対応する。ロボットというメンタルモデルを利用して、インスタンスの振る舞いを理解できるように意図している。

ロボットに命令を送るという動作は、メッセージパッシングに対応する。指定したロボットに、命令を実行させるということを通して、インスタンスに対してメッセージを送るというイメージを理解できるように意図した。

設計図を継承する事は、クラスの継承と対応する。継承した設計図を利用すると有利なることにより、クラス継承の意味を理解できるように意図している。

これらを体験学習として繰り返し行うことによって、最終的にオブジェクト指向が理解できるようになると考えた。

#### 4. ゲームの実現

本システムの主な特徴として、マップ管理と、設計図継承がある。

##### 4.1 マップ管理方法

ロボット操作処理で、次のようにマップを管理する。

マップ上のマスに対応した2次元配列を、3層のレイヤーとして用意した。1つのレイヤーには1つの要素しか、同じマスに重ねて配置することができない。それぞれのレイヤーは、上からオブジェクト、補助オブジェクト、地形を管理する。

##### (1) オブジェクトレイヤー

ロボットや研究所など、マップ上を動くオブジェクトを管理する。

##### (2) 補助オブジェクトレイヤー

ドアや橋など、オブジェクトレイヤーのオブジェクトと同じマスに配置可能で、マップ上を動くオブジェクトを管理する。

##### (3) 地形レイヤー

マップ上を動かない、道や海等の地形を管理する。

##### 4.2 設計図継承

問題を追加する際、ロボットの機能に依存関係を持たせることができる。設計図を継承する事により、前提とする機能を保持させて高度な機能を搭載させることが可能となる。この機能のため、ユーザは設計図の継承を行う。

#### 5. システム構成

図3にシステムの構成を示す。

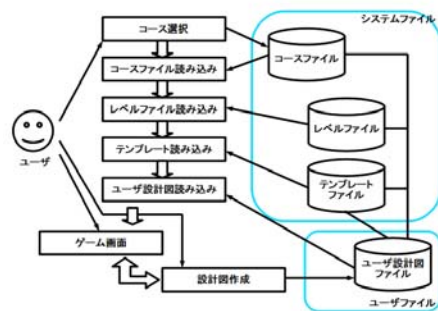


図3. システム構成

コースごとに、コースファイル、レベルファイル、テンプレートファイル、ユーザごとに、ユーザ設計図ファイルを持つ。各種ファイルの役割を次に示す。

##### (1) コースファイル

レベルファイル、テンプレートファイル、ユーザ設計図ファイルに関連付ける役割を持つ。

##### (2) レベルファイル

ゲームの要素の初期配置、ターン制限、コスト制限のデータを持つ。

##### (3) テンプレートファイル

設計図を作成する際の初期値、搭載できる機能リストのデータを持つ。

##### (4) ユーザ設計図ファイル

ユーザがゲーム中に作成した設計図のデータを持つ。作成した設計図分だけ存在する。

ユーザがコースを選択すると、関連するファイルを読み込み、ロボット操作画面、ロボット設計図作成画面を表示する。この2つの画面は、ユーザが自由に切り替えることができる。

設計図作成処理では、ユーザがロボットの設計図を編集し、利用できるようにする。また、ユーザ設計図ファイルとして保存し、継承して再利用できるようにする。

#### 6. おわりに

本研究では、オブジェクト指向を取り入れた、パズルゲームの開発を行った。オブジェクト指向の抽象的な要素を具現化し、ゲームとして動作させることができた。これによって、初心者も、オブジェクト指向を体験できるシステムを実現することができた。

学生に試遊してもらったところ、第一にオブジェクト指向との関連に気づいてもらうための工夫が足りないという指摘があった。また、面白い問題が不足している事、そして、初心者からの評価を得る機会が不足していた事がある。

#### 参考文献

- [1] 大村忠史 “Java GUI プログラミング Vol. II” カットシステム 2002