

手続き型プログラミング教育におけるオブジェクト指向の思考 The Thinking of Object-Oriented Programming in Procedural Programming Education

野口 孝文^{*1}

Takafumi Noguchi

^{*1} 釧路高専

National Institute of Technology, Kushiro College

Although object-oriented programming is often used to develop system, it is difficult for beginners to learn this. In our programming class, we have been teaching procedural program through making a game. However, our learning environment reduce the burden of programming, by using combination with a function object and a procedure object. We believe that learners can learn the concept of object-oriented such as management of objects and message between objects, by using our programming method.

1. はじめに

オブジェクト指向によるプログラミング開発が一般的に行われているが、プログラミングの初心者がこれを学ぶのは難しい。初心者のプログラミング教育においては、コンピュータ利用に関する知識やプログラミング環境利用の知識、プログラミング言語に関する知識、論理的思考力等の多様な知識が学習者に必要とされるばかりでなく、これらの知識に関して多様なレベルの学習者にも対応しなければならない。

これまで我々は、学習者の動機づけのためにゲーム作りを課題としてプログラミング授業を行ってきたが、手続き的プログラムの手法を学ぶ内容である。しかし、我々の学習環境は、絵を表示したりこれを移動したりする機能をオブジェクト部品で与えることによって、プログラミングにおける学習者の負担を軽減している。さらにそこでは、オブジェクト部品と、C 言語を用いたプログラムを記述した手続き部品を組み合わせた複合オブジェクトを 1 つの部品として扱うことによってプログラムを作成している。

この手法は、多数の複合オブジェクトを容易に組み合わせることができ、プログラム制作の過程でオブジェクトの管理やメッセージといったオブジェクト指向の考えを体感的に学ぶことができると考えている。これまで行ってきた授業には、数百個のオブジェクトを組み合わせる作品を制作する学生もいた[野口 2000, 2015]。

本論文では、釧路高専電気工学科 2 年生のプログラミングの授業で用いているプログラミング環境および授業の内容を紹介し、その特徴について述べる。本論文の 2 章ではプログラミング環境について述べ、3 章では学生の制作した作品と特徴について述べる。4 章でまとめを述べる。

2. プログラミング環境

2.1 IntelligentPad システム

IntelligentPad は、パッドと呼ばれるオブジェクトをダイナミックに組み合わせたり、変更したりできるシステムである。パッドは、ディスプレイ上に可視化され、マウスによる直接操作でパッドを自由に組み合わせることができる。パッド同士の結合は、標準化されたスロットの結合によって行う。

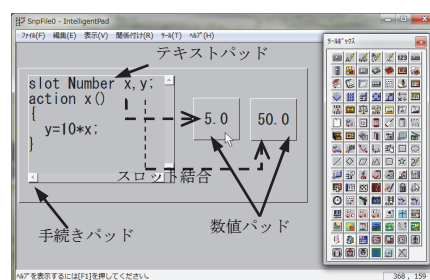


図1 手続きパッドを用いたプログラム

本システムでは様々な機能を持つパッドが用意されている。そしてパッドの一つに、C 言語のサブセットからなるスクリプトを記述することができる「手続きパッド」がある。授業では、この手続きパッドを用いている。手続きパッドと既存のパッドを組み合わせることで、多様なプログラムを作ることができる。図 1 は、入力した値を 10 倍して表示するプログラムの例である。キーボードからの数値の入力でプログラムは実行される。

2.2 イベントドリブンによるプログラム記述

図1に示したプログラムはキーボードからの入力によるイベントによって起動する。授業で作成するプログラムもイベントドリブンにより、タイマなどのイベントと組み合わせることで、繰り返し文をほとんど記述すること無しに多様なプログラムを実現している。

学生に提示しているもぐらたたきゲームとそのプログラムの一部を図 2 および図 3 に示す。また、シューティングゲームを図 4 に示す。図2と図4の下に、パッドの貼り合わせを示す。

図2のゲームは、図3に示したプログラムを記述した手続きパッドの上に、2 枚の絵を選択的に表示できるスイッチパッドを複

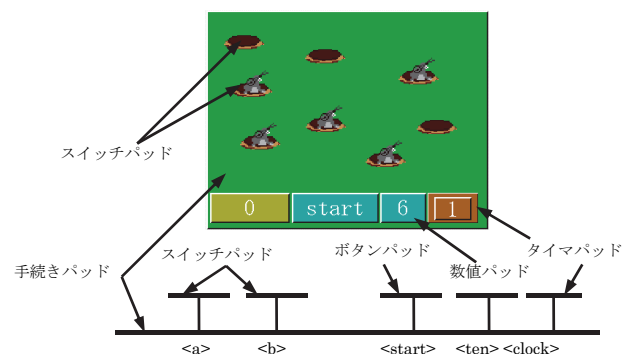


図2 もぐらたたきゲームとパッドの貼り合わせ

```

slot Number r,nokori,ten;
slot Command clock,start;
slot Boolean a;
Number random();
r=0.123456789;
Number random()
{
  r=123456789*r%1;
  return r*6+1-r*6%1;
}
action clock()
{
  if(nokori==0) return;
  nokori=nokori-1;
  if(random()==1) a=TRUE;
  else a=FALSE;
}

action start()
{
  a=FALSE; /*穴の絵*/
  nokori=10;
  ten=0;
}
action a()
{
  if(a==FALSE) ten=ten+1;
  else ten=ten-1;
  a=FALSE;
}

```

図3 もぐらたたきゲームのプログラム(一部)

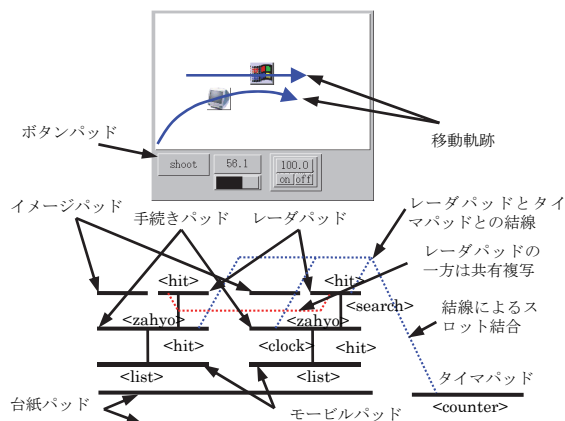


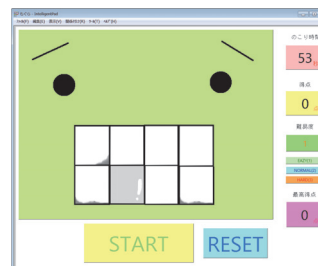
図4 シューティングゲームとパッドの貼り合わせ

数枚貼することで、もぐらたたきを実現している。もぐらの表示は、手続きパッドに貼付したタイマパッドによって起動されたアクション関数「clock」が乱数値に応じて変更する。得点は、絵を表示しているスイッチパッドをマウスクリックすることによって起動されるアクション関数「a()」が計数する。図3のプログラムは、もぐら1匹分のプログラムを提示しているが、スロット「a」とアクション関数「action a()」の記述方法に倣ってスロットと関数を増やすことで、何匹でもぐらを増やすことができる。

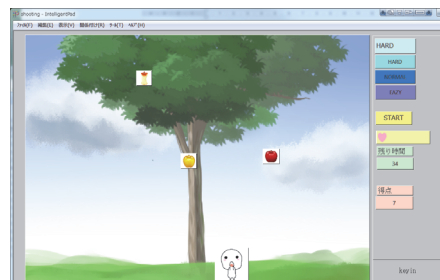
図4のシューティングゲームは、「shoot」ボタンを押すと左下から「弾」が放物状に打ち出される。一方、「移動物体」が左端から現れる。パッド同士が当たると、下から打ち出された弾は、左下隅に消え、当てられた移動物体は、下方に落下する。絵の表示や重なり判定、移動といった機能を部品で与えることで、学生は、軌道の計算や衝突時の動作プログラムを記述するだけでシューティングゲームを実現できる。

もぐらたたきゲームとシューティングゲームのプログラムの作成の仕方は異なっている。何れの場合にもタイマのイベントによりもぐらや移動物体が動き出すが、もぐらたたきが1つの手続きパッドの上に絵を表示するパッド(イメージパッド)を複数枚貼付しているのに対し、シューティングゲームでは、移動するパッド(モバイルパッド)の上に手続きパッドを貼付している。このようにすることでシューティングゲームでは、移動物体を1つの部品として扱えるようにしている。そのためシューティングゲームの移動物体は、複写するだけで簡単に増やすことができる。

このようにすることで、移動物体同士は独立して動きをプログラムすることができる。また、独立しているため、重なり(衝突)を判定するための部品として、レーダパッドを用意している。レーダパッド同士に重なりがあったときには、その情報がこれを貼った手続きパッドに送られる。手続きパッドは、重なり(衝突)に応じた動作を記述するだけで良い。



(a) もぐらたたきゲーム



(b) シューティングゲーム

図5 学生が制作したゲーム(2014)

2.3 もぐらたたきゲームとシューティングゲーム

プログラミングの授業では、初めに基本的な文法と手続き的プログラミングの例題を学習する。その後応用課題として、もぐらたたきゲームや電卓のプログラム、シューティングゲームのプログラムを順に制作する。

図5に学生が制作した作品の例を示す。図5の(a)は、もぐらたたきゲームである。図の中央部の歯がいろいろな虫歯の絵に変わるのをマウスでクリックして得点するという作品である。(b)は、シューティングゲームである。木から落ちてくるいろいろなアイテムをキーボードの操作で下方のキャラクタを移動させて受け取るという作品である。

シューティングゲームでは学生は、基本動作をする複合パッドを作成しこれを複写してプログラムを変更したり、初期値を変更したりして、さらにたくさんの動くパッドを作る。本プログラミング環境では、絵の上手な学生は絵を工夫したり、プログラムの得意な学生はプログラムを工夫したりと、それぞれの得意な点を発揮しプログラムへの興味を深めることができる。

前節でも述べたとおりシューティングゲームでは、それぞれのアイテムの動作を独立して作成することができるため、作品(a)で使われたパッドの数よりも、(b)で使われたパッドの数の方が多くなっている。これらの特徴については、3章で述べる。

3. 実習課題に見られる特徴

3.1 もぐらたたきに見られる使用オブジェクト

もぐらたたきに見られる特徴について述べる。その後3.2節でこの特徴とシューティングゲームの特徴を比較する。

学生は熟達度に応じて、パッドの数、スクリプト(手続きパッドに記述したC言語を用いたプログラム)の量や画像ファイルの数等で工夫をすることができている。2011年度から2014年度の過去4年間について、2年生が制作したもぐらたたきで使用したパッドの数を表1に示す。表中の数字は、使用したパッドの数と学生の人数の関係を示している。また、合計は学生の総数を表し、最大パッドはその年度の中で一番多くパッドを用いた作品のパッド総数を表している。

表1 もぐらたたきゲームで使したパッド数

パッドの数	2011年度	2012年度	2013年度	2014年度
40未満	26	22	37	24
40～50	13	14	4	11
50～60	5	5	0	7
60～70	3	2	0	3
70～80	0	1	0	3
80～90	1	2	0	0
90以上	0	0	0	0
合計	48	46	41	48
最大パッド	86	89	45	79

表2 シューティングゲームで使したパッド数

パッドの数	2011年度	2012年度	2013年度	2014年度
40未満	3	3	5	5
40～50	10	15	14	16
50～60	8	8	14	11
60～70	12	9	4	6
70～80	5	7	1	2
80～90	3	1	1	1
90以上	3	3	2	5
合計	44	46	41	46
最大パッド	146(15)	136(18)	135(18)	136(36)

年度によって、人数の分布に若干の差はあるが、使用しているパッドの数が50枚以下に分布していることが分かる。なお、手続きパッドの使用枚数は全員1枚である。中には、90枚近くのパッドを使用する学生もいるが、希である。

3.2 手続き型プログラムと一体化した複合オブジェクト

シューティングゲームに見られる特徴について述べる。2年生が制作したシューティングゲームで使したパッドの数を表2に示す。3.1節で述べたのと同様に表中の数字は、使用したパッドの数と学生の人数の関係を示している。また、合計は学生の総数を表し、最大パッドはその年度の中で一番多くパッドを用いた作品のパッド総数である。また、括弧の中はパッド総数の内の手続きパッドの数を表している。もぐらたたきでは、手続きパッドは1枚であったが、シューティングゲームでは、イメージパッドやモバイルパッドと組み合わせた手続きパッドの組が複数用いられている。

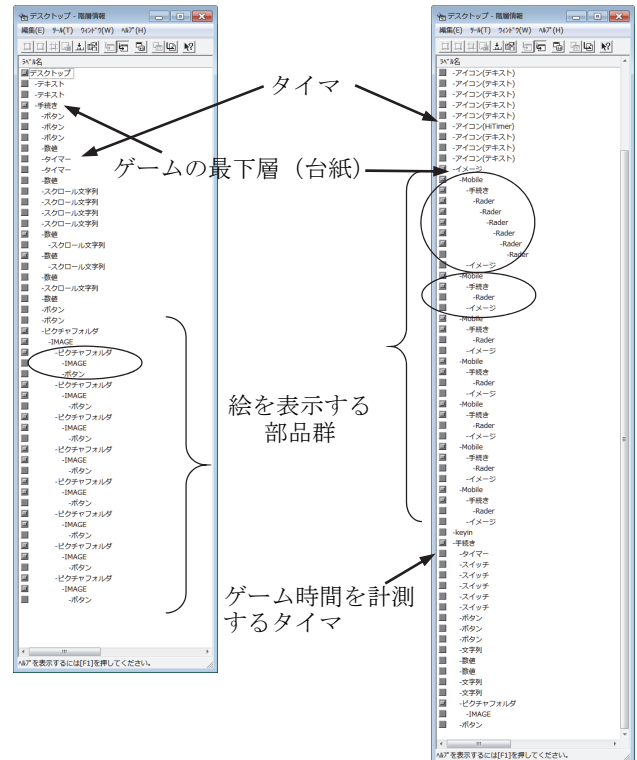
年度によって、人数の分布に若干の差はあるが、使用しているパッドの数が40枚から90枚の間に分布していることが分かる。多くの学生の手続きパッドの使用枚数は4枚から7枚程度であるが、中には、10枚以上使う学生もいる。

表1と表2を比較すると、人数の一番多い行は、もぐらたたきがパッドの数40未満であるのに対し、シューティングゲームではパッドの数50枚前後になっている。また、シューティングゲームでは90枚以上のパッドを用いたゲームが多くなっている。

3.3 パッドの構成に見られる特徴

表1および表2に見られる特徴は、ゲームプログラムに使用しているパッドの数であった。しかし、2.2節で述べたように、もぐらたたきとシューティングゲームでは、プログラムの作り方が異なる。図6に図5で示した2つのゲームのパッドの構成を示す。もぐらたたきでは、絵の動作を制御するプログラムを記述した手続きパッドは、図6の(a)の上方に示された1枚のみである。これに対し、シューティングゲームでは、8枚ある。

2.2節でも述べたが、もぐらたたきでは、モグラの数を増やすには、図6の(a)の丸で囲った部分の表示部品を増やすと同時に、手続き部品に記述したプログラムを変更する必要がある。



(a) もぐらたたきゲーム

(b) シューティングゲーム

図6 図5に示したゲームのパッドの構成

一方、シューティングゲームは、手続きパッドとモバイルパッドとの貼り合わせの関係をひっくり返すことで、1つの部品として扱えるようにしている。この一体化した部品は他の部品の動作と独立して考え、この動作を定義した手続きパッドのプログラムを記述することができる。内部状態は、他のパッドからは見えない。

従って、シューティングゲームにおけるプログラムでは、個々の部品に記述したプログラムが少ないため、その変更も容易で

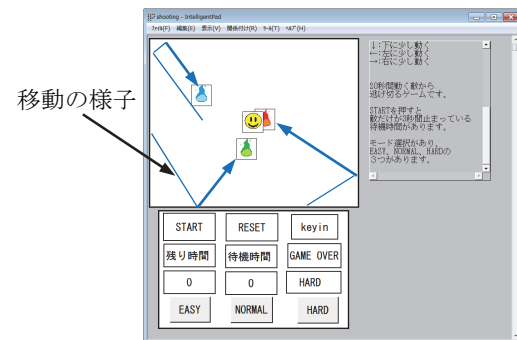


図7 鬼ごっこ



図8 いらいら棒

ある。図 7 と図 8 にシューティングゲームを変更して制作したゲームの例を示す。図 7 は、三つのキャラクターが四角い枠の中を反射しながら飛び回っている中を、キーボードの矢印キーを操作して画面中央のキャラクターが逃げ回るというゲームである。一定時間逃げ切れれば、勝ちになる。また、図 8 は画面左下のキャラクターを、図 7 のゲームと同様に操作して、途中動き回る障害物を避けながら、ゴールまで移動させるというゲームである。

図 8 のゲームは図 7 のゲームと同様にできているが、図 8 中の矩形の壁が通り抜けられないように、動き回る障害物と同様の複合オブジェクトが並べられている。動き回る障害物との違いは、移動するための座標値をプログラムで与えているか、一定値にしている(移動しない)かである。

図 8 のゲームでは、壁にしている複合オブジェクトの代わりに当たり判定に用いているレーダパッドを並べることで実現できるが、同じ部品を多用する図 8 に示したプログラムの方が単純である。この作品の場合学生は、プログラムを作成するというよりも、部品を並べることによってゲームを実現している。

3.4 多数のパッドを用いた作品の例

図 9 は、学生が制作したシューティングゲームである。このプログラムには、400 枚あまりのパッドが用いられている。図の左の UFO や落ちてくる多数の弾の移動制御を、図の右にある矩形のパッド群が行っている。このプログラムでは、パッドの制御を機能ごとにグループにまとめて管理実行している。

このゲームでは、7機の敵と1機の自機からなる。敵機は種類によって発射できる弾の数が1から5発あり、それぞれが自機に向かって撃ってくる。また、中には一定の高さまで下りてくると、自機に向かって来る敵機もある。手続きパッドは、自機および敵機、および弾ごとに用意されている。しかし、手続きパッドに記述されるプログラムは他のオブジェクトの位置関係などを意識していない。

この作品の場合、組み立てていく過程では、小さな機能を組み合わせ順に大きな機能へと組み立てることで、比較的容易に実現したと考えられる。一方でできあがった作品を見るとどのような構成になっているかは簡単には分からない。オブジェクト指向によるプログラミングの問題点が表れていると思われる。

4. おわりに

本論では、我々が開発してきた初心者のためのプログラミング学習支援システムを紹介し、これを用いて学生が制作したシューティングゲームに見られる特徴について述べた。本システムは、15 年近く高専のプログラミング実習に利用している。また、初心者のプログラミングばかりでなく、これを発展させた授業でも用いることができている。

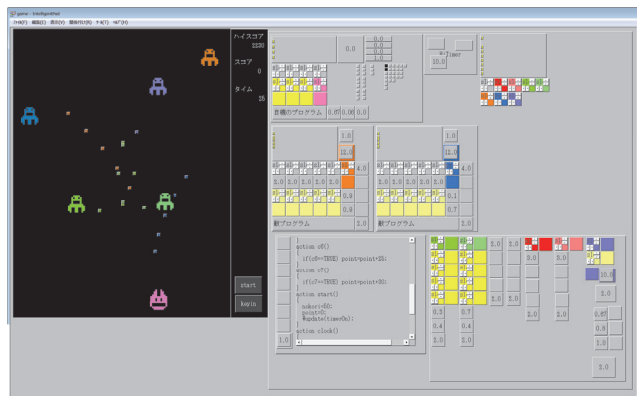


図 9 多数のパッドを用いた作品におけるパッドの管理

本論で示したシューティングゲームでは、モバイルパッドの上に手続きパッドを貼付することで、移動する複合オブジェクトを 1 つの部品として扱えるようにしている。そのためシューティングゲームの移動物体は、複写するだけで簡単に増やすことができるため、作品では多くの部品を用いたものが作られている。学生は、複合オブジェクトを部品として複写し、プログラムの一部を変更している。必要なイベントやデータのみを伝えることで、これらの部品同士を連携させている。学生はオブジェクト指向について認識はしていないが、継承に類似したことや、カプセル化といったことを体験的に学んでいると考えている。その結果、数百個のパッドを組み合わせたゲームを制作する学生もいる。

本研究の一部は、科学研究費基盤研究(C)(24501168)を受け推進している。

参考文献

- [野口 2000] 野口孝文, 田中譲: “プログラミング学習のためのツールキットシステムを用いたマイクロワールド”, 教育システム情報学会論文誌, Vol. 16, No.4, pp. 208-216, 2000.
- [野口 2006] 野口孝文, 田中譲: “プログラミング学習のためのツールキット野口孝文: “マイクロワールドにおける教材提示と管理”, 電子情報通信学会, 信学技報, Vol.106, No.437, pp.39-42, 2006.
- [野口 2015] 野口孝文, 千田和範, 稲守栄: “初心者から上級者までシームレスにプログラミングを学ぶことができる持続可能な学習環境の構築”, 教育システム情報学会論文誌, vol. 32, No.1, pp.59-69, 2015.