

プログラミング教育におけるオブジェクトの構成比率による課題プログラムの分類と特徴 Classification and Feature of Program by Composition Ratio of the Objects in the Programming Education

野口 孝文[†]
Takafumi Noguchi

1. はじめに

初心者にとってプログラミングは、言語に関する知識ばかりでなく開発環境に関する知識等多様な知識が必要とされるため難しい。我々の学習環境では学習者の負担を、図を表示したり移動したりする機能をオブジェクト部品で与えることで、軽減している[1][2]。そこでは、1つのオブジェクト部品に対しプログラムを記述した1つの手続き部品を対応させ、これを部品としてさらに他の部品と組み合わせ用いている。その結果、プログラムに用いたオブジェクトの種類と数に特徴が見られる[3]。

本論文では、プログラミングの課題として学生が制作したゲームプログラムを単純にオブジェクトの種類とその数で分類し、特徴によってグループ化されることについて報告する。この分析により、初めに与えたサンプルプログラムがどの程度改変されたかといった、プログラミングの進捗状況を知ることができるようになる。

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。

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。プログラミングの授業では、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シューティングゲームを図2に示す。図2の下に、パッドの貼り合わせを示す。こ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

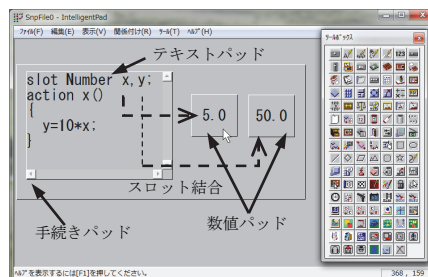


図1 手続きパッドを用いたプログラム

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。

図1のプログラムは、手続きパッドの上に入力や出力のパッドを貼付し、キーボードからの入力で結果が表示される。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。これにより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、複製するだけで簡単に増やすことができる。

また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。一方独立しているため、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。

2.3 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。

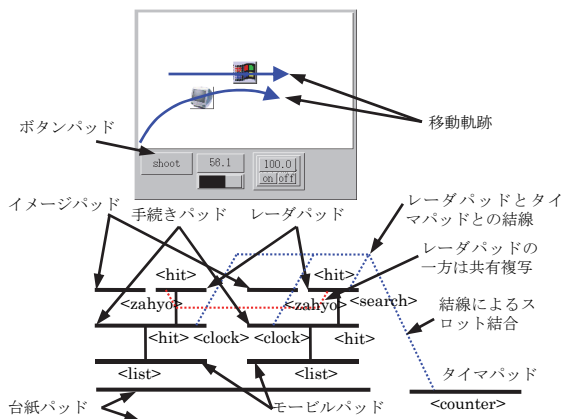


図2 サンプルのパッドとその貼り合わせ

[†] 釧路高専 National Institute of Technology, Kushiro College

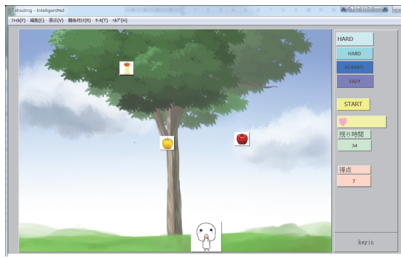


図3 学生が制作したゲーム(2014)

図3に学生が制作した作品の例を示す。図3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製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。

3. オブジェクトの構成

3.1 パッドの種類と数による分類

2章でも述べたように、IntelligentPadでは、可視化したオブジェクト(パッド)を貼り合わせることによってプログラムを作成することができる。図4に時計パッドを示す。この時計は、秒の表示に合わせて鳩時計のように動く目玉の絵を表示している。図4の左の表は、時計パッドを構成するパッドの種類と数を示している。右上は、パッドの数を多い順に対数で表したグラフである。右下は、上のグラフの傾きをパッドの数に關係付けて表したグラフである。このグラフを「傾き分布グラフ」と呼ぶことにする。

2章の2.2節でも述べたように、IntelligentPadでは、組み合わせたパッドを1つの部品として扱うことができるため、同じ種類の作品は「傾き分布グラフ」において近い位置に配置されると考えられる。図5は、シューティングゲームのサンプルプログラムの部品構成と傾き分布グラフを示している。また、傾き分布グラフに図4に示した時計のデータも表示している。サンプルプログラムと時計は、パッドの総数はほぼ同じであるが、構成するパッドの数が異なっているため、グラフの表示位置が異なっている。

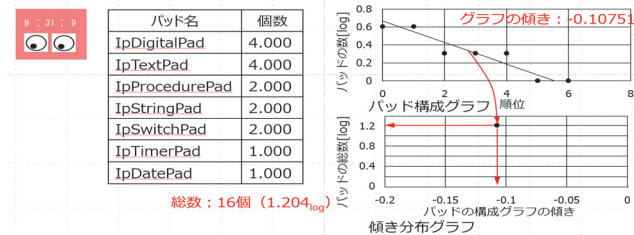


図4 時計パッドにおけるオブジェクトの構成と特徴

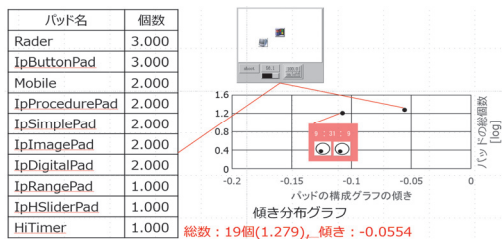


図5 シューティングパッドと時計パッドのグラフ傾き

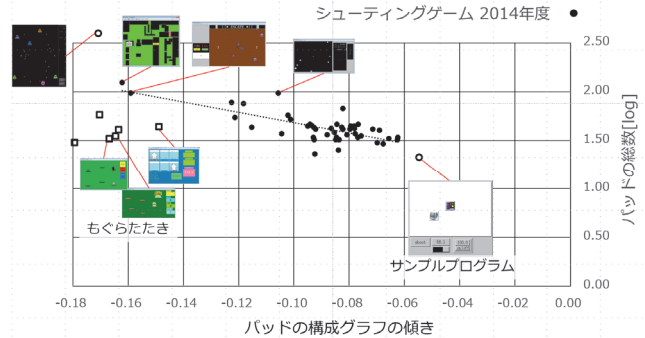


図6 学生の作品における傾き分布グラフの特徴

3.2 シューティングゲームに見られる特徴

図6は、2014年に学生が制作した45個のシューティングゲームを、前節に述べた方法を用いて作成したグラフである。図中●点が2014年のデータである。点線で示した線が近似線で、各点は近似線の近くに分布している。近似線の延長上右に表示される○点は、授業の中でサンプルとして示したプログラムのデータである。また、近似線の延長上左上方に示される○点は、過去の作品で最もパッドを多く使用した作品のデータである。近似線の左側や下に、もぐらたたきの課題データについて、□点で示してある。

図6の結果から、同じ種類のゲームはほぼ近似線の周辺に分布する事が分かる。また、サンプルプログラムも近似線の延長線上にあることから、各作品はサンプルプログラムを進展させていったようにも見える。実際、グラフは左上がりに分布し、使用しているパッドの数も多くなっているプログラムの得意な学生の作品が左方に配置されている。

4. おわりに

我々は、学習者の負担を軽減しながらも多様な学生に対応することができるプログラミングの学習環境について提案してきた。本論文では、学生が制作したゲームプログラムを単純にオブジェクトの種類とその数で分類し、特徴によってグループ化されることについて報告した。この分析により、初めに与えたサンプルプログラムがどの程度改変されたかといった、プログラミングの進捗状況を知ることができる可能性について示した。

本分析方法は、極めて単純な方法であるが、これによって学生が制作した作品の評価ができるばかりでなく、時系列的にデータを求めることによって作業の進展状況を把握できる可能性があることを示した。

謝辞

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

- [1] 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).
- [2] 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).
- [3] 野口孝文, 田中譲: “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).