

情報系高校におけるセンサを利用した プログラミング教育の実践

大見 嘉弘[†] 滑川 敬章^{††} 永井 保夫[†]

情報系高校の生徒が、加速度センサ等を用いたプログラム作品を制作する授業を5年間実践した。変数や制御構造からイベント処理、図形描画、センサ処理などプログラミングの初歩から本格的な内容まで網羅する授業を行い、大多数の生徒はゲーム等の作品を完成した。授業では、幾何学や力学を取り上げることでプログラミングと数学、科学との関連を意識させるよう工夫した。結果として、生徒は自信をつけ、その後の課題にもその経験を生かした。

A Practice of Programming using Sensors on a High School of Informations

Yoshihiro Ohmi[†], Takafumi Namekawa^{††}
and Yasuo Nagai[†]

We practiced lectures for 5 years which creates a program using an accelerometer. These lectures are for students of high school of informations. This lectures covered from introduction to advanced programming and they consist of variables, control structure, event processing, graphical drawing and sensor processing, etc. As a result almost students finished programs such as video game. In the lectures, it was made to make a student conscious of the relation of programming, and mathematics and science by showing geometry and mechanics. Consequently, the students gave confidence and applied to later subjects.

1. はじめに

近年、コンピュータの利用が浸透しており、初中等教育においても様々な取り組みが行われている。しかし、プログラミング教育については各所で取り組みがなされているが[1][2][3][4]、高等教育と比較するとまだまだ取り組みが少ないといえる。プログラミングは、コンピュータを意のままに操るといった知的な好奇心を満たす創造的活動であり、完成した時の達成感も大きい。また、コンピュータは何ができるか、何に向いているかを見極めることや、論理的思考能力を高めるという副次的効果も期待できる。

今回、我々は高大連携事業の一環として、大学教員が高校一年生に対してプログラミングの授業を5年間実施した。本授業は大学で行っていた加速度センサ等を用いたプログラミング演習の教材とノウハウを活用したものであり、プログラミングの初心者や苦手な者でも習得しやすい開発環境を提供した。本授業では、いわゆるフィジカルコンピューティング[5]の一例として加速度センサを利用した対話型のプログラムを作成し、最終的に大多数の生徒はゲームを完成させることができた。

以降では、本実践の特徴、授業で利用した教材、授業内容、結果についてそれぞれ述べる。

2. 本実践の特徴

前述のように、我々は初中等教育においてもプログラミング教育を推進すべきだと考えている。しかし、大学におけるプログラミング教育の内容をそのまま持ち込むとうまくいかないことが予想される。そこで、生徒の意欲がわく題材としてゲームなどの対話性の高いプログラム作りに絞り、その目標のために必要な概念を順を追って取り上げることにした。また、加速度センサを操作デバイスとして使い、手で振ったり傾けたりするという体感的な操作を行うことで、誰でも楽しめるプログラム作品を作ることを目指した。この特徴も生徒に意欲を持たせることを意図している。

ただし、変数や制御構造などを教える授業の初期では、そのまま行くと生徒が退屈しそうな内容が多い。このため、なるべく興味を持てるような例や演習になるように工夫した。

また、高校の特徴に合った工夫も行った。柏の葉高校 情報理科科は、情報と理数科目を専門とし、情報系、理工系の大学に進学することを目標とした学科である。このため、本実践では単にプログラミングを教えるだけでなく、要所で幾何学や力学など

[†] 東京情報大学
Tokyo University of Information Sciences

^{††} 千葉県立柏の葉高校
Chiba Prefectural Kashiwaoha Senior High School

を取り上げ、情報科学と理工系の学問との関連の深さを認識させるように工夫した。

3. 本実践の教材

本実践の教材は、プログラミング環境、ハードウェア環境、解説資料から構成される。

3.1 プログラミング環境

本実践では、プログラミング環境として Processing[6][7]を用いた。Processing は画像、アニメーション、インタラクションを扱うプログラムを手軽に作成できるプログラミング言語・環境である。基本的には Java 言語であるが、クラス定義や main メソッドの定義を省略でき、グラフィックス表示などの、頻繁に利用する機能を簡便で分かりやすく書くことができるライブラリが提供されている。内部的には「Java への変換 Java のコンパイル 実行」という過程を経てプログラムが実行されるが、シンプルなお見かけのツールで、Run ボタンを押すだけで上記の過程が実行される。そのためスクリプト言語で書くのに近い感覚でプログラミング作業が行える。また、音響処理、ビデオ処理、3D CG などを平易に扱う外部ライブラリも充実している。これは、膨大な Java の資産によるものが多い。そして、フリーソフトウェアであるため、教育機関で導入しやすい。

Processing は当初、メディアアート、インタラクティブアートの開発者向けに作られた。このため国内でも芸術系大学などで先駆けて活用が始まったが、平易にプログラム作成できライブラリが充実しているという特徴から、他の分野でもプログラミングの入門教育に導入されつつある[8]。

我々が所属する東京情報大学情報システム学科では、Java 言語を中心にプログラミング教育を行っており、その一環として 2006 年から Processing を研究室内で導入し、2007 年から演習授業で用いてきた。Processing はグラフィック描画やイベント処理などを平易に扱えるという特徴があるため、高校における初学者教育にも向いていると考えた。

3.2 加速度センサ

加速度センサは、任意の物体の動きを検出できるセンサであり、本実践を開始した 2007 年頃には安価になり、入手も容易になった。本実践で用いた 3 軸加速度センサは 3次元の加速度を検出でき、動加速度に合わせて重力加速度も検出する。したがって、静止状態でのセンサの地表面に対する傾きも検出できる。

加速度センサは手に持つなりして動かせば即座に反応するため、非常に直感的で身体性に富む。このため、生徒の興味を引きやすいと考える。従来のセンサを用いたプログラミングでは、光量センサのように 1 点の値(光量)しか分からないものや、温度センサのように時間変化が緩やかなものなど、生徒の興味が引きにくい傾向にあった。

3.3 センサの接続

本実践では、加速度センサの値を PC で読み取るために Gainer[5][9]を用いた。Gainer は、PC とセンサやアクチュエータの間をつなぐ I/O モジュールの一種であり、Processing で制御できるライブラリが提供されている。本実践を開始した 2007 年当時、類似したものに Arduino があったが、国内での Arduino の入手性が悪かったことや、PC から制御するには Gainer より手間がかかることや、ブレッドボードと一体にして使う形態ではないことから、Gainer を用いた。

Gainer は、PC とは USB で接続し、USB バスパワーで動作し手軽である。また Gainer は、電子部品の詳細や回路図、プリント基盤図、ファームウェアを含めたソースが全て公開されている、いわゆるオープンソースハードウェアであり、電子工学の知識があれば容易にクローンを製作できる。初期の製品版 Gainer はやや高価であったため、本実践の当初(2007~2009 年度)はクローンを製作して用いた(図 1)。制作費はセンサ等を含め 1 台 4,000 円程度であった。

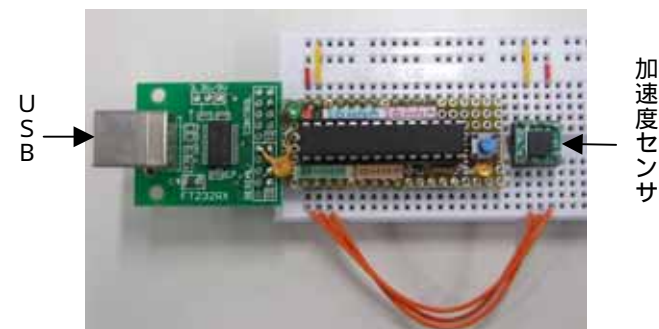


図 1 Gainer のクローン
Figure 1 A clone of Gainer.

3.4 配線部材(ブレッドボード、配線)

Gainer とセンサの接続には Gainer の思想を取り入れてブレッドボードを用いることにした。ブレッドボードは半田付けの必要がなく、配線を抜き差しすることで容易に結線が変更されるため、違うセンサを使用したり、簡単な電子回路を追加したりする場合に作業が容易である。また、配線が目に見えるため、センサと Gainer 間が接続されている様子が一目瞭然で、さらに USB ケーブルで PC に接続されていることからデータの流れを理解させやすい。配線は、生徒自身に行わせることで、より実感できるようにした。また、ブレッドボードは手で握って操作しやすいように、細長い形状のものを採用した(図 1)。

3.5 解説資料

本実践を行うにあたり 講義や演習をすすめるための解説資料を 2008 年に用意した。解説資料は PowerPoint で作成したスライドであり、講義中に教員が示しながら解説したり、生徒が適宜復習するために用いた。資料の内容を表 1 に示す。

本実践を開始した 2007 年については、この資料の作成が間に合わず、専ら板書により解説を行った。さらに授業以外の時間に復習するために当時唯一の Processing の解説書であった、文献 10) を適宜用いた。文献 10) は Processing による視覚表現や対話的プログラミングの魅力的な例が多い良著であったが、変数、制御構造、配列などのプログラミングの基礎の解説が少なく、プログラミング初学者には難しい印象があった。このため解説資料を作成し、2008 年から用いた。

解説資料の大部分は、東京情報大学情報システム学科の 3 年次向けの授業「メディアシステム」向けの資料を流用している。ただし、表 1 の 3. 基本概念と 8. 配列については、新たに加筆した。大学の授業では前もって Java の授業を 2 年間行っており、基礎的な解説を省いていたためである。また、資料全体を見渡して、初学者には難しいと思われる箇所に適宜加筆修正を施している。

以下に本解説資料の特徴について説明する。まずはじめに、取り上げる概念の順序を考慮した。資料の最初(表 1 の 1. および 2.) においては変数や制御構造を一切使用しないで、各種のグラフィックス描画の方法について説明した。Java などの従来の言語を教育する場合、早々に変数や制御構造を扱うことが多いが、それらの概念は初学者にはやや難しいと考える。そこで、より具体的で生徒が興味を持ちやすい図形描画などを最初に充実させ、生徒が冒頭でつまづきにくいようにした。また、画面の座標(X, Y) について生徒が位置感覚をつかむには、やや時間がかかるように思えたため、冒頭で集中的に座標に慣れる訓練をさせる意図もある。

2 番目に、変数や制御構造といった基本概念(表 1 の 3.) を取り上げるが、ここでも図形を描画するプログラム例を専ら用いることで、視覚的に概念を理解させ、かつ、なるべく生徒の意欲を削がないように工夫した。図 2 にその一例を示す。この例では、ある結果を実現するために複数の実装方法があることも示している。このように、なるべく多くの実装手段を示すことで、生徒が自ら考えて手段を選択したり、手段を考案することを促す狙いがある。

3 番目に、アニメーション(表 1 の 4.) について取り上げる。Processing では、ここでプログラムの記述方法が以前と比べ大幅に異なるため注意を要する。Processing では、一枚絵を表示するといったアニメーションが必要ない場合は、スクリプト言語のように、処理を行う文のみを記述すれば良い。言い換えれば、Java の main メソッドの中身のみ書けば良い。しかし、アニメーションを行うには、実行開始時の初期化を setup メソッドとして、定期的(通常 1/60 秒毎)に処理する内容を draw メソッドとして記述する必要がある。このため、アニメーションのプログラム例を数多く用意することで、

表 1 解説資料の内容

Table 1 The contents of textbook

1.	Processing の概要と基本 概要, 類似物, 最初の一步, 図形描画(点, 線, 四角, 円, 多角形), 描画モード, 画面の大きさ, 背景描画, 色指定(RGB, HSB), 不透明度, 画面の変数
2.	文字表示, 画像表示 文字表示(英語, 日本語), 画像の表示, 拡大縮小, 透明色の利用
3.	基本概念 変数, 代入, 加算, 四則演算と省略形, 繰り返し, for 文の例, 増分, 二重ループ, 乱数, 条件分岐
4.	アニメーション 基礎, 変数の利用, 自由落下, 壁の反射, 円運動, 残像効果, 画像のアニメーション, フレームレート
5.	イベント処理 概要, マウス移動, マウスクリック, ボタンの種類, 押した離れた, 論理型変数, マウスイベントの例, 画像のスクロール, キーボードの処理
6.	ゲームの製作モグラ叩きのアニメーション, 当り判定, シューティングゲームの製作, 背景のスクロール, 自機の操作, 敵の出現と移動, 当り判定
7.	配列 概念, 使い方, 配列を使ったアニメーション, 過去の値を記憶する, 過去の平均, 二次元配列, 迷路の応用
8.	座標系の変換 平行移動, 回転, ラジアン, 拡大縮小, 利用例
9.	センサ 概要, 三軸加速度センサ, フィジカルコンピューティング, Gainer, Gainer の組み立てと準備, センサ値のグラフ表示, センサ値の校正
10.	加速度センサの利用 センサ値の正規化, キャラクターの移動, 画像のスクロール, Gainer のボタンの利用, 振った量を表示, 振った回数を表示
11.	センサ値の信号処理 待ち行列の導入, 平均値の表示, 値の平滑化, 重力加速度の除外, 簡単な物理シミュレーション

なるべく生徒が戸惑わないように配慮した。その後のイベント処理(表 1 の 5.) についても実例を多めに用意した。本実践では最終的に加速度センサを用いるが、従来のマウスやキーボードのイベント処理は活用範囲が広いので、ここで取り上げている。

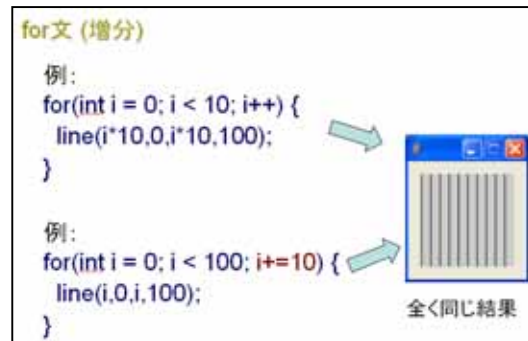


図2 解説資料の例

Figure 2 An example of textbook

次にゲームの製作に入る(表1の6.)。生徒の多くはゲームが作りたいために本実践に参加しているため、極力早めに取り上げている。ここでは、短いプログラムのみを示し、生徒がいきなりつまづかないように配慮している。制作したゲームは、決まった場所にモグラが出てくるモグラ叩きと、自機と敵機だけで弾のないシューティングゲームである。

4番目に、配列(表1の7.)を取り上げる。プログラミング教育の多くの事例では、もっと早期に配列を取り上げるが、ここでは敢えて後半に置いた。プログラミング初学者にとって配列は特につまづきやすい概念だと考えるからである。直前のゲームの作成では、配列を使わない例のみを用いている。例えば、シューティングゲームの敵機は一度に1つしか表示できないようにし、表示位置を示す2つの変数のみを使用している。そして配列の説明で、この変数の代わりに配列を用いることで、複数の敵機を表示する例を示した。このようにゲーム製作に対する配列の恩恵を示すことで、配列の価値を実感させ、なるべく理解することを諦めないよう配慮した。二次元配列の解説では、ゲームの迷路マップデータとして二次元配列を用いる例を示した。

5番目に、座標系の変換(表1の8.)を取り上げる。これは概念を理解すれば便利に使える機能であるが、使いこなしが難しいため、余裕があれば取り上げる、あるいは必要がある場合のみ取り上げるという立場を取る。

最後に、センサについて解説した(表1の9~11)。まず、各種のセンサについて解説し、加速度センサ、Gainerの利用法について解説している。次に、加速度センサの値をプログラムで扱う際にセンサ値の校正と正規化を解説している。これは、使用した加速度センサの出力値の個体差が大きく、センサ値をそのまま使うと微妙な操作に支障をきたすため、センサ値から0Gの時に0.0、1Gの時に1.0となるように数値を変換

して個体差を吸収しているためである。WiiリモコンやiPhoneのような製品の場合、加速度センサの値はこのような校正と正規化を行った値が提供されているが、それは利用者に見えないところで校正作業が行われている結果である。このように一般の人が普段意識しないことが多い計測時の校正の重要性について認識してほしい意図もあり、この部分は詳しく解説している。

次に、加速度センサの典型的な利用法としてセンサを傾けた方向に画面上の物体が移動するプログラム、センサを振った場合の回数等を数えるプログラムを示している。終わりに高度な例として、信号処理プログラミングによって、センサ値の平滑化や動加速度のみの抽出、重力加速度のみの抽出の方法について取り上げている。

4. プログラミング講座の実践

4.1 実践の概要

ここでは、柏の葉高校で行ったプログラミング教育の実践内容について述べる。

本実践は柏の葉高校情報理科が開設された2007年度に合わせて「プログラミング講座」として開始された。以降、この講座を本講座と呼ぶ。本講座の対象は情報理科の1年生であり、受講を希望する生徒のみが参加する課外授業として実施した。期間は概ね6月~2月であり、その中で放課後の2時間半程度の講座を計12~15回行った[11]。

本講座はだまかに前半、後半に分けられる。前半は、3.5節に示した解説資料を使用して、プログラミングの各種概念を順に理解する一斉授業の形式を取った。後半では、最終的に作成するプログラム作品の仕様を決め、各自が開発を進めながら、個別指導するという形式を取った。

次節以降は、2011年度に実施した本講座の各回の詳細について説明する。

4.2 実践の前半(一斉授業)

(1) 導入、図形描画、色指定

最初に本講座の全体の概要を説明し、Processingの特徴と使い方を説明した。解説資料(表1の1.)を順に使用しながら、図形描画や色指定の方法を説明し、簡単な課題を適宜行わせた。ここでは、線や円といった図形の書き方を単に習得するだけでなく、座標の感覚に慣れることに注力した。つまり、画面上の位置をイメージし、特定の位置がX,Y座標のどの辺りの値になるかを把握するための訓練といえる。最初は、この把握が一番重要であると考え、締めくくりとして、国旗を描画する課題を与えた。どの国の国旗にするかは自由とし、簡単な国旗を選び早めに完成した生徒には、他の国旗も描画するように指導した。その結果、ユニオンジャックを短時間で完成させる生徒の存在も見られた。この課題は大多数の生徒が意欲的に取り組み、図形描画、色指定、座標の把握を習得する効果的な方法であったと考える。

(2) 文字表示, 画像表示, 変数, 繰り返し

まず, 文字表示と画像表示を説明した. 画像表示では, 事前に用意しておいた画像ファイルだけでなく, ペイントソフトで生徒自身に画像を作成させ, それを表示する課題も行った. また, 画像中に透明色を設ける方法も習得させた. これはゲームで表示するキャラクターを画像で表現する際に多用するためである. ここでは, ペイントソフトで画像を作る作業ばかりに熱中し, プログラムの理解が疎かになる生徒が数名現れ, 適宜注意した. 次に, 基本的な概念として変数と繰り返し構造を説明した. 3.5 節で述べたように, 退屈と感じがちな内容のため, 図形描画と組み合わせることで工夫した. 残念ながら, 多くの生徒は概して他の内容ほどは意欲的ではないように見受けられた.

(3) 乱数, 条件分岐, アニメーション

まず, 乱数と条件分岐について説明した. 乱数はゲームのプログラムに用いられることが多いため, 多くの実例を取り上げた. 条件分岐は繰り返しと組み合わせる図形描画する例を取り上げ, 退屈にならないよう配慮し, 実用的な例を示すよう工夫した. 次に, アニメーションの方法を説明した. 3.5 節で述べたように, Processing ではこの部分からコーディング方法が大きく変わるため, 時間をかけて説明した. 最初は, 乱数を使用してランダムな位置に図形を表示するアニメーションを示した. これは変数を使用する例よりも理解しやすく実行結果も魅力的なためである. ただし, この課題で表示する図形を巨大にし, ランダムな色で変化させる生徒が例年数名現れ, 視覚に悪影響を与える恐れがあるため, 見つけ次第注意を行なった. その後は, アニメーションを実現するための変数の使い方を数多く紹介した. 玉が壁を反射するアニメーションでは条件分岐を使う例を示した. アニメーションを始めると表示が変化するため, 多くの生徒が積極的に取り組むが, 急に難しい内容になるため戸惑う生徒も多く見られた. ここで, 円運動のアニメーションを実現するために三角関数を用いた. プログラムの説明と併せて三角関数自体の説明を行い, 座標を扱うプログラムと幾何学の関係を意識させるようにした. また, 弾が落下し床で反射するプログラムについては, 力学を簡単に説明することで, 物理学の初歩との関係を意識させるようにした.

(4) イベント処理

ここでは, マウスとキーボードのイベント処理に触れた. 特徴的な課題として, 表 2 に示す, ペイントソフトのプログラム例を紹介した. これは, マウスボタンを押した際に点を描画する単純なプログラムであるが, この内容をここまで簡潔に記述できるプログラミング環境は数少なく, Processing の長所を実感させる良例と考える. 課題では, `point(mouseX, mouseY);` の部分を好きなように記述させて, ペイントソフトのブラシを自分で作成させた. この課題はほとんどの生徒が意欲的に取り組んだ. しばらくは各自で自由にブラシを作成させ, その後, 画像ブラシ, ブラシの大きさに変化を付ける, 半透明にする, 透明度を変化させ墨がかすれるようにするなどの効果

表 2 ペイントソフトのプログラムコード

Table 2 A code of painting tool.

```
void setup() {  
  background(255);  
}  
void draw() {  
  if (mousePressed) {  
    point(mouseX, mouseY);  
  }  
}
```

を出すといった実例を示した.

(5) ゲームの製作

マウスやキーボードで操作する簡単なゲームの作成方法を説明した. まずは, モグラ叩きのモグラが出て引っ込む一連の動作のプログラムを紹介し, モグラが出ている時にクリックすると, 引っ込んでいる時にクリックすると x を表示する例を紹介した. この例では決まった場所にしかモグラは出てこない. したがってこのゲームでは, クリックした時にモグラの状態を判別するだけの最も簡単な当り判定を行っていると言える. 次に, シューティングゲームを紹介した. ただし, 簡単のため, 自機 1 個と敵機 1 個が表示され, 弾が出ない敵を避けるだけのゲームである. このため, 自機と敵機との当り判定の方法のみを説明した. 弾を発射する機能は今後の課題とし, 意欲のある生徒は作るように薦めたが残念ながら自力で作成した生徒は現れなかった.

(6) 加速度センサの利用

ブレッドボード上に Gainer と加速度センサ等を挿してから PC に接続し, センサ値を PC 上で読み取れることを確認した. 確認には, 加速度センサの X,Y,Z 値をグラフ表示する Processing プログラムを使用した(図 3). このプログラムは, センサ値の性質や意味を理解しやすくするために, X,Y,Z 軸の値を別々の場所に表示し, 現在の値を棒グラフ状に大きく表示しつつ, 過去の値を波形状に表示するなどの工夫を行なった. 生徒がこのグラフを観察しながら, 加速度センサを振ったり傾けたりすることで, センサ値の意味や特徴を実感させるようにした. その後, X,Y,Z 軸がそれぞれ 0G,1G になるのはどのような状態かを考えさせ, それぞれのセンサ値(8bit)を控えさせた. そして, プログラム中でそれらの値を定義している部分を書き換えさせることで校正作業が完了する. その後, プログラムを再実行させ, グラフが 0G,1G の目盛りにはぴったり合うことで, 校正が正しいことを確認させた.

最後に, 加速度センサを傾けてゲームキャラクターを移動する方法や, 3.5 節に示した高度な例を紹介した.

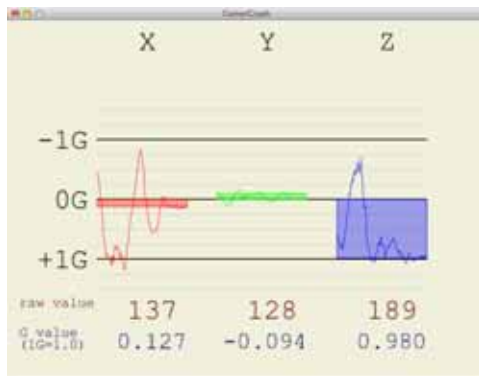


図3 加速度センサの値をグラフ表示するプログラム
Figure 3 A program which draws graphs of accelerometer values

4.3 実践の後半(作品製作)

後半では、加速度センサ等を使った対話型のプログラム作品の製作を行っていく。まず、作品の企画発表を行わせた。発表前には、2~4週間程度の長い期間を与え、作りたい作品の概要を考えさせ、製作班のメンバーを決定させた。本講座の当初(2007,2008年度)は参加者が少ないことから一人で一作品を作ることにしたが、2009年度から参加者が大幅に増えたため、班を構成して作るようにした。ただし、希望により、班を組まず一人で作ることも認めている。2011年度の場合、メンバーが二人の班が3つ、三人の班が3、四人の班が1つで、一人で製作したのは2名であった。

企画発表は、WordやPowerPoint等で作成した企画書を提示しながら口頭で説明する方法をとった。ここで、問題点や改善点を指摘し、期間内に完成できそうな仕様に調整した。その概要を付箋紙に記入させ一覽したものを図4に示す(2011年度)。この一覽は、我々が企画の全容を把握したり、効率良く指導するために用いている。

その後は、作品の完成に向けて簡単に実現できる部分から実現し、そのプログラムを徐々に拡張しながら完成に導いていった。原則として各班を回って個別に指導したが、共通のジャンル、例えばシューティングゲームを作りたい数班をまとめて講義をするなど、効率良く柔軟に指導することを心がけた。指導では、可能な限り助言に留め生徒自身がコーディングするよう努めたが、期間の制約もあり、一部に我々がコーディングしたことがある。このプログラミングの指導を6回行い、作品を完成させた。

完成させた作品は、2011年度は全てゲームであった。内訳は平面シューティング2つと、一人称視点シューティング、平面のアクション、迷路脱出、ダーツ各1つである。また、それ以前の作品もほとんどがゲームであり、いろいろ棒、弾幕シューティ

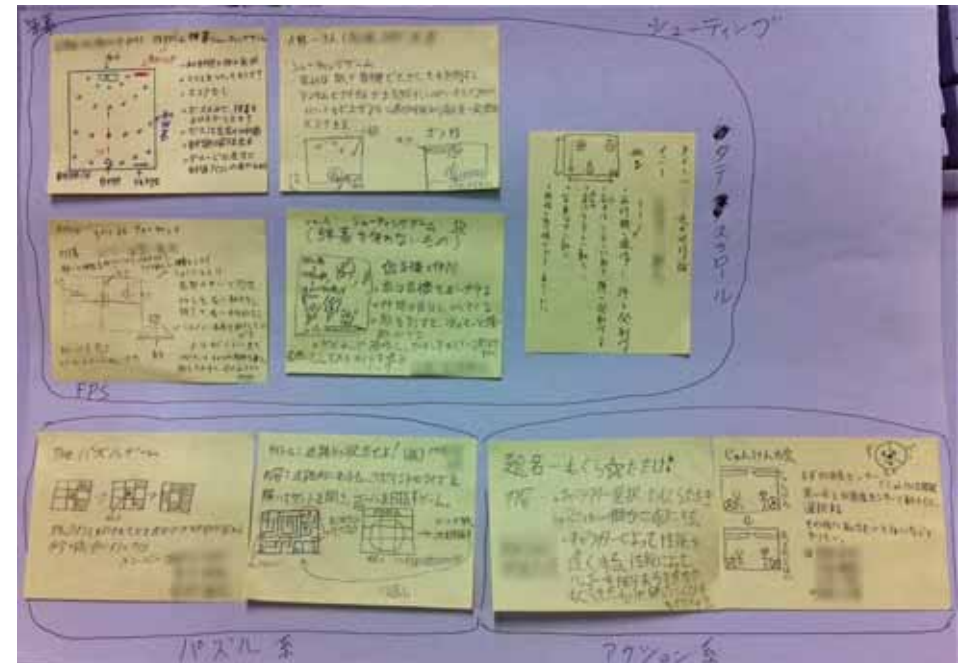


図4 プログラム作品の企画の一覽
Figure 4 An overview of plans of program works



図5 ウクレレ状の電子楽器
Figure 5 An ucelele typed instrument



図6 陣取りゲーム
Figure 6 An encampment game

ング、パズル、虫取り、魚釣り、ブロック崩し、平面カーレースなどであった。ゲーム以外では、ウクレレ状の電子楽器を作成した(図5)。これは、加速度センサを手で振って弦をはじいたかのように演奏するという企画に基づき、我々がスイッチとセンサを用いてハードウェアを作成した。生徒は、演奏するプログラムとウクレレのボディをダンボールで作成した。結果として、フレットに相当するスイッチを押しながら、指輪状にした加速度センサを振ることで演奏する楽器が完成した。

本講座で作成したプログラム作品は、高校祭で実演したり、プログラミングコンテストに応募している。そのうち2007年度の作品と、2009年度の作品が東京情報大学ソフトウェアコンテストにおいて最優秀賞を受賞した*。図6にその一つを示す。これは、マス上に駒を正方形に配置して点数を競う陣取りゲームであり、ルールを生徒自身が独自に考案したことが高く評価された。

5. 類似研究との比較

コンピュータの初中等教育において、センサーを利用したプログラム教育としては、紅林らによる自律型3軸制御ロボットによる制御教育[4]がある。対象とするロボットは、センサ等を利用して適切な場所に移動し、対象物を持ち上げるなどの仕事をおこなうために、複雑な制御が必要になる。このような動作をプログラムとして作成し、自律的な動作を実現するために、ドリトルを使ったプログラミング教育をおこなう学習意欲を高めたものづくりとプログラミング学習をおこなえる教材を提供している。

本研究でも、初中等教育に対して、コンピュータのソフトウェアとハードウェアの両方を興味を持って経験させることを目的としている。具体的には、オープンソースハードウェアとオープンソースソフトウェア(開発環境)を利用できるツールキットとともに、さまざまなアプリケーション開発用のレシピ(教材)を提供している。今回は、これらを利用したプログラミング講座の実践について説明した。本研究では、初中等教育においてフィジカルコンピューティングの考え方を導入し、さまざまなセンサーを利用したインタラクティブなシステムを実現するための教材を、ハードウェアとソフトウェア両方の観点から支援することを目的としている。初中等教育において、このようなフィジカルコンピューティングを実現していくという考え方で、学生に対してものづくりに興味を持たせつつ、プログラミング教育の実践を支援しているアプローチは今後の情報教育において有益であると考えている。

6. まとめ

本実践は、高校の情報と理数系科目を専門とする学科において、プログラミング導入教育を5年を通して行ったものである。生徒が意欲を持つと思われるゲーム等の対

話性の高いプログラムの製作をテーマにしたことにより、密度が高い内容にも関わらず大多数の生徒が作品を完成させるまで辛抱強く取り組んだという印象がある。ただし、放課後の課外授業であり、他の部活との両立で支障をきたしたり、強制ではないため途中で断念する生徒が少なからずおり、年々配慮を重ね改善に努めている。本実践は全て1年生に対して行っているが、その後、2年生、3年生でより高度な課題に取り組む際にProcessingで開発する機会が多く、本講座で習得したプログラミング能力が生かされている。

生徒の理解度については、制御構造や配列などの基本概念を工夫して教えたつもりではあるが、我々が期待したほどには理解していない生徒が多く見られた。今後は原因を把握して、本講座の改善を図りたい。本講座で使用した解説資料は、生徒の反応を見ながら年々改善しているが、元が大学向けのためまだまだ分かりにくいところが残っている。今後も改善を続け、初学者に向けたものとしたい。

本講座で製作したプログラム作品は、ほとんどがゲームであるが、我々はゲームに限らず4.3節で述べた電子楽器など幅広い作品を製作することを意図している。原則として、生徒が作りたい作品を作るという方針は変えないが、今後は企画発表の前にゲーム以外の作品例などを示して、作品の幅を広げたいと考えている。

謝辞 本実践は、2010、2011年度にサイエンス・パートナーシップ・プロジェクト事業として科学技術振興機構より支援を受けた。ここに記して謝意を表す。

参考文献

- 1) 兼宗進, 阿部和広, 原田康徳: プログラミングが好きになる言語環境, 情報処理 Vol.50, No.10, pp.986-993 (2009).
- 2) 兼宗進: 教育用プログラミング言語の動向, 情報処理 Vol.48, No.6, pp.589-593 (2007).
- 3) 紅林秀治: 制御教育への利用, 情報処理 Vol.48, No.6, pp.602-606 (2007).
- 4) 辰巳丈夫: 教育用プログラミング言語を利用した教科教育と情報教育, 情報処理 Vol.48, No.6, pp.612-615 (2007).
- 5) 小林茂: フィジカルコンピューティング概論, 情報処理 Vol.52, No.8, pp.914-917 (2011).
- 6) Processing, <http://processing.org/>
- 7) R., Casey Reas and F., Ben Fry: Getting Started With Processing, O'Reilly Media(2010) (船田 巧(訳): Processingをはじめよう, オライリージャパン(2011))
- 8) 菊池誠: プログラミング, 何をどう教えているか: Processingによるプログラミング教育, 情報処理, Vol.52, No.2, pp. 213-215(2011).
- 9) 遠藤孝則: Gainer 入門, 情報処理 Vol.52, No.8, pp.918-921 (2011).
- 10) 前川峻志, 田中孝太郎: Built with Processing, ビー・エヌ・エヌ新社(2007).
- 11) 千葉県立柏の葉高校 プログラミング講座(SPP 採択事業), http://cis.kashiwanoha.ed.jp/?page_id=695

* コンテストの審査は作品の内容によって公平に行われており、我々は審査は一切関わっていない。