

過去体験を重視した横断型プログラミング授業の設計と評価

伊藤一成^{1,2} 吉田葵¹ 安彦智史³ 竹中章勝⁴ 中鉢直宏⁵

青山学院大学では、後期（9月から1月）に、社会情報学部2年次以上選択科目「オブジェクト指向プログラミング」を開講している。授業では、プログラミングの知識や技術を身に付けるだけでなく、主体的に学ぶ姿勢を身につけることを目的とした。そこで「(教師は)教えない」を合言葉とし主体的に学ぶことを促した。また、学生の興味を惹き、アイデアを引き出せるよう、センサーボードをはじめとした様々なアイテムを提示し、自らのアイデアを形にした同学部1年次必修科目「社会情報体験演習」の過去体験を想起させる工夫を施した。本稿では、その授業設計について報告するとともに、履修学生に対するアンケート結果及び成果物から、技術の習得及び学ぶことに対する意識の変容について考察する。

A Trial of Cross Programming Class Design with Awareness of Past Experience

KAZUNARI ITO AOI YOSHIDA
SATOSHI ABIKO AKIMASA TAKENAKA
NAOHIRO CHUBACHI

School of Social Informatics, Aoyama Gakuin University offers a course titled “Object oriented programming”. This course is offered as a selective course for the sophomore or higher in second (September-January) semester. A major objective of this course is not only to teach basic study skills such as programming skills and ability but also to learn proactively. For this purpose, we employed a method of learning by “making”. We prepared a slogan in learning proactively, “We (lecturers) do not give you (students) any instruction.” And then, we also prepared a lot of gadgets in stirring students’ imagination for creation and capture their hearts and mind, and then we offered “making” experience. This course also aims that students look back on past experience of the course “Hands-on Practice in Social Informatics” that is compulsory course of first year. This paper describes the class design, and we consider the change of students’ mind from the results of questionnaires and student’s final products.

1. はじめに

青山学院大学社会情報学部は、文理融合を掲げる学際系学部である。本学部では、大学2年生以上を対象に、オブジェクト指向の概念理解を主眼に置いた授業を設定している。使用言語はJavaである。この講義の目的は、「オブジェクト指向分析・設計に基づいたオブジェクト指向開発におけるプログラミング技法に必要な基本概念について総合的に学習する。」ことにある。例えばはじめにクラスの特徴を学習する際、オブジェクト指向プログラミングの特徴をふまえて教える必要がある。現実社会をオブジェクトとクラスに投影することを実践させ、その重要性を学習者に理解してもらうことである。例えば、表1のように、人間をクラスとして投影させたり、自動車をクラスとして捉

え、フィールドやメソッドを考えさせたり、現実世界のオブジェクトを投影させるような演習を設ける。

表1 通常の課題例

1:「名前・身長・体重などをメンバとしてもつHumanクラスを作成せよ」(フィールドやメソッドは自由に設計すること)
2:「自動車クラスCarに対して、フィールドやメソッドを自由に追加せよ。たとえば:タンク容量をあらわすフィールドを追加する・ナンバーを表すフィールドを追加する/燃費を表すフィールドを追加する/移動による燃料残量の計算に燃費を反映させる/給油のためのメソッドを追加する etc…」

人間や自動車を使ったクラスの説明の例は、オブジェクト指向に関する本の解説[1]にも扱われており一般的なオブジェクト指向プログラミングの課題例と思われる。このような現実世界のオブジェクトを学習者によってクラスなどに投影させている。

例えば、表1の例に示すような人間クラスの場合、一般性が高く、考えやすい題材であるが、社会的視点、身体的視点、様々な視点から捉えることができ、課題として持て余す可能性がある。また、自動車クラスの場合は、そもそも自動車に興味がない学生にとっては難しい。結果、ど

1. 青山学院大学 社会情報学部
School of Social Informatics, Aoyama Gakuin University
2. 青山学院大学 ヒューマンイノベーション研究センター
Human Innovation Research Center, Aoyama Gakuin University
3. 青山学院大学 附置情報メディアセンター
Institute of Information and Media, Aoyama Gakuin University
4. 青山学院大学大学院 社会情報学研究科
Graduate School of Social Informatics, Aoyama Gakuin University
5. 島根大学 教育・学生支援機構
Organization for Education and Student support, Shimane University

ちらも教科書の誘導に従って提案された項目をフィールドやメソッドに追加するだけの単純作業となってしまう危険性を往々にはらんでいる。

また文献[2]の書籍をはじめ、大学のプログラミング科目での利用も想定した Java 入門書籍の多くは、コンソール上の文字ベースでの入出力を想定しており、課題テーマを想像したり、興味関心を喚起したりするのは難しい。この問題点は、岡本らも指摘しており[3]、何らかの改善策が求められる。

そこで、実物を Java プログラミングに反映させた実践や、ゲームの要素をオブジェクト指向プログラミングの課題に組み込んだ実践が報告されている。例えば、土肥らは、Java 初学者のつまずきの要因を分析するとともに、定義されたクラスを表現する実物を見せながらオブジェクトの振舞いや状態を理解させる工夫について言及している[4]。また、水口は、Java 言語演習科目において、受講者のモチベーションを維持させるために、継承やオブジェクト変数などオブジェクト指向プログラミングに関する理解を深めるためにゲーム要素を取り入れた課題を設計している[5]。

2. 振り返りを意識した横断型授業設計

2.1 方針

岡本らは、プログラミングの概念理解に関して、出力の動作を視覚的に顕在化することの重要性を指摘しており、視認性、判別性、予測可能性、独立性の4つの状態から評価し有効性を示し、プログラミング教材作成の際の指針として提案している[3]。ここで岡本らが提示する、4つの側面とその具体的解決方法について表2に示す。

表2 視覚的顕在化と具体的解決方法（文献[3]より抜粋要約）

視覚的顕在化の4つの状態	具体的解決方法
視認性（大きさ、速さなどが視認可能な動作である）	表示サイズや動作速度を適切に変更する
判別性（周囲の視覚的要素を区別して認識可能）	表示位置を分離するか、他の視覚的要素から際立たせる
予測可能性（動作および動作位置が予測できる）	事前に明示するか、既存の知識や経験をもとに容易に予測できるようにする
分離化（他の命令に基づく動作と区別分離できる）	複数の動作を区別が可能なかたちに分離する

これに基づき、本手法では以下の3点を指針とする。

1. 過去の経験をふまえた相互学習
2. Processing の利用
3. コンテンツにピクトグラムを採用

Processing 及び人型ピクトグラムは、特に高い「視認性」、

「判別性」を担保するために用いる。また「予測可能性」は、学習者の知識、過去経験に基づくトピックにすることで解決する。この際、学習者の知識および過去体験には、「独立性」は人型ピクトグラムを用いた日常生活動作とプログラミング片をマッピングさせることで区別させる。岡本らは、独自に開発したマイコンボードの実習をテーマに評価しているが、本研究では、ソフトウェア、ハードウェア、過去体験という、より総合的な視点で視覚的顕在化の4状態実現を試みる。以下2.2節から2.4節で具体的な方策を説明する。

2.2 過去の経験をふまえた相互学習

学習者が持つ知識や状況から自身の体験や物事を想起させる実習型課題により、間接的にオブジェクト指向の理解を深めることを目指した。これによりオブジェクト指向プログラミングの再確認を促し、オブジェクト指向プログラミングを理解する有用な手段になるのではないかと考えた。そこで、2012年度より本学部1年生の春学期必修科目である「社会情報体験演習」に着目し、この授業体験を再現した授業設計を行なった。この科目では、複数ある演習項目の一つに、Scratch を用いて、センサ情報に基づく処理やハードウェアの制御などフィジカル・コンピューティングを通じてのものづくりの楽しさを体得する授業を3週×2時限(90分×2)設定している。その中で、Scratch の基本、制御ボード経由でのモーター制御により LEGO®で作られたオブジェクトを Scratch から制御、グループによる自由作品作成と相互評価からなる実習を行なっている[6][7]。

2.3 Processing の利用

Processing はビジュアルデザイン、インタラクションデザインなどの分野に特化したプログラミング環境である。Processing は Java のフロントエンドアプリケーションであり、内部的には Java コードを生成している。そのため、Java との親和性が非常に高い。またホームページから無償でダウンロード可能であり、教育の現場でも国内外問わず Processing を使った取り組みが数多く報告されている[8]。

2.4 人型ピクトグラムの使用

ピクトグラムとは日本語で絵記号、図記号と呼ばれるグラフィックシンボルであり、意味するものの形状を使ってその意味概念を理解させる記号である。特に具象物を表現したピクトグラムはその抽象度の高さから自分自身や本人に関わる人物事物と同一視する効果があると言われている。有名な「非常口」のマークは、デザイン策定の段階で、視

a <http://processing.org>

認性に加え、実際に避難中の人が如何に出口へ向かう人型のピクトグラムと自身とを同一視するかにデザインの労力が払われたという[9]。人型ピクトグラムの例を図1に示す。

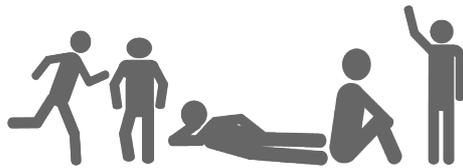


図1 人型ピクトグラムの例

つまり使用している人型の標準的なピクトグラムは知覚レベルでは、人自体や人の特定の状態を想起させる。

筆者らの研究グループでは、この特性をふまえ、人型ピクトグラムを活用した Scratch プロジェクトをこれまでに数多く手がけている。

図的要素を組み合わせることでプログラムを記述する方式は、これまでに多くの提案がなされている[10]が、あくまで本件では、コンテンツ表現としてのピクトグラムの同一視効果に注目する。

3. 実践概要

3.1 科目（オブジェクト指向プログラミング）シラバス概要

実施科目は、2015年度秋学期青山学院大学社会情報学部2年生以上の選択科目である「オブジェクト指向プログラミング」「オブジェクト指向プログラミング演習」の2時間続きの(90分+15分休憩+90分)授業である(以下まとめて「オブジェクト指向プログラミング」と記す.)。同じく春学期開講2年生以上の選択科目である「プログラミング基礎」(内容はJavaによる構造化プログラミング)を学習していることが前提となっている。履修登録者は38名であったが、実際の授業出席者は31名であり、以後この31名を履修対象者として話を進める。講義の概要を表3に示す。下線を引いた第9回から第13回が本研究の実践部分である。実質的な講義開始となる第2回目から第7回目までは、オブジェクト指向プログラミングの基本事項である、クラス、継承、多相性などについてJavaの書籍[1]を参考にした講義を毎回冒頭30分程度行い、その後与えられた実習課題を数問順番に解いていく、一般的な個別実習スタイルである。

表3 講義の概要(下線部が今回設計対象)

全15回	内容
1	ガイダンス, 構造化プログラミングとしてのJava, IDEの使い方
2~3	クラスとインスタンス
4	クラス変数とクラスメソッド, パッケージ

5	継承
6	多相性
7	抽象クラス, インタフェース
8	中間試験
9	<u>Processing 入門</u>
10(1 限目)	<u>Processing, NanoBoardAG によるハードウェア制御,</u>
10(2 限目)	<u>作品制作(個人作品)</u>
11(1 限目)~13(1 限目)	<u>作品制作(グループ作品)</u>
13(2 限目)	<u>作品発表会</u>
14	動画像処理で学ぶオブジェクト指向プログラミング(1)
15	動画像処理で学ぶオブジェクト指向プログラミング(2)

今回対象とする「オブジェクト指向プログラミング」の第9回から第13回(2時間×5回=10時間)と社会情報体験演習(2時間×3回=6時間)のシラバス比較を表4に示す。

表4 シラバス比較

時間目	オブジェクト指向プログラミング	時間目	社会情報体験演習
1~2	1. Processingの概要 2. Processingプログラミングの基本 3. Processing基本課題	1	1. Scratchの概要 2. Scratchプログラミングの基本 3. Scratch基本課題
3	4. センサーボード(NanoBoardAG), モーター等の配布物一式の説明 5. センサーボードを用いたProcessingプログラミング	2	4. センサーボード(NanoBoardAG), モーター等の配布物一式の説明 5. センサーボードを用いたScratchプログラミング
4	6. Processingによるモーターの制御 7. NanoBoardAGを用いた個人作品の作成(個人)	3	6. Scratchによるモーターの制御 7. NanoBoardAGを用いた個人作品の作成(個人)
5~9	8. グループ作品作成(2~4人1組, 11グループ)	4~5	8. グループ作品作成(基本4人1組, 16グループ)
10	9. 品評会 10. 授業アンケートの実施	6	9. 品評会 10. 授業アンケートの実施

b <https://scratch.mit.edu/users/GOSEICHO/>

以下 3.2 節から 3.5 節までこのシラバスの時系列に沿って具体的内容を説明する。

Processing では、起動時 IDE 上にすでに、PApplet クラスの派生クラスが 1 つ自動的に生成されており、学生によるクラス定義は実際には、自動生成されるクラスのインタークラスとして定義される。自動生成されるクラス宣言部自体は Processing IDE 上には表示されないで、一見すると Java のプログラミングとは異なった印象を学生は抱いてしまう。そもそも Processing は PApplet クラスで定義されている豊富な描画関係のメソッドの御陰で、明示的なクラス定義をしなくても手軽にグラフィックスプログラムが実装できるのが特徴だが、Java からの移行という視点では、学生が混乱しやすい点となるため、注意が必要となる。そのため、1,2 時間目の「2. Processing プログラミングの基本」では、はじめに Java から Processing への移行について留意すべき点を集中して短時間で解説する。他は個人作品作成や、グループ作品作成の実習時間に配布資料や書籍を用意し、自学習させることをねらいとする。社会情報体験演習では、「2. Scratch プログラミングの基本」で「ネコ逃げ」プログラムcを使って 30 分程度で進めている。これに対して「2. Processing プログラミングの基本」では、この「ネコ逃げ」に相当する「ピクトグラム逃げ」プログラムdを作成し、これを使用する事とした。「ネコ逃げ」とは壁に反射して直線運動する猫からマウス操作によって鼠を操作し逃げ続ける単純なミニゲームである。「ピクトグラム逃げ」と「ネコ逃げ」のスクリーンショットを図 4 に示す。キャラクタと使用プログラミング言語は異なっているが、いずれも同一機能のプログラムを作成している。

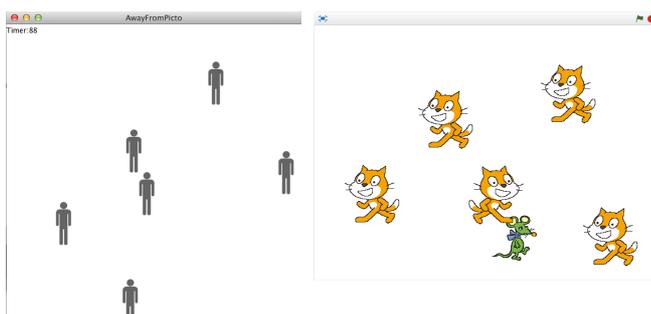


図 4 ピクトグラム逃げ (左) とネコ逃げ (右)

3.2 専用ハードウェアの利用

物作り感や、実装感が、プログラミングの教育効果を高めるといふ報告がいくつかなされている。「社会情報体験演習」では、「4.センサーボード」「5.センサーボードを用い

たプログラミング」「6. Scratch によるモーターの制御」で NanoBoardAG というセンサーボードと LEGO を併用した実習を取り入れており、当授業でも可能な限り忠実に再現させることに力点を置いた。

3.2.1 センサーボード NanoBoardAG

NanoBoardAG は MIT メディアラボが開発した Scratch 用センサーボード(PicoBoard)の互換機である[11]。新村とも氏が個人で開発している。スライダ、ライトセンサー、サウンドセンサー、タッチセンサー (ボタン)、抵抗センサ入力端子(4 系統)を有し、さらにモーター駆動用の IC や回転速度制御のための機構もボード上に備え付けられているため、直接モーターとボード上のモーター出力端子をジャンパーケーブルで接続するだけで制御可能になっている。NanoBoardAG の外観を図 5 に示す。

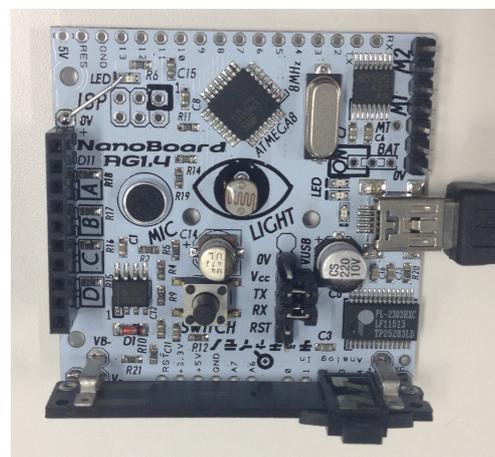


図 5 NanoBoardAG 1.4

Scratch SensorBoard Protocol に準じた Scratch 用センサーボードは高価で、さらにブレッドボード上にモーター制御のための回路を別に自ら構築しなければならなかった。そのため、限られた時間で学生が実習を遂行する上でつまずきやモチベーション低下の要因になると考え、2012 年度開始の「社会情報体験演習」に合わせ NanoBoardAG の開発をお願いした。NanoBoardAG は現在も改良が続けられており、2016 年 2 月現在の最新バージョンは 1.4 である。本学をはじめとして、多くの大学、初等中等の教育機関やワークショップ等で利用実績があるe。さらに、NanoBoardAG は汎用マイコンボードである Arduino fとして転用可能なことが大きな特徴である。そのため、Arduino 用の IDE を使い任意のプログラムを NanoBoardAG に書き込んで使用することができる。例え

c <http://swikis.ddo.jp/abee/77>
 d <https://github.com/goseicho/AwayFromPictoForProcessing>

e 作者ホームページによれば、津田塾大学、CANVAS、仙台高等専門学校、生理学研究所、日本科学未来館などで使われているとある。
 f <https://www.arduino.cc/>

ば、NanoBoardAG を Arduino として転用したドリトルの実践も報告されている[12]。本取組みでは、本学部の複数科目や PBL 型科目で横断的に利用することを想定している。従って NanoBoardAG 側に書き込まれているプログラムは頒布時の Scratch 用の状態で変更せず、Processing 側で Scratch SensorBoard のプロトコルに準じた通信を解析し、動作する NanoBoardAG クラスを独自に実装し、テスタークラス NanoBoardAGTester クラスを含むサンプルファイルを授業中に提供した^g。このサンプルファイルを使用して、「5.センサーボードを用いた Processing プログラミング」～「6.Processing によるモーターの制御」を行なった。図 6 にサンプルプログラム実行画面と Scratch でセンサ監視板（各センサの値を表示させるパネル）表示時のスクリーンショットを示す。実際に Processing によるサンプルファイル実行前に、Scratch でセンサ監視板（各種センサ値を表示するパネル）を表示させる操作を学生にさせ、また Processing と Scratch で類似の画面構成にすることで、振り返りを促す。

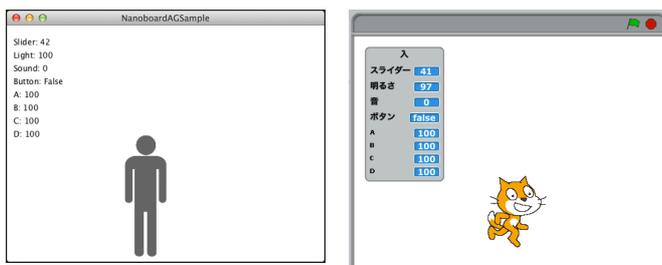


図 6 Processing のサンプルプログラム実行画面(左)と Scratch でセンサ監視板を表示させた画面 (右)

一般的に諸処の大学では情報系の専門科目で C, Java, Processing, Ruby, PHP など様々なプログラミング言語を用いた授業が用意されているが、それぞれ独自に様々なハードウェアと連携した授業実践が多数報告されている。一方、ハードウェアをハブに様々なプログラミング言語を適正やプログラミングパラダイムに基づき学習者が自律的に選択しながら学習する環境を講義の枠を超えて横断的に構築できるカリキュラムづくりが重要であると考えている。

3.2.2 LEGO®ブロック

通常の LEGO®ブロックでは、ポッチと呼ばれる突起を組み合わせてオブジェクトを作ることができる。LEGO® テクニックシリーズのブロックが付属している書籍である LEGO® Crazy Action Contraptions を用いている。LEGO®テクニックシリーズではポッチに加え、ブロック側

面に空いた穴の部分にペグと呼ばれるコネクタを使用して連結できる。また LEGO®テクニックシリーズは歯車やバネなど様々なパーツがあるのが特徴的である。

LEGO®を使ったオブジェクト指向プログラミングの教材では、Butterworth が、LEGO® Mindstorms を使ったオブジェクト指向プログラミングのスキル習得方式を提案している[13]。しかし LEGO® Mindstorms を使った試みはその自由度から、あくまでロボット制御の視点からのアプローチになっている。本件では、次に続く「グループ作品作成」のヒントとして、実際に手を動かすものづくり演習の前準備と位置づけているのが特徴である。

3.3 グループ作品作成と使用ハードウェア

「8. グループ作品作成」は第 11 回の 1 時限目から第 13 回の 1 時限目まで合計 5 時限×90 分の計 450 分を要して、2～4 人一組になりグループで作品を作成する。グループは教員側であらかじめ決めるのではなく、学生側で自由にグループメンバーを選択できるようにした。

「社会情報体験演習」でも合計 2 時限×90 分の計 180 分を要して同様のグループ作品作成を行なっている。ただし「社会情報体験演習」では、1 年次春期必修授業ということもあり、グループメンバーは教員側であらかじめ決定している。

「社会情報体験演習」と同様に PC 実習室に整備されている PC では、以下 3.3.1 項から 3.3.3 項に挙げるハードウェアや、独自で動作するハードウェアも複数用意し自由に利用できるようにした。

3.3.1 外付け Web カメラ

USB により PC と接続し、撮影した映像を PC にリアルタイムに取り込むことができるカメラである。

3.3.2 Leap Motion

手と指の位置や動作を、赤外線カメラと赤外線 LED により感知する、小型のジェスチャーセンサーであるⁱ。2012 年に Leap Motion 社から発売された。両手の 10 本の指を同時に感知できることに加え、スワイプや円を描くなどの動作を感知することができる。USB により PC と接続し利用する。

3.3.3 ポケットミク

ポケットミクは、ヤマハが開発した音声合成 LSI「NSX-1」を搭載し、スピーカやリボンコントローラ、MIDI インタフェースを備えたハードウェア NSX-39 の商品名である。2014 年 4 月に学研から発売された^j。スタイラスを利用して、ボーカロイド初音ミクの声を演奏できる。初音ミクと

^g 以下の URI で公開している。
<https://github.com/goseicho/NanoBoardAGForProcessing>
^h ISBN 978-1-59174-341-5

ⁱ <https://www.leapmotion.com/>
^j <http://otonanokagaku.net/nsx39/>

は、クリプトン・フューチャー・メディア株式会社が展開するバーチャル・シンガーであり、ヤマハの開発した音声合成システム「VOCALOID」に対応したボーカル音源のことである。

このうちいくつかは、ドライバインストールの制約から PC 教室の PC では利用できないが、当該授業では、持ち込んだ自身のノート PC を使って実習をしている学生も多い。そのため持ち込みノート PC を使って授業を履修している学生はこれらハードウェアを使った作品作成も許可した。「8. グループ作品作成」では、必ずしも対象授業で教授した Processing に制限せず、個々のグループで、使用するプログラミング言語は自由に選択してよいことを講義内でアナウンスした。また作品製作において「コミュニケーションを活性化する作品」というテーマを設定した。

3.3.1 項の外付け Web カメラは Processing がカメラ入力を標準でサポートしている。また 3.3.2 項の LeapMotion と 3.3.3 項のポケットミクは Processing で動作させるためのライブラリが公開されているため、Web で検索するとコーディング例を参照できる。このように高い自由度を許容することで、学習者の関心を喚起させる学習環境構築を目指している。またこの授業では、「オブジェクト指向プログラミング」「オブジェクト指向プログラミング演習」を履修済みで、さらに「社会情報体験演習」の SA (Student Assistant) も担当済みの学部学生 SA3 名がグループ活動のサポートを行なう。3 名はいずれも「社会情報体験演習」の演習内容や、「社会情報体験演習」で 3.3.1 項～3.3.3 項のハードウェアがどのように扱われたかについて把握している。授業のサポートにあたっては、「社会情報体験演習」と同じく、可能な限り学習者自らが資料を探したり考えたりするなど思考を促すアドバイスにとどめ、教えないことを心がけた。

3.4 品評会と振り返り

「9.品評会」も「社会情報体験演習」と全く同方式で行なう。品評会に先立ち、「作品紹介シート」を作成し、発表場所に置くか、または手で持って発表してもらおう。記載項目は以下のとおりである。

- ・グループ名
- ・グループメンバー
- ・作品タイトル
- ・作品紹介（操作方法、どういう作品か）
- ・利用したアイテム
- ・どういうことができるか
- ・今後どのように発展させられるか

「作品紹介シート」は、色画用紙に手書きで記述するか、

PC で作成したポスターを印刷して掲示するように指示した。「作品紹介シート」の例を図 7 に示す。

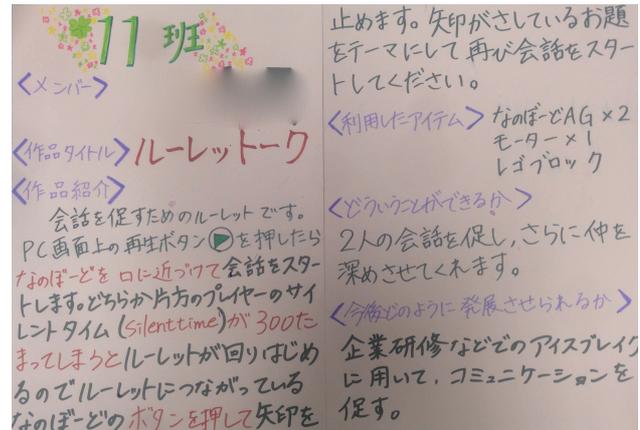


図 7 「作品紹介シート」の例

品評会では、グループ内で作品紹介担当と出資担当に分かれる。紹介担当とは、グループの持ち場で作品の紹介をする担当であり、出資担当とは、全グループの作品を見てまわって出資する担当のことをいう。出資担当者は、用意された仮想通貨を使って出資したいグループに出資する。品評会は 40 分間行い、前半終了の 20 分後にグループ内で紹介担当と出資担当を交代させる、よって履修者全員が自グループ作品を紹介・説明することになる。履修者各人が 50GOSEICHO, 20GOSEICHO, 10GOSEICHO(ここで GOSEICHO は仮想通貨単位)札を各 1 枚ずつ所持し、自分以外のグループのいずれかに投資し、最終的に最も多くお金を集めたグループを表彰する。仮想通貨の裏面には、投資する理由を記述する欄を設けた。仮想通貨の例を図 8 に示す。



図 8 仮想通貨 GOSEICHO 札 (上: 表面 下: 裏面)

4. 実習全般の考察

本章では実習全般について、観察と、提出課題プログラムの定量的評価の両面から考察する。

4.1 実習中の観察に関する考察

当該授業では、SA がヘッドマウントカメラを装着し、教室を巡回した。その講義動画を元に定性的な考察を行なった。

k <http://www.crypton.co.jp/>

4.1.1 1週目の行動分析

1週目は図4の左に示した「ピクトグラム逃げ」を提示した。このピクトグラムを題材としたクラス定義や継承関係の簡単なサンプルを参照しつつ、同時に Processing で定義されている PImage (画像クラス) などのクラスについて Processing の API サイトを見ながら作業する様子が確認された。また、1年次の「社会情報体験演習」の Scratch ではスプライト (表示されているキャラクタ, オブジェクトに相当) が, ステージ (表示画面領域) の端を跳ね返る処理は「もし端についたら, 跳ね返る」というブロックを使えば良かったので, 学習者は速度の概念や跳ね返りの条件処理を記述しなくてもよかった。

一方今回の Processing では, 座標系や, 速度の定義, 跳ね返り条件の記述が必要になるため, サンプルプログラムに関するその部分の挙動に関する質問や学習が見受けられた。本来はオブジェクト指向プログラミングと直接の関係はないが, 跳ね返り判定のように, 数学や物理学の諸原理に基づいた実世界上に起こる現象をモデル化しシミュレーションする体験には関心が高いようである。

4.1.2 2週目の行動分析

2週目は図6の左に示した NanoBoardAG を Processing から制御する提供プログラムを改変する課題が課された。

1年次の「社会情報体験演習」の Scratch では, NanoBoardAG を USB ポートに接続するだけで, 各種センサ値が変数ブロックとして登録されるので, 学生は NanoBoardAG を利用しない通常のプログラムとの差異を意識する事はない。一方今回の提供プログラムでは, 動作原理やプログラムソース自体の解読が求められる。そのため提供プログラム自体の把握に履修者は一定時間を要したようだが, センサの挙動と出力の関係を試行錯誤的に確認しながら少しずつプログラムを変更する作業が頻繁にみられた。入力はハードウェアのセンサなので具体的に知覚しやすく, また出力に関しても画面上のピクトグラムの表示変化の形で認識されやすいという題材自体の特性が影響していると思われる。これは岡本ら[3]がいう「分離化」に相当する効果と推察される。

4.1.3 3週目～5週目までの行動分析

グループ作品作成では, 2週目に提供した NanoBoardAG を Processing から制御するサンプルプログラムが広く利用された。このサンプルは, NanoBoardAG1 台のみを制御できる。複数台同時に制御が必要なアイデアを提案するグループ (グループ10) の中で, 実際にサンプルプログラムを複数台同時制御に作り変えた学生がいた。その後, SA の誘導もあり, 他の2グループ (グループ2, 7) の学生が実装方式をその学生から学び, 各自で実装した。

また, ポケットミクを制御するためのサンプルプログラムは事前に提供されたが, PC 教室環境に合わせてコード

を書き換えなければならなかった。グループ2の学生がポケットミクの制御に成功し, 他の2グループ (グループ6, 8) にコードの書き換え方法を伝授した。このように, 知識, ノウハウやスキルがグループの学生間で伝播される光景が局所で見られた。

4.2 提出作品の分析と考察

4.2.1 1週目の課題分析

前述のように1週目は「ピクトグラム逃げ」のインタラクティブ性のあるゲーム風サンプルプログラムを提示した上で, 「3. Processing 基本課題」として自由テーマでプログラムを作成させた。課題提出のあった28人の作品中, 作品内容は表5のようになった。

表5 作品内容の分類

種別	人数	比率
インタラクティブ性のあるもの	14	50.00%
アニメーション	9	32.14%
動作不良 (エラー)	5	17.85%

例示されたピクトグラム逃げと同様のインタラクティブ性のある作品を作成した学生が50%であった。

また, 例示したサンプルプログラムと学生作品のクラス内容などの差異は表6のようになった。

表6 提出プログラムとサンプルプログラムの差異

種別	人数	比率
サンプルと同じ構造	3	10.71%
サンプルを元に改変したもの	11	39.29%
独自構造のもの	14	50.00%

サンプルプログラムと全く同じ構造及び少し改変した構造のものと, 独自構造で作成したものが50%ずつに分かれた。半数の学生が例示をもとに改変して作品作りあげているが, 半数は独自に作品を作成しようとしていることがわかる。さらに独自構造の作品の内訳を表7に示す。

表7 独自構造の作品の内訳

種別	人数	比率
基本動作を試す (アニメーション作成)	7	53.85%
インタラクティブ性の動作	2	15.38%
エラー	4	30.77%

基本動作を試す (アニメーション作成) はオブジェクト指向的構造というよりは, 図形描画自体を試した作品となっている。いずれもインタラクティブなものではなく, 画面端部で跳ね返るなどのアニメーションやシミュレーションのようなプログラムとなっており, オブジェクト指向的な作品ではなく「Processing によるプログラミング」から

学ぼうとしていることがうかがえる。

逆にインタラクティブ性の動作のある独自構造のプログラミングを行った 2 作品は独自のクラスおよび継承を行った作品で理解度が深いことを表している。

またすべての作品をクラス数と継承クラス数の差に着目して分類した内訳を表 8 に示す。

表 8 クラス数と継承クラス数の比較

分類	人数	比率
継承クラス数の方が多いもの	8	28.57%
クラス数の方が多いもの	6	21.43%
クラス数と継承クラス数が同数	7	25.00%
クラスがないもの	7	25.00%

75%の作品はサンプルプログラムや独自に考えた構造でオブジェクトを意識した作品作りを行っていることがわかる。継承数の方が多いものの中でも前述の独自構造でインタラクティブ性のある 2 作品はクラス数-継承クラス数が -5 と -11 でありオブジェクトの継承をうまく活用した作品を作成しており理解度が深いことが推察される。

これらのことから、短い実習時間であっても、わかりやすいサンプルプログラムの提示によって自ら思考したプログラムの作成が行えていることがわかった。また、サンプルプログラムで使用されたピクトグラムではなく、他のピクトグラムを使用し独自のテーマでピクトグラムから誘発されるイメージをもとにインタラクティブな動きのあるプログラムやアニメーションを作成しており、自ら創造するプログラムテーマを作成しやすくしている。

4.2.2 グループ作品

「8. グループ作品作成」で最終的に作成された作品について解説する。

はじめに、各グループの人数構成および、使用したハードウェア、ソフトウェアの一覧を表 9 に示す

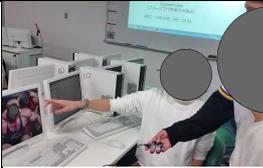
表 9 人数、使用ハードウェア及び使用ソフトウェア

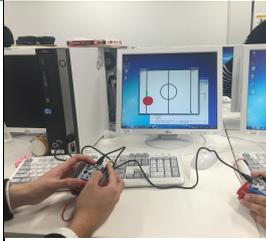
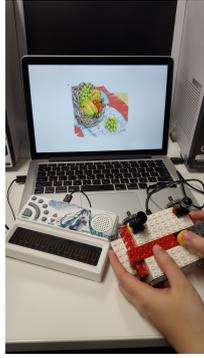
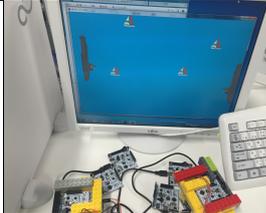
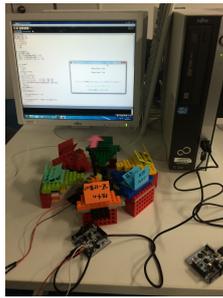
グループ番号	人数	ハードウェア	ソフトウェア
1	女性 2	LeapMotion	Processing, Julius
2	男性 2	NanoBoardAG ×2	Processing
3	女性 3	NanoBoardAG, LEGO®	Processing
4	男性 1 女性 2	LeapMotion	Unity
5	男性 4	NanoBoardAG, ウェブカメラ	Processing

6	男性 2	NanoBoardAG,, ポケットミク	Processing
7	男性 3	NanoBoardAG ×2	Processing
8	男性 2 女性 2	Web カメラ, ポケットミク, NanoBoardAG, LEGO®	Processing
9	男性 2	Web カメラ	Processing, OpenCV
10	男性 2	NanoBoardAG ×6, モーター×2, LEGO®	Processing
11	男性 1 女性 3	NanoBoardAG ×2, モーター, LEGO®	Processing

最終的に各グループが作成した作品のスクリーンショット及び概要説明を表 10 にまとめる。

表 10 作品のスクリーンショット及び概要説明

グループ番号	スクリーンショット	概要説明
1		Julius (フリーの音声認識ソフトウェア) で認識されたテキストを順次、手紙を模した白色矩形に表示して行く。スワイプの手振りを Leap Motion が感知すると手紙 (白色矩形) が画面上部に飛んで行くアプリケーション
2		画面上に女性が表示されており、最初は不機嫌そうに様々な音声をしゃべる。こちらからなだめると NanoBoardAG の音声センサが反応し次第に機嫌が直ってくる。また 2 台の NanoBoardAG のどちらが話したかで女性の向きが変化する。
3		釣りゲームを作成。NanoBoardAG と LEGO® で作成された釣り竿のリールを回転することで、NanoBoardAG のスライダが移動させることで、画面が遷移する。
4		街中にゴリラが大量に出現し、それを人間の手で実際に振り払うゲーム。手の動きは、Leap Motion を使って認識している。
5		ウェブカメラで撮影した画像とあらかじめ用意された画像を重ね合わせ、ユーザが顔を適切な位置に表示させるように移動する。すると顔出し看板に顔を はめ込んだ画像を保存出来るプログラム
6		ポケットミクを使って音声発話するプログラムを実行すると、近くに置かれている NanoBoardAG の音声センサに反応して、画面上のビクトグラムが踊る。

7		2 人対戦用のホッケーゲーム、それぞれのプレイヤーのバーを NanoBoardAG のスライダで操作する。
8		一見記憶力ゲームを装い、NanoBoardAG と LEGO® で作られたコントローラを操作すると次々に記憶すべき画像が表示されていく。途中突然ホラー画像が表示され、ポケットミクで生成された不気味な音楽が鳴り響くどっきりプログラム。
9		Processing と OpenCV を用いて、顔画像領域を抽出し、その上部にサンタの帽子を表示されるプログラム
10		NanoBoardAG を 6 台用いた 3 対 3 の対戦型海戦ゲーム。各チーム戦艦 1、戦艦に対応するコントローラはモータを活用した振動機能を有する。
11		LEGO® で作られたルーレット台に矢印が NanoBoardAG のモータを使って回転する。矢印が止まった所に記載されているお題で 2 人が会話する。会話する 2 人は NanoBoardAG の音声センサにより会話が一定時間止まると再びルーレットの矢印が回転し始め、次のお題が決定される。

5. アンケートの結果と考察

アンケートの結果と考察について報告する。

5.1 アンケート回答者属性

アンケートの回答者の属性は表 11 の通りである。これらの人数は、実際履修している学生数とは異なる。作品を提出し、アンケートに回答した人数である。

表 11 アンケート回答者の属性

学年	男子	女子
2	11名	6名
3	7名	6名
4	1名	0名

5.2 学習効果に関するアンケート結果

授業実施後のアンケート調査では、授業に対する満足度「この体験演習型授業には、全体的に満足している」と授業後の継続的な学習意欲「プログラミングについてもっと学習したいと思うか」「もっとものづくりをしたいと思うか」を調査した。回答は「4:とてもそう思う」「3:そう思う」「2:そう思わない」「1:まったくそう思わない」の4段階尺度を用いた。結果を表 12 に示す。満足度に関する設問の回答者 1 人あたりの平均は 3.52 と高い結果であった。

表 12 授業満足度

満足度	ものづくり		プログラミング	
	平均	SD	平均	SD
2年生	3.41	0.62	3.36	0.70
3,4年生	3.64	0.50	3.50	0.52
全体	3.52	0.57	3.42	0.62

5.3 学習コミュニティに関するアンケート結果

また、Rovai の Classroom CommunityScale をもとに学習コミュニティに関する調査項目を作成した[14]。回答は「4:とてもそう思う」「3:まあそう思う」「2:そう思わない」「1:まったくそう思わない」の4段階尺度を用い、以下 8 設問を実施した。回答の平均と標準偏差を示したのが表 13 である。

設問 1. 体験演習型（第 9 回から第 13 回）の授業では講義型（第 2 回から第 7 回）の授業より、クラスのメンバが互いの立場を尊重していると感じた

設問 2. 体験演習型の授業では講義型の授業より、クラスのメンバとのつながりを感じた

設問 3. 体験演習型の授業では講義型の授業より、クラスのメンバの心意気を感じた

設問 4. 体験演習型の授業では講義型の授業より、クラスのメンバに対して親しみを感じた

設問 5. 体験演習型の授業では講義型の授業より、他のメンバが自身をサポートしてくれるという確信がもてた

設問 6. グループの話し合いで他者の意見を聞いて、自分の意見を考え直すことがあった

設問 7. グループの話し合いでは、他者の考えと自分の考えを比べながら聞いた

設問 8. グループで話し合いをすることによって、自分の考えがまとまることがあった

表 13 学習コミュニティに関する認識

	問 1	問 2	問 3	問 4	問 5	問 6	問 7	問 8
平均	3.48	3.68	3.61	3.74	3.67	3.61	3.58	3.39
SD	0.51	0.48	0.50	0.44	0.54	0.50	0.50	0.56

5.4 自由記述によるアンケート結果

アンケートでは、今回の体験型演習に関して自由記述による感想も求めた。質問は、「他のグループの作品をみての感想を自由にお書きください」、「教えない」ということに対してどう思いましたか?」、「作品の相互評価を行って気が付いた点があれば記述して下さい」、「この授業に関するご意見、ご要望をお聞かせ下さい」の 4 項目である。自由記述のアンケート結果を表 14~17 にまとめる。

表 14 他のグループの作品をみての感想

自分が考えつかないようなアイテムの使い方をしている班が多かったので、非常に面白かった。
いろんなアイテムを用いた面白いアイデアがたくさんあって楽しかった。なのぼどでゲームが作れたり、LeapMotion と Unity で本格的な RPG が作れたりするのはすごいなあと思った。今まではプログラミングがどういうものに応用できるのかまいち実感がわかなかったけれど、今回の授業で体験できてよかった。
自分の班の作品がなかなかうまくいったように感じたが、ほかの班を見るととてもいいものがたくさんあったので、もっと手の込んだ作品にしたかったと思った。みんな、コミュニケーションというものをグループみんなで作っているような作品ができていたのがとても面白いなと思った。
できる人は今の時点でものすごくできるんだなと思いました。
すべてのグループがいろんなアイデアの作品を作っていてとても楽しかった。
それぞれの作品に特徴があり、参考になったし良い刺激をもらった。
リープモーションなどを使っているグループの発想が面白かったです。
発想が豊かなものが多く面白かったと思います、またいい刺激にもなりました。
自分には思いつかないような奇抜なアイデアがたくさんあり面白かった
レゴとの組み合わせを利用した班に面白さを感じた。
現実世界のもの、パソコンの画面の向こう側に反映させることをプログラミングで、しかも、子どもでも作れるレゴを使って行うところが面白かった。
どのグループもよくかけていたと思います。
言語や発想力の違いでここまでいろいろ違ったものができることに驚きました。
個々のグループでは画像を利用してその上にオブジェクトを乗っけるだけでしたが、もっともっと加工なんかができるといいのではないかと思います。
様々な視点からコミュニケーションをとる作品を作っていて、非常に面白かった。
ほかの班も自分たち同様にいろいろ工夫して作品作りしていたので自分たちとは違う作品の形をたくさん見られてよかった。
アイデアがすごく、技術がすごいと思った
とてもレベルが高かった。
オープンソースをうまく使えば作品の幅が広がったのかな、と思いました。
グラフィックにもかなりこだわっていると思いました。
またツール・コンテンツ自体もこだわっておりよく作られたものが多か

ったです
みなさんとでもアイデアと技術がすごいと思った。
コンセプトがはっきりしていて、より特殊なものをもっている作品が、より面白く思えた。
自分たちでは思いつかないようなアイデアが沢山あってよかった。ゴリラの班はゲーム性だけでなくストーリー性もあってよかった
みんなアイデア力があって面白い
正直みんなそこまでプログラミングはうまくないと思うけどこういう作品においては考え方を少し変えて違う観点から作ることが大事だしほかの人には思いつかない作品になっていた
やはり発想力がすごく大切だなと感じました。簡単なプログラムでも見せ方や、普段と違う遊び方によって面白く感じる場合が多い。
先生やTAから何も教わっていないのに、そこまでできるとはすごく不思議に思いました。中に3Dゲームを作ったグループもあったので、実際のオンラインゲームのように遊べたのでとてもいいと思いました。
OOPを理解して、コードをかけるだけで様々な作品を作ることができたので、技術的なハードルが低く感じられた。
ストーリー性のある作品は強いと思いました。
webカメラやLEAP MOTIONなどを使った作品を作っていたのが、すごいと思いました。メール送信の機能の作品はすごく良いと思いました。
なのボードとモーターしか使わなかったのですが、ほかの機器を使って面白そうなのが一杯あったのでいろいろ使ってみたいなと思いました。
よくこんな発想が出来るなあと感じる作品がたくさんあった。
発想が自分たちとは全然違うものばかりでとても刺激を受けました。発表会がとても面白かったです。自分たちが使っていないアイテムを使っているところも見れて感心するばかりでした。
チュートリアルまで作っている作品もあり、凝っていると思った。笑いを誘うやり方やゲームと一緒にやって楽しさを共有するやり方もあるのだなと思った。
リープモーションを使った作品などは、どうやってプログラミングを組んだのかも分からないし、すごく憧れました。他にも、面白いアイデアの作品がたくさんあって刺激を受けた。

表 15 教えないことへの感想

少し慣れていました
自分の考えとして、わからないところをネットで調べながら解消していくのは、「自分で考える」ということを補助すると思うので絶対に必要だと思います。
しかし、考えた上でわからないというものに関しては教えがあっても良いと思います。
的確なアドバイスをくれたことで、アイデアが作品にまとまったのでよかったです。
自分なりに調べようという気はおこったが、やはり、期限があるため、教えないで調べるのに時間がかかってしまっとうまくいかないことがあるのであまり肯定的には思わなかった。
情報系らしいなと思いました。
苦戦しました。
が、メンバーと試行錯誤しながら自力で作品を作れたときはうれしかったです。
でもヒントはくれたのでそこまで気にならなかったです。
1からというのは難しいものはあるが、一番の勉強だとは思う
自分の力で考えなければならぬので、力試しのように感じた
あえてこうしろと言われないことで自由性があったところが班の個性につながっていたと考える。
教えられないということで迷うところも多かったが、結果として面白さにつながっていたのでよかったのではないかと考えている。
もっと知りたいと思えるようになりました。
人が面白いと思えるような企画を練りたいと考えていたので、分担をよく考え、チーム内で自分ができるようなことを取り組みました。
自らが考えるという技術を習得できることはとても良いと思いました。
最初は不安だったが、やってみると何とかなるものだと感じた。
自分たちで考えなければいけないことがたくさんあり、コミュニケーションをたくさん取れた。
自分で学ぶ力がつくし、絶対に忘れない気がした
自分で考える良い機会になった。
初めのうちはとても苦労したが、自分で調べて解決する力が少し身についたのではないかと思います。
自分でやらないと出来るようにならない分野なので、そうであるべきだと思います。
個人的にはやりがいのある講義だと思った。
自分で考えることの楽しさと苦しさを味わえてよかったと思った。
モノ作りには答えがないので答えを提示するよりも自分たちで見つけた

ものが重要だと思った。
少し冷たい
自分から調べるのはとても大事だけど、人によっては考えが偏るから質問にはある程度応じてほしい
常に自分で答えを探す力を養うことができ、とてもよかったと思います。
自分で考えて学ぶのもいいと思うが、学ぶのに時間がかかるので、やはり先生が直接知識を教えたほうが時間の短縮ができると思います。
目的に対して、自分の思いつかない、もっと簡単な方法があったときは、教えてもらえるとよかった。
自分だけでやると、自分のわかる範囲で作ろうとしてしまった。
ただ、複数人でやると、お互いの得意・理解が深いところを説明し合っ、モチベーション高く解決しようとした。
自分なりに考えるのは筋道を自分で立ててはいけなかったのが難しかったです。
自分でなんとかしようとするのは面白かったです。ただ、私はいくらやってもプログラムがうまくいきませんでした。
悩むことでプログラムの能力が上がるということは日々実感しているので悩ましてくれるいい授業だと思いました。適時サポートしてくれたので助かりました。
ヒントを教えてくれることがあったが、自分で考えることが多く、苦労したことが多かった。
難しいけれど試行錯誤する中で新たなアイデアが生まれたりして、自分たちで考えるということもとても大事だと思いました。でもとても大変だと思いました。
自分で考えなくてはいけなくて真剣に取り組まなきゃいけないと思えて、やる気が出た
もうすこし教えてほしいと思うことが多々ありました。

表 16 作品の相互評価で気付いた点

評価に関しては特に思ったことはないですが、様々な作品を見て、「この作品を作ってみたい!」や、「この作品のここを拡張したい!」と思ったものがあって、この先も沢山いじってみたいと思います。
自分たちが作ったものをほかの人に認めてもらえるのがとても嬉しかった。ほかのグループの作品を見て、これからのプログラミングに生かしていきたいと強く思った。とても楽しかったです。
ほかの作品をみて、考えが改まるころがあった、とてもよかったと感じている。しかし、感想を伝えてあげると評価のみでよかったのではないかと思つた。
経験者の人と、未経験の人での力の差がはっきり見えました
もう少しゆっくりほかの人の作品を見たかったです。
かきこまった講評ではなく、率直な意見が聞けたのでよかったです。
ほかのグループの作品もとても面白かったです。
コミュニケーションの取り方が各班で考え方が違うなと思つた
他者が自分の作品に対して反応してくれることがうれしかった
評価してほしいと思つて全力で取り組んだ部分が評価されていて伝わるものだなあと感じた。他の作品については、独創性の高いものが評価されているように思つたし、私自身も独創性の高いものを評価したいという風を感じた。
どのチームもよくまとめられているなと思いました。
テーマが抽象的だったので、これだけの一人一人の考え方の違いに驚きました。
プログラミングが固いイメージが強かったので、もっといろいろやってみたいと思えました。
単純にオブジェクトを乗っけるだけでなくそれ自体を加工することが大切だと感じました。
実際に評価されることで、作品の質をより高めようという向上心につながったと感じる。
発想というものは重要であると感じた。
ほかの班が自分よりすごいと認めねばならないことがあった
現実の厳しさを知りました。
ほかの人の作品を見ることによって、自分たちの作品に何が足りなかったのか考えさせられました。
全体として適切な評価がなされていたように感じました
みんなすごい
ほかの班の作品のレベルが高い
人によって面白いと感じたり感じなかったり様々なので正解なものはいくつかあるのだと感じた。
人の好みはだいたい変わると思つた
ここを押してるんだけどここが面白いという意見もある
より改善できる部分を気が付けました。
ないです
最初は、自分たちのできる範囲で作品を考えているところが多かったと思うが、次第に、自分たちが本当に面白いと思うものをつくるということに変わっていったのは面白かった。

他人の作品にリスペクトを持つことができました。
いろんな人に作品を見てもらって、いろいろなことを言ってもらえたのでよかったです。
特になかったです。
一番多く投資をしてくれる人がいると、やった甲斐があったなと感じた。
ほかのグループの方や先生に見てもらって、新たな進歩のためのヒントをもらったり、質問されたりして自分たちでは気づけなかったことも気づくことができ、相互評価はとてむためになりました。
自分たちは自分たちの作品に対してそこまで自信がなく、ほかの班がすごいなと感じていたが、ほかの人たちからの評価が思ったよりもよく自信が持てた。
他の人の作品を深く知ることができた。
班番号がすぐにわかるようにしてほしいと感じた。

表 17 授業に対する意見, 要望

デバイスの中にスマートフォンを混ぜたら面白かったかもしれません。
定期的に課題が出されて、グループワークもあって、とてもやる気になりました。
何かを作り上げたり、学ぼうとすることに対する姿勢がよくなる授業だとは感じた。この雰囲気の授業がもっといろいろな授業に生かさればとてもいい大学生活をすごせると感じた。
他人と協力してものをつくれるのは大学だからできることですのでいい経験になりました。ありがとうございました。
楽しかったです。
一年のときの体験演習のように面白かったです。
最後のグループワークでいろいろ考えて作るのが楽しかったです。もっとLEGO やっていたかったです。
初めての体験ができて楽しかったです
いろいろなことが出てきて楽しかったです。ハチャメチャなもの作ってすみませんでした。
自分はプログラムが苦手で、どのようにグループ内で動けるか、自分の立ち位置を考えました。 モノづくりができる職に就きたいと考えているので、企画がモノとして動く姿が感動しました。 ありがとうございました。
もっと長い時間をかけて実習を行うとより完成度の高いものができるのではないかと思います。
難しい授業だが、非常に役に立つ授業であったと感じた。
退屈しない授業であった。
プログラミングがもっとできればすごく楽しいと思う もっと知識を増やしたいと思った
やはり難しいけど得られることは多かった。
楽な時間ではなかったけれどその分成長できたのではないかと思います。
個人的には講義型でプログラミングをする方が楽しかったです。 直接何かの役には立たないかもしれませんが、モノをつくるよりもプログラムの流れなどを頭を使って考えることが、個人的には好きなためです。 ただ、グループでプログラミングすることで複数人でのプログラミングの大変さやコメントをすることの重要性などに気付けたように思います とても楽しかった。
なのぼーどのよさが再確認できてよかったです。
web カメラやポケットミクなどのマニュアルがないのでそこに時間がかかってしまった
授業難しいいいです
特になし。
ないです
自分たちで実際にものづくりできたのは良かったと思う。
使えるアイテムを増やして欲しいです、(kinect や romo など)
プログラミングが苦手なんですけど、グループワークではプログラミングではない部分で頑張れたのでよかったです。
いい授業だと思います。
講義形式の授業ももう少しやってほしかった。
グループでみんなの力を合わせて一つのものを作るというのがとても楽しかったです。充実していました。
自分で考えて学ばなくてはいけない授業でとても楽しかったです。
特になし。

5.5 アンケート結果の考察

表 14 の「他のグループの作品をみての感想」では、基本「アイデア」「発想」といった、創造性に関する回答が大

半を占めたが、特定のハードウェアやソフトウェアの視点からの言及も目立った。これは、目にしたことのない未知のデバイスに対する興味関心や、子ども時代に慣れ親しんできた LEGO に言及している回答もあり、このようなデバイスの使用が学生に強く意識付けされると考えられる。

表 15 の教えないことへの感想では、自分で考える力や調べる力がついたという意見をはじめ、試行錯誤するというプログラミングで大切な要素に関する意見も見られ、肯定的な感想が大半を占めた。無下に全くコミュニケーションを取らないのが適切な訳ではなく、どのように「教えない」でコミュニケーションをとりながらアドバイスを行うかは授業担当者や、SA 側として状況や学習者の能力を見極めながら対応していく能力が要求される。

また、表 16 の作品の相互評価では、他者との比較がまず見られた。上方社会的比較により、より高い目標や向上心が発生しているのわかる。また他者から作品の評価を受ける事で、自己有用感や自己肯定感の高揚に繋がっている回答も見受けられる。

表 17 の「授業に対する意見, 要望」でも、特定のハードウェア、ソフトウェアに言及して意見や要望を述べる学生が何名かいた。また、1 年次の「社会情報体験演習」の振り返りに関して言及する学生もおり、振り返りを目的としたカリキュラム設計の意義は認められる。

少数ではあるが、講義型の授業や指定された内容の課題を望む者、プログラムの処理の流れやアルゴリズムについて強い関心を表明する学生もいる。自由度が高いグループ活動であっても、このような学生に適切なテーマ設定を事前に想定し、必要に応じて設定する準備も必要である。

そもそもグループで活動する事自体が苦手な学生もいるので、今回は 2 人以上のグループでという制約を課したが、本取組みでは、Web カメラ、NanoBoardAG、LEGO などのハードウェアは十分な数量が用意されているので、一人のグループを許容し、また複数人のグループ活動においても あえてグループを離脱させ一人での作業に変更させるなど柔軟な運用も考えられる。これは、プログラミング関連授業に関わらず広く言えることで、グループワークが不得意な学生や、消極的な学生に対する配慮や対応可能な授業設計も今後の課題である。

6. まとめと今後の課題

本稿では、学際系学部におけるプログラミング授業設計として、過去体験に着目した横断型プログラミング授業設計を行なった。学生の興味を惹き、アイデアを引き出せるよう、センサーボードをはじめとした様々なアイテムを提示し、自らのアイデアを形にした同学部 1 年次必修科目「社会情報体験演習」の過去体験を想起させる工夫を施した。

今回の授業で、新たな課題や改善すべき点も発見されたので、それらの点をふまえて 2016 年度以降のシラバスに反映

させる予定である。また、他の科目群も加えた一層の横断的な授業設計をさらに進めていく予定である。

謝辞 本研究はJSPS 科研費 26330140 の助成をうけたものです。また Scratch「ネコ逃げ」プロジェクト作者の阿部和広先生, NanoBoardAG の開発者である新村とも様に深く御礼を申し上げます。授業の運営をサポートしてくれた SA 諸氏に感謝いたします。

参考文献

- 1) 柴田望洋:明解Java入門編,ソフトバンククリエイティブ(2007)
- 2) 高橋麻奈:やさしいJava,ソフトバンククリエイティブ (2013)
- 3) 岡本雅子, 村上雅之, 吉川直人, 喜多一:「視覚的顕在化」に着目したプログラミング学習教材の開発と評価, 日本教育工学会論文誌, Vol.37, No.1, pp.35-45 (2013)
- 4) 土肥紳一, 宮川治, 今野紀子:初学者におけるオブジェクト指向プログラミングの難しさの分析と教授の工夫,情報教育シンポジウム論文集,Vol. 2012, No. 4, pp.21-28 (2012)
- 5) 水口充:Java 言語演習科目における対戦型ゲーム課題の設計と実践, 情報教育シンポジウム論文集,Vol. 2013, No. 2, pp.233-240 (2013)
- 6) 伊藤一成, 新目真紀, 阿部和広:世代や組織を超えた相互学習を促進するためのプログラミング導入教育の実践報告,情報処理学会研究報告,コンピュータと教育,CE116,pp.1-7 (2012)
- 7) 吉田葵, 伊藤一成, 阿部和広:ものづくり体験を通じたプログラミング授業の設計と評価, 情報処理学会研究報告,コンピュータと教育,CE134,pp.1-10 (2016)
- 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) 太田幸夫: 国際安全標識ピクトグラムデザインの研究
<http://www.tamabi.ac.jp/soumu/gai/hojo/seika/2003/kyoudou-ota1.pdf>
- 10) E.P. Glinert, S.L. Tanimoto: Pict: An Interactive Graphical Programming Environment, Computer, vol. 17, no. 11, pp. 7-25 (1984)
- 11) Kazuhiro Abe, Tomo Niimura, Koji Yokokawa, Kazunari Ito, Daisuke Kuramoto:NanoBoardAG: An Inexpensive Sensor Board Compatible with PicoBoard and WeDo, Scratch@MIT (2012)
- 12) 鎌田敏之, 大村基将, 兼宗進:「プログラムによる計測と制御」概念形成を意識したコースカリキュラムとその評価,情報処理学会研究報告,コンピュータと教育,CE131, No.10, pp.1-15, (2015)
- 13) Butterworth, David T:Teaching C/C++ programming with lego mindstorms. Proc. 3rd Int. Conf. on Robotics in Education (RiE 2012), Prague, Czech Republic. (2012)
- 14) Rovai, A. P. : Development of an instrument to measure classroom community, Internet and Higher Education, Vol. 5, pp.197-211 (2002)