

文科系大学の情報科学教育における試みと考察

伊 藤 則 之*

1. はじめに

本来理科系科目であるコンピュータ基礎や情報処理論などの情報科学を、文科系大学の学生が理解しやすいように教えることは容易ではない。1999年から高等学校で教科「情報」が取り入れられているが、大学に入学した時点での情報についての知識のレベルは、出身高校毎に異なる。また、高校で情報についてある程度勉強している状況はあっても、文科系大学において、90分の講義形式授業の中で、コンピュータや情報処理の理論だけを説明したのでは、学生にはなかなか内容を理解してもらえない。理科系の学生はもともと科学的なものに興味を持っているが、文科系の学生は科学的なものを敬遠する傾向がある。こうした文科系の学生に、情報科学などの理科系の授業を理解してもらう前に、まず興味を持ってもらうことが重要である。

そこで、筆者が文科系大学で担当するコンピュータ基礎および情報処理論の授業の中で、いくつかのフリーソフトを実際を使って、コンピュータのハードウェアの仕組み、ソフトウェア開発手法、情報処理の具体的方法についての教育を試みた。ただ話して終わる授業ではなく、学生が実際にソフトを使って授業内容を確認することによりどのような効果があったのか、また講義形式の授業の中で行うこうした取り組みの課題は何なのかなどについて、これまでの試みを振り返りながら考察する。

2. 講義形式授業の双方向型・参加型への取り組み

文科系大学で理系科目を担当する教員の悩みは、題名の中に「文系学生」や「文系学科」などの言葉が使われている教育関係論文からも読み取ることが出来る¹⁻⁷⁾。

たとえば、文系における理科系科目の教育に関する論文²⁾では、下記のような記述がある。

- ・文系学生は一般に理科系科目に関心がなく、授業に引きつけるのは非常に難しい。
- ・学生は、原理的・抽象的な話よりは、実際の・現象的な話に興味を持ちがちである。
- ・学生の眠気を吹き飛ばすだけの迫力ある講義が期待される。

また、文科系・理科系に関係なく、教師が一方的に話をして板書する講義型の授業の問題点が指摘されるようになり、教師と学生が授業の中でコミュニケーションを持つ双方向型授業やディスカッションなどを取り入れた参加型授業が行われるようになってきた。理科系科目では、講義形式の授業ながら、MITのWalter Lewis教授の物理学の授業のように、講義内容を実験で確かめることのわかりやすさも注目を集めている⁸⁾。Walter Lesis教授の物理学の授業の中では、振り子の周期が重さや振幅に依存しないことを説明したあと、教壇で自ら振り子になって実験している。

3. 試みおよび考察の対象となる授業内容

* 広島経済大学経済学部教授

経済学部において筆者が担当した科目の中

で、情報科学教育における試みと考察の対象にした科目は以下の通りである。いずれの科目の授業も、約150人を収容出来る教室において、講義形式で行われた。また、この教室には1人に1台のパソコンが机上に備えられている。

[前期]

- コンピュータ基礎 I
コンピュータのハードウェアの原理を教える授業
- 情報処理論 I
情報処理のための基本的手法および実適用例を教える授業

[後期]

- コンピュータ基礎 II
各プログラミング言語に共通するソフトウェア開発技法を教える授業

4. 一年目前期

4.1 コンピュータ基礎 I

第6回目の授業の冒頭で、第1回から第5回までの授業内容についての理解度確認テストを抜き打ち的に実施した。50点満点の選択式のテストであり、60名の学生が受けた。点数の悪い方から並べてみると、図1のように6点から47

点まではほぼ一様に分布している。

このように理解力に大きなばらつきのある学生になるべく授業内容を理解してもらうために、説明資料ではプログラム実行の流れや回路の動きをアニメーション風にした。アニメーションの中には出来るだけ文字による説明も入れることにした。一度説明したあと、各学生にスライドショーとして自分のペースで見てもらった。このようなかたちの授業を15回行い、毎回の出席カードには感想とともに各自の理解度を下記のような5段階で記入してもらった。

- 5：よくわかった
- 4：だいたいわかった
- 3：なんとなくわかった
- 2：あまりよくわからなかった
- 1：まったくわからなかった

記入なしの学生もいるが、記入してもらった15回分の理解度の分布は図2のようになった。毎回の授業のあと、この分布を見ながら次回の授業内容のレベルを調整したが、図2の線から下にある理解出来ない学生の割合は意図したようには減らなかった。

図2の15回分の理解度の分布について、各回の平均をグラフ化すると図3のようになる。理

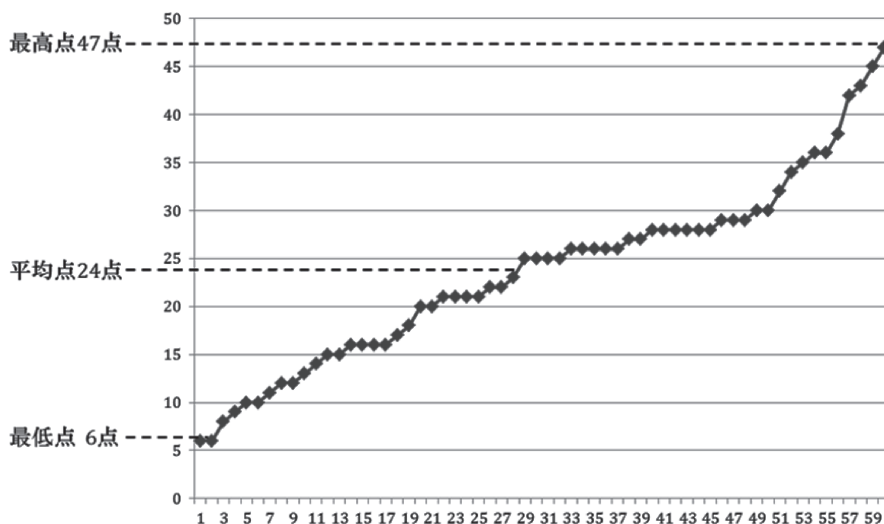


図1 学生の理解力の分布

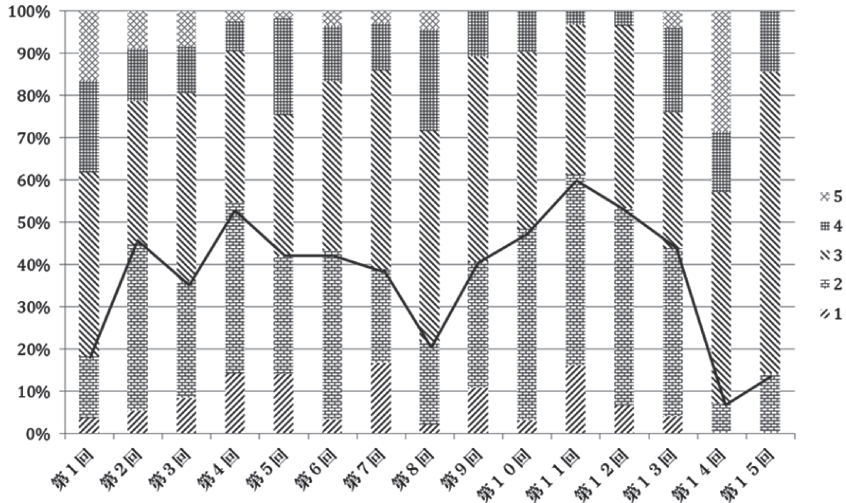


図2 15回の授業の理解度の内訳

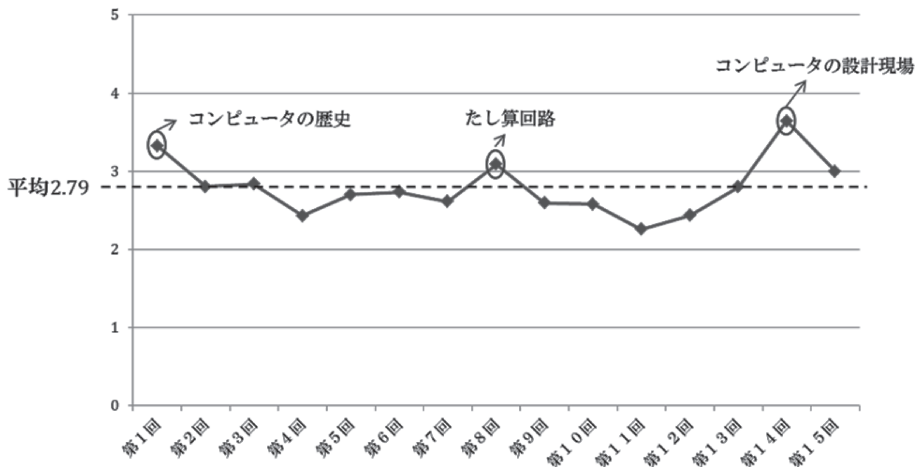


図3 15回の授業の理解度平均の推移

理解度の平均が高いのは、コンピュータの歴史やコンピュータの設計現場といった物語的な内容の場合や、たし算回路のように馴染みのある筆算をベースにした内容の場合となった。

4.2 情報処理論 I

情報処理論 I でも、授業は説明にアニメーションを取り入れたかたちで進めた。情報処理論 I では、データの構造を表現する方法とそこから情報を取り出す手順が重要となる。この手順を実現するプログラミング手法の中で、プロ

グラムの中で自分自身を呼び出す再帰プログラムというものがある。この再帰プログラムは情報処理においてはとても重要な項目であるが、理解することは難しい。この再帰プログラミングを説明する授業では、図4のようにデータ構造の各データ項目に1人の学生を割り当て、各学生に同じ指令書を渡し、制御と書かれた紙が回ってきたら持っている指令書にしたがって行動することで処理が進むという実験を行った。

この実験のあとで授業に出席した学生が書いてくれた感想の主なものを図5に示す。実験に

参加した学生は理解出来るようになったが、参加していない学生にまだ理解が難しいことがわかった。

情報処理論 I でもコンピュータ基礎 I で行っ

たデータ集計と同じことを行った結果は、図6、図7および図8に示す通りである。情報処理論 I のデータはコンピュータ基礎 I のデータとほぼ同じ傾向を示している。理解度の面で

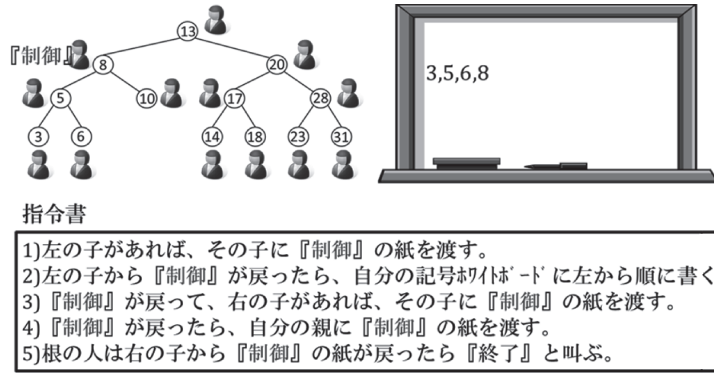


図4 学生複数名による再帰プログラムの実験

肯定的感想

- ・実験がわかりやすかった。
- ・実験とてもわかりやすかったデス。
- ・前でやっているののでわかりやすかったと思います。
- ・実験で動くと、二分木の再帰的プログラムがどういう物かよく分かった。
- ・人を使って番号を整理することによって分かりやすかった。
- ・実験は選ばれてめんどうだったが、参加したことで理解できるようになった。参加してよかった。

否定的感想

- ・実験は時間がかかる割にはあまり理解に役立っているとは思えなかったです。普通の説明でも十分だと思いました。
- ・実験はやっている人は分かりやすいと思うけれど、やっていない人は良く分からないんじゃないかなと思います。

図5 再帰プログラムの実験への感想

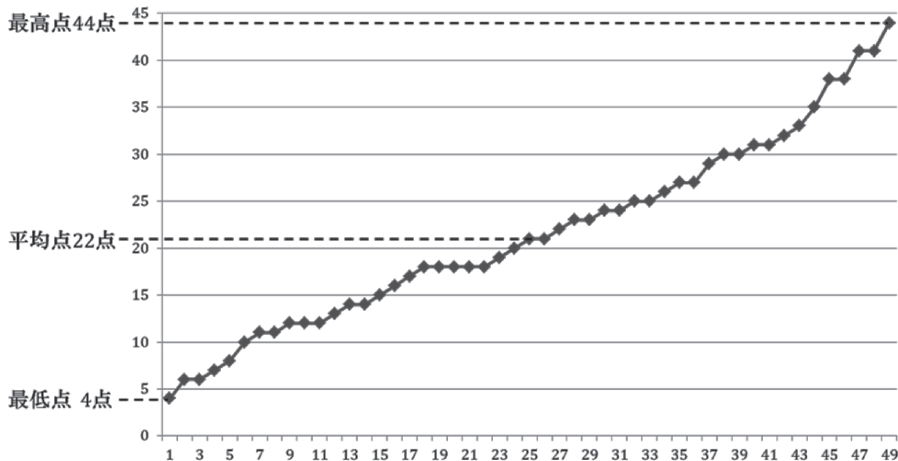


図6 学生の理解力の分布

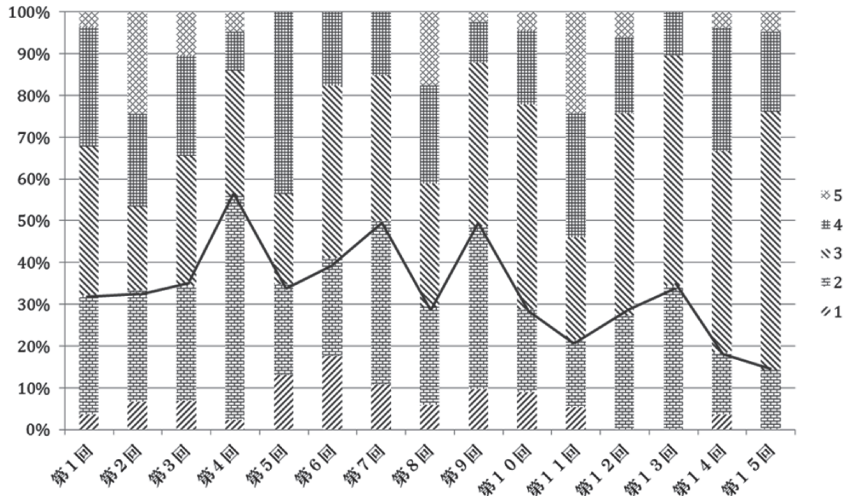


図7 15回の授業の理解度の内訳

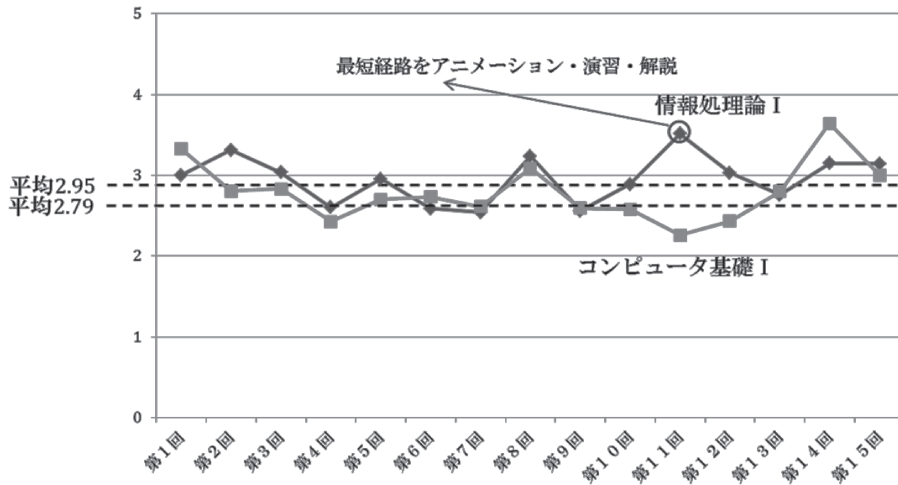


図8 15回の授業の理解度平均の推移

は、コンピュータ基礎 I より多少高くなっている。これは、コンピュータ基礎 I での 2 進数演算や論理回路という内容より、情報処理論 I での情報を処理するソフトウェア的な内容のほうが多少理解しやすいということが要因と思われる。また、情報処理論 I の第 11 回の理解度平均が全体で一番高いが、この回では最短経路問題をアニメーションを使って詳しく説明し、授業中に演習問題も解いてもらい、それを解説したことによるものと思われる。

4.3 授業アンケート結果とその対応

大学教員一年目の前期の授業に対する学生の評価は厳しく、学生が記入したコメントの概要は下記の通りである。

- ・面白さが伝わらない
- ・ひたすら先生がしゃべる
- ・プリントが難しすぎる
- ・授業が形式的
- ・ペースが速い

そこで、前期授業が完全に講義型の一方通行の授業であったことを反省し、後期の授業に向

けた改善として、下記の取り組みを実施することとした。

- ・パワーポイント資料とは別に文字で説明した資料を配る
- ・内容は、網羅するより、重要なポイントに絞る
- ・学生にパソコン上で作業をさせる

5. 一年目後期

5.1 コンピュータ基礎Ⅱ

パワーポイント資料は学生がアクセス出来る授業関連の Web サイトに授業資料として登録し、授業では配布はしないこととした。そのかわり、文字で説明した図9のような資料を配布して復習しやすくした。

コンピュータ基礎Ⅱではソフトウェア開発手法を教えるのであるが、第1回目の授業でプログラミングの経験のある人に挙手してもらったところ、数名しかいないことがわかった。ソフトウェア開発手法を学ぶには、少しでもプログラミング経験がある方が理解しやすい。

そこで、毎回の授業の最後の20分から30分を使ってプログラミングも教えることとした。67名の履修登録者がいるため、簡単なプログラミング言語を使って短時間でスムーズに進める必要がある。そのために、MITのメディアラボが開

発した初心者向けプログラミング言語 Scratch⁹⁾を使うこととした。

Scratch はパソコンへのインストールが不要であり、学生が各自のドキュメントフォルダに1つの Scratch フォルダをコピーするだけで利用出来るため、教育現場で適用されている例もいくつかある^{10,11)}。起動した画面では、図10のように、背景や登場キャラクタをまず設定し、これらに対してプログラミング部品から必要部品を取り出してプログラミングする。実行すると、ステージ上でキャラクタがプログラム通りに動く。

毎回の授業の最後で行う Scratch を使った実験に対する学生の感想は、図11のように肯定的なものがほとんどであるが、プログラムの入力ミスで意図したようにキャラクタが動かない学生もおり、そのことを感想に書く学生も毎回数名いた。うまく動かない学生には挙手をさせて個別に原因を指摘したが、挙手せずに自分で解決しようとして時間切れになる学生もいたので、こうした学生への対応が課題として残る。

コンピュータ基礎Ⅱでも、コンピュータ基礎Ⅰで行ったものと同じ理解度データの集計を行った。結果は、図12および図13に示す通りである。理解度の面では、コンピュータ基礎Ⅱで取り入れた新たな施策により、コンピュータ基

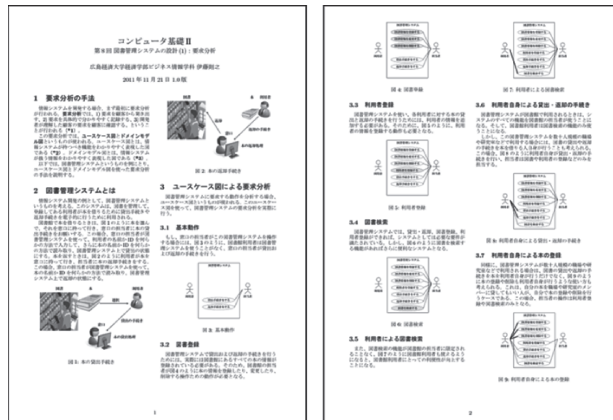


図9 文字を入れた配布資料



図10 Scratch を使ったプログラミング

- ・プログラミングは話を聞くだけより、実際にやったほうがすごくわかりやすかったし、おもしろかったです。
- ・授業内容とプログラミングへの自分のイメージの差があったけど、わかりやすかった。
- ・プログラミングで様々な動きをさせられるのがおもしろかった。
- ・かんたんなソフトですが、奥が深いようにも見えました。たのしみです。
- ・少しパソコンにふれることができて楽しみながら授業を受けることができました。
- ・自分でキャラクターを簡単に操作できて楽しかった。もっといろんな動かし方を学びたい。

図11 Scratch を使った実験への学生の感想

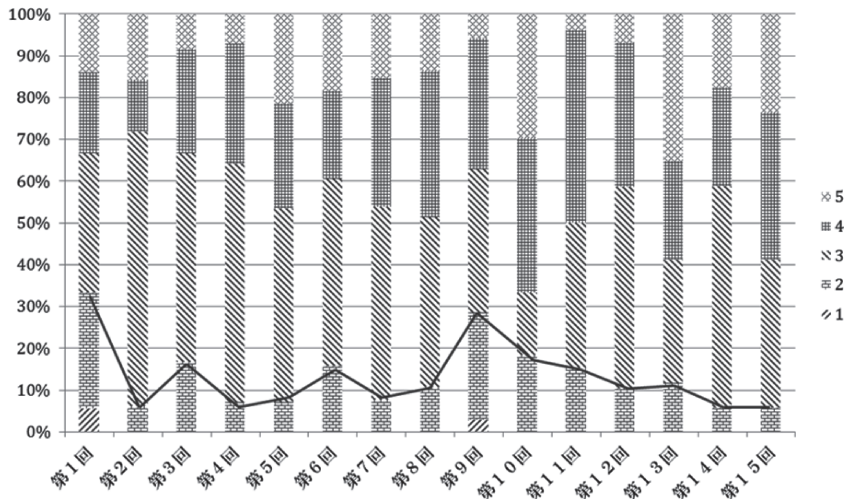


図12 15回の授業の理解度の内訳

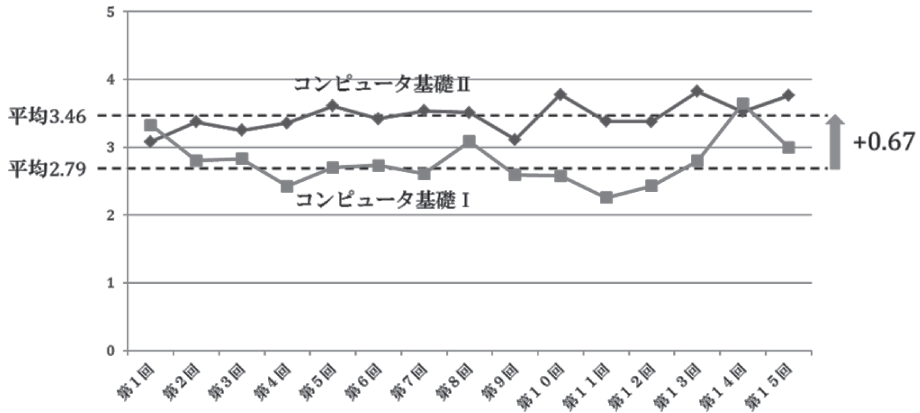


図13 15回の授業の理解度平均の推移

礎Iより平均で0.67だけ高くなった。これは、コンピュータ基礎IIではソフトウェア開発に関するものであるため、チームによる開発方法やプログラムに誤りを入れないための行動など、人間が多く登場する内容であり、理科系の内容が少ないことが大きく影響していると考えられる。また、新たに取り入れた施策による効果もどの程度かを切り分けることは出来ないが、要因として含まれていると思われる。

5.2 一年目の総括と二年目での取り組み

前期の反省に基づき、後期で実験を取り入れたことにより学生の興味を引き出すことは出来た。ただ、課題として、ソフトウェア開発手法の講義部分と実験との強い関連付け、そして講義部分の理解度をさらに上げる施策の実現というものが明らかになった。そのため、二年目では、その日のうちに授業内容を再度振り返って確認させ、どこがわからないのかを学生および教員が把握することとした。図14は、実験の導入も含めた二年目の施策の一覧を示したもので

		コンピュータ基礎I	コンピュータ基礎II	情報処理論I
2011	前期	<ul style="list-style-type: none"> スライドショーによるアニメーション 		<ul style="list-style-type: none"> スライドショーによるアニメーション 学生による再編プログラムの実験
	後期		<ul style="list-style-type: none"> 初心者用プログラミング言語Scratchによる実習 	
2012	前期	<ul style="list-style-type: none"> 疑問点の把握とその解消 アニメーションの改善 実験の導入 <ul style="list-style-type: none"> CPUエミュレータ 論理設計・検証ツール 授業内での復習と理解度確認 		<ul style="list-style-type: none"> 疑問点の把握とその解消 アニメーションの改善 実験の導入 <ul style="list-style-type: none"> ハノイの塔,HSP, アルゴリズムク,オセロ 授業内での復習と理解度確認
	後期			

図14 二年目の取り組み

ある。

6. 二年目前期

6.1 コンピュータ基礎 I

6.1.1 疑問点の解消と理解度の確認

どこがわからないのかを学生および教員が把握するために、毎回の授業の最後に図15の左側のような書式で疑問点と感想・要望を記入してもらい、また授業の難易度も5段階で評価してもらったこととした。疑問点については、次の授業の最初で図15の右側のような文字にした資料を配布して疑問点として挙げられた項目につ

いて解説を加えた。

授業の理解度の向上と把握のために、図16の左側のような確認問題を授業の最後に記入させることにした。その日のうちに授業内容を再度振り返ることにより記憶を定着させるとともに、どこが理解出来なかったのかを教員側が把握出来る。この確認問題を採点添削して、5回毎にまとめて学生に返却して、定期試験前に復習してもらうこととした。

6.1.2 GMC-4 コンピュータ

コンピュータ基礎 I でコンピュータのハードウェアを学ぶ際には、コンピュータの機械語を

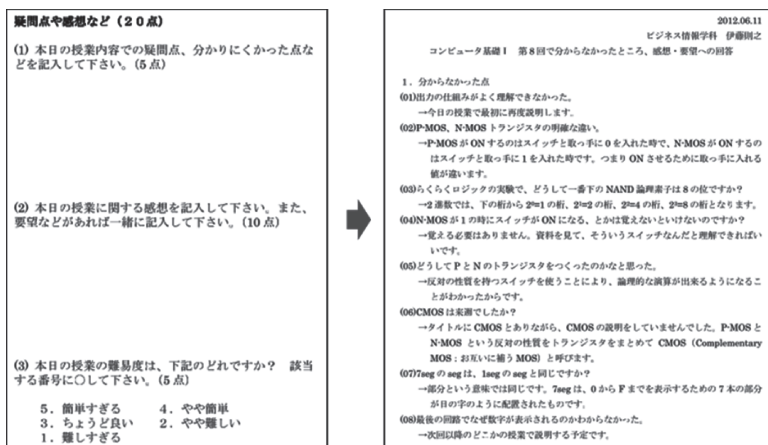


図15 疑問点の把握とその解消

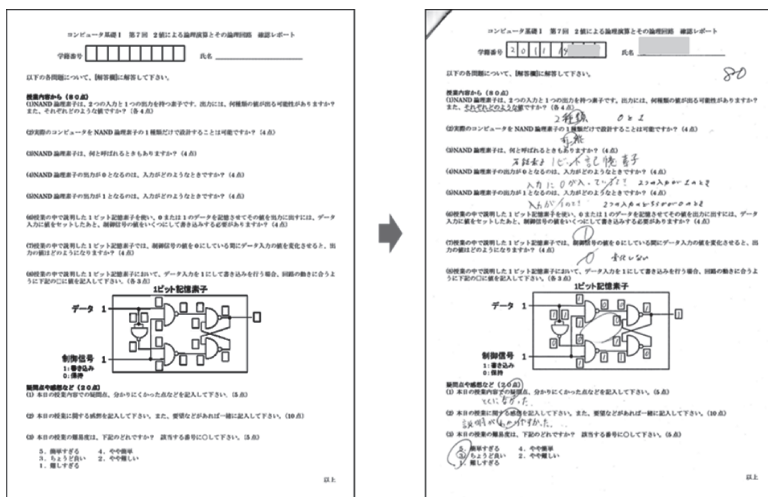


図16 授業内での復習と理解度確認

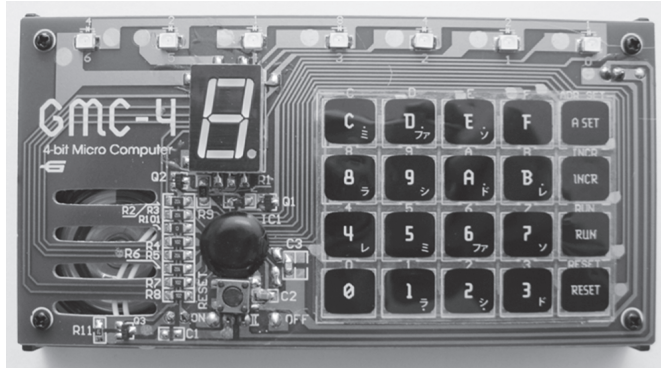


図17 GMC-4 の実物

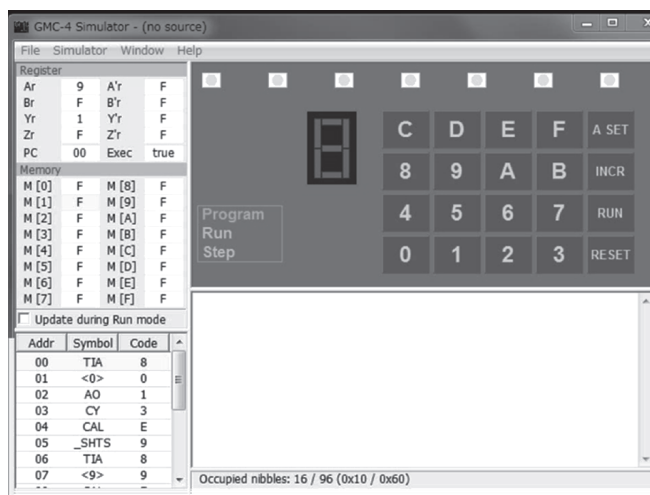


図18 GMC-4 のエミュレータソフト

理解することが必要となる。一年目ではコンピュータサイエンス分野において有名な教育用プロセッサである DLX¹²⁾ というものを題材として、その機械語の一部を学生に教えた。

しかし、授業の中で学生が実際にプログラミングして動かすことが出来ないのが、学生にはかなり難しいという評価になった。そのため、二年目では、大人の科学マガジンという雑誌¹³⁾ の付録になっている4ビットコンピュータである GMC-4 という図17のコンピュータを題材として使うこととした。

この実物の GMC-4 を各学生に配布して操作させることが理想であるが、この雑誌は2,500円であるため、この GMC-4 をソフトで実現して

いる図18のような GMC-4 エミュレータ¹⁴⁾ というフリーソフトを使ってもらうこととした。このソフトもパソコンへのインストールは不要であり、あるフォルダを各自のドキュメントフォルダにコピーするだけでよい。このエミュレータを使って、学生には様々なプログラムを機械語で記述してもらい、それをこのエミュレータに実物と同じ操作法で入力してもらい、動作を確認させた。

GMC-4 のエミュレータソフトを使った授業に対する学生の感想は、図19のように肯定的意見とともに否定的意見もあり、全員が楽しいと思う状態にはならなかった。

肯定的感想

- ・あそび感覚で使えるソフトでたのしかった。
- ・音を自分でアレンジして、音のプログラムを作るのがおもしろかった。
- ・命令の中身やサブルーチンについて理解が深まりました。
- ・初めて使ったときは理解できなかったが、命令コードの意味がわかってからおもしろくなった。
- ・PCよりシンプルなのでコンピュータの構造を理解しやすかった。

否定的感想

- ・アセンブラ言語を覚えるのが大変だった。
- ・何をどう入力したらいいのか理解できないことがあった。
- ・どこをどう操作すればいいのかわからなかった。
- ・プログラム入力方法が難しく苦労した。

図19 GMC-4を使った実験への学生の感想

6.1.3 らくらくロジック

GMC-4のエミュレータのソフトを使って機械語を理解してもらったあと、コンピュータはどのようにしてその機械語を実行しているかを理解してもらうために、らくらくロジックというフリーソフト¹⁵⁾を使って、図20や図21のような回路を入力してもらい、ソフト上でその回

路を動かして仕組みを理解してもらった。図20は1ビットの情報を記憶させるための回路であり、回路を画面上で作成後に、シミュレーションという手法により回路に1ビットの情報を記憶できることを学生に確認してもらった。図21は2進数4桁のカウンタ回路であり、この回路についても同様に回路を作成してもらったあと

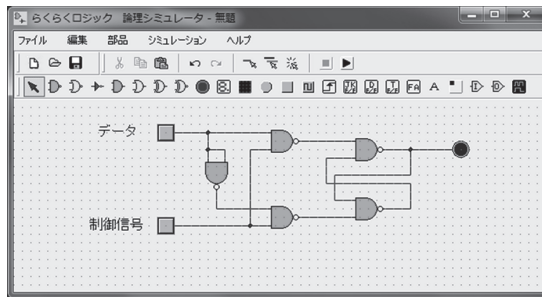


図20 1ビット記憶回路の設計と動作確認

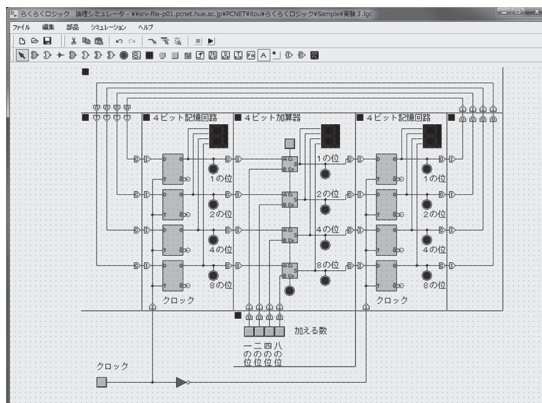


図21 4ビットカウンタの設計と動作確認

肯定的感想

- ・回路図などを見るのが苦ではなく好きなのでおもしろかった。
- ・コンピュータ回路を自由に設計して、どんな風になっているかがわかるとても楽しく、とてもためになるソフトでした。
- ・実際に回路を組み立てたことで理解が深まったと思います。
- ・クロック信号がこれをつかったことでより理解がふかまった。
- ・実際に部品を配置して、接続して実行してみるというのは、本当に専門家になったみたいで楽しかった。

否定的感想

- ・全然使いこなせなかった。使い方がわからなかったです。
- ・やっていることはよくわからなかったが、図を作っているみたいで楽しかった。
- ・どんな部品があって、どこどこをつなげば動作するのか分かりづらかった。

図22 らくらくロジックを使った実験への学生の感想

に、動作についても確認してもらった。

らくらくロジックを使った授業に対する学生の感想は、図22のように肯定的意見とともに否定的意見もあるが、GMC-4のエミュレータに比べて肯定的意見が多かった。

6.1.4 授業分析

授業の最後に、学生に授業の難易度を下記のような5段階で記入してもらったものを集計した結果が図23である。約半数の学生は難しいと評価している。

- 5：簡単すぎる
- 4：やや簡単
- 3：ちょうど良い

2：やや難しい

1：難しすぎる

一方、毎回の授業の最後で行う確認問題の平均点数から理解度平均の推移と、図23から計算した難易度平均の推移は図24のようになる。理解度平均は難易度平均と強い相関を持っている。理解度平均は4.46であるが、確認問題は復習を兼ねるために資料参照可能としているため、学生自身が感じる理解度とは同じではない可能性がある。また、確認問題の最後に授業の難易度を記入してもらっているため、確認問題の平均点数と難易度が強い相関性を持っていると考えられる。前期と同じように、学生自身が

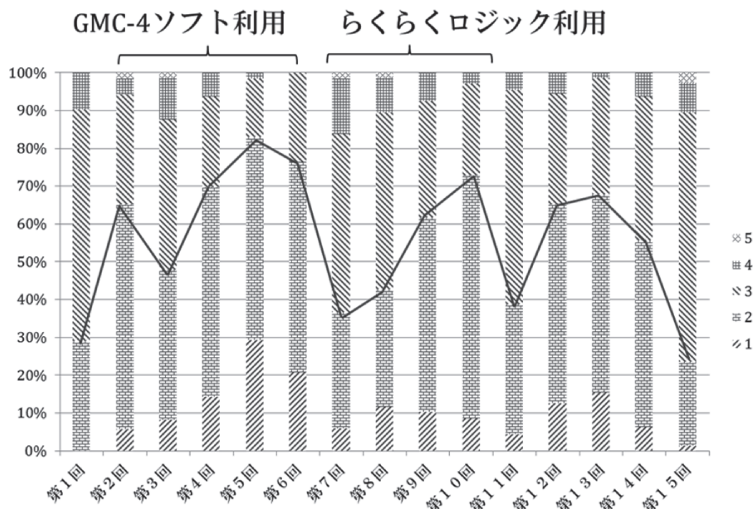


図23 15回の授業の難易度の内訳

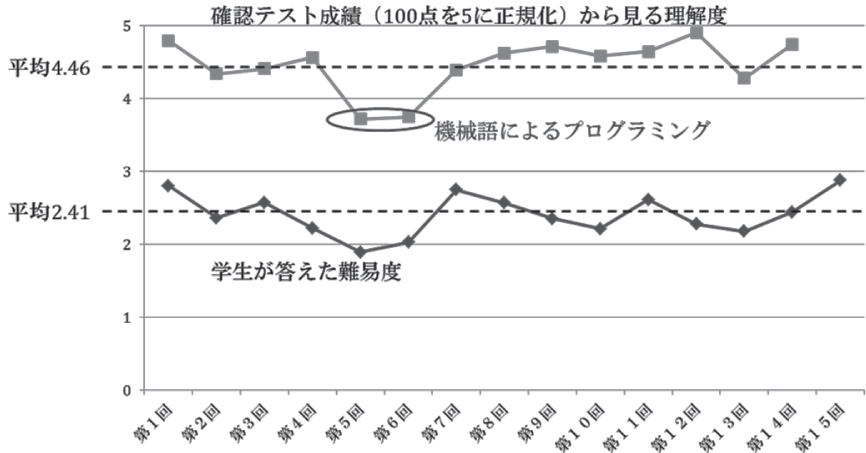


図24 15回の授業の理解度平均と難易度平均の推移

5段階で授業の理解度を記入するようなかたちも残すべきであったと感じている。

確認問題の平均点数は、機械語によるプログラミングの授業の場合に極端に低いことから、コンピュータの機械語の理解が学生にとって非常に難しいことがわかる。

6.2 情報処理論 I

二年目前期の情報処理論 I では、学生の興味と理解を高めるために複数の実験を導入した。

6.2.1 ハノイの塔

まず、ハノイの塔の問題に適用した。ハノイの塔とは、すべてのフロアを左の柱から右の柱に、下記ルールを守って移動する問題であり、情報処理の中では有名な問題の一つである。

- 1) 各柱の一番上のフロアからしか移動出来ない
- 2) 小さいフロアの上にそれより大きいフロアは積めない

ハノイの塔の問題は、ルールは一見簡単であるが、フロアの数が多くなると手番が多くなり、人間には煩雑な問題に見える。しかし、フロア数が少ない問題について手順を考えることにより、フロア数が多い場合でも容易に解けることがわかる。このような問題解決の手法

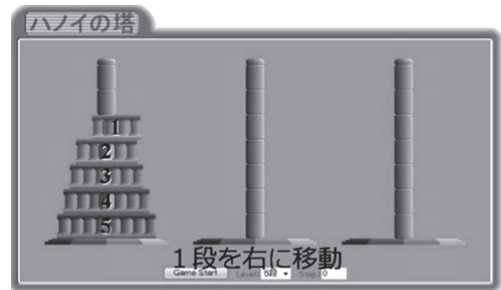


図25 ハノイの塔

を学生に理解してもらうことが、ハノイの塔を題材とした理由である。

このハノイの塔の問題を、図25のような Web 上のソフト¹⁶⁾ を使って、学生自身が考えて解いてもらう。そのあとで、解決方法を説明して、その解法を適用してもらう。

このハノイの塔の実験への学生の感想は図26に示すように、頭で考えるだけでなく、考えたアイデアを実際に適用することで楽しみと納得性もあるため、効果が大きかった。

6.2.2 アルゴロジック

アルゴロジック¹⁷⁾とは、電子情報技術産業協会 (JEITA: Japan Electronics and Information Technology Industries Association) が開発した学習目的のソフトであり、プログラミングの基本となるアルゴリズムをゲーム感覚で学習する

ことが出来る。実際の教育現場でも適用されている^{18,19)}。

図27のように、ロボットが旗をすべて取るように、命令のブロックを組み立ててプログラム

を作り、実行して結果を確認することが出来る。

このアルゴリズムの実験への学生の感想は図28に示すように、問題を解くために考えることも楽しくなり、プログラムを作ることに興味

肯定的感想

- ・あそび感覚で使えるソフトでたのしかった。
- ・入れかえに頭を使ったがおもしろかった。
- ・置く順番に法則があったビックリした。
- ・公務員試験の数的の問題にも出てきたので驚いた。
- ・やっていくうちにパターンが理解できたので簡単だった。
- ・法則を見つけたときはうれしかった。

否定的感想

- ・5段目から頭がごちゃごちゃになり苦戦した。
- ・発想が大切で、数がふえていく程むずかしかったです。
- ・何度も同じような繰り返しでうまくいきませんでした。

図26 ハノイの塔を使った実験への学生の感想

応用1 知恵の輪

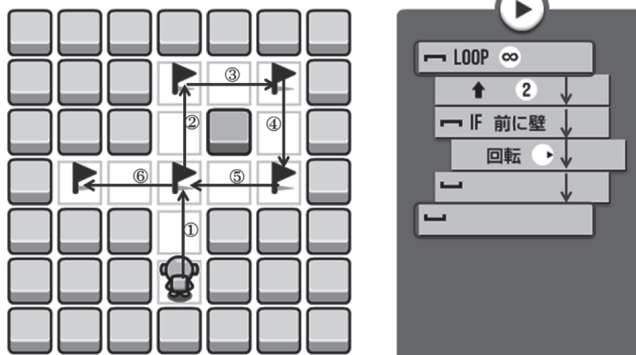


図27 アルゴリズム

肯定的感想

- ・プログラムの楽しさを覚える良いソフトだと思った。
- ・非常に頭を使われ気づけばクリアするのに夢中になっていた。
- ・自分で考えて、攻略法を見つけ出すから楽しい。
- ・◎つくまでひたすら考えました。とても楽しかったです。
- ・アルゴリズムをゲームと同じ気分で理解でき、おもしろかった。
- ・解いたときは自分も成長した感じです。

否定的感想

- ・よく考えないと分からなく難しかった。
- ・向きを変えたりして頭がこんがらがった。
- ・クリアできないとかなりストレスがたまる。
- ・elseの使い方がわかりにくかった。
- ・なかなか思い通りの動作をしてくれず、残念だった。

図28 アルゴリズムを使った実験への学生の感想

がわくという学生も多く、効果が大きかった。ただ、否定的意見にあるように、解けないとイライラするケースもあり、進め方の工夫は必要である。

6.2.3 オセゲーム

第13回目の授業では、実際の問題の解決Ⅱ（ゲームに勝つ戦略）がテーマであり、ゲームを情報処理として扱う場合の一般的な手法を説明したあと、オセロゲームを例に取ってプログラムが次の一手を決定する方法を説明する。

プログラムがオセロゲームで次の一手を決定する戦略を教える前に、学生に自分のやり方でオセロゲームをしてもらい、その勝敗を記録してもらう。そのあとで、プログラムが行う基本的な処理手法を解説して、その手法を実際のゲームに適用してその勝敗を図29のように記録してもらう。ゲームには図30のようなオセロゲームのフリーソフト²⁰⁾を使う。

オセロゲームを使った実験への学生の感想は図31に示すような肯定的なものが多いが、授業

最初は、これまでの経験だけでゲームして、結果を記録する。

第1回目 ●：___ ○：___
 第2回目 ●：___ ○：___

次に、ゲームに勝つ戦略を使ってゲームして、結果を記録する。

第1回目 ●：___ ○：___
 第2回目 ●：___ ○：___

図29 オセロゲームの手順

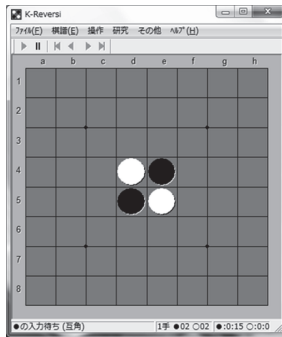


図30 オセロゲームのソフト

肯定的感想

- ・必勝法のおかげで少しはマシになったが、やはり強かった。
- ・オセロをすることがあれば、少しは勝率は上がるだろう。
- ・戦略を見ながらだどこに打つとまずいなど分かり勝てるようになった。
- ・何度もする内に、新しく必勝法も見つけることができた。
- ・ただのゲームではなく、どうすれば勝てるかなどいつもと違う視点で見ることができ、興味深かった。

否定的感想

- ・我流の方が勝てた。
- ・勝ち方がわかっても勝てなかった。
- ・いろんな方法、戦術をためしたがかてなかった。
- ・勝つ方法を見つげるためにやったものの勝てませんでした。

図31 オセロゲームを使った実験への学生の感想

で説明した戦略を適用しても勝てない学生の場合は、楽しかったもののその戦略の効果が確認出来ないままとなってしまった。

6.2.3 プログラミング言語 HSP

HSP (Hot Soup Processor)²¹⁾とは、プログラミング言語の一つであり、次のような特徴を持っている。

- ・ ONION software というグループが開発・公開
- ・ Windows で動作するプログラミング言語
- ・ フリーソフトウェアとして誰でも無償で入手可能
- ・ BASIC 言語をベースにした簡潔な文法
- ・ ゲームなどの開発に適している
- ・ マルチメディア再生などのアプリも簡単に作れる
- ・ サンプルプログラムも豊富

この言語を使って、授業の中で説明したいいくつかの手法をプログラミングして実際に確かめ

てもらった。HSP を使う最後の授業では、3つの数を乱数で発生させて、ボタンを押して数字を止めて、3つの数字が同じだと得点が増えるというゲームの作成を段階を踏みながら行ってもらった。完成すると図32ようになる。

HSP を使った実験への学生の感想は図33に示すようなものになったが、数としては肯定的なものが少なく、否定的なものが多かった。文字を多く入力することへの抵抗感、また1文字でも入力を間違えると動かないという厳密性への嫌悪感を抱く学生が多い。

6.2.4 授業分析

授業の最後に、学生に授業の難易度を下記のような5段階で記入してもらったものを集計した結果が図34である。毎回の授業の最後で行う確認問題の平均点数から理解度平均の推移と、図34から計算した難易度平均の推移は図35のようになる。コンピュータ基礎 I の場合と同様に、理解度平均は難易度平均と強い相関を持つ

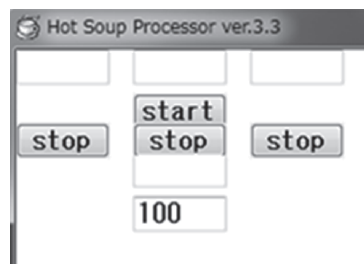


図32 HSP を使って作ったゲームの画面

肯定的感想

- ・ プログラムの意味を理解すると見ためほど難しくなかった。
- ・ ちゃんと動いた時うれしかった。
- ・ スロットのときはボタンを"Start"ではなく"激熱"にしてあそびました。
- ・ VBAとは違い、とても扱いやすいプログラミング言語だと思った。
- ・ 打つのは大変だったけど、達成感があった。

否定的感想

- ・ プログラムを打ち込むのに苦労した。
- ・ 一文字でも間違えると、上手く機能しない。
- ・ 手順と英語の意味を覚えるのが難しいです。
- ・ 少しの文字をだすのにいっぱい入力するのはとてもたいへんだった。
- ・ 文字をうつだけで、おもしろくなかった。
- ・ 細かいミス一つでダメになるのが厳しかった。

図33 HSP を使った実験への学生の感想

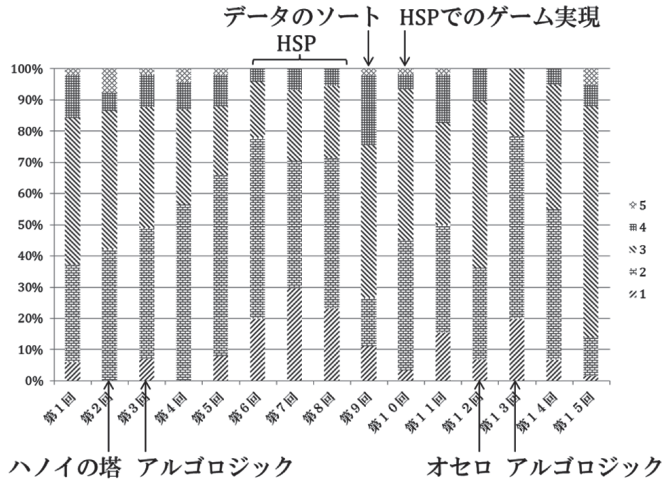


図34 15回の授業の難易度の内訳

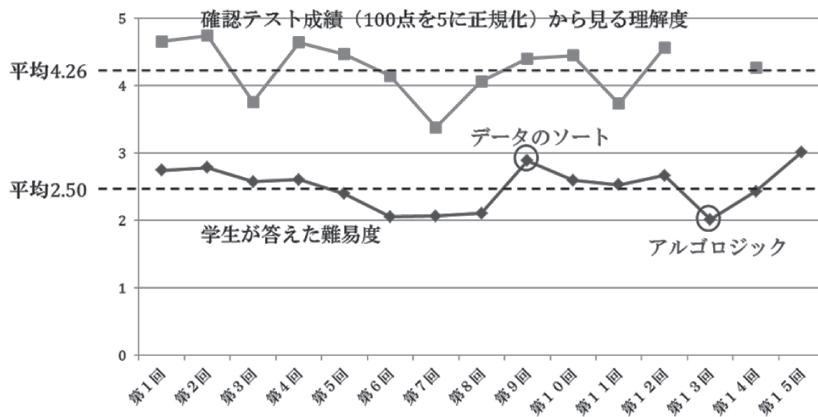


図35 15回の授業の理解度平均と難易度平均の推移

ている。理解度平均は4.26であるが、確認問題は復習を兼ねるために資料参照可能としているため、学生自身が感じる理解度とは同じではない可能性がある。学生が答えた難易度平均は、第9回のデータのソートが一番高く、第13回のアルゴリズムが一番低い。このように、第13回のアルゴリズムでの難易度平均が一番低いが、学生の感想は肯定的なものが多い。したがって、前期と同じように、学生自身が5段階で理解度を記入するようなかたちも残し、理解度と難易度の関係を考察する必要があることがわかった。

7. 考察と今後の取り組み

7.1 授業アンケート結果からの考察

教員になって一年半が過ぎ、コンピュータ基礎 I および情報処理論 I については、昨年度と今年度での授業アンケート結果を比較することが出来るデータがそろった。授業アンケート結果は5点満点で数値化されるが、これまで担当した科目で一番悪い評価の値を0として、他の科目の評価を相対的にプロットすると図36のようになる。

コンピュータ基礎 I および情報処理論 I については、昨年度の授業に対する授業アンケート

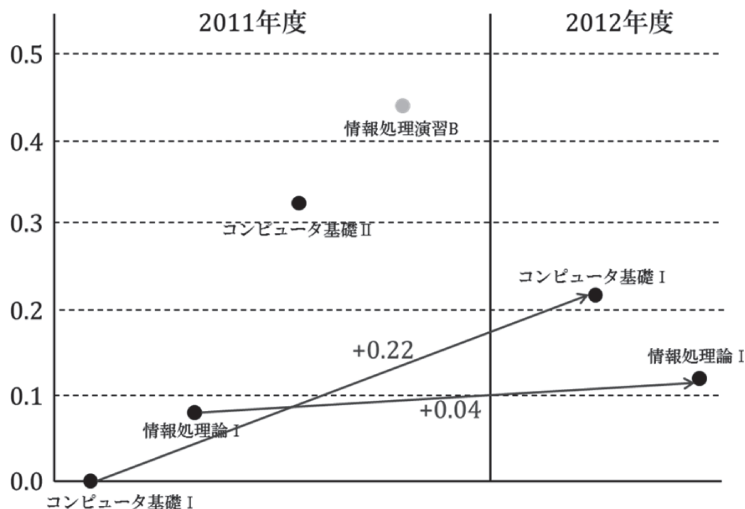


図36 授業アンケート結果の相対的比較

結果からの反省に基づいて、今年度は演習や実験を含めて様々な試みを取り入れてきた。コンピュータ基礎 I では授業アンケート結果の評価は0.22だけアップしているの新たな試みは多少効果が出ているが、情報処理論 I では0.04のアップのみとなっている。

情報処理論 I において今年度は0.04しかアップしていない理由は、今年度新たに取り入れたHSPというプログラミング言語が文字で入力する言語であるため、1文字でも入力を間違えるとプログラムが動かないなど、学生にとっては満足度を下げる題材だったからではないかと考えている。1文字の入力間違いも許さないという厳密性、物語性のない英単語を入力しながらプログラムを作る無機質さ、こうした要素が学生の満足度を下げる要因となっているのではないかと推測している。

科目は違うが、昨年度後期のコンピュータ基礎 II の授業アンケート結果と、今年度のコンピュータ基礎 I の授業アンケート結果を比較してみると、施策を多く取り入れた今年度のコンピュータ基礎 I の授業アンケート結果の方が0.1だけ評価が低い。これは、コンピュータ基礎 I はハードウェアに関する内容であるために内容

表1 学生の参加度と評価の関係

参加度	(2012年度の評価) - (2011年度の評価)
高い	-0.39
普通	+0.33
低い	+1.94

に人間は登場しない。一方、コンピュータ基礎 II は、すでに触れたようにソフトウェア開発に関する内容であり、チームによる開発方法やプログラムに誤りを入れないための行動など、人間が多く登場する内容であり、理科系の内容が比較的少ない。

この結果から、人間が登場しない無機質な理科系の内容は、学生にとって理解度も満足度も低いのではないかと考えられる。

コンピュータ基礎 I の今年度の授業アンケート結果は昨年度のものに比べて0.22だけアップしているが、授業への参加度が高い学生、普通の学生、低い学生それぞれのグループでの評価を昨年度と今年度を比較してみると、表1のようになる。なお、学生の授業への参加度は、授業アンケートの中に受講の熱心さと出席回数を聞く質問があり、その2つの質問への回答に基づいて数値化したものを使って分類した。

全体平均としては今年度0.22だけ評価がアップしているが、表1を見ると参加度の高いグループの学生の評価平均は今年度-0.39と下がっている。一方、参加度が普通のグループでは+0.33、参加度が低いグループでは+1.94と上がっている。

このようになった理由を考えてみる。一年目ではコンピュータサイエンス分野において有名な教育用プロセッサである DLX という少し難しいものを題材として教えた。この一年目では、授業アンケートの結果かなり難しいという評価になったため、二年目では、大人の科学マガジンという雑誌の付録になっている4ビットコンピュータである GMC-4 という初歩的なコンピュータを題材として使った。つまり、参加度の高い学生にとっては、一年目のような本格的なプロセッサの方が学問的満足度が高く、初歩的な4ビットコンピュータでは学問的満足度が低いのではないかと推察出来る。一方、参加度が普通か低い学生にとっては、初歩的な4ビットコンピュータの方が理解しやすかったと思われる。授業アンケート結果の全体平均は今年度の方が高いが、参加度の高い学生の評価平均は今年度の方が悪かったという事実は、厳しく受け止める必要がある。

7.2 授業内容についての考察

授業の中で取り入れたプログラミングについて学生の感想から考察すると、下記のようにまとめることが出来る。

□好きになる要因

- ・音が出る、絵が動くなどの Audio Visual 性
- ・文字入力が不要なプログラミングの簡易さ
- ・パズルのように考えて解く面白さ
- ・実際に完成したときの達成感
- ・ゲーム性のある題材

□嫌いになる要因

- ・英語がベースのプログラミング言語で、

意味把握の難しさ

- ・文字入力の多さ
- ・1文字の打ち間違えでも動かない厳密性
- ・実行して表示されるものの無味乾燥さ
- ・プログラム記述の分かりにくさ

授業の中に取り入れた実験について学生の感想から考察すると、下記のようにまとめることが出来る。

□肯定的な要因

- ・授業で聞いたことを自らの実験で確認するわかりやすさ
- ・手を動かすことで、眠くならず、時間が忘れる
- ・ゲーム性のある実験の楽しさ

□懐疑的な要因

- ・実験に使うソフトの操作性の悪さ
- ・話で理解出来たものを再度実験することの意味
- ・実験しても意図した通りにならない場合の実験の意味
- ・実験がうまく行かないときにすぐに聞けないもどかしさ

7.3 課題

今回のこのような試みから得られた課題としては、下記のようなものがある。

□内容面

- ・学生の興味をもっと強く引き付けること
- ・理解できない学生の割合が減るような難易度の設定
- ・身近な話に置き換えたり、実際の利用場面の説明を多くすること
- ・参加度の高い学生の満足度の向上

□運用面

- ・プログラミングや実験などを出来るだけ毎回の授業への適用
- ・全学生への手厚いサポート
- ・新たな題材の発掘（見つからない場合に

は、自分で開発することも含めて)

□その他

・プログラミング言語は文字入力不要なものを利用

・実験や演習と授業内容との強い関連付け
内容面において、「理解できない学生の割合が減るような難易度の設定」と「参加度の高い学生の満足度の向上」という2つの課題を同時に解決することはかなり難しいと思われるが、何とかして解決しなければならない課題である。

7.4 今後の取り組み

今後の取り組みとしては、下記のような3点を検討している。

1. 学習の意欲と効果を高めるための測定
難易度, 理解度 (テストによるもの, 学生の印象), 参加度, 満足度
2. さらなる実験教材の調査および作成
探してもない場合は自作するなど
3. コンピュータサイエンスアンブラグド^{22~24)}
などの利用

コンピュータの電源プラグを抜いて、コンピュータ科学を身近なものに置き換えた体験を通してコンピュータサイエンスを学習するため

の教育手法

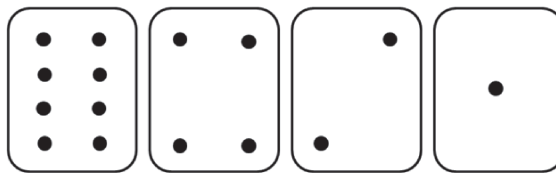
(たとえば2進数は図37のようなカードで説明する)

コンピュータ基礎Iはハードウェアに関する内容であり、人間が登場しない典型的な理科系で科目であり、今年度の授業の中でハードウェア構成を図38を使って説明した。この説明を行った授業の出席カードに、ハードウェア構成は難しいので何かに例えて説明して欲しいという要望があった。

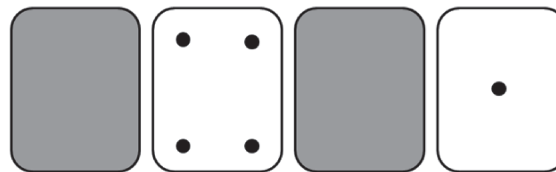
そこで、次の回の授業では図39のように、顧客から依頼を受けた仕事を会社の組織がどのように処理して行くかに例えてハードウェア構成を再度説明したところ、その授業の出席カードには、例えて説明してもらえたのでわかりやすかったという感想があった。

このことは、人間が一切登場しないコンピュータ基礎Iのような科目においても、なるべく人間社会の仕組みと対応させて身近なものとして説明することにより、文科系大学の学生も情報科学系科目を少しは身近なものとして親しみを持つ一つの例かもしれない。

2進数の意味を身近なカードで教える



表のカードは1、裏のカードは0と表記したものが2進数。
1のカードにある●の合計の数が10進数になる。



0 1 0 1

図37 コンピュータサイエンスアンブラグドの例

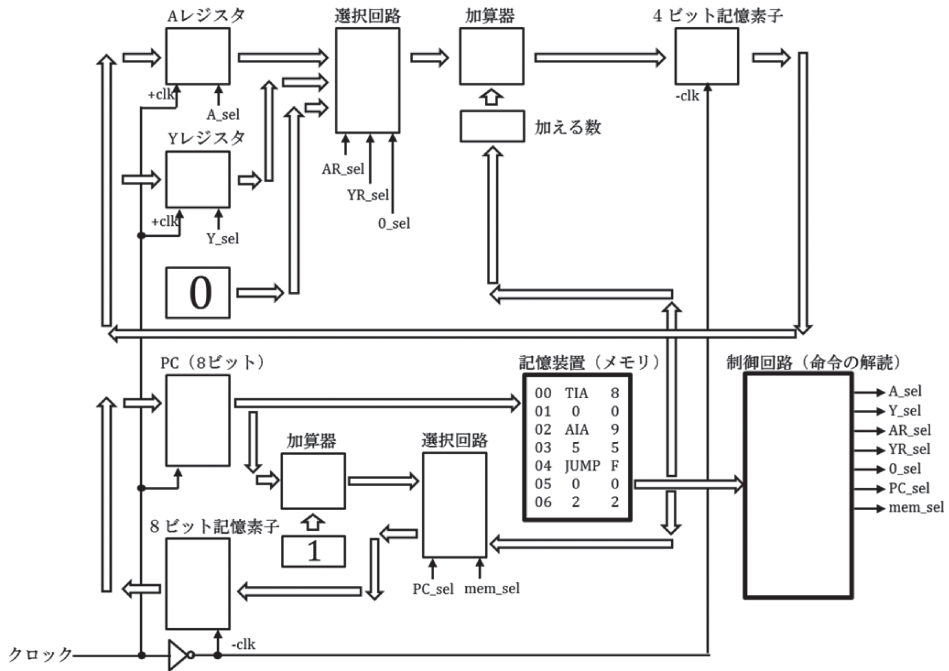


図38 ハードウェア構成の説明図

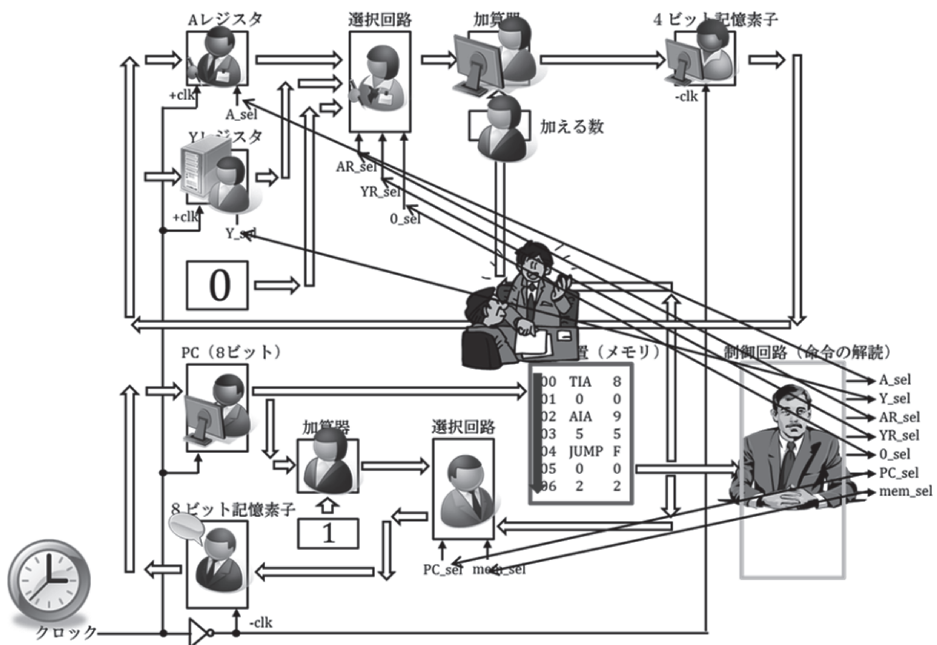


図39 会社組織に例えたハードウェア構成の説明図

注

- 1) 三浦信宏 (1995) 「企業教育から見た文系学生のための情報処理教育への提案」『情報処理学会全国大会講演論文集』第50回平成7年前期(1), pp. 73-74.
- 2) 畠山武道 (1996) 「文系における理科系科目の教育 —法学部の観点から—」『高等教育ジャーナル(北大)』, 第1号.
- 3) 河村一樹 (1996) 「文科系学科におけるコンピュータサイエンス教授法: データベース教育を事例にして」『情報処理学会論文誌』37(12), pp. 2438-2446.
- 4) 海老澤成享 (1997) 「大学における文系学生への情報基礎教育に関する一考察」『東京家政学院筑波女子大学紀要』1, pp. 129-132.
- 5) 中平勝子 (2002) 「文系女子短期大学生に対する情報処理教育実践」『早稲田教育評論』16(1), pp. 109-124.
- 6) 国分道雄 (2008) 「文系学生へのプログラミング教育」『聖学院大学論叢』20(2), pp. 197-206.
- 7) 福井正康, 細川光浩, 奥田由紀恵 (2009) 「文系学部における数理系教育の試み」『日本教育情報学会学会誌』(増刊), pp. 121-122.
- 8) MIT OpenCourseWare における Walter Lewin 教授の Vide Lectures を参照. <http://ocw.mit.edu/courses/physics/8-01-physics-i-classical-mechanics-fall-1999/video-lectures/>.
- 9) MIT Media Lab の Scratch ホームページを参照. <http://scratch.mit.edu/>
- 10) 森 秀樹 (2009) 「Scratch を用いた文系大学生向けプログラミング教育」『日本教育工学会論文誌』34, pp. 141-144.
- 11) 伊藤一成 (2009) 「プログラミング, 何をどう教えているか: Scratch を用いた授業実践報告」『情報処理』52(1), pp. 111-113.
- 12) David A. Patterson (原著), John L. Hennessy (原著), 富田真治 (翻訳), 新実治男 (翻訳), 村上和彰 (翻訳) (1994) 『コンピュータ・アーキテクチャー設計・実現・評価の定量的アプローチ』日経 BP 社.
- 13) 大人の科学マガジン編集部 (2009) 『コンピュータとプログラムの基礎がわかる大特集!』大人の科学マガジン Vol. 24, 学習研究社.
- 14) GMC-4 シミュレータの Web ページを参照. <http://dansan.air-nifty.com/blog/gmc4-simulator.html>.
- 15) らくらくロジックの Web ページを参照. <http://www.te-com.biz/delphi/rakuraku/LogicFrame.html>.
- 16) ハノイの塔の Web ページ参照. <http://goldsaucer.sakura.ne.jp/PicturePuzzle/Hanoi/hanoi/hanoi.html>.
- 17) アルゴロジックのホームページを参照. <http://home.jeita.or.jp/is/highschool/algo/index.html>.
- 18) 大山 裕 (2012) 「アルゴリズム体験ゲーム「アルゴロジック」(べた語義(第11回))」『情報処理』53(3), pp. 316-320.
- 19) 佐賀孝博 (2012) 「アルゴロジックとプログラミングを用いたプログラミング学習」『稚内北星学園大学紀要』(12), pp. 99-111.
- 20) リバーシソフト K-Reversi の Web ページ参照. <http://www.geocities.co.jp/SiliconValley-SanJose/8766/misc.html>.
- 21) プログラミング言語 HSP のホームページ参照. <http://hsp.tv/>.
- 22) 奥村晴彦 (2007) 「特集 教育用プログラミング言語と授業利用 (3) 情報科学教育への利用」『情報処理』53(12), pp. 1310-1313.
- 23) 兼宗 進, 久野 靖 (2009) 「コンピュータサイエンスアンプラグドの状況と今後の展開」『情報処理学会研究報告. コンピュータと教育研究会報告』(15), pp. 155-162.
- 24) 井戸坂幸男 (2012) 『中学校における情報教育—校内の情報教育と技術・家庭科の授業—』情報処理48(6), pp. 598-601.