

迷路脱出を題材としたプログラミング入門 (続報)

川上 孔明¹ 鈴木 貢¹

概要: 本報告では, 118 回研究発表会での報告に引き続き, 中等学校以上の導入プログラミング教育の題材として迷路脱出を用いることを提案し, 実践結果の報告と考察を行う. 前回からの主な改良点は, 次の 3 項目である. 第 1 に表現言語を C 言語から Java に変更し, プラットフォームに依存せずに結果をグラフィカルに提示するようにしたことである. これにより, 生徒がプログラミングの結果を把握し易くなり, 結果的に自主的に課題に取り組むことができた. 第 2 に, プログラミング技法を必要に応じて教示するようにすることで, 技法習得面での消化不良を解消できた. 第 3 に, アルゴリズムの説明や検証のためにフローチャートを用いたことで, アイディアからプログラムへの橋渡しが円滑になった. 以上について, 昨年度の実践成果と比較しながら今回の実践の結果を報告する.

Introduction to programming dealing with maze escape (Follow-up)

Abstract: In this report, we present a programming instruction method for novice programmers of students in secondary school or more, using maze escaping as a subject, following the presentation at IPSJ SIGCE 118, and we discuss the practice results. Compared to the last practice, following three are the improvement. First, we change the representation language from C to Java, which eased to show the results graphically with platform independent manner. This helped the students to grasp the result of their programming, and they independently could challenge the problem. Second, we instructed necessary and sufficient programming techniques along with every phases of this education program on demand, and this helped the students to avoid "indigestion" of learning techniques. Third, we employed flowchart to describe/verify algorithms and according program codes, and this smoothed transferring from ideas to program codes. Comparing with the result of last year, we report the result of the practice this year.

1. はじめに

IT に関する知識やスキルは, 情報関係に限らず銀行, 弁護士, 事務や販売等の幅広い業界の従事者が, 仕事で役に立つ知識として社会人が高校や大学で学んでおきたかったことの上に挙げている [1].

ユーザがプログラムの稼動を認識可能な機器という意味でのパーソナルコンピュータ類は, 家庭普及率は 75% を超え [2] 増加傾向にあり, もはや家電製品の一部になっている. この潮流は, スマートフォンやタブレットの爆発的な普及により, さらに加速すると考えられる. そのために, ソフトウェア業界は多くの優れた人材が必要となるが, 専門学校や大学への入学が, キャリア形成の大きな影響を与えることが多いことを鑑みると, 人材確保に向けた活動では高等学校以前の学生を対象にすべきである. コンピュータ

サイエンス アンブラグド [3] は, その活動の一例である.

しかし, 高等学校以前における情報教育の実情は, ワープロ・表計算・プレゼン+ Web 検索+ ネット程度に留まっている [4]. プログラミングに関連する内容を拡充し興味を持たせる施策は, 情報関連業種の従事者の火急の課題であると考えられる. こうした背景の下に, 我々は高校以前の情報教育におけるプログラミング学習を支援する教材を開発している.

スーパーサイエンスハイスクール (SSH) は, 高校生が大学の内容の導入部を学習する教育プログラムである [5]. 我々は, 島根大学 総合理工学部における SSH 事業の一環として, 2012 年度に引き続きプログラミング入門の授業を実施した. 大学におけるプログラミング教育では, 一般に半期から通年の科目として, 言語機能を体系的に習得する授業や演習, そして, それらに立脚してプログラミング技法やアルゴリズムの実現を学ぶ授業を開講し, ゼロからプログラムを構築していける人材の育成を目指す. しかし,

¹ 島根大学大学院 総合理工学研究科
Interdisciplinary Faculty of Science and Engineering of Shimane University

大学での SSH が 2 泊 3 日という短期間の合宿形式であり、利用可能な時間が実質的に 12 時間程度であるので、この方法論は使えない。そのために、この種の授業では「数列の和」等の数学的な題材で数行で完成するものが選択されることが多いが、生徒の達成感や問題考察の深化を考えると最適とは言い難い。

そこで、我々はプログラミング導入教育において、鳥瞰的ではない、現在位置における状況の把握、回転、それに 1 歩前進のみが許されている迷路からの脱出を題材とすることを提案している。

本稿では、2012 年度の実践 [6] の反省に基づいた 2013 年度の SSH 事業における実践について、2012 年度の実践の反省とそれに対する改良点を明確にしながら報告する。

2. 2012 年度の実践の反省と改良

2012 年度の実践の結果からの反省点は以下の通りであった。

- (1) 事前学習では必修課題しか解答しない。(発展課題は素数列を求める C プログラムの解説。)
- (2) プログラミングの講義で高度な迷路脱出アルゴリズムの表現に必要なものまですべてを詰め込んだので、これに時間をとられた。
- (3) 我々の予想に反して、プログラミング言語の関数を、数学の関数の延長として理解できなかった。
- (4) キャラクタ端末での結果の提示は直観的でなく、紙とエンピツで検討する方法を指導するのに手間取った。
- (5) 自分で考えた迷路脱出アルゴリズムをなかなかプログラムとして表現できなかった。

前回の実践を省みると、「プログラミング技法習得の題材としての迷路脱出」に傾きがちであったが、今回の実践では「アルゴリズムを考える題材としての迷路脱出」に舵を切るように努めた [7]。この節では、それぞれの問題点に対する今回の実践での改良を示す。

2.1 事前学習の内容の検討

(1) から高度な発展課題はハードルを高くするだけだと判断して削除することにし、事前課題を以下の 4 点に絞った。

- Windows 以外であなたが知っている OS (オペレーティングシステム) を最低で 3 つ以上挙げよ。また、そのうちのいずれか 1 つについて、Windows とどのような点で異なっているかを述べよ。
- 「テキストエディタ」の働きを調べ、実際にプログラミングするのに使いやすいものを 1 つ選んで紹介せよ。また、それが他のテキストエディタに比べてどういう点で使いやすいのかを説明せよ。
- プログラミングにおけるソースコードとコンパイラと実行の関係を説明せよ。
- 自分の名前を表示する Java プログラムを示せ。

表 1 SSH プログラミング学習内容 (教えた順)

	2012 年度	2013 年度
基礎	○	○
表示	○	○
演算	○	×
型 (int 型, double 型など)	○	×
条件文 (if 文)	○	○
等価演算子	○	○
関係演算子	○	×
論理演算子	○	×
関数 (メソッド) 呼出し	○	○
while 文	○	○
for 文	○	×
多重ループ	○	×
配列	○	×
関数 (メソッド) 定義	○	×

2.2 プログラミング学習の内容の検討

2012 年度の実践では、トレモアやオーアのアルゴリズムの実装に必要な内容まで詰め込んでいたので、「配列操作」や「関数の定義」等を理解させるのに時間がかかり、消化不良を起こしていた。生徒からも「学習内容が多かった」という指摘があった。これは、1 節で述べた大学での専門プログラミング教育の方法論の縮小版であったと反省する。

そこで、内容を減らし、理解の深化や、迷路脱出アルゴリズムを発案する時間を増やすことを狙った。今回の実践では「2 泊 3 日の合宿では右・左手法の発案と実装が限界」と割り切り、表 1 に示すように、それに必要十分な内容に絞った。そして、発案するアルゴリズムの高度化に沿って、プログラミング技法も高度化していくように計画した。これにより、教師にとって鬼門といえる上記 2 点を回避できた。

2.3 表現言語の選択

アルゴリズムを表現する言語はこの種の実践で最も重要な事項であるが、利用できる環境や教師の習熟度といった制約条件があり、理想的な選択は難しい。また、Windows 系 OS (自習) でも Linux 系 OS (大学の計算機設備) でも実習できる体制にする必要があった。我々の状況では、C 言語, Java, CommonLisp, JavaScript の中から選択しなければならなかった。

2012 年度の実践では C 言語と Java が候補の最後に残ったが、Java のオブジェクト指向の言語機能を「初学者向きでない」と判断して C 言語を用いた。しかし反省点の (4) にあるように、結果を直観に訴えるように提示すること、つまりグラフィカルな結果の提示は生徒の自主性を引き出すのに重要である。

そこで、今回は Java を用いることにし、オブジェクト指向の部分は「おまじない」として納得させることにした。これにより、プラットフォームに依存しないでグラフィカ

表 2 2012 年度と 2013 年度の比較
実施仕様

	2012 年度	2013 年度
使用言語	C 言語	Java
フローチャート	×	○
実行結果の提示	文字端末	グラフィック
教師の人数	1	3
生徒の人数	2	3
計算機の台数	2	3

授業内容（実施順）

	2012 年度	2013 年度
プログラミング講義（基礎項目）	○	○
フローチャートの学習	×	○
フローチャートで右・左手法を実装	×	○
プログラミングで右・左手法を実装	○	○
教師側から生徒に対し 右・左手法で解けない迷路の提示	○	○
右・左手法で解けない迷路の 脱出法をフローチャートで実装	×	○

ルな結果提示が可能になった。

ドリトル [8] も、センター試験用手順記述標準言語 DNCL を拡張したプログラミング環境の Pen [9] も初学者向け、かつ、グラフィカルな結果提示向きであるものの、残念ながら指導者が不慣れであったために利用を断念した。

2.4 フローチャートの導入

反省点 (5) に関して、我々が蒐集した高校の情報の教科書を調査した。アルゴリズムに関連させて、「情報と科学」では 4 冊のすべてがフローチャートを解説しており、「社会と情報」でも 6 冊中 1 冊がフローチャートを解説していた。一方で、プログラミング言語としては、「情報と科学」では 2 冊が JavaScript, 2 冊が構造化 BASIC と別れていた。

よって、高校生向けのアルゴリズムの「制御」の側面の説明に限れば、フローチャートを用いるのが適切であると判断し、今回の実践で用いることにした。特に、今回重点を置いた if 文と while 文の流れを理解させるには、矢印で次にどこに行くのかを追えるので、その動作がイメージしやすいと考えた。

3. 実施計画

2012 年度と 2013 年度の実践の比較を表 2 に示す。新規に加わった点は、フローチャートの学習を加え、アルゴリズム学習にフローチャートによる説明を追加したことである。

プログラミング講義の基礎項目では、基本的な Java の書き方を学習し、迷路に使う条件文やループ等の知識を学習する。

3.1 プログラミング学習のテーマの選定

今回は、個々の言語機能の使い方よりは、コード書法の選び方に力点を置いた。

迷路脱出の基本的な手順は以下の通りである。

- (1) 自分の状態を把握
- (2) 状態に対応してアクションする
- (3) (1) に戻る

(3) の「(1) に戻る」という項目から、ループの考え方を学ぶ。また (2) の「状態に対応して」ということから、条件文を学ぶ。

3.2 Java プログラミングの基礎学習

プログラミングの基礎学習では図 1 のようなスライドを用いて行った。Java 言語の基礎部分をおおまかに理解させることがここでの目的である。この講義だけで Java をマスターすることは想定せず、「講義した内容は忘れる可能性が高い。」として講義の計画を立てた。忘れた場合はネットワーク等で調べ、自主的に問題を解決させる。生徒間で話し合うことを禁じなかったが、プログラミング演習は生徒が自分の計算機を独占できるので、個別に実習に取り組むことになる。どうしても分からない場合は、教師側からどのようにすればよいか指示を出す。Java の参考書として文献 [10] を用意した。

3.3 授業の進め方の方針

授業の進行速度や最終的にどの程度までのプログラミングを要求するか等は、状態を見て個別に適宜判断する。生徒から見てプログラミング自体がとても困難なものに見える場合は、目標を下げてなるべく右・左手法に誘導するようにしていく。

逆の場合は、右・左手法を廃案し実装した後は、Web 上（学生が使う計算機はネットワーク環境に接続してある）にあるアルゴリズム等を見て、それを基にしてフローチャートやプログラムの作成しても構わないとした。

時間に余裕があれば、他の脱出法も検討したり実装し、効率を比較する方向に誘導することにした。

3.4 評価の観点と目的

今回の実践では、迷路脱出という題材の有用性を評価するために、4 つの観点から評価をする。

- Java の基礎事項のプログラミング学習効果
 - フローチャートによるアルゴリズムの学習効果
 - 迷路脱出を用いることによるアルゴリズムの学習効果
 - 迷路脱出を用いることによるプログラミング学習効果
- 大学の体系的な講義と異なり、SSH の目的はプログラミングを身近に感じさせることである。初学者にとってプロ

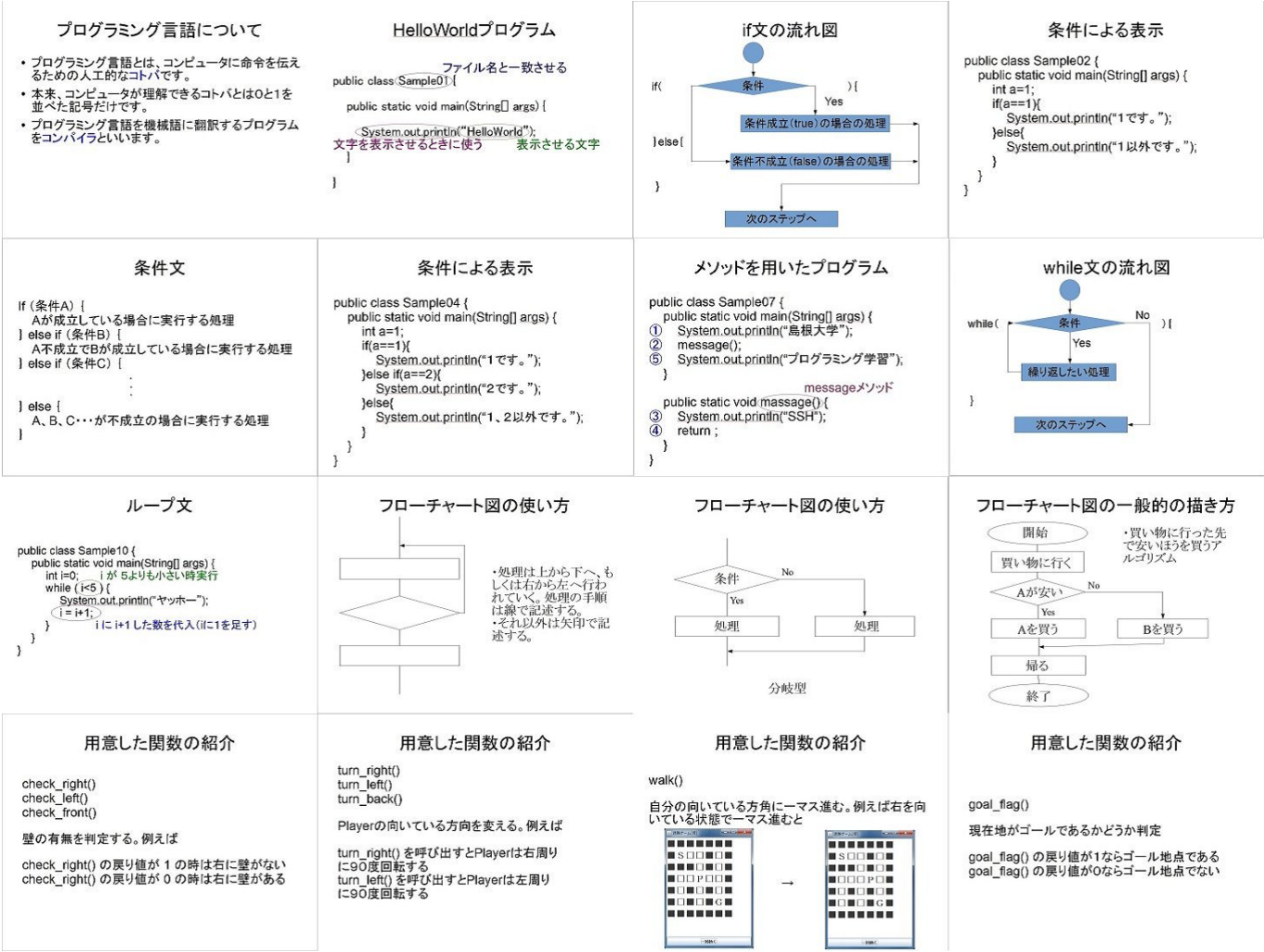


図 1 プログラミングの基礎学習で用いたスライド（抜粋）

プログラミングは身近なものではなく、抵抗が少なからずある。そのためなるべく簡単な問題から入りプログラムは簡単に作れることを教え、生徒にプログラミングに興味を持たせることに注力する。

4. 実施内容

4.1 生徒の様子とスケジュール

本授業に参加したのは島根県立の公立高等学校2年生男子生徒1名、女子生徒2名で、2012年度と同様に「パソコン操作は両手で操作はできるが、タイピングは人差し指のみ」という程度で、プログラミングスキルは皆無であった。1名はパソコン操作には慣れていたが、残りは素人同然であった。

今回は学科の実習設備をそのまま利用し、プログラミング環境が完備している Ubuntu Linux 12.04 の環境を用いて、1人1台を占有できるようにした。

教師は川上を含めて3名で、川上が講義を行い、残り2名がTAとしての役割を果たした。Javaのテキストとし

表 3 SSH 予定表

日付	時間	内容
8/21	11:20～	島根大学 着 学部案内、見学、昼食
	13:30～	事前学習内容の発表
	14:30～	講義、実験など
	17:00 頃	各宿舎 着
	20:00～	実習内容の整理等
	22:30 頃	就寝
8/22	9:00 頃	島根大学 着
	9:00～	講義、実験など
	17:00 頃	各宿舎 着
	20:00～	実習内容の整理・プレゼン作成等
	22:30 頃	就寝
10/18	9:00 頃	島根大学 着
	9:00～	実習プレゼン完成
	13:00～	実習成果発表 着
	15:30 頃	島根大学 発

て [10] を使用したが、用意した配布資料だけで問題に取り組めるようにした。

SSHのスケジュールを表 4.1 に示す。

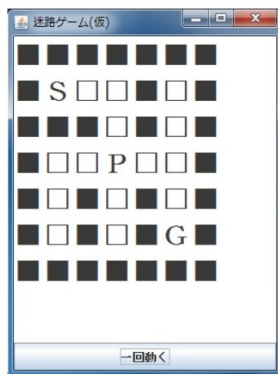


図 2 検証用のウィンドウ（「一回動く」を押すと呼び出し側に戻る）

4.2 プログラミング講義

川上が基本事項（条件文やループなど）を解説し、生徒から疑問点を抱い上げ、その場で問題を解決する形式で行った。前回の経験から、プログラミングは数学等への関連付けが難しいテーマであることがわかったので、具体例を用いて説明した。

最初に、教師側が用意したソースコードをコンパイルし実行して結果を見ることから始めた。次に、それらをすこしずつ変更したものを実行させ、その変更が意図した通りにプログラムの振る舞いを変えることを検証させた。前回の実践で得た、「プログラミングは真似からだと取り組み易い」という経験をさらに徹底している。例えば、if 文の講義では

```
if(a==1){
    S1
}else{
    S2
}
を
if(a==2){
    S1
}else{
    S2
}
```

と書き換えることによって、条件文で条件を変えると振る舞いが変わることを体験する。

書法を教える際には、汎用性を重視するようにした。例えば if 文の講義では、必ず文括弧をつけさせるように指導した。

4.3 迷路脱出によるプログラミング演習

「迷路オブジェクト」を表に出すことを差し控えながら、次の機能を有するメソッドがあることを前提にして、既習の if 文と while 文で迷路脱出を構成する。

- (1) 左に壁があるかチェック（`check_left()`）
- (2) 右に壁があるかチェック（`check_right()`）

*	*	*	*	*	*	*
*	G					*
*						*
*			*	S		*
*						*
*						*
*	*	*	*	*	*	*

図 3 右手法で脱出できない迷路

※ [S] はスタート, [G] はゴール, [*] は壁



図 4 授業風景

- (3) 前に壁があるかチェック（`check_front()`）
- (4) 90 度左を向かせる（`turn_left()`）
- (5) 90 度右を向かせる（`turn_right()`）
- (6) 180 度後ろを向かせる（`turn_back()`）
- (7) 向いている方向に 1 マス進ませる（`walk()`）
- (8) ゴール地点であるかどうか判定する（`goal_frag()`）

位置を変えるメソッドを呼び出すと、図 2 のような表示が更新され、直ちにプログラムの動作を検証できる。上記 8 つの組み合わせで、最初のステップである右・左手法を実装し検証できる。フローチャートを使って考えたアルゴリズムを、きちんとプログラムとして記述できるかどうか、ここでの観点である。

4.4 追加課題

図 3 のようなまず左・右手法では脱出不可能な迷路を提示する。そして、次のステップである高度な迷路脱出プログラムの必要性を理解させ、新たな脱出法を検討するように誘導する。

5. 考察

5.1 講義

まず第 1 段階では、座学でプログラムのソースコードの書き方や、if 文、while 文等の構文の使い方を学習した。説



図 5 演習風景

明時にはわざと配布資料を配らず、一通り説明し終わった後に配布したが、資料を用意し配布することを宣言したのでメモはあまりとっていなかったようである。既習の延長上にはない事項を学ばせる場合は、学習内容を資料としてできるだけまとめておくことが重要であると実感した。今回は、プログラミングする際に必要な情報をすべてプレゼン資料としてまとめていたので効果的に進められた。図 4 に授業の様子を示す。

5.2 演習

第 2 段階では、実際にプログラムを生徒自身に作成させた。ただし、この作成するプログラムは講義で取り扱ったプログラムの一部を変えるだけで動く簡単なプログラムのみを作成するものであったこともあり、そのためそれほど苦勞せずに問題に取り組むことができていると感じた。今回は教師も人数が多いことからワンツーマンで指導もできたので効果的に学習できた。図 5 に演習の様子を示す。

5.3 アルゴリズム学習

フローチャートを用いてアルゴリズムを学習していくのに、最初に学習したのは基本的なフローチャートの描き方である。その後、例題として次の問題を与えた。

図 4 の条件の元に、1 週間の満足度が高くなるように週間の「遊びの計画」を決める。(カッコ内は満足度を示す。) このときどこにも行けない曜日は作らない。(○は営業日、×は休み)

図 6 のような図が正解で、全員が正解できた。

5.4 実験

実験はまずフローチャートを記述し、それを Java で実装し、ランダム生成される迷路を実際に脱出して検証する。

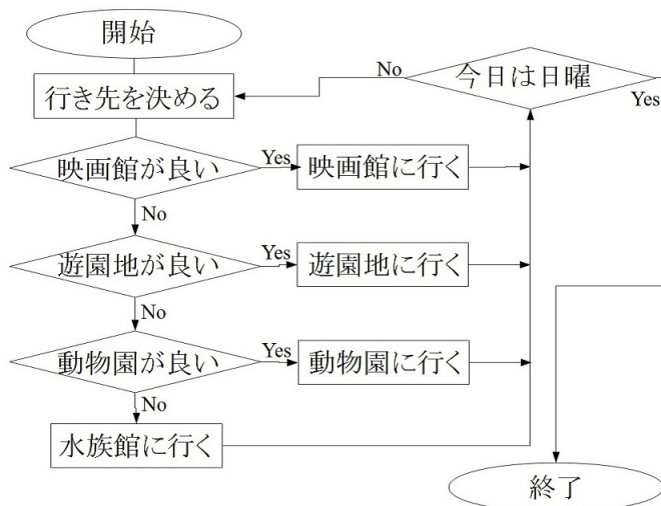


図 6 「遊びの計画」を決定するフローチャート

表 4 「遊びの計画」の条件

	月曜	火曜	水曜	木曜	金曜	土曜	日曜
映画館 (4)	×	×	×	×	×	○	○
遊園地 (3)	×	○	×	○	×	○	○
動物園 (2)	×	×	○	○	○	×	○
水族館 (1)	○	○	○	○	○	○	×

何パターンか試行して、ダメな迷路があれば別の手段を考える。

ただし 4.3 節の通り、可能な行動のパターンは限られているので、ランダム生成であっても、右・左手法で脱出できる迷路のみ生成させるものとする。またデバッグコードをはさみ、まず if 文のみで構成させ、一回ずつ動かすようにさせた。これはバグやミスがあった場合でも、この方が簡単に直せるためである。それに while 文を加え最後まで動かすという手順をとった。最終的に生徒は、次のような右手法にたどり着くことができた。(なぜか 3 人とも右手法で左手法を作った生徒はいなかった。)

```
while(goal_flag()==0){
    if(check_right()==1){
        turn_left();
        walk();
    }else if(check_front()==1){
        walk();
    }else if(check_left()==1){
        turn_right();
        walk();
    }else{
        turn_back();
        walk();
    }
}
```



図 7 発表風景

5.5 追加課題

発展課題であったためか、ヒントなしでは回答不可能であった。そこでヒントとして通ったマスのチェック (実世界でいうと通った場所に飴玉をおいていく) ようにしたらどうなるかと問いかけてみた。その結果として、1人の生徒がトレモアのアルゴリズムに類似したアルゴリズムのフローチャートを描くことができた。

発展課題として考えていたが、フローチャートを描くだけなら、それほどな難易度ではなかったといえる。

5.6 成長観点

直後に行ったアンケートでは、「プログラミングスキルやパソコン操作スキルがあがった。」や「プログラミングは難しいが、あきらめず作る忍耐力などがあがった。」という意見があった。また、「フローチャートの考え方が身についたのかパソコン内部の動き方が分かって、違う方向から物事を見ることができるようになった。」という意見もあった。

3週間後に高校側からの届いたアンケートによると、「プログラミングは自分には程遠いものであって関わることはないと考えていたが、意外とすんなりプログラムを作ることができるものであると分かり、プログラミングに対する興味が増した。」や、「自分の進路選びの参考となる体験ができた。」という意見があった。

アルゴリズムやプログラミングの学習とは直接関係がないが、前回のSSHよりもプレゼン作成に及び発表練習にも力を入れた。その甲斐あって、スライドを作る力や発表の力がついたように感じた。図 7 に最終日の成果発表の様子を示す。

6. 評価

アルゴリズムとプログラミング双方において生徒自身の理解度は 50 % 以上と自己評価した。ただ教師側から見れば問題にすべて正解したことから考えても、ほぼすべての内容を理解できたのではないかと信じる。ただし生徒自身

も難しいと考える点は三者三様であるため、やはり授業では生徒の個々を見る必要があると考える。今回はその点に関して、ワンツーマンで指導できたことが良かったと感じた。またプログラミングに対するイメージも身近なものへ変わったという意見があり、プログラミングに親しみを持つことができたのではないかと考える。生徒の「親しみが持てる点」として考えているのは、自分が作ったものが自分が思った通りに動くという点であるようである。このように動く楽しみを持たせることがプログラミングに興味を持つための近道であると実感した。

6.1 問題点・改善点

事前課題の自分の名前を表示する問題では、C 言語と Java を間違えた生徒がいた。ここで垣間見れるのは、プログラミング言語も自然言語と同様に多様であることを、現在の情報教育では教わっていないことである。また、全員が作成したプログラムを動かす機会を持たなかった。次回は、Java プログラムを試せるサイト [11] を教示し、確認させることを考える。

前回同様に、プログラミングのバグに対する解決策を自力で見つけられなかった。しかし今回は、バグの意味を教えるようにしたところ、意外にすんなりバグを解決できるようになった。これは有効な手段ではないかと考える。次の段階の学習で、バグの意味の解説を加えると良いかもしれない。

迷路での動作に関するメソッドは資料で解説してあったが、プログラムする段階では動作が分からず困惑していた。これは、プログラミングを真似から始めていて、自力で書く力がついていなかったため、つまり、フローチャートをプログラムにつなげる力がついていなかったためだと想像する。

迷路プログラムの直前でフローチャートを学習したが、一旦図 6 を下記のような擬似コードに変換する段階を挟むべきだと考える。

```
while(今日は日曜日!=0){  
    if(映画館が良い==1){  
        映画館へ行く  
    }else if(遊園地が良い==1){  
        遊園地へ行く  
    }else if(動物園が良い==1){  
        動物園に行く  
    }else{  
        水族館に行く  
    }  
}
```

※ Yes を 1, No を 0 としておく。

前回に比べ悪かった点は、教師と生徒がワンツーマンに

なり，生徒同士での話し合いが少なかった点である．教師が助言をせず見守る時間を設け，生徒同士の交流を促すよう授業を進めていくべきかもしれない．

6.2 プログラミング学習における迷路脱出の有効性

迷路脱出は他のプログラミング学習教材に比べ自由な発想が期待できる教材である．これは迷路脱出の方法が一通りだけでなく，いくつか迷路脱出の方法がある．今回の実践では，三人三様の脱出法を提案した．

「実際にやってみるとプログラミングは難しかった」等の意見もあったが，「意外と簡単にプログラムを作ることができた」という生徒がいたり，フローチャートレベルではあるがトレモアのアルゴリズムを検証できた学生もいた．これらのことから，GUIによる結果提示等の適切なお膳立てがあれば，迷路脱出はプログラミング初学者向けの題材として適切であると考えられる．

7. 関連研究

迷路ゲームを用いた関連研究として，プログラミング導入を支援する教育用教材の開発 [12] が挙げられるが，この実践では鳥瞰的な迷路状態が最初から提示されており，その状態をみてどのように動いたらよいのかを記述するといった，迷路解法の手順をとっている．

それに対し本研究では，鳥瞰的な迷路の情報は与えず，現在位置での壁や床の認識して，迷路脱出を考えるという形で行う．この設定にしたのは，マウスのような問題への発展を期待しているためである．

8. まとめ

本稿では迷路脱出を用いたプログラミング学習を考察し，実際 SSH での実践の内容を報告し評価した．主に前回との違いを詳しく考察し，前回より良くなった点や悪くなった点を考察しまとめた．前回に比べてアルゴリズムの考察に力点を置いたが，フローチャートの利用は正しい選択であった．

今後の実践として，大人数授業への展開も検討している．ワンツーマンに近い形でこの実践の手法は，本来であればこのままでの大人数での展開は困難である．2012 年度の生徒の意見として，教師に直に教わることが，本や Web など学習するよりも良かったというものがあった．それを解決する手段として，「早くできた生徒はまだできていない生徒に教える．」となるような体制で講義を進行することが考えられる．これは単純に教師役を増えるという効果だけでなく，教える側もより理解を深まることと，生徒間の進行度合いの均等化を促すことが期待される．上記のような体制で授業を進めるには，課題をクラス 1/3 程度の生徒が簡単な説明で解けるような容易さのものにする必要がある．

謝辞

この実践にあたり，TA として生徒の指導に協力して頂いた鈴木研究室の松尾昇君と角澤慶紀君に感謝します．また，SSH 事業の実施にご尽力頂いた関係各位に感謝します．

参考文献

参考文献

- [1] 教育研究部: “キミのミライ発見”, <http://www.wakuwaku-catch.net/>, 河合塾教育研究開発本部 (2012).
- [2] 本川 裕: “パソコンの普及率”, 社会実情データ図録, <http://www2.ttcn.ne.jp/honkawa/6200.html>.
- [3] Tim Bell, Ian H. Witten, Mike Fellows (兼宗 進 監訳): “コンピュータを使わない情報教育 アンプラグドコンピュータサイエンス”, イーテキスト研究所 (2007).
- [4] 辰己丈夫, 江木啓訓, 瀬川大勝: “大学 1 年生の情報活用能力と ICT 機器やメディアの利用状況調査”, 学術情報処理研究, 国立大学法人情報系センター協議会, pp.111-121 (2012).
- [5] SSH, “<https://ssh.jst.go.jp/>”
- [6] 川上 孔明, 鈴木 貢: “迷路を題材としたプログラミング入門”, コンピュータと教育 研究会報告, 情報処理学会, 2013-CE-118, No.5 (2013).
- [7] 新開 純子, 宮地 功: “プログラミング学習支援システムを用いた入門教育の実践”, 日本教育工学会論文誌 33, 2009-12-20.
- [8] 兼宗 進, 御手洗 理英, 中谷 田哉子, 福井 真吾, 久野 靖: “学校教育用オブジェクト指向言語「ドリトル」の設計と実装”, 情報処理学会論文誌 プログラミング, Vol.42, No.SIG11(PRO12), pp.78-90 (2001).
- [9] 西田 知博, 原田 章, 中村 亮太, 宮本 友介, 松浦 敏雄: “初学者用プログラミング学習環境 PEN の実装と評価”, 情報処理学会論文誌, Vol.48, No.8, pp.2736-2747 (2007).
- [10] 川場 隆: “わかりやすい Java”, 秀和システム (2004).
- [11] ideone.com, “<http://ideone.com/>”.
- [12] 立川 文春, 石川 賢: “プログラミングへの導入学習を支援する教育用教材の開発”, 教育システム情報学会 研究報告書, Vol.24, No.3, p.28-35.