オンラインプログラミング環境Bit Arrowを用いたC言語プログラミングの授業実践

長 慎也[1,a)]　長島 和平[2]　堀越 将之[1]　兼宗 進[3]　並木 美太郎[2]

概要：
筆者らは現在，Webブラウザを用いてプログラミング学習が可能な環境「BitArrow」を，初学者向けのプログラミング学習活動に用いている．BitArrowは，複数のプログラミング言語の学習ができるように設計されており，C言語でのプログラミング学習にも対応している．また，文字の入出力を行なうような標準的なライブラリだけではなく，初学者にも簡単に扱えるグラフィックスライブラリを備えており，興味を持たせる題材を提供することができる．BitArrowのC言語を用いた授業は，昨年度より大阪電気通信大学において利用が始まり，今年度からは明星大学の情報学部においても行われている．本発表では明星大学で現在行われている授業の経過を報告する．この授業を通じては，C言語の制御構造，変数，関数，構造体，ポインタなどを扱っているが，それらの単元で扱うプログラムを実行できることを確認している．

Learing C programming using “Bit Arrow”, an online programming environment.

CHO Shinya[1,a)]　NAGASHIMA Kazuhei[2]　HORIKOSHI Masayuki[1]　KANEMUNE Susumu[3]
NAMIKI Mitarou[2]

1. はじめに

筆者らは，プログラミング学習環境として，Webブラウザのみで動作可能なBitArrow[1]を開発し，初学者向けの授業におけるプログラミング環境に用いる実践を行っている．BitArrowは現在，JavaScript，ドリトル，C言語で書かれたプログラムを実行することが可能であり，今回の実践では，C言語の授業を実践した経過を報告する．

情報系の学部において，C言語は重要な学習項目となっている．その理由として，C言語はポインタや配列など，CPUのアーキテクチャに近い部分に直接アクセスする機能を有しているため，コンピュータのデータ構造，メモリモデルの理解に有用であることや，JavaやJavaScriptなどへ言語仕様の一部が引き継がれており他の言語の習得の助けになることなどが挙げられる．このため，情報系の学部ではC言語は早い時期に習得させることが多い．

一方で，C言語はプログラミングの初学者にとっては難しい点が多い．例えば，変数の初期化忘れや，メモリ範囲外のアクセスなど，実行時のエラーに対するチェックがないため，原因不明のエラーに遭遇してしまうこともある．情報系の学部の学生であっても，必ずしもプログラミングの経験があるとは限らないが，先述の通り早い時期にC言語を学ぶことが多いため，初めて学ぶ言語がC言語であることも多い．このような状況においてC言語の難しい点に遭遇した場合，つまずきの原因となる可能性がある．

また，初学者にとっては，プログラミングの題材は興味を持てるものである必要がある．例えば，グラフィックスやアニメーションを用いることで興味を惹くことが有用である．一方で，C言語に標準で用意されているライブラリでは，グラフィックスを利用するのは比較的難しい．この

[1] 明星大学
Meisei University, Japan
[2] 東京農工大学
Tokyo University of Agriculture and Technology, Japan
[3] 大阪電気通信大学
Osaka Electro-Communication University, Japan
[a)] cho@eplang.jp

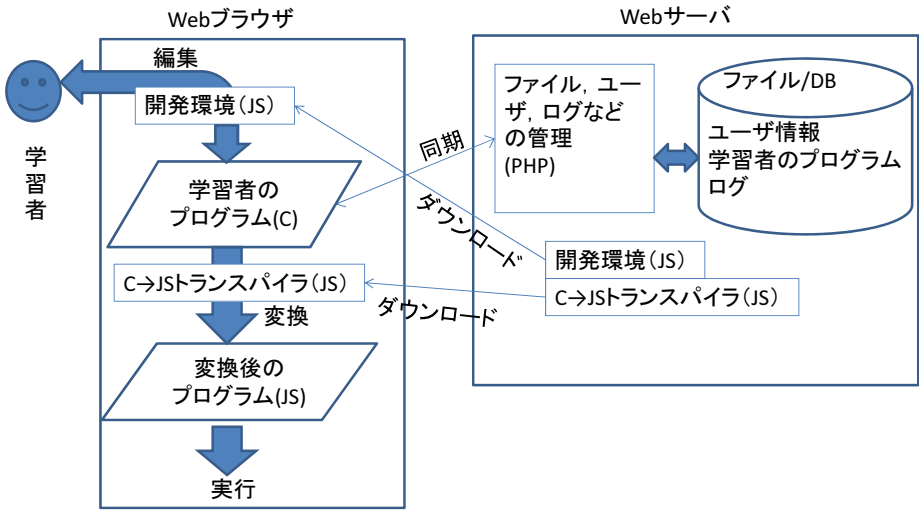

図 1 BitArrow の構成

ため，ほとんどの C 言語の教材では文字やファイルの入出力のみを扱うことが多い．これらを学習することも重要ではあるが，それだけで興味を持続させるのは難しい場合もある．

BitArrow 上で動作する C 言語では，C 言語の初学者が陥りやすい間違いを指摘してくれる機能や，グラフィックスやアニメーションを用いた興味をもたせやすい題材で実習を行なうことができる．

BitArrow 上で動作する C 言語は，昨年度から著者の 1 人が所属する大阪電気通信大学での授業に用いられており，今年度からは別の著者が所属する明星大学においても実践を行っている．本発表では，明星大学における授業実践の経過を報告する．

2. BitArrow の設計

BitArrow の構成を**図 1** に示す．BitArrow は，Web アプリケーションとして動作し，開発環境本体および，学習対象となる各種言語の処理系はすべて JavaScript（JS）で記述され，Web ブラウザ側にダウンロードして実行する．また，学習者が書いたプログラムは処理系により JavaScript に変換（トランスパイル）され，Web ブラウザで実行される[*1]．サーバ側のプログラム（PHP）の機能は，Web ブラウザ側で作成・編集したファイルの同期やログの記録などに限定されており，言語の処理系はサーバ側で動作しない．次に，この構成を採用することの利点を挙げていく．

2.1 Web ブラウザのみで動作する

BitArrow は，言語の処理系がすべて JavaScript で実装されているため，学習者が用いる PC に Web ブラウザさえ入っていれば，他の開発環境を一切導入することなく利用することができる．

演習を行なう教室の環境には必ずしも希望したアプリケーションを導入できない場合があるが，Web ブラウザであればほとんどの環境に導入されているため，多くの場所で実習が可能になる．

また，自宅など演習室以外での演習も，新たに PC にアプリケーションを導入する必要がないため，学習者は授業時間外の実習も容易に行なうことができる．

2.2 Web ブラウザの機能を利用したインタラクティブなプログラムを開発できる

JavaScript は Web ブラウザのスクリプト言語であるため，グラフィックスをはじめ，Web ブラウザが持つ豊富な機能を利用することが可能であり学習者にとって興味ある題材でプログラミングができる．また，すべてのプログラムが学習者の手元で動作するため，アニメーションなどのインタラクティブ性の高い作品にも対応できる．

2.3 サーバ側の処理の負荷・管理の手間を減らす

BitArrow が動作する Web サーバを設置する管理者（教員等）側の利点として，BitArrow は実行をすべて Web ブラウザで行ない，サーバ側で処理系を動作させる必要がないため，処理の負担が少ない．これにより，一斉にアクセスが発生する多人数の授業でも処理を滞りなく行うことが可能である．また，サーバ側に処理系をもたないため，Apache や PHP などの標準的に導入されているソフトウェアだけでサービスを始められる．これはサーバを新たに設置する場合（現状の BitArrow は単一のサーバを用いてサービスしているが，学校内や地域ごとにサーバを設置す

[*1] 図 1 では学習者が C 言語のプログラムを書いている場合で説明しているが，他の言語でも同じ方式で動作する

ることも検討中である）においてセットアップや管理の手間の軽減につながる．

3. 関連研究

オンラインでプログラミングの学習を行える環境には CloudCoder[2]，CodeWrite [3]，Online python tutor[4] などがある．これらはいずれも，特定の関数を仕様に合うように作成させる演習を行わせる形式が多く，実行および結果の検証をサーバ側で行なう．この方式では，結果がブラウザに返されるのに時間がかかるため，インタラクティブ性の高い作品を作ることが難しい．また，gcc などのコンパイラをサーバに設置する必要があるので，サーバへの設置の手間と処理の負担がある．

C 言語に代わり，大学でプログラミング授業の導入として用いられることが多いのが Processing である．Processing はグラフィクスやアニメーションを手軽に利用できることからプログラミングの入門に向いており，Web ブラウザで編集・実行することも可能である [5] が，逆に C 言語で容易に作成可能なコンソールアプリケーションの作成が難しいという点が指摘されている [6]．また，言語仕様についても C 言語よりかは後発の Java などに似ているため，C 言語特有の機能（ポインタなど）を学習させるのには適していない場合もある．

4. C-JS の実装

本章では BitArrow 上で動作する C 言語（以下「C-JS」と呼ぶ）の実装を示す．

4.1 変換規則

C 言語を JavaScript に変換する際には，JavaScript と C 言語で似ている点についてはなるべくそのまま変換し，異なる点についてその差異を吸収するような変換を行うよう設計した．

- **制御構造**
  JavaScript は C 言語の文法を踏襲しているため，式文，条件分岐，繰り返し，関数などの制御構造はそのまま JavaScript でも同様に扱えるものが多く，それらについては対応する同じ要素に変換している．
- **数値の型**
  C 言語が扱う数値には整数，実数などの区別や，符号の有無やビット長の区別などがあるが，JavaScript で扱う数値は Number 型という 1 種類しかない．このため，コンパイル時に検出した型の情報を用いて，型の違うもの同士での演算が発生した際に変換操作を行なうようにしている．これにより，整数同士で割り算をした場合に余りの部分を無視して整数の結果を得るなど（JavaScript では小数点つきの値になる）C 言語本来の挙動に近い動作を実現している．
- **変数とポインタ**
  C 言語においては，任意の変数についてそのポインタを取得することができる．一方，JavaScript はポインタと同等の機能は直接用意されていないため，オブジェクト[*2]への参照を用いてポインタ相当の機能を再現している．変換後の JavaScript においては，関数（正確にはブロック）ごとに変数を「変数オブジェクト」というデータ構造に格納しており，ポインタが必要な場合は「ポインタオブジェクト」を生成する．ポインタオブジェクトは，ポインタが指し示す対象の変数に対する操作（読み取りや書き込み）機能を提供する．**図 2** に例を示す．`main` 関数のローカル変数は `v_m` という変数オブジェクトに格納される（生成後の JS2 行目から 4 行目）．このうち変数 p についてはポインタオブジェクトを生成・参照し（4 行目），変数 a に対する操作を提供する．これを関数 f に渡して，f から変数 a の値を書き換える（10 行目）ことが可能である．

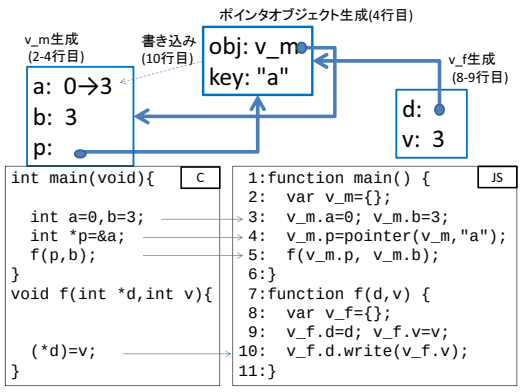

図 2 C-JS におけるポインタの表現

- **配列**
  C 言語における配列とは，メモリ空間のある領域を指すポインタであるが，JavaScript にはメモリ空間を直接扱う機能はなく，代わりに動的に確保可能な配列オブジェクトが利用できるため，これで配列を再現する．前述のポインタオブジェクトは配列の要素を指し示すことも可能であり，特定の要素をポインタを用いて読み書きすることや，そのポインタから指定されたオフセット分前後した要素を指し示すポインタを生成すること（本稿ではこれを「相対アドレッシング」と呼ぶ）も可能である．
- **構造体の値渡し**
  C-JS では構造体を JavaScript のオブジェクトで表現するが，C 言語の構造体はそれ自身が値であり，他の

[*2] JavaScript のオブジェクトの実体はキー（文字列）と値の対応づけであり，どのオブジェクトに対しても任意のキーと値とを複数格納可能である．また，キーと同名の文字列を用いて任意の値を読み書きできる．

変数へ代入したり，関数に渡したりするときにはコピーが発生する（値渡し）．一方，JavaScript のオブジェクトは参照であり，代入したりや関数に渡したりしたときにコピーが発生しない（参照渡し）．これらの差異を吸収するため，コンパイル時に型を判定し，構造体同士の値の受け渡しをする際には事前にコピーを行なうようにしている．

後述する授業においては，構造体が値渡しであることにより期待した動作をしない例 (**図 3**) を挙げたり，その解決法として，参照渡し（ポインタ）を導入する例 (**図 4**) を挙げたりしている．

```
#include <x.h>
typedef struct {
    float x, y, vx, vy;
} Rect;
void draw(Rect r) {
    fillRect(r.x,r.y, 20, 20);
}
void move(Rect r) { //四角形を動かす:NG
    r.x+=r.vx;
    r.y+=r.vy;
}
int main (void) {
    Rect r;
    r.x=10;r.y=20;r.x=2;r.y=3;
    while (1) {
        clear();
        draw(r);
        move(r);//NG(動かない)
        update();
    }
}
```

図 3 参照渡しと値渡しの違いを示す題材 (うまく動かない)

```
void move(Rect &r) { //四角形を動かす:OK
    r->x+=r->vx;
    r->y+=r->vy;
}
//---中略
        move(&r); //OK
```

図 4 参照渡しと値渡しの違いを示す題材 (修正方法)

- **同期処理の実行**

  C 言語は，処理中に I/O などで待ち時間が発生した場合に，任意の箇所で処理を中断し，入力などのイベントがあるまで待機した後に処理を再開するという，いわゆる同期処理を行なうことができる（scanf など）．

  一方，JavaScript はイベントの処理を非同期に行なう．すなわち，イベントに対するイベントハンドラを定義して，イベント発生時にイベントハンドラの呼出を行なう．さらに，イベントハンドラの処理中に待機をし，他のイベントを受け取ってから処理を再開する仕組みを基本的にはもっていない．そこで，C-JS では一部の Web ブラウザに搭載されているジェネレータ[*3] の仕組みを利用して，1 つの関数を複数のイベントに渡って呼び出せるようにしている．ただし現時点（2017 年 5 月）では非対応の Web ブラウザも残っているため，それらの Web ブラウザにおいては代替の命令（例えば，scanf は JavaScript の prompt 関数で代用）を用意したり，プログラムによっては正しく動作しなかったり（後述の update など）する場合もある．

4.2 初学者へのサポート機能

C-JS では，一般的な C 言語との互換性を保つだけでなく，初学者がつまずきやすい点に関しては独自にサポートを行なうようにした．

- **メモリの範囲外へのアクセス対策**

  C 言語のポインタは，相対アドレッシングを用いることで任意のメモリ領域にアクセスができるが，使い方を誤ると，正規に確保されたメモリ領域以外へのアクセスを行ってしまい，思わぬ動作を引き起こす．初学者にはその原因を特定するのが困難なことが多い．

  そのため，C-JS では相対アドレッシングを利用できるポインタは配列へのポインタだけに限定している[*4]．つまり，単純変数や構造体のメンバへのポインタは，生成はできても相対アドレッシングをしようとするとエラーが発生するようにしている．例えば**図 5** のプログラムはエラーになる．

```
typedef struct {
    int x,y;
} Vec2;
int main(void) {
    Vec2 v;
    int *pv=&v.x;
    *pv=3;
    pv++;//エラー（&v.y を指すことはできない）
    *pv=5;
}
```

図 5 相対アドレッシングが禁止される例

相対アドレッシングが可能な配列へのポインタにつ

[*3] https://developer.mozilla.org/ja/docs/Web/JavaScript/Guide/Iterators_and_Generators

[*4] 動的メモリ確保には未対応 (現状の C-JS は動的メモリ確保の仕組み自体が未実装)

いても，JavaScript の配列オブジェクトが実行時に要素数の情報を保持していることを利用して，相対アドレッシングによって配列の範囲外を指すポインタオブジェクトが生成され，それを使ったアクセスが行われた場合は実行を中断し，学習者に注意を促すようにしている．

- **無限ループ対策**
  学習者が無限ループを起こすようなプログラムを書いた場合，Web ブラウザが固まってしまい演習を続けることができなくなる場合がある．これを防止するため，実行されてからの経過時間をチェックし，5 秒以上経過すると自動的に実行を停止するようにしてある．このチェックを行なうための処理を，繰り返し構造のたびに挟み込んである．なお，前述のような途中で処理を中断し，次のイベントを待つ処理が実行された場合は，待っている時間は経過時間をカウントしないようにしてある．
- **初期化忘れ対策**
  C 言語は，ローカル変数に対して明示的に初期化を行わない場合の変数値は保証されておらず，いわゆる「ゴミ」の値が入っている．C 言語の動作を正しく学ばせるために，C-JS はこの動作も再現しているが，初学者にとってはゴミの値が表示されることで混乱を来す可能性があるため，代入や演算の処理中にゴミの値を検出した場合は変数が初期化されていない旨の警告を行なうようにしてある．

4.3 標準関数

現在，一般的な C 言語と共通して C-JS で使える関数は次のものになる．

- 標準入出力（stdio.h）は `printf` と `scanf` に対応している．`printf` のフォーマット指定は`%f`, `%d`, `%c`, `%s` に対応し，桁数指定子には未対応．`scanf` のフォーマット指定は`%f`, `%d`, `%c`, `%s` に対応し，桁数指定子には未対応．変数は 1 つだけ指定できる．
- 数値関数（math.h）は次の関数に対応している：`abs`, `acos`, `asin`, `atan`, `atan2`, `ceil`, `cos`, `exp`, `floor`, `log`, `max`, `min`, `pow`, `round`, `sin`, `sqrt`, `tan`
- 文字列関数（string.h）は次の関数に対応している：`strlen`, `strcpy`, `strncpy`, `strcmp`, `strncmp`, `strcat`, `strncat`, `memset`, `index`, `rindex`, `memcmp`, `memcpy`, `strstr`
- 標準ライブラリでは (stdlib.h) は `rand` が利用できる．`srand` は用意されていないが，乱数系列の初期化は自動的に行われる．

なお，これらの命令を利用するためには，対応するヘッダファイルを`#include<stdio.h>` などの形式でインクルードしないとエラーになる（一部の C コンパイラでは警告が出るだけの場合もあるが，C-JS では厳密に書かせるようにしている）．

表 1 グラフィックスライブラリの主な関数

関数	機能
fillRect	四角形を描く
fillOval	楕円を描く
clear	画面を消去する
setColor	描画色を設定する
drawGrid	座標を判別するための枠線を描画する
setPen	線分を描画する（開始点の指定）
movePen	線分を描画する（終了点の指定）
getkey	キーボードの押下状態を取得する
update	一定時間（1/60 秒程度）待機する

4.4 グラフィックスライブラリ

C-JS は標準関数のほかにグラフィックスやアニメーションの制作を支援するライブラリが用意されている．これは，HTML5 の Canvas 機能を利用して図形の描画を行なうものである．**表 1** にグラフィックスライブラリの命令を示す．なお，これらの命令を利用するには`#include<x.h>` というインクルードが必要になる．**図 6** にグラフィックスライブラリの使用例を示す．このプログラムでは，四角形が右に移動するアニメーションを表示している．

```
#include<x.h>
int main(void) {
    int x,y;
    setColor(255,0,0);
    x=30; y=50;
    while (x<300) {
       clear();
       fillRect(x,y , 50, 50);
       x+=2;
       update();
    }
}
```

図 6 アニメーションの例

4.5 未対応の機能

現状の C-JS では，次のような機能は未対応である．

- 関数ポインタ
- 共用体
- 列挙型
- プロトタイプ宣言における変数名の省略．例えば `void f(int, int);` は `void f(int x, int y);` と記述する必要がある．
- 引数つき`#define`，`#include` と`#define` 以外のプリプロセッサ命令

表 2 授業内容

回	学習内容	題材
1	入出力，変数，演算	日付の計算，グラフィックスの入門
2	繰り返し	合計計算，繰り返しの図形表示，アニメーション
3	条件判断	数当て，アニメーションの動作変更
4	関数	三角形の描画，面積の計算，再帰によるフラクタル描画（任意課題）
5	配列	複数の図形からなるアニメーション，2 次元配列（任意課題）
6	構造体	前回のプログラムを構造体に置き換える
7	ポインタ・参照渡し	ゲーム制作:プレイヤーと敵の作成
8	マクロ・配列の参照渡し	ゲーム制作: 衝突判定
9	並べ替え	ゲーム制作: スコアランキング
10	ファイル	ゲーム制作: スコアランキングの保存
11	文字列	ゲーム制作: ランキングに名前の登録
12	配列の探索	ゲーム制作: 弾の発射
13	総合	作品制作
14	総合	コンテスト（プログラミングの問題を複数出題し解答数を競う）
15	総合	まとめ

- 三項演算子

5. 授業実践

現在，明星大学の情報学部における「プログラミング I」という授業において C-JS を使用している．授業内容を**表 2** に示す．本授業は，2 年生前期の授業であり，1 年次ですでに学んでいる C 言語の基本（変数，条件分岐，繰り返し，配列など）の発展として，関数，ポインタ，構造体の概念を学習することを目標としている．なお，今回の受講者は 45 名であり，授業中には TA2 名によるサポートを行っている．この原稿の執筆時点で 5 回までが終わっている．また，この後の授業で扱うプログラムについても動作を確認済である．

5.1 課題の出題・採点

授業は 2 時限続きで行われ，最初の時限に学習内容についての説明を行い，次の時限に課題を解く演習を行っている．課題には，全員に解答を要求する基本課題と，余力のある受講者向けの任意課題がある．

課題の採点は，予め課題を作成するためのソースコードを BitArrow の教員支援機能により配布し，学生が編集した後のプログラムをまとめて実行し，出力結果とソースコードの内容を教員がチェックして行う．採点作業は当日の演習中に行っており，学習者に即日フィードバックされるようにしている．ただし，現状ではプログラムをまとめて実行する指示は手動でコマンド入力している．これらの詳しい仕組みについては，本シンポジウムの別の発表 [7] を参照されたい．

4 回目までに基本課題 16 題を出題した．基本課題の題数×受講者数 (16 × 45=720) に対する，完了した基本課題の延べ数は 609(81%) である．ただし，その週のうちに課題が終わらない人は次回以降に着手したり，授業外に行ったりしてもよいことにしており，すべての課題の締切日は半期の授業全体が終わるまでとしてあるため，現時点で完了していない課題も今後完了していくと考えらえる．また，毎回，受講者の約半分にあたる 20 人余りが任意課題にも着手している．

TA が受けた質問の内容としては，課題のプログラムの書き方自身がわからないというものがもっとも多く，次いで，スペルミスに起因する未定義の関数・変数の直し方，演算時や関数に値を渡すときの型の不一致に関するエラーの直し方などの質問が多く，単純なセミコロン忘れなどのエラーについての質問はほとんど出ていなかった．また，配列の範囲外のアクセスの通知については，TA が原因を素早く特定するのに有用であった．

5.2 ログの分析

BitArrow は利用者の作成したプログラムを逐一ログの形式で Web サーバに保存する機能をもっている．ログのデータは，利用者がプログラムファイルを保存したり，プログラムを実行したりしたときに送信される．送信される 1 つのログ項目に含まれるものは次の通りである．

- 実行または保存を行った日時（タイムスタンプ）
- ファイル名
- 保存されたプログラムのソースコード
- コンパイルエラー・実行時エラーの有無．ある場合はエラーメッセージ

今回の授業の受講者のログ（第 4 回目まで）に含まれるデータから，次のことを調査した

- コンパイルエラーの発生する頻度とその原因
- コンパイルエラーからの復帰時間と，復帰時間が長いエラーの原因
- 授業時間外のアクセス状況

表 3 コンパイルエラーの内訳（第 4 回まで）

コンパイルエラーの有無・種類	ログの件数 (授業時間内)	ログの件数 (授業時間外)
コンパイルエラーなし	12450(77%)	413(88%)
文法エラー	1689(11%)	35(35%)
未定義の変数・関数	1140(7%)	14(3%)
二重定義	450(3%)	6(1%)
引数の数が違う	315(2%)	2(1%未満)
その他	32(1%未満)	0
合計	15606	470

表 4 エラー復帰時間の分布

復帰時間の区間 (秒)	件数	累積%
≤ 30	750	51%
30-60	209	65%
60-90	110	72%
90-120	68	77%
120-150	60	81%
150-180	40	84%
180-210	19	85%
210-240	19	86%
240-270	17	87%
270-300	18	89%
>300	168	100%
合計	1478	-

5.2.1 コンパイルエラーの発生する頻度とその原因

表 3 に，コンパイルエラーの内訳を示す．今回の授業の受講者によって生成されたログ項目は，第 4 回終了後の 2017 年 5 月 15 日の時点で 16076 件であり，コンパイルエラーが発生した旨を含むログ項目（以下，「エラーログ項目」と呼ぶ) はそのうちの 3626 件（全体の約 23%）であった．その原因として多いのは記号の入力忘れなどに起因する「文法エラー」であり，次いで変数定義忘れ，スペルミスに起因する「未定義の変数・関数」であった．

5.2.2 コンパイルエラーからの復帰時間

コンパイルエラーが連続して発生しているのは，受講者がプログラムの間違いの原因を解決できないでつまずいている可能性を示している．このような状態がどの程度発生しているかを調べるために，次のように「コンパイルエラーからの復帰時間」を定義した．

- ある受講者 1 人のログを時系列に見たとき，エラーログ項目が連続している箇所を「エラー列」と定義する．
- あるエラー列の最初のエラーログ項目のタイムスタンプと，そのエラー列の直後に出現するエラーログ項目でない（コンパイルエラーがない）ログ項目のタイムスタンプとの差を，そのエラー列の「復帰時間」と定義する．

全受講者のログに登場したエラー列について，その復帰時間の分布を**表 4** に示す．「累積%」の項目は「件数」を上からその区間まで足した場合の，全体 (1478 件) に対する割合である．例えば，復帰時間が 120 秒以内であったものは 77%と読み取ることができる．なお，表 4 のうち，時間外のアクセスにおけるエラー列は 6 件で，復帰時間はそれぞれ 380, 350, 476, 646, 758 ,3324 秒であった．3324 秒かかっているエラー列は，5 個のログ項目からなり，最初と 2 番目のログ項目が 50 分ほど開いており，残りの 4 個は 4 分以内に収まっていた．

約半分のエラー列については 30 秒以内に復帰ができている．一方，復帰に 5 分 (300 秒) 以上を要したエラー列も 10%ほど存在している．復帰に 5 分以上かかっているケースを個別に調べた結果，次のような原因で時間がかかっているケースが見つかった．

- **`#include` し忘れによる「未定義の変数・関数」**
  先述した通り，標準関数を利用するには対応する `#include` を追加しないと，関数が未定義である旨が表示される．例えば，`#include<stdio.h>` を書かずに `printf` を呼び出そうとすると「printf は定義されていません」と表示され，さらに `printf` を書いた行の付近を調べるように促される．しかし，実際には `printf` を書いた行は間違っていないため，修正方法がわからなくなってしまっていたと考えられる．
  そこで，5 回目以降は，標準関数を使おうとしているときに対応する`#include` がない場合はそれを追加するように促すメッセージを表示するようにしている．
- **大文字小文字の違いによる「未定義の変数・関数」**
  C 言語では変数・関数名の大文字小文字を区別するが，学習者が大文字小文字の差を意識しないで書いていると，例えば `int day;Day=3;` のようなプログラムを書いたときに「Day は定義されていません」というメッセージが出たとしても，「day は宣言したはずだ」と思ってしまい，修正のしかたがわからないことがある．そこで，5 回目以降は，未定義の変数・関数があり，かつ，大文字小文字を区別しないと仮定して同じ変数が見つかった場合，大文字小文字を区別するよう表示をしている．上の例では「Day は宣言されていません．大文字小文字は区別されます．day の間違いではないですか？」と表示されるようにしてある．

5.2.3 授業時間外のアクセス状況

BitArrow は Web ブラウザが動作する環境であれば，どこでも演習ができるため授業中に終わらなかった課題などを授業時間外に行なうことも可能である．そこで，ログから授業時間外のアクセスとみられるログ項目を集計した (ただし，授業終了後 1 時間以内は教室に残って作業している可能性が高いため，授業時間外には数えていない)．

その結果，5 月 17 日時点での 16302 件のログ項目中，696 件のログ項目が授業時間外におけるアクセスであった．また，1 度でも授業時間外にアクセスを行った受講者は 45 人中 13 人であった．その活動内容は主に，授業中に終わら

なかった任意課題を解いたり，C-JS の機能を（授業で触れた機能以外に何ができるか）試したりするものであり，プログラミングの能力が高い中上級者の受講者によるアクセスが多かった．

一方で，授業時間外に数回アクセスしてやめている受講者もいることがわかった．授業で触れた以外の機能を試したところ，対応していない機能 (malloc や ファイル操作関数など) を利用しようとしてエラーが出てしまったためと考えられる．

6. 授業実践の考察

今回の実践は本稿執筆時点で継続中であるが，現時点での実践の考察を行う．

まず，言語処理系 C-JS は，表 2 に挙げたような，半期の授業で教える C 言語の要素が動作することを確認しており，授業での実用に耐えうるものとなっている．4.5 で挙げたような未対応の要素もあるが，C 言語の基礎を教える際に利用される頻度は少ないため，今後必要に応じて実装を続けていく．

演習中に受講者が起こしやすい間違いについては，C-JS のエラーメッセージの表示を工夫することで，ある程度サポートをしているが，それでも長時間エラーが出て先に進めない受講者がいることがログの分析結果からわかった．エラーの量・内容は単元が進むにつれて変わってくると考えられるため，今後もログ分析を続けて随時 C-JS の機能に追加し，機能追加以降のログを見て，エラーからの復帰時間の改善ができているかどうかを検証する．

また，標準の C 言語をそのまま学習させるだけではなく，グラフィクスやアニメーションを題材としているのことで受講者が興味をもって演習を行えていると考えられる．1 年次に受けた授業では標準の C 言語を用いていたため，授業終了後にアンケートをとって，授業への興味関心の変化を調査する予定である．

課題を解かせる演習においては，その成否を素早くフィードバックすることができ，その日のうちに課題を終わらせることができる学習者が多い．一方，現状では，課題の進捗状況の把握や採点結果の通知がリアルタイムには行えていないため，採点結果が受講者に遅れて届いたり，教員の採点中に受講者が対象のプログラムを書き換えてしまったりして，採点結果を正しく伝えられないケースも見受けられる．今後，UI や通知の仕組みを改善していく．

最後に，BitArrow の特徴である Web ブラウザのみでどこでも実行できる点が活かされているかどうかを，授業時間外のアクセスの有無で検証したところ，一部の中上級者による時間外のアクセスを認めることができたが，初級者は授業時間外の 1 人での対処が難しいことも読み取れた．今後は初心者の間違いやすい点を自動的に指摘してくれるような機能の開発も必要となってくる．

7. まとめ

本発表では，Web ブラウザ上で動作可能な学習環境 BitArrow における C 言語の処理系である C-JS の機能を紹介し，実際の授業で使用した経過を報告した．

BitArrow の設計方針は，学習の対象となる言語で書かれたプログラムを JavaScript に変換することにより，Web ブラウザ上でプログラムを実行させるものであり，C-JS もその方針に則って作られている．これにより，従来の C 言語の機能に加えて，グラフィックス機能などの Web ブラウザ特有の機能を利用した演習を可能としている．また，学習者の間違いやすい誤りに対してわかりやすく指摘する機能を有している．

これらの機能を題材にしつつ，C 言語の主要な要素である構造体やポインタなどの学習項目を網羅した授業を展開することができている．授業はまだ継続しており，授業終了まで継続して C-JS の改善，およびログなどのデータをもとに改善の効果を検証していく．

謝辞 本研究は JSPS 科研費 17K00989 の助成を受けたものです。

参考文献

[1] 長島和平，本多佑希，長 慎也，間辺広樹，兼宗 進，並木美太郎：オンラインで複数言語を扱うことができるプログラミング授業支援環境，情報教育シンポジウム 2016 論文集，Vol. 2016, pp. 1–9 (2016).

[2] Papancea, A., Spacco, J. and Hovemeyer, D.: An Open Platform for Managing Short Programming Exercises, *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*, ICER '13, New York, NY, USA, ACM, pp. 47–52 (online), DOI: 10.1145/2493394.2493401 (2013).

[3] Denny, P., Luxton-Reilly, A., Tempero, E. and Hendrickx, J.: CodeWrite: Supporting Student-driven Practice of Java, *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education*, SIGCSE '11, New York, NY, USA, ACM, pp. 471–476 (online), DOI: 10.1145/1953163.1953299 (2011).

[4] Guo, P. J.: Online Python Tutor: Embeddable Web-based Program Visualization for Cs Education, *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, SIGCSE '13, New York, NY, USA, ACM, pp. 579–584 (online), DOI: 10.1145/2445196.2445368 (2013).

[5] 三浦元喜 : Processing Web IDE を用いたプログラミング基礎教育の試み，情報教育シンポジウム 2013 論文集，Vol. 2013, No. 2, pp. 225–231 (2013).

[6] 白井達也 : Processing 上で古典的なコンソール対話型プログラミングから高度なグラフィックスプログラミングまで学習可能なフレームワーク Crowbar+Tomahawk の紹介，情報教育シンポジウム 2014 論文集，Vol. 2014, No. 2, pp. 227–232 (2014).

[7] 長島和平，堀越将之，長 慎也，間辺広樹，兼宗 進，並木美太郎：プログラミング学習支援環境 Bit Arrow の教員支援機能の設計と試作，情報教育シンポジウム 2017 論文集（発表予定），(2017).