

ソフトウェア開発プロセスの体得方法としての プログラミングによる動画制作

中 島 豊四郎
松 山 智恵子
岩 田 員 典

1. はじめに

近年、大学におけるプログラミング教育は理系学部にとどまらず、文系の様々な学部において情報教育の中の一つの科目として実施されている。このようにプログラミング教育が一般化している中であって、教育を受ける側の学生の中には、プログラミングに興味を持たない者も多々見受けられ、そのため魅力あるプログラミング教育のための種々の試みがなされている¹⁻⁶。一方、プログラミングは、ソフトウェア開発の一つプロセスであるものの、この視点に立ったプログラミング教育までには至っていない。

筆者も担当している文系女子学生のプログラミング教育において、プログラミングに興味を持たせるため理系的題材に代えて、アニメーション的動画を題材に取り入れることによって、履修者が積極的にプログラミングに取り組みことができることを明らかにしてきた^{7-9, 11}。また、履修者のプログラム作成力の定量的評価について提案した^{10, 11}。しかし、プログラミングがソフトウェア開発の一つプロセスであることを、動画制作を通して体得できることについては

言及していなかった。そこで、ここではプログラミングによる動画制作がソフトウェア開発の一つプロセスであることを体得させることができることについて述べる。

2. プログラミング演習の目的と概要

2.1 プログラミング演習の目的

プログラミングが初めてである文系女子学生に対して、事務系、制御系に限らず広範囲な分野で利用されている C 言語を用い、特に数値計算などの難しい数学的知識を必要としないで、かつ、学生が興味を持ってプログラミングに取り組めるように、プログラミング演習の中でグラフィックス関数を使い、課題として動画を制作することを試みた。また、この動画制作においては、言語の習得を目的とするのではなく、プログラムで作成する対象物の仕様作成から設計、コーディング、デバッグ等の一連のプロセスとその繰り返し、すなわち図 1 に示すようなシステム進化の過程全体¹²を理解し、体得させることを目的とした。

2.2 プログラミング演習の概要

動画課題を取り入れたプログラミング演習は女子大学文系学科の情報関係科目の選択科目の一つとして、3 年次前期（2 コマノ週）に開講されている。その概要を表 1、演習内容を表 2 に示す。

本演習の履修者は、40 名で、Word、Excel 等のアプリケーションソフトの使用経験はあるが、プログラミングについては初学者がほとんどである。言語処理系として、Ultra-C Pro を用いたのは、本ソフトが基本的なグラフィックス関数をサポートしていることとインタプリタ方式で、1 ステップ実行やトレース、変数のモニタリング等のデバッグ機能を有し、かつソースプログラム、標準入出力、グラフィックス、エラー画面等が独立したマルチウィンドウになっ

ソフトウェア開発プロセスの体得方法としてのプログラミングによる動画制作

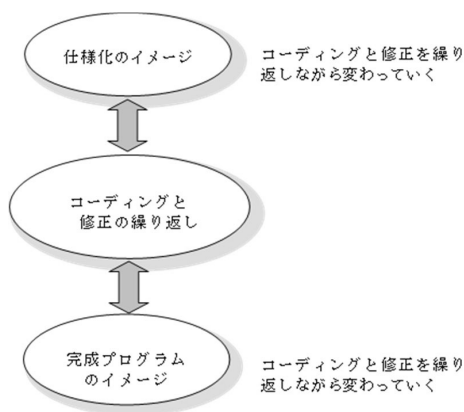


図1 システム進化（開発）のプロセス

表1 演習の概要

学 生 数	3年次 40名
総 時 間 数	2コマ (1.5h×2) / 週 ×14回
T A	2名
言語処理系	Ultra-C Pro (株式会社ソフトボード) Windows 環境

ており、初学者でも扱いやすいためである。パソコン (PC) は履修者 1人1台で、PC 2台に1台の割合で、教員の PC 画面や資料、また、学生の PC 画面が配信できるモニタ (教材モニタ) が学生の PC の間に配置されている。演習の進め方は、前半 (1コマ目) と後半 (2コマ目) の始めに、その時間に学習する内容の説明をし、その後、履修者がそれに従って演習をする。TA (Teaching Assistant) は2名で、履修者の質問に適宜対応できるようにしている。演習の内容は、表2に示すように9回目の前半までは、プログラミングの概説、言語処理系の利用方法、C言語の基礎を毎回単元毎に説明し、それぞれ簡単な例題を与える方式であるが、9回目の後半より、動画制作の基本とな

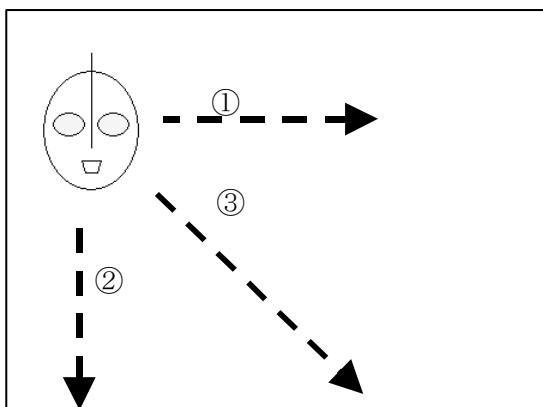
表2 演習の内容

回	内 容
1	履修上の注意、演習の進め方、プログラミングとは、アルゴリズム・プログラム・プログラミング言語の関係、プログラム言語、C言語とは、Ultra-Cの使い方、Cプログラムの概要
2	Ultra-Cの操作方法、Cプログラミングの基礎、定数と変数
3	変数、データ型、画面への出力
4	キーボードよりの入力、演算子
5	演算子
6	分岐、繰り返し、関数
7	関数の作り方
8	記憶クラスと通用範囲、配列（数値）
9	配列（文字）、C言語のまとめ、グラフィックスの説明、課題（C言語の基礎的な問題と動画制作）の提示、動画およびグラフィックスのテキスト配布
10	グラフィックス（動画の例題）、Web教材の提示
11	動画制作
12	動画制作
13	動画制作
14	動画およびソースプログラムの発表、アンケート調査など

るグラフィックスの説明と例題（動画の例を含む）を1.5回行い、その後課題として「動画制作」を課した。動画の制作には3回（3H×3=9H）をあてた。ここで、第1回目の演習時に教員は、「プログラミングは難しい」と言う先入観を捨てることと「プログラミングについて何も知らなくても大丈夫、これから少しずつ身に付けていく」と言うことを強調した。演習の最終回には、履修者全員の制作した動画とそのプログラムリストを各自のPCから教材モニタへ配信しながら、発表を行った後、この動画制作を通して学生が興味を持ってプログラミングに取り組めたか、また、プログラムの作成プロセスを理解できたか等についてアンケートを実施した。

3. 動画制作

動画制作の前に、グラフィックスの基礎として、グラフィックスに関する用語、色や線種の種類等を表すマクロ名、関数の種類の説明をし、すべての関数について一通りの使い方を練習させた。その後、図2に示す例題（ソースコードを図3に示す）で静止画を動いたように見せる動画の制作方法を学習させ、課題制作に取り組みさせた。



《動画の例題：ウルトラマンの顔を動かす》

ウルトラマンの顔が順次3つの方向へ移動する。

始点は画面左上とする。

始点から右へ移動し、

始点から真下へ移動し、

始点から右斜め下へ移動する。

(注) Ultra-C Pro のグラフィックス画面は、座標 (0, 0) は画面左上、
上図 が X 軸の正方向、 が Y 軸の正方向となる。

図2 動画の例題

<ソースプログラム>

```
#include "graphic.h"
void Ultraman(int i, int j);

void main()
{
    int i, j, k;

    for(i = 100; i < 500; i += 10) /* 右へ移動 */
    {
        Ultraman(i, 100);
        for(k = 0; k < 10000; k++) ;
        DrawRectangle(i, 100, i+85, 215, WHITE, PSOLID, 1, WHITE, BSOLID, 0);
    }
    for(j = 100; j < 500; j += 10) /* 下へ移動 */
    {
        Ultraman(100, j);
        for(k = 0; k < 10000; k++) ;
        DrawRectangle(100, j, 185, j + 115, WHITE, PSOLID, 1, WHITE, BSOLID, 0);
    }
    for(i = 100, j = 100; i < 500, j < 500; i += 10, j += 10) /* 右斜め下へ移動 */
    {
        Ultraman(i, j);
        for(k = 0; k < 10000; k++) ;
        DrawRectangle(i, j, i + 85, j + 115, WHITE, PSOLID, 1, WHITE, BSOLID, 0);
    }
}
/* ウルトラマンの顔を描く関数 */
void Ultraman(int i, int j)
{
    POINT point[4];

    DrawEllipse(i, j+15, i+80, j+115, BLACK, PSOLID, 1, WHITE, BSOLID, 0); /* 輪郭 */
    DrawEllipse(i+8, j+50, i+35, j+70, BLACK, PSOLID, 1, YELLOW, BSOLID, 0); /* 右目 */
    DrawEllipse(i+45, j+50, i+72, j+70, BLACK, PSOLID, 1, YELLOW, BSOLID, 0); /* 左目 */

    point[0].x = i + 32;   point[0].y = j + 90;   /* 口の座標を指定 */
    point[1].x = i + 48;   point[1].y = j + 90;
    point[2].x = i + 45;   point[2].y = j + 100;
    point[3].x = i + 35;   point[3].y = j + 100;
    DrawPolygon(point, 4, BLACK, PSOLID, 1, WHITE, BSOLID, 0); /* 口 */
    DrawLine(i + 40, j, i + 40, j + 80, BLACK, PSOLID, 1); /* 角 */
}
```

図 3 動画例題のソースプログラム

3.1 グラフィックス関数

静止画を作成するためにシンプルなグラフィックス関数、11個を Ultra-C でサポートされている関数を基に履修者が使いやすいように再定義した。それらの関数を表 3 - 1、3 - 2 に示す。

3.2 課題の提示と動画の制作

課題として、全履修者に対して動画を制作することを課した。内容は自由とし、制作の目標として通常の授業の演習量から想定し、図 2 に示す例題 (図 3 より LOC : 31、動画のファンクションポイント* : 4 *後述) 以上のものとした。履修者はこの指示に基づいて、各自動画の仕様を考え、それをコード化し、デバッグをし、動画を完成させる。完成までの過程においては、仕様で考えたことが用意されたグラフィックス関数では実現できないことや、もっと出来映えをよくする等の理由から、図 1¹²に示すような仕様の見直し、プログラム (コード) 変更、デバッグが何回も繰り返し行われていることが観察された。また、制作に集中している姿や隣同士教え合ったりする履修者もみられた。さらに、完成した時の満足感や充実感も観察された。履修者が制作した動画の概要を表 4 - 1 から表 4 - 3 に示す (同表には後述の LOC およびファンクションポイントによる動画制作の評価を併記してある)。表 4 - 1 から表 4 - 3 より、制作された動画は一人ずつ異なっており、履修者の一人ひとりが他人のプログラムを真似しているのではなく、各自オリジナリティを發揮しながら、動画を制作していることがわかる。このことは、履修者が興味を持って動画作成に取り組んだことを示し、この点からも動画制作がプログラミングに取り組むための動機付けになっていると評価できる。

表 3 - 1 グラフィックス関数 (その 1)








No.	関数	機能・説明	例
1	DrawBackground (色)	機能：画面を塗りつぶす。 説明：グラフィックス画面全体を指定した色で塗りつぶす。	
2	DrawPixel (x, y, 色)	機能：点を描く 説明：座標 (x, y) に指定した色で点を描く。	
3	DrawLine (x1, y1, x2, y2, 色, 線の種類, 線の幅)	機能：線を描く 説明：座標 (x1, y1) から座標 (x2, y2) に指定した色で線を描く。	
4	DrawRectangle (x1, y1, x2, y2, 輪郭線の色, 輪郭線の種類, 輪郭線の幅, 内部の色, 内部のスタイル, ハッチング)	機能：長方形を描く 説明：左上座標 (x1, y1)、右下座標 (x2, y2) を対角の頂点とする長方形を描く。	
5	DrawEllipse (x1, y1, x2, y2, 輪郭線の色, 輪郭線の種類, 輪郭線の幅, 内部の色, 内部のスタイル, ハッチング)	機能：円・楕円を描く 説明：左上座標 (x1, y1)、右下座標 (x2, y2) を対角の頂点とする長方形に内接する楕円を描く。	
6	DrawRoundRect (x1, y1, x2, y2, w, h, 輪郭線の色, 輪郭線の種類, 輪郭線の幅, 内部の色, 内部のスタイル, ハッチング)	機能：角の丸い長方形を描く 説明：左上座標 (x1, y1)、右下座標 (x2, y2) を対角の頂点とする長方形に内接する小さな楕円を描き、角の丸い長方形を描く。	
7	DrawArc (Lx, Ly, Rx, Ry, x1, y1, x2, y2, 線の色, 線の種類, 線の幅)	機能：円弧を描く 説明：座標 (Lx, Ly)、座標 (Rx, Ry) を対角の頂点とする長方形の内接する楕円の円弧を描く。座標 (x1, y1) と楕円の中心を結ぶ直線が交わる点 a が円弧の開始点に、座標 (x2, y2) と楕円の中心を結ぶ直線が交わる点を b が円弧の終了点になる。	

表 3 - 2 グラフィックス関数 (その 2)



8	<p>DrawChord (Lx, Ly, Rx, Ry, x1, y1, x2, y2, 輪郭線の色, 輪郭線の種類, 輪郭線の幅, 内部の色, 内部のスタイル, ハッチング)</p>	<p>機能：弓形を描く 説明：座標 (Lx, Ly)、座標 (Rx, Ry) を対角の頂点とする長方形の内接する楕円の弓形を描く。座標 (x1, y1) と楕円の中心を結ぶ直線が交わる点 a が円弧の開始点に、座標 (x2, y2) と楕円の中心を結ぶ直線が交わる点を b が円弧の終了点になる。</p>	
9	<p>DrawPie (Lx, Ly, Rx, Ry, x1, y1, x2, y2, 輪郭線の色, 輪郭線の種類, 輪郭線の幅, 内部の色, 内部のスタイル, ハッチング)</p>	<p>機能：扇形を描く 説明：座標 (Lx, Ly)、座標 (Rx, Ry) を対角の頂点とする長方形の内接する楕円の扇形を描く。座標 (x1, y1) と楕円の中心を結ぶ直線が交わる点 a が円弧の開始点に、座標 (x2, y2) と楕円の中心を結ぶ直線が交わる点を b が円弧の終了点になる。</p>	
10	<p>DrawPolygon (配列の名前, 頂点の数, 輪郭線の色, 輪郭線の種類, 輪郭線の幅, 内部の色, 内部のスタイル, ハッチング)</p>	<p>機能：多角形を描く 説明：POINT 配列で指定された n 個の頂点を順に結ぶ多角形を描く。</p>	
11	<p>DrawText (x, y, “出力する文字”, 文字の色, “フォントの名前”, フォントサイズ, 文字の種類, 下線)</p>	<p>機能：文字を出力する 説明：座標 (x, y) に指定された色、フォント、フォントサイズで文字を出力する。</p>	<p>ABC</p>

表 4 - 1 履修者が制作した動画プログラム (その 1)

No.	タイトル	内 容	LOC	FP
1	不思議な ステージ	小玉が画面のふちに並べられていく。 2人のピエロが玉乗りをしながら、玉をけりあう。けている玉だけが左右に飛び交う。 画面ふちの小玉が順に消えていく。	157	17
2	ドラえもん	ドラえもんが絵描き歌のように順に描かれる。 ドラえもんがドラ焼きを食べ、 右へ竹コプターで飛んで行く。 画面中央に「おわり」を表示し、後姿のドラえもんが上の方へ飛んで行く	229	12
3	くすだま	クスダマが割れ、紙ふぶきが散り、 垂れ幕があり、「お疲れ様！」の表示。	64	9
4	UFO	UFO が飛んできて、 小包を落とす。 小包が地面へ届くと、「Happy Birthday!」の文字が表示される。	80	12
5	おにぎり	おにぎりが水溜りに落ちていくのをピンクの熊が傍らで見ています。	30	7
6	クジラ	くじらがしおをふくのくりかえす。	69	10
7	魚	魚が右へ泳ぐ。 魚が左へ泳ぐ。	90	11
8	ひよこ	ひよこがとことこ歩いてきて、 木にぶつかり、 りんごが落ちる。	47	11
9	お花	お花の横を4匹の蝶々が飛んでいくと同時にクモも動く。 夜になると月と星がでる。	105	12
10	雪だるま	雪だるま (は動かない) と降ってくる雪。	92	7
11	ちょうちょ	帽子のわきを右下へ蝶が飛んでいき、 次に蝶が右上へ飛んでいく。	24	5
12	気球	空を気球がふわふわととんでいく。	120	7
13	夜空に UFO が	白 UFO 点滅 水色、ピンク、紫 UFO 点滅 の UFO が動く パールグリーン、プラム UFO 点滅 の UFO が動く 水色 UFO 点滅 オレンジ UFO 上に上がる。(たくさんの UFO が夜空に点滅している。そのうち、複数の UFO があちらこちらに飛んでいく。)	312	16
14	空飛ぶ ・・・!?	風船が上へ飛んで、 気球が左から右へ飛んでいく。 その後、右から左へピンクの動物が飛んでいく。	46	9
15	うさぎの シャボン玉	うさぎがシャボン玉を膨らまそうとして 一回目は失敗、 2回目はうまく膨らみ、 上へ飛んで行く。	81	21

ソフトウェア開発プロセスの体得方法としてのプログラミングによる動画制作

表 4 - 2 履修者が制作した動画プログラム (その 2)

16	くま	くまとりんごの木。 太陽が昇り、 りんごが落ちて、くまの手の上に。	141	12
17	冬の夜空	冬の夜空に星が見える。星を線で次々と結び、星座の名前を表示して行く。 こいぬ座 おおいぬ座 オリオン座 冬の大三角	149	19
18	「星空」	星空の星が一通り点滅し、 流れ星がひとつ流れる (わかりにくい)	115	7
19	色玉	同心円の半径をだんだん小さくし、カラフルな色で見せる。 その後、星が流れ、 半径をだんだん大きくし、もとの円の大きさまで描く。 その後、クスダマがわれ、が出る。	176	22
20	ブクブク ワンワン	風船を持った犬。 風船が飛んでいく。 おなかのあたりで別の風船が膨らみ、 しぼむ。	41	8
21	うららか	雪だるまのカップルがキスすると溶けてしまう。 太陽と虹と蝶々、花がでて「春がきた」と表示する。	214	20
22	まる	四角の線で囲まれた通路を 2 つの が通過し、途中出会うが、通過してゴールまでいく。	142	12
23	ドラえもん 気球の旅	気球に乗ったドラえもんを、 竹コプターをつけたねずみが追いかける。 途中で背景が夜に変わる。	61	7
24	くじら	くじらがしおを吹いた後、 海面へ沈んでゆき、 ふたたび浮かんでくる。	64	8
25	宇宙飛行	と UFO がとんでいく。 上陸	136	11
26	うさこ	うさこ (うさぎ) が持っていた風船を鳥が飛んできて割ってしまう。 うさこが泣く。 そのあと鳥が風船を持って飛んできて、うさこに渡す。 うさこは風船とともに空へ飛んで行く。	177	15
27	陽子さんへ	ドアを開けて人が入ってきて、 台の乗り、 紐を引っ張ると クスダマが割れて、「陽子さんお誕生日オメデト」の垂れ幕がでる。	424	22
28	ピリヤード	白い玉をつくると、 9 ボールが散りころがる。	92	6
29	ねずみ	ねずみの親子がチーズに寄っていく。 はねかえる。 また近づく。	52	8
30	パンダちゃん	パンダがじゃんけんを繰り返す。	88	9

表 4 - 3 履修者が制作した動画プログラム (その 3)

31	koala	3匹のコアラがスピードを変えて、順に緑の綱を 下り、上る。最後は寝ている。	173	18
32	星座	星座を表示し、上へ移動。赤い線が出て、太陽がでる。	278	10
33	たまご	たまごにひびが入り、何が生まれるかなと期待させて、パンダが生まれる。文字が大きくなる	158	11
34	雪だるま	白い玉が左から動いてきて中央でとまる。上からやや小さめの白い玉が動いてきて、雪だるまができる。横一列にならんだ雪のつぶが 降ってくる。 2回	55	10
35	花火	同心円の半径を順に描いていき、花火にみたてである。花火が複数出る。	155	7
36	ボーリング	ボーリングのボールが転がってくる ピンを全部倒して「ストライク」	40	7
37	cake	ショートケーキの上に立っているローソクの炎が 点滅し、まわりがピカピカする。	96	9
38	折り紙	折鶴を順に折っていく過程の図を見せる 完成した鶴が飛んでいく。	564	19
39	作品	雪だるまと月。雪の粒が点滅	26	6
40	海	船とカモメ。海面が動く。船の汽笛から煙がもくもくと出る。トビウオが海面から出る。	85	13

4. プログラム作成力の評価方法と結果

履修者がどのくらいプログラムを作成する力がついたかを客観的にみるために、ソフトウェアの生産性を計る指標として用いられている LOC^{13,14}とファンクションポイント¹³⁻¹⁸の2つの方法で評価した。ここで、ファンクションポイントについては、動画を単機能に分解し、評価基準を定義した。

4.1 LOC による評価

表 4 - 1 から表 4 - 3 に示した履修者が制作した動画のソースリストの LOC

ソフトウェア開発プロセスの体得方法としてのプログラミングによる動画制作

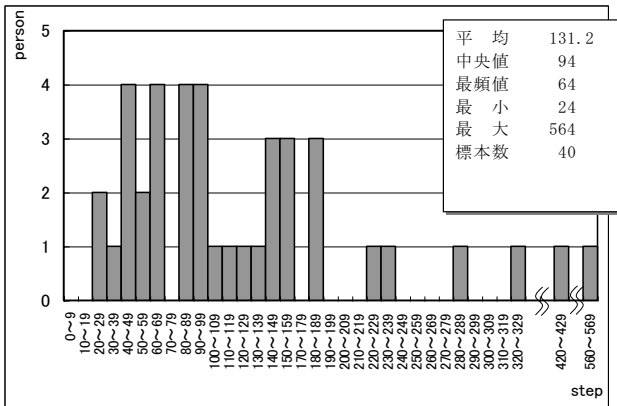


図 4 LOC の分布

を度数分布にしたものを図 4 に示す。ここで、LOC は、ソースの総行数から空行、コメント行、余分な ‘ { ’ または ‘ } ’ の行を除いたものである。たとえば、図 3 に示した動画の例題のソースプログラムの場合、(総行数：42) + (1 行に 2 命令の行：4) - (空行、コメント行、{、} の行：16) から LOC は 31 ステップと計数する。

表 4 - 1 から表 4 - 3、図 4 より、全履修者の LOC の平均値は 131.2、中央値は 94、最頻値は 64 であり、また、最小値は 24、最大値は 564 と LOC は広範囲に分布していることがわかる。

動画制作における LOC の目標は、図 3 の例題 (LOC : 31) を超えることとしたが、目標を超えた履修者は全体の 92.5% である。また、LOC が目標の約 3 倍である 100 を超える履修者は約半数の 47.5%、200 を超える履修者は 6.7% である。本課題が仕様作成、検討を含めて全てを履修者自身がクリエイトする必要があることを考慮すれば、これらの履修者はプログラミングの初学者で、かつ制作期間が短いにもかかわらず、相当のプログラムを作成する力がついたと言える。このことから、履修者はプログラミングに興味を持ち、いかに積

極的に取り組んだかがわかる^{10,11}。

4.2 ファンクションポイントによる評価

履修者がプログラミングにより作成した動画の複雑さを評価するために、定義した動画のFPを表5に示す。ソフトウェア工学におけるFP法¹³⁻¹⁸は、ソフトウェアの規模をそれが実現する機能 (Function) の数によって見積る方法で、ソフトウェアの機能を入出力の数やマスターファイルの数として捉えるものである。ここでは、動画を形成する複雑さを機能として数えるために、表5に示すような8つの機能を定義し、評価基準とした。動画を形成するためのファンクションは、描画しているオブジェクトの数、オブジェクトの移動回数、オブジェクトの色の変化、複数のオブジェクトの描画、オブジェクトを順次描き足し、オブジェクトの一部だけの変化などに分類し、その評価基準に従って、履修者の動画がいくつのファンクションで構成されているかを集計している。ただし、FPの定義において、見る人の感性のよって評価が異なると思われる描画の美しさや動画のストーリーのおもしろさ等については除外している。

表5の基準により全履修者の動画(表4-1から表4-3参照)を評価し、それを度数分布に表したものを図5に示す。図5より最小値は5、最大値は22と広範囲に分布していることがわかる。ここで、例題の動画は、一つの図形が移動を繰り返したもので、そのファンクションポイントは4であるが、履修者の作成した動画のファンクションポイントの平均値は11.6、最頻値は7と例題を上回っており、表4-1から表4-3に示したようにその多くは複数の図形が描かれ、そのうちのいくつかが移動したり、色を変えたり、形を変えたりとその機能も多種組み合わせられて作られていることがわかる。

このようにオリジナリティのある動画が作られるということは、学生が自分なりに考えた仕様をプログラムで実現できることが理解でき、かつ、「こんな図形を追加しよう」、「こんなふうに見えるようにつくりたい」と言うように、いろいろな形や機能を組み合わせることで作ってみようと言う創造的な態度でプログ

表5 動画ファンクションの定義と評価基準

ファンクション	評 価 基 準
描 画	<ul style="list-style-type: none"> ・一まとまりのオブジェクトについて1とする。 ・背景とみなされるものはまとめて1とし、一色で塗りつぶすのも1とする。
移 動	<ul style="list-style-type: none"> ・移動するオブジェクト（もしくはオブジェクトの一部）一つを1とする。 ・オブジェクトの移動方向が変わったときには、それも1とする。
色 変 化	オブジェクトまたは背景に対して色の変化があれば1とする。
複数描画	同じオブジェクトが複数配置されているとき、1とする。
順 次	オブジェクトに順次付け足して描画しているものは、そのまとまりを1とする。(eg. 絵描き歌の場合、絵 + 絵描き歌 = 2)
点 滅	オブジェクトを描いたり、消したりし、繰り返すものを1とする。繰り返す回数は考慮しない。星の輝き等はまとめての点滅を1とする。
一部変化	オブジェクトの中の一部だけが変形したり、追加されたりする場合1とする。
そ の 他	<p>描 画) 同じオブジェクトを複数配している場合は、「複数配置」という一まとまりを1とする。</p> <p>移動1) ただし、同じ方向に同じ物が移動する場合は「複数移動」という一まとまりを1とする。 (eg. 雨、雪、流れ星の場合は、移動 + 複数移動 = 2) ソースプログラムに関係なく同じ処理だと思われる場合はそれらを1とする。</p> <p>移動2) 往復移動の場合は、「移動方向の変化」で1とする。</p> <p>色変化) 色の変化は何色変化しても1とする。(eg. 花火の色変化) ただし、場面の移り変わりを示す色の変化は各変化をそれぞれ1とする。 (eg. 朝 昼 夜、春 夏 秋 冬における木や背景等)</p> <p>その他) 乱数などを使って、まばらな処理をしている場合は、その度合いにより考慮し、単純に「乱数使用」で1とする。(eg. 雨や雪で1、星の輝きで1とする)</p>

ラミングに取り組めた結果であると思われる。このことから、履修者はプログラミングに興味を持ち、いかに積極的に取り組んだかがわかる。

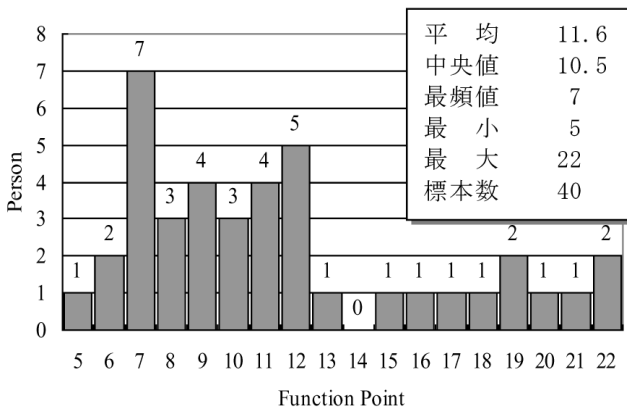


図5 FPの分布

また、ファンクションポイントとLOC間の相関係数は0.62で、おおむね相関があると言え、動画の評価方法として適当であると判断できる^{10,11}。

5. 履修者のプログラミング演習に対する評価

本演習の中で課題として取り入れた動画に興味を持って取り組めたか、また、その制作を通してプログラム完成までのプロセス等を理解できたかを最終回の授業の中で、履修者にアンケート形式で調査した。その結果を表6、表7に示す。

表6より、「動画制作は楽しかった」と回答した履修者は82.5%、否定的な履修者は0%、「動画を制作してプログラムを作成する力がつきた」は65%、力がつかなかった履修者は5%、「動画制作は難しかった」は90%、難しくなかった履修者は2.5%である。また、「プログラミングに興味を持った」履修者は70%、否定的な履修者は30%である。このように大半の履修者は、動画をプログラムで制作することは難しいと思ったものの、興味を持ち楽しみながら取り組み、かつ、プログラムを作成する力もつきたと認識していることがわ

ソフトウェア開発プロセスの体得方法としてのプログラミングによる動画制作

表6 アンケート調査結果1

項目 \ 選択肢	はい	どちらとも いえない	いいえ	無回答
グラフィックス（動画制作） は楽しかったですか	82.5%	17.5%	0.0%	---
グラフィックスで動画を制作 してプログラムを作成する力 がついたと思いますか	65.0%	30.0%	5.0%	---
グラフィックス（動画作成） は難しかったですか	90.0%	5.0%	2.5%	2.5%
プログラミングに興味を持ち ましたか	70.0%	30.0%	0.0%	---

表7 アンケート調査結果2

項目 \ 選択肢	5 わかつた	4 わかつた どちらかといえ ば	3 どちらとも 言えない	2 どちらかといえ ば わからなかつた	1 わからなかつた	無回答
Q1	30.0%	27.5%	30.0%	7.5%	5.0%	---
Q2	37.5%	45.0%	15.0%	0.0%	0.0%	2.5%

Q1：プログラミングとはアルゴリズムを特定の言語（たとえば、C言語）で表現するものということが理解できましたか

Q2：実行できるプログラムができるまでの作成過程がわかりましたか

かる。ここで「プログラムを作成する力がついた」と、学生自身が認識していることは、学生が自分なりに考えながらプログラミングができる自信がついたと捉えられる。また、表7より、プログラミングとはアルゴリズムを特定の言語（例えば、C言語）で表現するものということを57.5%の履修者が理解でき、

実行できるプログラムができるまでの作成プロセスを 82.5%が理解できたとしている。これらの要因としては、動画すなわち、グラフィックスは視覚的に出力結果が確認できるので、学生自身が自分の考えた仕様に対して、完成プログラムが一致しているかどうかを直感的に理解しやすいためと考えられる。

これらより、プログラミング演習に動画制作を導入したことは、履修者のプログラミングに対する興味を持たせることにつながり、また、学生がプログラミングの理解やプログラムを完成させるまでの作成プロセスを体得することができたと評価できる^{7,11}。

また、一般的なソフトウェア開発工程は基本的にはウォーターフォールモデル¹⁹が採用されている。このモデルの開発工程の概要を表 8 に示す。表 8 に示すようにプログラミング（プログラムの作成）は、ソフトウェア開発工程の一部であるが、このモデルは、工程ごとに区切りをつけて次の工程に進むが、図 1 のシステム化のプロセスは区切りがないものの区切りを入れたらウォーターフォールモデルとほぼ同じである。これより 8 割強の履修者は、この演習を通してその概要を体得したと評価できる。

表 8 ソフトウェア開発の工程の概要

工程名	工程の作業内容
1 構想設計	いわゆるシステムエンジニアリング業務である。顧客要件を分析し、ソフトウェアで実現する項目を開発要件（ソフトウェア仕様書）として具体化する。
2 設 計	CD 工程で作成された開発要件（ソフトウェア仕様書）にしたがって、ソフトウェアの機能分割・ソフトウェアのモジュールの組み合わせ・ソースプログラムの作成などの設計業務を行う。
3 デバッグ	設計工程で作成された成果物が、設計どおりに作成されていることを実機を使って動作確認する。設計作業を行った当事者が実施する。
4 テ ス ト	デバッグが完了したソフトウェア（プログラム）が、顧客要件を満足していることを確認する。設計・デバッグ作業とは異なる第三者が実施する。

6. おわりに

プログラミングの初学者が興味を持ってプログラミングに取り組むことができるように、課題として動画制作を取り入れ、実施した結果、履修者はプログラミングを難しいと認識しながらも、楽しみながら、かつ、プログラミングに興味を持って積極的に取り組めたこと^{7,11}、また、プログラムの作成プロセスがどのようなものを体得できたことが明らかになった。さらに、出来上がった動画の評価をLOCやファンクションポイントによって行った結果、初学者でも相当な量のソースプログラムを書くことができ、動画も工夫されたものであった。このことは、動画制作が仕様検討を含め全てを履修者自身がクリエイトする必要があることを考慮すれば、履修者がプログラミングの初学者で、かつ制作期間が短いにもかかわらず相当のプログラムを作成する力がつき^{7,11}、また、この演習を通してソフトウェア開発の工程の概要を体得したことは、評価できる。今後は、より良い教材の開発や環境、また、プログラミングに求められる知識・能力・資質等について検討すると共にソフトウェア開発の工程をより詳しく教育できる方法について検討していきたい。

参考文献

- 1 朝倉宏一、渡邊豊英：「情報工学系学部学生に対する並列プログラミング演習教育」、情報処理、Vol. 40, No. 5, pp. 2235-2245, 1999.
- 2 米沢宣義、志村武、南敏：「多人数初心者向きプログラミング教育システムの基本設計について」、情報処理、Vol. 27, No. 1, pp. 96-102, 1986.
- 3 関本理佳、海尻賢二：「プログラミングスタイルの診断システムの構築」、教育システム情報学会誌、Vol. 17, No. 1, pp. 21-29, 2000.
- 4 「プログラミング教育における制御構造のイメージと理解度について」、情報処理、Vol. 39, No. 4, pp. 1190-1183, 1998.
- 5 不破泰、中村八束、山崎浩、大下眞二郎：「Webを用いたCAIシステムによる大学講義の高度化とその評価」、教育システム情報学会誌、Vol. 20, No. 1, pp. 27-38, 2003.
- 6 高橋参吉、佐野繭美、橋本はる美、牧野純、松永公廣：「Web問題集を使ったCプログラミングの授業設計」、教育システム情報学会誌、Vol. 20, No. 4, pp. 392-397, 2003.

- 7 松山智恵子、中島豊四郎、石井直宏：「プログラミング初心者における動画製作の試みと評価」、電気学会論文誌 C、Vol. 124, No. 12, pp. 2482-2488, 2004.
- 8 松山智恵子、中島豊四郎、石井直宏：「プログラミング教育のための Web ベース教材の開発と評価」、電気学会論文誌 C、Vol. 125, No. 12, pp. 1900-1905, 2005.
- 9 松山智恵子、中島豊四郎、石井直宏：「プログラミング初心者のソースプログラムの分析と傾向」、電気学会論文誌 C、Vol. 125, No. 12, pp. 1914-1915, 2005.
- 10 中島豊四郎、松山智恵子、「プログラム作成力の評価における LOC とファンクションポイントの相関についての検討」、椋山女学園大学 文化情報学部紀要、第 10 巻、pp. 71-81, 2010
- 11 TOYOSHIRO NAKASHIMA, CHIEKO MATSUYAMA, NAOHIRO ISHII: "Development of Teaching Materials for Students to Tackle Programming with Interest and its Effectiveness", "Java in Academia and Research", pp. 37-60, iConcept Press Ltd., 2012
- 12 Mint (経営情報研究会)：「図解でわかるソフトウェア開発のすべて」、日本実業出版社 (2000)
- 13 R. S. Pressman: "Software Engineering", McGraw-Hill, New York, 1992.
- 14 Ali Behforooz, Frederick J. Hudson: "Software Engineering Fundamentals", Oxford University Press, New York, 1996.
- 15 Capers Jones 著、鶴保征城、富野壽 訳：「ソフトウェア開発の定量化手法」、共立出版、1993.
- 16 児玉公信：「実践ファンクションポイント法」、日本能率協会マネジメントセンター、1999.
- 17 榎アレア：「失敗しないファンクションポイント法」、日経 BP 社、2002.
- 18 河村一樹：「改訂新版ソフトウェア工学入門」、近代科学社、2003.
- 19 Boehm, B. W.: "Software engineering", IEEE Trans. Softw. Eng., Vol. C-25, No. 12, pp. 1226-1241, 1976.