

3DCG を活用したプログラミング教育教材開発の試み

相馬 佑哉 桑野 文洋

高等学校の学習指導要領が新学習指導要領に改訂され、2025 年度大学入学共通テストから「情報 I」が新たな出題科目として追加される予定となっている。「情報 I」では、すべての生徒が、プログラミングの基礎知識を学ぶこととなる。そのために、多くの生徒がプログラミングに興味を持つ教材が必要となると考えている。筆者はそのような教材の題材として 3DCG に着目した。プログラミングによって高精細な 3DCG を操作できる教材を生徒に提供できれば、興味関心を引き出し、内的動機づけを上げることが期待できる。本研究では、中等教育を対象とした基礎的なプログラミングによって、3DCG を操作できるシステムおよび教材を開発している。本システムでは日本語プログラミング言語であるドリトルや python のプログラムから Unity で実装している 3DCG を操作することが可能となっている。今後は本システムと教材を高等学校での教育現場に試験的に導入し、その有効性を検証してゆく予定である。

1 はじめに

初等中等教育の学習指導要領の改訂 (以下、新学習指導要領)[1]に伴い、小学校、中学校、高等学校とプログラミング教育が順次必修化され、プログラミングは全ての学生が学ぶべき教育内容となった。

そうした状況の中、本格化するプログラミング教育の中でいくつかの課題も上がっている。プログラミング教育を受けたことがある学生に調査を行ったプログラミング教育実態調査[2]では、高校生を対象とした「プログラミング教育は楽しかったか」という質問で『とてもそう思う』や『そう思う』という回答が合計で 55%、それ以外の項目が 45%という結果となった。楽しいと答えた高校生が過半数を超えているが、それ以外の回答も多い結果となっている。

教育を受ける学生がより積極的に学ぶためには、興味関心を高めるための工夫が一層必要であると考えられる。実際にプログラミングの興味を持たせるために

様々な工夫が教材に施されている。塾の満足度を調査している民間企業が公表しているプログラミング教室の人気ランキング[3]では、ロボット制御をプログラミングで行うものやゲーム性を持たせた教材など、魅力向上に工夫を凝らしている教材を用いている教室が上位にあがっている。

その一方で 3DCG 技術の発展はめざましく、様々な魅力的な CG コンテンツが作成されるようになり、公開されている。また、高等学校の情報の授業においても「MAYA」が利用されている学校もある。

そこでプログラミング教育に 3DCG を組み込んだプログラミング教材を導入することによって教育の魅力向上につなげることができるのではないかと考えた。本研究ではこの仮説に基づき、初歩的なプログラミング教育に 3DCG を活用する仕組みとその仕組みに基づいた教材を提案する。

以降の論文構成は次の通りである。2 章では、まず初等中等教育で利用されているプログラミング言語の現状を概観する。3 章で提案する教材システムの概要を示し、4 章ではこのシステムを使って開発した教材を提示する。5 章において関連研究に言及し、6 章で今後の課題をまとめる。

3DCG-based Learning Materials for Introductory Programming

Yuya Souma, Fumihiko Kumeno, 日本工業大学 大学院工学研究科 電子情報メディア工学専攻, Dept. of Information Technology and Media Design, Nippon Institute of Technology.

2 初等中等教育で利用されるプログラミング言語の現状

前項で述べた通り、新学習指導要領によってプログラミング教育が本格的に導入されつつある。ここではそれぞれの段階で使用されているプログラミング言語について概説する。

2.1 小学校向けのプログラミング言語

小学校では「Scratch」[4]や「Viscuit」[5]等のビジュアルプログラミング言語がよく用いられている。小学生、とりわけ低学年の場合、文字をキーボードでタイプするのがハードルになることが多い上、日本人のように英語が母国語でない場合、アルファベット表記の入力も難しい。そのため、小学生向けには、文字をたくさんタイプ・入力しなくて良いように工夫された、ビジュアルプログラミング言語が利用されている[6][7]。

2.2 中学校向けのプログラミング言語

中学校の指導要領では、「ネットワークを利用した双方向性のあるコンテンツのプログラミング」や「計測・制御のプログラミング」などを学ぶことが明記されている。そのため、「Scratch」や「Viscuit」等のビジュアルプログラミング言語だけでなく、JavaScriptやHTML等の実用的なプログラミング言語を使用されている例もある。

中学校の「技術」は4つの分野で構成され、授業が実施されている。限られた時間数の中での実施になってしまうことからプログラミングに関する内容は幅広く用意されているものの、実際には深い内容を教育することが難しいのが現状である[8]。そのため、Scratch等のビジュアルプログラミング言語を採用している学校が多く、JavaScriptやHTMLのような実際のシステム開発で用いられるプログラミング言語を導入している学校は限られているようである。

2.3 高等学校向けのプログラミング言語

高等学校では、2022年の学習指導要領改訂によって普通科における「情報I」が必修となり、高校普通科

のすべての生徒がプログラミング学習は必須となった。その一方で教育に利用するプログラミング言語の指定はされておらず、どの言語を授業に取り入れるかは担当教諭に一任されている。そのため、複数の種類のプログラミング言語が授業の場で用いられている。

ここでは、普通科における「情報I」、工業高校における情報科の「専門科目」で使用されている言語についてまとめる。

「情報I」については、文部科学省がホームページにてJavaScript、VBA、ドリトル、swiftの4つの言語について紹介をしている[9]。また、実教出版が出している「情報I」の教科書においてはJavaScript版の教科書及びpython版の教科書の2種類の教科書が用意されている。「専門科目」においては、python、Javascript、C言語、ビジュアルプログラミング言語などのプログラミング言語が利用されている[10]。

こうしたことから、中学校と比較するとビジュアルプログラミング言語の利用からテキストメインのプログラミング言語への移行がみられると推測できる。

2.4 プログラミング教育の段階に関する課題

2.1から2.3にあるように中学校まではビジュアルプログラミング言語が主に使われているが、高等学校や工業高校で主に使用されているプログラミング言語はC言語やpython等のテキストプログラミング言語になっている。

ビジュアルプログラミング言語は感覚的にプログラムを構築することができるため、スタートとして用いることは効果的であるが、実際のプログラミングは半角英数字のテキストベースである。

そのため、ビジュアルプログラミング言語からテキストプログラミング言語へ移行する段階でギャップが大きく、プログラミングに対して苦手意識を増加させていることが憂慮される。実際に教育実態調査[2]にあるように小学校から高等学校に進むにつれて「プログラミングは楽しかったか」という質問に対する『とてもそう思う』『そう思う』の割合が少なくなっている傾向にある。

このことから小中高と教育の段階が進むにつれ、プログラミングに対する興味が薄れている可能性が

高い。興味が薄れている原因の究明には、より踏み込んだ調査分析が必要と考えられるが、筆者はこの状況を改善するためにはプログラミングの楽しさをより訴求する魅力的な教材が重要という仮説を立て、本研究を実施した。

3 提案する教材システムの概要

本研究ではテキスト型プログラミング言語に対応し、3DCGを動かすことができるコンテンツ教材の開発を目指す。対象となる言語は高校の教科書でよく取り上げられるpythonや入門用のテキスト型プログラミング言語として日本語でプログラミングができる「ドリトル」とした。pythonやドリトルでは、MAYAやBlenderのようなツールで作成した3DCGのモデルを制御することは困難である。そのため、これらのプログラミング言語でも3DCGを制御できる仕組みを構築する必要がある。

図1はシステムの全体構成である。システムはUnity、Webサーバ、プログラミング環境で構成される。

(1)Unity MayaやBlender等の3DCG制作ソフトを用いて制作した3DCGを制御する環境を構築している。C#のscriptによって(2)のWebサーバとデータの送受信を行う。

(2)Webサーバ プログラミング環境とUnityをつなぐ役割を担っている。(3)のプログラミング環境から利用可能なAPIを実装しており、APIを介して(3)のプログラムと(1)のUnityの間でのデータを送受信することができる。

(3)プログラミング環境 学習者が使うプログラミング環境であり、pythonやドリトル等の複数のプログラミング言語を想定している。WebAPIをサポートしているプログラミング言語であれば、(2)のAPIを使って、(1)のUnityとデータの送受信が可能である。

(1)(2)(3)を組み合わせることで3DCGモデルを複数のプログラミング言語で制御することができるようになっている。制御を実現する手順は次のとおりである。

1. Unityの3DCGアプリを起動する。このアプリでは起動後は待機状態となっており、Webサーバからのデータを受信することで、3DCGモデルが操作するようになっていく。

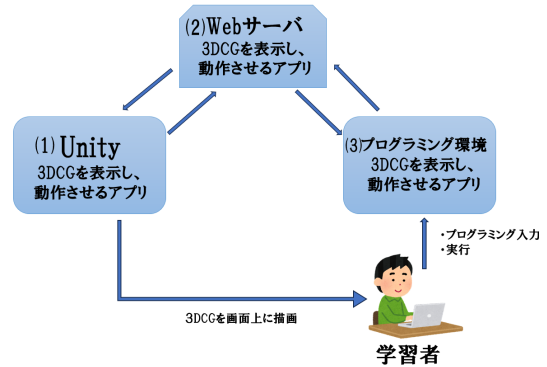


図1 システムの全体構成

バからのデータを受信することで、3DCGモデルが操作するようになっていく。

2. プログラミング環境において学習者が作成したプログラムを実行する。このプログラムの中に(2)のAPIを実行する命令が含まれている。APIが実行されることで学習者のプログラムからUnityの3DCGアプリにデータが送信される。
3. 待機状態になっているUnityの3DCGアプリにデータ(文字列)が届くと、その文字列に従って、3DCGがアニメーションとして動作する。動作した結果をプログラミング環境側に返り値として返送することも可能である。

4 開発したプログラミング教材

前節で説明した教材システムを用いて、2種類の教材を開発した。一つはプログラムの基本要素を学ぶための教材、もう一つは「モデル化とシミュレーション」の単元での利用を想定した教材である。それぞれの概要を示す。

4.1 車の運転を行うプログラミング教材

3DCGでモデリングした車をドリトルやPythonなどの学習用プログラミング言語で制御し、動かす教材である。

図2は3DCGの画面である。街の中を車が動く想定となっている。プログラミングの基本要素である「順次」「反復」「分岐」を車の操作を通じて学習する。車を進ませる、右折する、左折する、信号で停止す

る, などの基本的な操作コマンドと if 文, for 文を組み合わせ, スタート地点から目的地にたどり着くことを演習課題としている。

Python によるプログラミング例を図 3 に示す. このプログラムで import している carcontrol は車の基本操作 (直進や左折など) を提供するライブラリである. 図 4 が carcontrol のソースプログラムである. このプログラムにおいて (2)Web サーバの API を利用している. 学習者はこのライブラリを使って車の基本操作を図 3 のように組み合わせて車を操作す



図 2 車を操作する 3DCG 画面

```

1 import carcontrol1 as p
2
3 def start():
4     p.chokushin2()
5     p.chokushin2()
6     p.chokushin2()
7     p.chokushin2()
8     p.chokushin2()
9     p.left3()
10    p.chokushin2()
11    p.chokushin()
12    p.chokushin()
13    p.chokushin()
14    p.left3()
15    p.chokushin2()
16    p.chokushin2()
17    p.chokushin2()
18    p.chokushin2()
19    p.chokushin2()
20    p.left3()
21    p.chokushin2()
22    p.chokushin()
23    p.chokushin()
24    p.left3()
25
26
27
28 if __name__ == "__main__":
29     start()

```

図 3 python のプログラミング例

```

3
4 import requests
5
6 def chokushin():
7
8     _res = requests.get('http://127.0.0.1:8080/command/0')
9
10 def chokushin2():
11
12     _res = requests.get('http://127.0.0.1:8080/command/00')
13
14
15 def left():
16     _res = requests.get('http://127.0.0.1:8080/command/1')
17
18 def left2():
19     _res = requests.get('http://127.0.0.1:8080/command/11')
20
21 def left3():
22     _res = requests.get('http://127.0.0.1:8080/command/111')
23
24
25 def back():
26     _res = requests.get('http://127.0.0.1:8080/command/2')
27
28 def back():
29     _res = requests.get('http://127.0.0.1:8080/command/22')
30
31
32 def raight():
33     _res = requests.get('http://127.0.0.1:8080/command/3')
34
35 def raight2():
36     _res = requests.get('http://127.0.0.1:8080/command/33')
37
38 def raight3():
39     _res = requests.get('http://127.0.0.1:8080/command/333')
40

```

図 4 carcontrol のプログラムコード

るプログラムを作成する (したがって学習者が API を使う必要はない)。

図 5 はドリトルによるプログラミング例の抜粋である. ドリトルでは python のようにライブラリを別途定義することが基本的には難しいため, 車の基本操作を定義したプログラムをひな型とし, その下に学習者のプログラムを付け加えるスタイルとなる。

4.2 自由落下シミュレーション用教材

「情報 I」の単元である「モデル化とシミュレーション」で使用することを想定した教材である. この単元では, 実際の物理現象などのモデル化を行い, プログラミングを通して再現をし, シミュレーションを学ぶ内容となっている。

従来の教材では写真をもとに学習をするしかなく, 動きや速さなどの変化を学習することができない. プログラムによるシミュレーションにおいても, 2 次元

```

8 //前方に進むコマンド
9 かめた:進む = 「ret=w! "http://127.0.0.1/command/0" 8080 読む」。
10 かめた:大きく進む = 「ret=w! "http://127.0.0.1/command/00" 8080 読む」。
11
12 //左に曲がるコマンド
13 かめた:左に曲がる = 「ret=w! "http://127.0.0.1/command/1" 8080 読む」。
14 かめた:左に大きく曲がる = 「ret=w! "http://127.0.0.1/command/11" 8080 読む」。
15
16 //後ろに進むコマンド
17 かめた:戻る = 「ret=w! "http://127.0.0.1/command/2" 8080 読む」。
18 かめた:大きく戻る = 「ret=w! "http://127.0.0.1/command/22" 8080 読む」。
19
20 //右に曲がるコマンド
21 かめた:右に曲がる = 「ret=w! "http://127.0.0.1/command/3" 8080 読む」。
22 かめた:右に大きく曲がる = 「ret=w! "http://127.0.0.1/command/33" 8080 読む」。
23
24 //信号判定コマンド
25 かめた:信号見る = 「結果=w! "http://127.0.0.1/command/999" 8080 読む」。
26
27
28 //移動させたい量
29
30 かめた!進む。
31 かめた!進む。
32 かめた!大きく進む。
33 かめた!左に曲がる。
34 かめた!左に大きく曲がる。
35 かめた!左に大きく曲がる。
36 かめた!左に大きく曲がる。
37 かめた!戻る。
38 かめた!大きく戻る。
39 かめた!右に曲がる。
40 かめた!右に大きく曲がる。
41 かめた!左に大きく曲がる。

```

図5 ドリトルのプログラミング例(抜粋)

のプロットのための描画で表示を行うなど機能が限定される。

これに対し、本教材では物体の自由落下を 3DCG によって立体的に表現することによって視覚的な学習効果が期待できる。また、速度等を変更することで速度による違いなどを学習することもできる。図6は自由落下の様子を街の 3DCG モデルで球が落ちる様子として表現したものである。

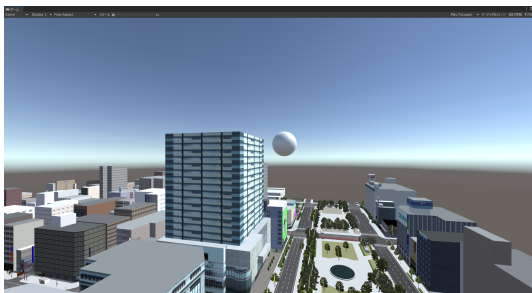


図6 3DCG で球の自由落下を表示している画面

図7はpythonで物体の自由落下のシミュレーション計算を行った結果(座標)を3DCGで表示する部分のプログラム例である。

```

1 import plot3d as p
2
3 def start():
4     p.show2d(10.0,10.0)
5     p.show2dString(20.0,20.0,"2DTEST")
6     p.show3d(25.0,25.0,25.0)
7     p.show3dString(30.0,30.0,30.0,"3DTEST")
8
9
10 if __name__ == "__main__":
11     start()
12
13
14

```

図7 pythonによるプログラム例
(座標を3DCGで表示する部分)

車の場合と同様に 3DCG と連携するためのライブラリ plot3d を用意している。plot3d では (2)Web サーバの API を使って、座標や文字データを (1)Unity に送信する機能を提供している。plot3d のプログラムコードを図8に示す。

5 関連研究

関連研究として 3D エンジンを利用したプログラミング教育用教材 [11] が存在する。この論文は高等教育を対象として C 言語を学習できる教材となっている。この研究論文では 3DCG を題材としており、ゲーム感覚で C 言語を学習することができる。ただし、C 言語のなかで比較的難易度の高いポインタや構造体等も教育対象となっており、プログラミングを触ったことがない学習者や C 言語を全く知らない学習者においては難易度が高い。そのため、高校生(プログラミング学習初期段階)には教材としての難易度が高く向いていない。

一方で同研究において実施したアンケートによれば、高等教育における 3DCG における動機付けは効果的であるという回答が多いことが報告されている。したがって、3DCG 教材が動機付けには効果的であると考えられる。

その他にも初等中等教育において、3DCG をプログラミングに活用する試みの研究は筆者の調べた限り見つからなかった。

```

test > plot3d.py > ...
1 # requests はwebにアクセスするためのモジュール
2 # 管理者権限のコマンドプロンプトで pip install requests を実行する
3
4 import requests
5
6 # 2次元座標,たとえば(1.0,1.0)を送る場合,仲介サーバにhttp://127.0.0.1/command/2D:x=1.0&y=1.0でアクセスする.
7 # unityには文字列"2D:x=1.0&y=1.0"が渡されるので,この文字列から座標を抽出して利用する,unity内では固定値のz座標を使い,3次元空間で表示
8
9 def show2d(xpos,ypos):
10     _res = requests.get('http://127.0.0.1:8080/command/'+x+''+str(xpos)+'&y=''+str(ypos)+'&z=')
11
12 #Unityで表示する文字列も渡す
13 def show2dString(xpos,ypos,string):
14     _res = requests.get('http://127.0.0.1:8080/command/'+String:x'+str(xpos)+'&y=''+str(ypos)+'&z=''+string'+string)
15
16 # 3次元座標,たとえば(1.0,1.0,1.0)を送る場合,仲介サーバにhttp://127.0.0.1/command/3D:x=1.0&y=1.0&z=1.0でアクセスする.
17 # unityには文字列"3D:x=1.0&y=1.0&z=1.0"が渡されるので,この文字列から座標を抽出して利用する
18
19 def show3d(xpos,ypos,zpos):
20     _res = requests.get('http://127.0.0.1:8080/command/'+x'+str(xpos)+'&y=''+str(ypos)+'&z=''+str(zpos))
21
22
23 #Unityで表示する文字列も渡す
24 def show3dString(xpos,ypos,zpos,string):
25     _res = requests.get('http://127.0.0.1:8080/command/'+String:x'+str(xpos)+'&y=''+str(ypos)+'&z=''+str(zpos)+'&string'+string)
26

```

図 8 plot3d のプログラムコード (一部)

6 今後の課題

本研究で提示した教材の検証をおこなっていく。本教材を利用したプログラミング演習を勤務先の高等学校にて実施し、参加者に対してアンケート調査を行う予定である。アンケートでは教材を使用することで楽しく学習することができたかを明らかにし本教材の有効性を確認する。

また、本研究で開発したシステムは高等学校でのテキストベースのプログラミング言語を対象としている、Web API をサポートしている言語であれば、ビジュアルプログラミング言語でも同様な教材として使用することができる。したがって従来のビジュアルプログラミング言語の単独機能では実現できない 3DCG の操作をこのシステムでは実現することができる。小学校や中学校において既存のビジュアルプログラミング言語で 3DCG の制御といったようなことも可能にすることができ、初等教育での活用も期待ができる。

参考文献

- [1] 文部科学省. 新学習指導要領の全面実施と学習評価の改善について. https://www.mext.go.jp/content/20201023_mxt_sigakugy_1420538_00002_004.pdf. (Accessed on 08/04/2023).
- [2] みんなのコード. プログラミング教育実態調査報告書. <https://code.or.jp/news/10370/>. (Accessed on 08/04/2023).
- [3] 株式会社イード. イード・アワード 2023「プログラミング教育」. <https://www.iid.co.jp/news/award/2023/programming-school.html>. (Accessed on 08/04/2023).
- [4] Scratch 財団. Scratch 公式サイト. <https://scratch.mit.edu/>. (Accessed on 08/04/2023).
- [5] 合同会社デジタルポケット. Viscuit 公式サイト. <https://www.viscuit.com/>. (Accessed on 08/04/2023).
- [6] 河原和好. 小学生を対象にしたプログラミング教育について. 新潟国際情報大学情報文化学部紀要, Vol. 3, pp. 27-35, 2017.
- [7] 原田康徳, 勝沼奈緒実, 久野靖. 公立小学校の課外活動における非専門家によるプログラミング教育. 情報処理学会論文誌, Vol. 55, No. 8, pp. 1765-1777, 2014.
- [8] 一般社団法人日本産業技術教育学会. 中学校プログラミング教育の実態調査の報告—令和元年度 技術・家庭科技術分野「d 情報の技術」の現状—. https://www.jstet.jp/main/teigen/210127_jr_chosa.html. (Accessed on 08/04/2023).
- [9] 文部科学省. 高等学校情報科「情報Ⅰ」教員研修用教材 第3章 他プログラミング言語版. https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1421808.htm. (Accessed on 08/04/2023).
- [10] 加納寛子. 工業高校の生徒のプログラミングに対する意識と経験. 日本科学教育学会年会論文集, Vol. 46, pp. 451-454, 2022.
- [11] 堀田忠義. 3d エンジンを利用した c プログラミング教育用教材. 技能科学研究, Vol. 35, pp. 58-65, 2019.