Example ベースプログラミングの導入

各回の演習ではカメラ設定等を施したフレームワークプログラムに、課題内容を挿入し表示の違いを確認することで進めた。例えば、図 5 は実際に提示している資料からの抜粋である。このようなサンプルソースを各回平均 7～8 プログラム用意した。

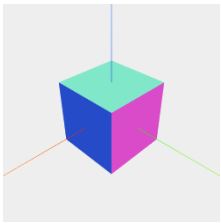
プログラムソース 1　　objects()の実装

```
//////////////////////////////////////////////////
//モデルの設定

//モデル設定で利用するグローバル変数
var axis; //軸オブジェクト
var cube; //直方体オブジェクト
function objects() {
                    //軸オブジェクトの生成
                    axis = new THREE.AxisHelper(100);
                    //軸オブジェクトのシーンへの追加
                    scene.add(axis);
                    //軸オブジェクトの位置座標を設定
                    axis.position.set(0, 0, 0);

                    //形状オブジェクトの宣言と生成
                    var geometry = new THREE.BoxGeometry(50, 50, 50);      //クラスリファレンスを参照
                    //材質オブジェクトの宣言と生成
                    var material = new THREE.MeshNormalMaterial();
                    //直方体オブジェクトの生成
                    cube = new THREE.Mesh(geometry, material);
                    //直方体オブジェクトのシーンへの追加
                    scene.add(cube);
                    //直方体オブジェクトの位置座標を設定
                    cube.position.set(0, 0, 0);
}
```

BoxGeometry クラス

親クラス：Geometry
対象レンダラー：WebGL, Canvas2D

直方体オブジェクトの形状を生成するクラスです。



コンストラクタ

```
var geometry = new THREE.BoxGeometry(
                    width, height, depth,
                    widthSegments, heightSegments, depthSegments )
```

プロパティ名	データ型	デフォルト	説明
width	<float>	なし（必須）	直方体のＸ軸方向、Ｙ軸方向、Ｚ軸方向の幅
height	<float>	なし（必須）	
depth	<float>	なし（必須）	
widthSegments	<int>	1	直方体のＸ軸方向、Ｙ軸方向、Ｚ軸方向の分割数。大きくすると表示は滑らかになるが演算量が増え負荷が大きくなる。
heightSegments	<int>	1	
depthSegments	<int>	1	

プロパティ(メンバ変数)

プロパティ名	データ型	デフォルト	説明
type	<string>	'BoxGeometry'	どのクラスのオブジェクトか判断する文字列。
parameters	<object>		コンストラクタの引数で指定された直方体オブジェクトを生成するのに必要なパラメーターを保持するオブジェクト。

メソッド(メンバ関数)

なし

図 5：配布資料の一部

各回の演習の流れとしては、立方体の表示を学ぶ場合、まず最初に、図5上のプログラムソース1に示すサンプルプログラムをカメラ設定等を施した既存のプログラムにコピー&ペーストする事で表示結果を確認させた。ソースには極力コメント文を用意してそれぞれの行を説明した。同時に立方体の形状を作るクラスであるBoxGeometryクラスに関して詳細を記したクラスリファリンスも図5下のように提示した。

次に、各演習ではいくつかの課題を用意し、例えば、ソース 1 の立方体を球体(SphereGeometry)に変更し、分割数を増して滑らかな面を表示する課題などを出題した。課題を解くには、クラスリファレンスを読んで各種パラメータの設定を知る必要がある。学生は、初回は苦労してリファレンスを読むことになるが、読み方が分かると応用できるようになり、オブジェクト指向の基礎であるクラスとインスタンスの概念や有用性を理解していた。

つまり、小学校課程で正三角形を描くプログラム（図 1 中央）に相当するものを、図 5 上のプログラムソース 1 として提示し、小学校課程で五角形や六角形への応用（図 2）に相当するものを、図5下のクラスリファリンスで提示している。クラスリファレンスでパラメータの意味を理解することで学生が自由に応用できるところがねらいである。

また、メンバ関数（メソッド）の使い方を理解すると親クラスのメンバ関数を調べ応用する力がついた。例えば形状は親クラスがGeometryクラスであり、「形状を移動させたい」と考えた場合は、Geometryクラスのリファレンスを調べメソッド名から「.translate (x : Float, y : Float, z : Float)」が利用できるのではないかと予測しながら試す力が付いた。さらに、three.jsのサイトにあるDocumentationも紹介した。このページは英語で記述されているため、学生にとっては敷居が高いが少し手助けすることで読もうとする学生もあった。

学生は、リファレンスを読み、クラスに秘められた機能を知ることで、オブジェクト指向プログラミングの有用性を理解した。ここで学生は「three.jsの開発者がクラスのカプセル化をしてくれたからスムーズに作れる」ことを理解し始めていると考える。

学生作品例と考察

学生は、それぞれが最終提出作品を目標とすることで学ぶモチベーションが増し、演習で学んだ技術を復習しつつ制作を進めていた。中には、演習で指導していない技術をthree.jsのサイトから、サンプルプログラムやリファレンスを調べて実装する学生もいた。

図6は学生による作品例である。課題の評価基準として、“アニメーションが施されていること”や“インタラクティブに操作できること”とした。例えば雲の流れや宇宙船の動き、マウスのドラッグによる視点変更等が実装されたリアルタイムCGを制作した。

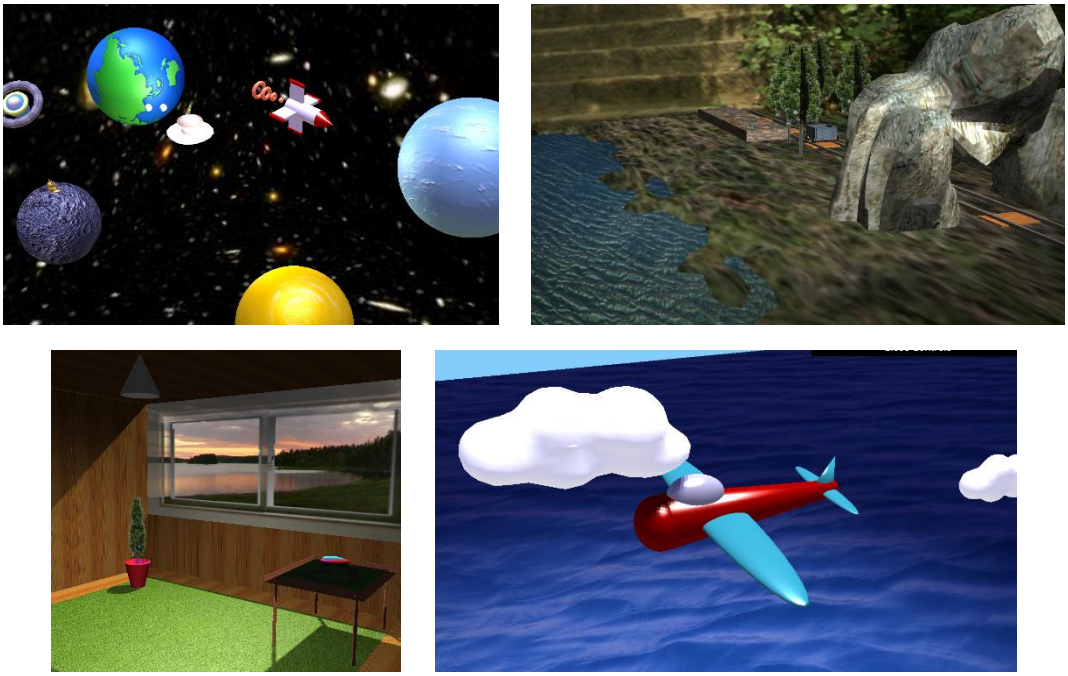


図 6：学生による作品例

2019 年度の履修者に対して、最終作品の提出後に表 2 に示す学習に関するアンケートを実施した。プログラミングに関して現時点での自己評価を訪ねた「③ あなたはプログラミングが苦手ですか」という問いに関しては、11 人中 5 人が「とても苦手」6 人が「どちらとも言えない」を選択し「そうは思わない」は 0 人であった。ここから、プログラミングを得意としている学生は 0%であることが分かる。対して「⑤ オブジェクト指向プログラミングの便利さは理解できましたか」の問いに対しては 11 人中 6 人が「強くそう思う」5 人が「そう思う」を選択し、「どちらとも言えない」「そう思わない」「全くそう思わない」は 0 人であった。ここから、ほぼ 100%の学生がオブジェクト指向の便利さに気づいたことが分かる。

これらの結果よりプログラミングにあまり自信がない学生が、オブジェクト指向の有用性を理解できたことに一定の成果があったと考える。

また、「(ア)　3DCG が目標であるため結果が数値や文字でなく絵で確認でき、従来のプログラミング演習よりは抵抗が少なかった」の問いに対しては 11 人中 3 人が「強くそう思う」8 人が「そう思う」を選択し、「どちらとも言えない」「そう思わない」「全くそう思わない」は 0 人であった。ここからも映像による結果表現に効果があったと考えられる。

表 2： アンケート結果

設問	解答群	回答数
① 3DCG により興味を持つことができた （時間があれば作り込んでみたい）	強くそう思う	8
	そう思う	3
	どちらとも言えない	0
	そう思わない	0
	全くそう思わない	0
② 今回の演習で興味を持てた（面白いと感じた）項目を 教えてください **（複数選択　可）**	光源設定	0
	影設定	3
	アニメーション	8
	カラーマッピング	1
	法線などバンプ系マッピング	6
	環境マッピング	4
	トゥーンシェーディング	1
	Blender	4
	JavaScript	0
③ あなたはプログラミングが苦手ですか	とても苦手	5
	どちらとも言えない	6
	そう思わない	0
設問③で「どちらとも言えない」・「とても苦手」を選択された方にお尋ねします (ア)　3DCG が目標であるため結果が数値や文字でなく絵で確認でき、従来のプログラミング演習よりは抵抗が少なかった	強くそう思う	3
	そう思う	8
	どちらとも言えない	0
	そう思わない	0
	全くそう思わない	0
④ JavaScript は慣れていないと思いますが、 C++や Java を学んできた流れで抵抗なく利用できましたか	全く抵抗はない	3
	抵抗はない	6
	どちらとも言えない	0
	苦労した	2
	かなり苦労した	0
⑤ オブジェクト指向プログラミングの便利さは理解できましたか	強くそう思う	6
	そう思う	5
	どちらとも言えない	0
	そう思わない	0
	全くそう思わない	0
⑥ three.js ライブラリは、便利ですが細かな設定ができません。 難易度は上がりますが、以下のライブラリから学んでみたいものはありますか **（複数選択　可）**	OpenGL(WebGL)	5
	DirectX	4
	それらのシェーダプログラミング	3
	詳しく知らないのでわからない	3
講義「コンピュータグラフィックス」を履修している方にお尋ねします。 講義では透視投影や座標変換、シェーディングなどの計算方法を学びました。 three.js ではそれらがクラスとして用意されており改造できません。 講義で学んだアルゴリズムをプログラミングして、自分の 3DCG プログラムを作ってみたいですか	強くそう思う	0
	そう思う	4
	どちらとも言えない	6
	そう思わない	1
	全くそう思わない	0

まとめ

本報告では、今後、義務教育等で実施されるプログラミング教育、とりわけ小学校課程におけるプログラミング教育を理解し、その接続を意識したプログラミング教育を提案した。中学・高等学校課程の生徒が学びたいというモチベーションをもって、積極的にリファレンスを調べることを目指した。教材としてリアルタイム 3DCG での作品制作をゴールとし、その難易度を下げる方法として three.js の導入を提案した。three.js は従来の 3DCG ライブラリと比較して少ない記述量で見栄えよい 3DCG を作成でき、幾何学等の概念を理解していなくとも、パラメータ変更でさまざまな応用ができることを紹介した。結果、プログラミングが得意でない学生もオブジェクト指向の便利さを理解することができた。

今後は、three.js のパラメータ変更のみで様々な表現の CG に変更できる点は、Scratch 等のブロックを利用した初心者プログラミングツールにも応用できる可能性があり、小学校課程での 3DCG プログラミング導入を考えていきたい。また、最新の GPU ではレイトレーシングをリアルタイムで実行できるようになってきており合わせて調査していきたい。

引用・参考文献

1. 文部科学省『学習指導要領「生きる力」』、(2019/10/14 現在)，
   http://www.mext.go.jp/a_menu/shotou/new-cs/1384661.htm
2. 文部科学省『小学校プログラミング教育の手引き（第二版)』、(2019/10/14 現在)，
   http://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1403162.htm
3. 文部科学省『小学校プログラミング教育に関する研修教材』、(2019/10/14 現在)，
   http://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416408.htm
4. 近藤邦雄, 『作品製作を中心とした CG プログラミング教育』, 図学研究 38, 2004,1-6.
5. 近藤邦雄, 『Example Based Programming にもとづく CG 制作技法の基礎教育』, 情報処理学会　研究報告, 2009, 177-182.
6. 高山文雄, 大表良一, 『Java 環境下におけるコンピュータグラフィックスを用いたプログラミング教材の試作』, 図学研究 41, 2007, 211-212.
7. 高山文雄, 『Web 環境を利用した CG 重視のプログラミング教育支援システム』, いわき明星大学科学技術学部研究紀要, 30, 2017, 40-45.
8. 辻合秀一,『インタラクティブアートプログラミング教育の一考察』, 富山大学芸術文化学部紀要 2, 2007, 86-92.
9. 辻合秀一,『芸術系学生の制作を含むプログラミング教育』, 第 54 回自動制御連合講演会, 2011,

837-840.

10. 坂田圭司, 高橋隆男, 『ソフトウェア開発教育における Processing と Arduino の活用』, 情報処理学会第 75 回全国大会, 4, 2013, 387-388.
11. 米元聡, 『グラフィックス課題を用いたプログラミング教育』, Common: 九州産業大学総合情報基盤センター広報誌, Vol.33, 2013, 44-49.
12. 山田聖也, 原田拡輝, 長谷川隼兵, 椿郁子, 桑原明栄子, 『小学校段階におけるプログラミングのための円滑な導入教育に関する考察』, 情報メディア学会　技術報告, Vol. 42, No. 12, 2018, 65-68.
13. 長谷川隼兵, 原田拡輝, 山田聖也, 椿郁子, 桑原明栄子, 『小学校段階におけるプログラミングの教育方法に関する考察』, 情報メディア学会　技術報告, Vol. 42, No. 12, 2018, 265-268.
14. MIT メディアラボ『Scratch』、(2019/10/14 現在),
https://scratch.mit.edu/