

three.js による

オブジェクト指向プログラミング教育の一考察

～ 小学校課程プログラミング教育からの接続 ～

河崎雷太

本報告では、基本的なプログラミング教育を受けた学生を対象に、より実践的にオブジェクト指向プログラミングを学ぶ方法を提案する。題材として three.js によるリアルタイム 3D コンピュータグラフィックスのプログラミングを用い、表示結果の印象を良くすることで、学習者のモチベーションを高めることを目指す。

はじめに

プログラミング言語や手法は時代によって変遷してきているが、プログラミングの導入教育は、普遍的な教育内容である。プログラミング教育でコーディング（プログラムの記述法）を学ぶ際には、プログラミングの概念を理解し、各言語の文法（記述ルール）に沿った順次進行プログラムから理解し、次に、条件分岐や繰り返しのアルゴリズムを理解している。2020 年度より小学校課程でプログラミング教育が導入され、手順を考える力の強化が予想できるが、中学校・高等学校課程でコーディングを学ぶ際には、ほぼ同じ導入教育がなされる事が予想される。

学ぶ言語によっては、「各言語の文法（記述ルール）」としてオブジェクト指向プログラムに則した記述を行うことになる場合があるが、オブジェクト指向の概念を理解するのは、前述した基本的な順次進行・条件分岐・繰り返しを理解した後に行われることが多い。よって、「クラス」や「メンバ」などのオブジェクト思考プログラミングで必要とされる技術用語とその定義方法は理解しているが利用価値は理解できていないことが多い。特に、クラスのカプセル化を学ぶ際に、ほとんどの学生が「なぜこのような面倒な仕組みを作るのか」といった感想を述べる事が多いように思う。

本報告では、オブジェクト指向プログラミングの有用性を理解できる教育方法を提案する。また、今後の教育方法に対応すべく、小学校課程でのプログラミング教育を理解し、その先の学習者が学ぶ意欲を持てるよう 3D コンピュータグラフィックスのプログラミングを導入する。必要なリファレンスを読み、クラスを活用できる技術習得を目標とする。

小学校課程でのプログラミング教育

小学校課程では、2017 年 3 月の新学習指導要領公示¹に続き、2018 年 11 月に「小学校プログラミング教育の手引き（第二版）」が文部科学省により公開された²。この手引きでは新学習指導要領の内容をより具体的に説明している。この手引きによると、小学校課程におけるプログラミング教育のねらいは、以下の 3 つであると記されている。

1. 「プログラミング的思考」を育むこと
2. プログラムの働きやよさ、情報社会がコンピュータ等の情報技術によって支えられていることなどに気付くことができるようにするとともに、コンピュータ等を上手に活用して身近な問題を解決したり、より良い社会を築いたりしようとする態度を育むこと
3. 各教科等の内容を指導する中で実施する場合には、各教科等での学びをより確実なものとする

また、資質・能力の 3 つの柱である「知識及び技能」、「思考力、判断力、表現力等」「学びに向かう力、人間性等」に対しても、それぞれに対応する説明がなされている。

「知識及び技能」に関して、小学校課程においては「コンピュータはプログラムで動いていること、プログラムは人が作成していること、また、コンピュータには得意なこととなかなかできないこととがあることを体験を通して気づかせること」と記されている。プログラミングの記述方法やアルゴリズムなど、具体的な知識・技能は、中学校や高等学校で学習するとされている。

「思考力、判断力、表現力等」に関して、小学校課程においては「論理的思考力、すなわち『プログラミング的思考』を育成すること」とある。ここで『プログラミング的思考』とは、「自分が意図する一連の活動を実現するために、どのような動きの組み合わせが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力」と記されている。また、コンピュータを動作させるための手順として、「① コンピュータにどのような動きをさせたいかという自らの意図を明確にする。② コンピュータにどのような動きをどのような順序でさせればいいのかを考える。③ 一つ一つの動きを対応する命令（記号）に置き換える。④ これらの命令（記号）をどのように組み合わせれば自分が考える動作を実現できるかを考える。⑤ その命令（記号）の組合せをどのように改善すれば自分が考える動作により近づいていくのかを試行錯誤しながら考える。」と記されている。

「学びに向かう力、人間性等」に関して、小学校課程においては「他者と協同しながらねばり強くやり抜く態度の育成、著作権等の自他の権利を尊重したり、情報セキュリティの確保に留意したりするといった、情報モラルの育成も重要」と記されている。

児童に対するプログラミング教材の動機づけ

前章で示した資質・能力の3つの柱を指導することは、それぞれに小学校教諭の指導法が問われるが、その中でも困難が予想されるのが「学びに向かう力、人間性等」に関しての「他者と協同しながらねばり強くやり抜く態度の育成」にあるように思う。児童が挫折せず、プログラミングに興味を持って、中学校・高等学校へ教育を接続するためには、児童にとって魅力ある教材やゴールを用意し、困難を乗り越えたいというモチベーションを築くところにあると考える。

文部科学省の「小学校プログラミング教育に関する研修教材」には、プログラミング言語教材として「Scratch」が挙げられている³。Scratchは、マサチューセッツ工科大学メディアラボが開発した子供向けのプログラミングツールで、ブロック状の命令をドラッグ&ドロップし組み合わせることで記述できる⁴。図1は「小学校プログラミング教育の手引き（第二版）」²にも示されているペン移動で図形を描くプログラム例をScratchで表現したものである。最初は図1左のように、同じ長さの辺（Scratchでは「200歩」と表現される）を120度で折り返すことを3回縦に繋いで描く順次進行プログラムを作る。プログラミングが理解できると図1中央のように3回繰り返しのブロックを利用することでより短く書ける事を学ぶ。どちらも図1右に示す同じ結果が得られる。

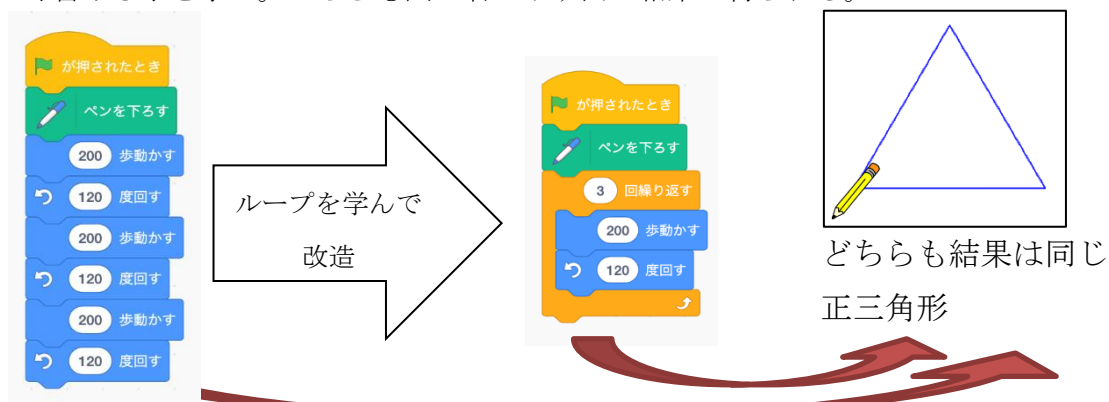


図1：Scratchでの正三角形を描くプログラミング

同じ結果であれば、どちらのプログラムでも良いが、ここで課題を四角形や五角形など多角形に応用することで有用なプログラムがどちらであるかを児童は理解する。図1左の順次進行プログラムで多角形を描く場合は、図形の角数に応じて縦に長いプログラムを書く必要がある。しかし、図1中央のループ構造プログラムは、図2に示すように構造は同じでもパラメータ（命令を設定する数値データ）の入れ替えで対応できる。例えば、図2左のように、繰り返し回数を4回とし、角度を90度に変更すると四角形が描ける。小学校課程では、この様な試行錯誤による気づきを重点的に指導することが予定されている。

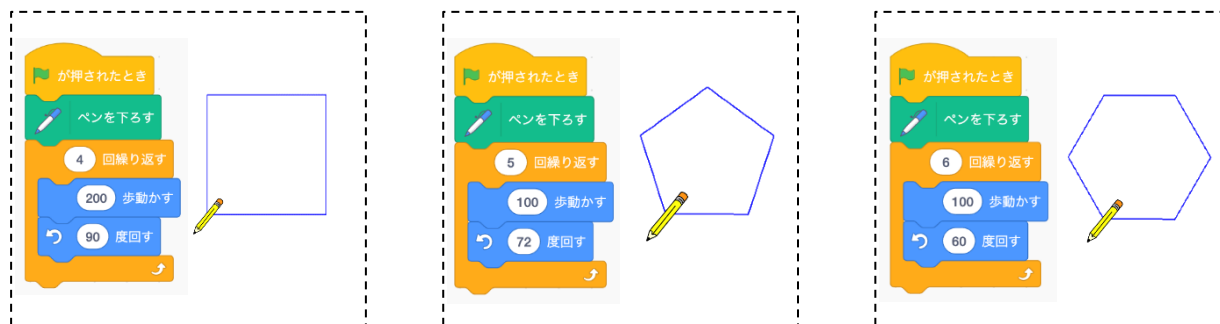


図 2：汎用性の気づき

この様に Scratch は、一般的な文字ベースのプログラミング言語と比較して理解しやすい仕組みが作られている。おそらく、一部の児童は、より自分の理想を実現したいというモチベーションから、授業の枠を越えた作品を制作する可能性があり、実際にそのような作品が Scratch の Web ページにいくつか掲載されている。

Scratch は導入教育として非常に強力な教材となり得るが、その先の中学校・高等学校課程のプログラミング教育において、新たな魅力を考える必要がある。スマートフォン・高性能ゲーム機、映画や TVCM などで見かける鮮やかなコンピュータグラフィックス（以下、CG）など、身近に高度な映像を観る機会の多い世代に対して、Scratch の魅力を超える教材を与える事が望ましい。

達成感あるプログラミング教育の研究

学習指導要領にプログラミング教育が導入される以前、つまり現在のプログラミング教育は、大学や専門学校で実施されてきた。おそらく今後は中学校・高等学校課程において同じようなプログラミング教育がなされることになるかと予測する。

高等教育では職業実践力を目標としており、Java や Python など実務で利用されているプログラミング言語を学ぶことになる。よって、Scratch のようなブロック状の命令ではなく CUI による文字ベースの命令を入力し、文字出力の結果を確認しながら学ぶのが一般的である。文字出力のプログラムは、シンプルな記述のみで作成でき、結果は定量的で確実な正解を示せるメリットがあるが、見た目に地味で達成感が低い。そこで出力結果を CG やロボット等とする事で達成感を上げる教材研究が 2000 年代初頭より行われてきた。本章では、それら先行研究のサーベイと本研究の位置づけを述べる。

辻合は、Pov-Ray による 3DCG プログラムの発展として、LEGO MINDSTORMS を教材としモーター制御とタッチセンサーによるインタラクティブな造形作品をゴールとする Java 言語のプログラミング教育を報告している^{8,9}。坂田らは、Scratch など教育用プログラミング環境から文字ベースの入出力によるプログラミング学習に接続するギャップを

埋める目的で、Processing 言語による 2D および 3DCG の演習と、応用として Arduino を用いたマイコンによる制御プログラミング教育を報告している¹⁰。山田らや長谷川らは、小学校でのプログラミング導入教育の方法として、Arduino で認識するタンジブルブロックによる遊び感覚のプログラミングを提案している^{12, 13}。

これらの研究からは、Arduino など時代の流行りを導入し学習者にとって魅力あるゴールの設定と、入出力で文字を利用しない方法を検討していることがわかる。しかし、LEGO MINDSTORMS のような汎用ツール以外の独自デバイスや教材は、まだまだ研究段階でありデバイスの一般化には困難が予測できる。とは言え、触覚を使ったデバイスは、プログラミング教育に新たな魅力を与えられる可能性がある。本報告では、触覚デバイスは用いないが、将来的には 3DCG から 3D プリンタ等で実体を出力することや、ブロックを組むことで 3DCG が作れる入力デバイスなども考えられる。その教育効果は興味深いと考えている。

他方、特別なデバイスを利用しない研究として近藤は、学生が同じ結果をゴールとし正確な解答を出力することではなく、各自の CG 表現をゴールとする事で Java 言語のプログラミングに興味を持たせている⁴。また後に、CG 表現をゴールとしつつ Processing 言語を用い、あらかじめ各プログラムの例となるソースを用意した Example ベースプログラミングを導入することで主に CG に関するアルゴリズムの理解を重視した教育法を提案している⁵。高山らは、軌道をトレースするロボットの動きを CG アニメーションとして再現したゴールを用意して Java 言語のプログラミングに興味を持たせている⁶。また、Web 上で Processing 言語のプログラムの実行やレポート管理が可能な独自の e-ラーニングツールを開発し CG 重視の課題を用いて、その有用性を報告している⁷。米元は、グラフィックス課題をゴールとした Java 言語の演習に対して、企業団体などの外部評価を実施し、その評価内容を報告している¹¹。

これらの研究からは、3DCG やアニメーションが、学習者にとって魅力あるゴールとなることが分かる。3DCG は前提知識として幾何学等の数学を理解する必要があり近藤や高山はその教育ツールとしてプログラミングを導入している。本報告では、3DCG の数学を学ばせるのではなく、プログラミング教育の為に 3DCG を用いて学習者のモチベーションを上げ、プログラミング学習の効果をあげることを目標としている。

DirectX から three.js への変更

筆者は、2009 年より 3 年次の演習科目「情報システム応用演習（旧科目名：プログラミング演習 3）」において 3 次元 CG プログラミングのテーマを担当している。本演習では他にも様々なテーマがあり、各テーマそれぞれ毎年 10 名程度の学生が選択している。

一般的に 3DCG は画像の表示であるレンダリングに計算時間をかけた手法（レイトレー

シング法等) と、リアルタイムで描画され動きを表現できるものがある。学生は動きのある映像やマウス等でインタラクティブに操作できる CG を好む傾向があることより、本演習ではリアルタイム 3DCG プログラミングを導入した。リアルタイム 3DCG のプログラミングの演習では、一般的に OpenGL や DirectX を学ぶことが多い。本演習でも 2016 年までは DirectX を利用していたが、2017 年より three.js を導入した。three.js は OpenGL を Web で表示できる WebGL を簡単に扱えるようにしたライブラリであり、少ないコード記述で見栄え良い 3DCG 表示を得ることができる。また、three.js を含む WebGL は Web 標準の技術であるため、ブラウザが対応していれば、どのような OS でも実行可能であり、スマートフォン等タブレット端末でも表示できる。three.js は細かな表現力が DirectX に劣るものの前述したマルチプラットフォームであることは学生の自習環境に影響を与えないメリットがある。



図 3：ティーポット形状に反射環境マップを実装

また、three.js は DirectX と比較してコードが非常にシンプルになることも強力である。例えば、図 3 のように Utah Teapot 形状に環境マッピングを適用する場合を考えると、DirectX では形状の法線データや頂点色を格納する領域を用意しなければならない。そのための格納領域を持った頂点データ構造体を新たに作成し、ティーポット形状の頂点データや頂点データを元に計算した法線データを再定義する手順が必要となる。当然、プログラムは煩雑となり 3DCG の数学的知識を持たない者にとっては理解しにくいソースとなる。他方、three.js では、基本的な手順は単色のティーポットの表示と同じで、パラメータとして環境マッピングを設定するのみで表現できる。

表 1 に、DirectX と three.js でのプログラムソースによるコーディング差を示す。両プログラム共に各種設定は省略し、環境マップ部分のみを比較提示する。また、省略した設定でテクスチャデータを「mTexture」に読み込んでいるものとしている。この表 1 より、DirectX は 3DCG の概念を理解し書く必要があり記述量も three.js より多いことがわかる。

表 1 : DirectX と three.js でのコーディング差 (形状定義部分抜粋)

DirectX による環境マップコード	three.js による環境マップコード
<pre>//手順1: ティーポット形状の準備 LPD3DXMESH pMesh = 0; D3DXCreateTeapot(pD3DDevice, &pMesh, 0); //手順2: 法線ベクトルなどを格納できる新たな FVF にコピー pMesh->CloneMeshFVF(pMesh->GetOptions(), MY_VERTEX_FVF, pD3DDevice, &m_pMesh); //手順3: 古いモデルデータを破棄 SAFE_RELEASE(pMesh); //手順4: 法線ベクトルを計算 D3DXComputeNormals(m_pMesh, NULL); //手順5: カメラ座標系における各頂点の法線の XY 座標がテクスチャの UV 座標として使用される ようにする pD3DDevice->SetTextureStageState(0, D3DTSS_TEXCOORDINDEX, D3DTSS_TCI_CAMERASPACENORMAL); //手順6: 4 象限から 1 枚に調整 D3DXMATRIX mTexture, m1, m2; D3DXMatrixScaling(&m1, 0.5, -0.5, 1.0); D3DXMatrixTranslation(&m2, 0.5, 0.5, 0.0); D3DXMatrixMultiply(&mTexture, &m1, &m2); pD3DDevice->SetTransform(D3DTS_TEXTURE0, &mTexture); //手順7: テクスチャの UV 座標系 (2 次元) と通 常座標系 (3 次元) の調整 pD3DDevice->SetTextureStageState(0, D3DTSS_TEXTURETRANSFORMFLAGS, D3DTTFF_COUNT2); //手順8: テクスチャステージ設定 pD3DDevice->SetTextureStageState(0, D3DTSS_COLOROP, D3DTOP_MODULATE4X); pD3DDevice->SetTextureStageState(0, D3DTSS_COLORARG1, D3DTA_CURRENT); pD3DDevice->SetTextureStageState(0, D3DTSS_COLORARG2, D3DTA_TEXTURE); //手順9: 描画 m_pMesh->DrawSubset(0);</pre>	<pre>//手順1: ティーポット形状の準備 var geometry = new THREE.TeapotBufferGeometry(50, 10, true, true, true, true, false); //手順2: 透過・反射の別 mTexture.mapping = THREE.EquirectangularReflectionMapping; //手順3: 反射するマテリアルの準備 var material_Equi = new THREE.MeshPhongMaterial({ envMap: mTexture, reflectivity: 0.75, combine: THREE.MixOperation, color:0xff5533, specular: 0x333333, shininess: 150}); //手順4: 形状にマテリアルを適用 mesh = new THREE.Mesh(geometry, material_Equi); //手順5: 描画 scene.add(mesh);</pre> <div style="border: 1px solid black; border-radius: 15px; padding: 10px; margin-top: 20px;"> <p>ただし、外部形状を読み込んだ場合は以下に示すコードで法線ベクトルの計算が必要となる。</p> <pre>geometry.computeVertexNormals();</pre> </div>

正確には、three.js でも標準で実装されていない形状、例えば Stanford Bunny などを読み込んだ場合、DirectX の手順 4 に相当する「法線ベクトルの計算」が必要になるが、DirectX と同じく 1 行 (`geometry.computeVertexNormals();`) の追加のみであり、手順の手間は増えない。また、three.js では、例えば「`mTexture.mapping`」の値を「`THREE.EquirectangularRefractionMapping`」に変えるなど、簡単な変更で図 4 のような透過環境マッピングに変更することができる。同じように、影の表示など様々な面で簡単にコードを書くことができる。この直感的に扱える記述性が three.js の強みである。

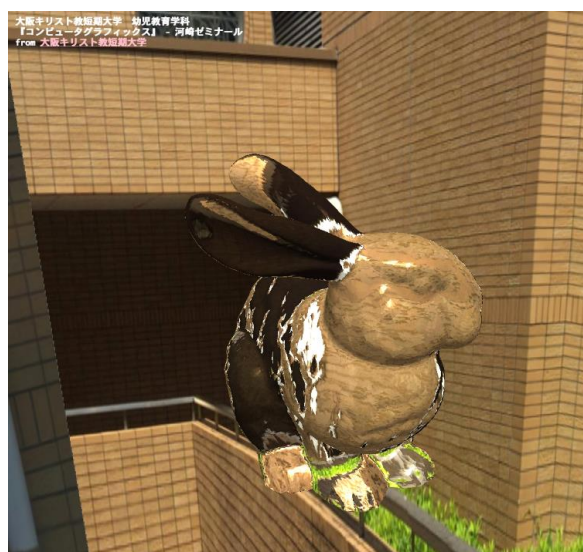


図 4 : Stanford Bunny 形状に透過環境マップを実装

演習の目的と進行

本演習では、three.js ライブラリの利用方法を学び、最終的に自分の作りたい作品を制作することを目標とした。プログラミングに関する演習の進行を以下に示す。

第 0 回	ガイダンス (JavaScript と three.js の説明)	
第 1 回	three.js 基礎 (頂点とカメラの配置)	資料数 14 ページ
第 2 回	プリミティブモデル表示・座標変換	資料数 19 ページ
第 3 回	アニメーション・シェーディング	資料数 11 ページ
第 4 回	ライティング・シャドウイング	資料数 16 ページ
第 5 回	キーフレームアニメーション・テクスチャ	資料数 11 ページ
第 6 回	外部形状読み込み・UV マッピング	資料数 14 ページ
第 7 回	バンプマッピング・ディスプレイスメントマッピング	資料数 14 ページ
第 8 回	環境マッピング・トゥーンシェーディング	資料数 22 ページ

履修者は 2 年次で C 言語と Java 言語を学んでおり、JavaScript への移行は 0 回目のガイダンス時に説明するのみで対応できた。本演習は 2 コマ連続で行われ 1 回分の学びと演習の時間が比較的多めに設定できる。実際には汎用の 3DCG ソフトでモデリングを学ぶ回を設けているが、それを除いた 3DCG プログラミングに関する演習は、以上に示す全 8 回の構成で、途中、5 回・6 回の間で、中間テストによる理解度チェックを 2 週に分けて実施した。資料は昨今、文字を読まない学生が増えたことも鑑み、画像例を多めに作成し 8 回目を除き各回 20 ページ以内の量で収めた。8 回目以降は 3 週にわたって自由課題の作成時間とした。全ての資料とサンプルプログラムは、学生が自宅からも予習・復習できるように moodle サイトにコースを作成して提示した。

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; //立方体オブジェクト
function objects0 {
    //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();
    //立方体オブジェクトの生成
    cube = new THREE.Mesh(geometry, material);
    //立方体オブジェクトのシーンへの追加
    scene.add(cube);
    //立方体オブジェクトの位置座標を設定
    cube.position.set(0, 0, 0);
}

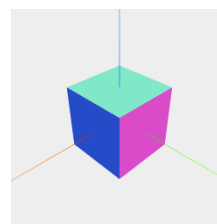
```

BoxGeometry クラス

親クラス：Geometry

対象レンダラー：WebGL, Canvas2D

立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)	立方体の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	立方体の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計算量が増え負荷が伏々になる。
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が申請する文字列
parameters	<object>		コンストラクタの引数で指定された立方体オブジェクトを生成するのに必要なパラメータ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レンスも図 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、初回は苦勞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 下のクラスリファレ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。

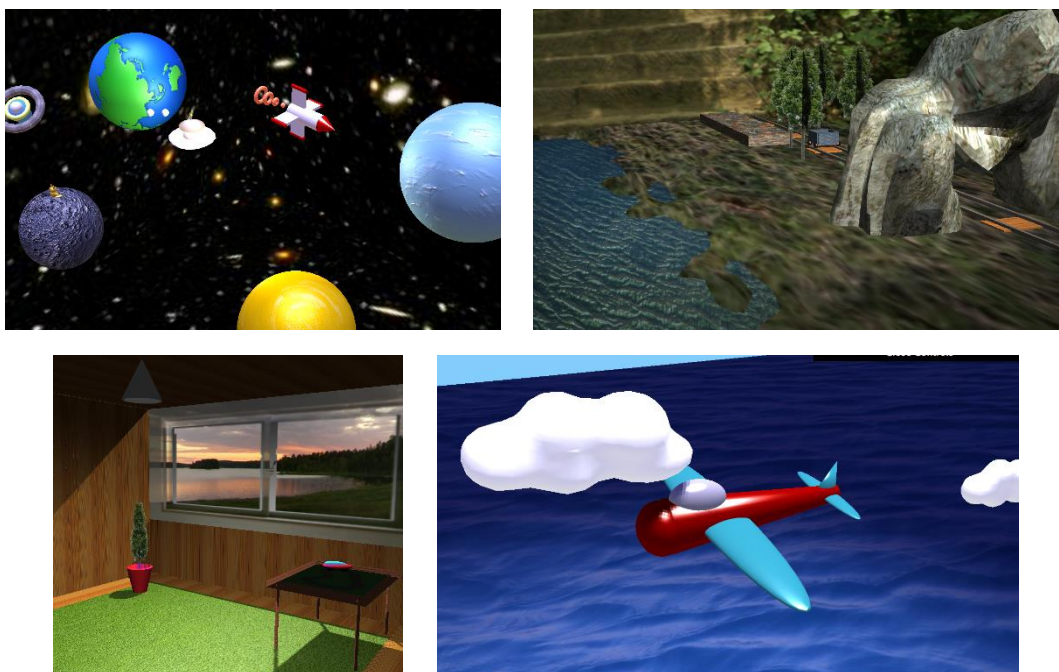


図 6：学生による作品例

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
③ あなたはプログラミングが苦手ですか	JavaScript	0
	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 のがありますか (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, 『芸術系学生の実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, 原田拓輝, 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, 原田拓輝, 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/>