

福山平成大学経営学部紀要
第 16 号 (2020), 55–63 頁

HTML5 および Javascript による 3 次元幾何アニメーション表示

尾崎 誠

福山平成大学経営学部経営学科

要旨：これまで Javascript により HTML5 上で動作する 3D Graphics Viewer^[1]というプログラムを開発し、それを利用して 2 変数関数グラフや 3 次元パラメータ関数を Web ブラウザ上で 3 次元表示させることを可能にした。さらに、WebGL を描画に利用することで、描画速度の高速化にも成功した。そのため今回は、3 次元幾何アニメーションの表示機能の開発に取り組み、基本的なオブジェクトの 3 次元幾何アニメーション表示を可能とした。簡単なマクロ言語によるプログラムにより、3 次元で図形を描き動かすことによって、数学的な関数や座標の概念を学ぶ環境を提供することが可能となった。本プログラムは、物理学シミュレーションにも応用可能であり、将来的にはシミュレーション言語に発展させる予定である。

キーワード：HTML5、Javascript、Web ブラウザ、3 次元幾何アニメーション

1. はじめに

これまで、社会システム分析のための統合化プログラムである College Analysis^[2]を Javascript により HTML5 上で動作するよう移植を進めてきた。今回は、数学的な関数や座標の概念をグラフィックの形状や動きによって学ぶことのできる 3 次元の幾何アニメーションについて紹介する。

幾何アニメーションは簡易的なマクロ言語によって描画されるが、言語仕様の作成には可能な限りの簡潔さを心掛けた。また、コマンド数もサンプルを考えながらできるだけ減らし、中学生程度の知識でも、かなりの描画が可能になるように努めた。今後もアニメーションの機能向上を図っていくが、マクロ言語の仕様はできるだけ上位互換になるように発展させる予定である。

プログラムデータは、1つのプログラムを1ページとし、パソコンであればローカルに保存されたデータの読み込みを可能としている。その際のデータフォーマットは College Analysis と同じとした。そのため、これまでに College Analysis で作成してきたデータを利用することができる。

2. 3次元幾何アニメーション

3次元幾何アニメーションプログラムを起動すると、図2.1のような画面が表示される。



図 2.1 3次元幾何アニメーションプログラムの初期画面

HTML5 および Javascript による 3次元幾何アニメーション表示

Webブラウザ上では650×450ドットの描画範囲に描画を行う。初期状態では、描画範囲の中心が中心座標 (0,0,0) となり、視点は (0,0,1000) に位置し、視野角は45°、注視点は (0,0,0) としている。マウスの左右へのドラッグにより、Y軸周りに視点を回転、上下へのドラッグにより、X軸周りへの視点を回転させることができる。また、ホイール操作によりZ軸方向への視点座標の増減を行える。照明は、座標 (0,0,1000) に配置し、平行光源としている。すなわち、太陽光とほぼ同じく光源から無限遠まで同じ強さで光線が平行に到達する光源である。プログラムは中央のテキストボックス内にコマンドを用いて記述する。

コマンドを用い、プログラムを記述または「ファイル選択」ボタンをクリックすることで、ローカルフォルダからプログラムデータを読み込み、「スタート」ボタンをクリックするとアニメーションが実行される。アニメーション実行後、「スタート」ボタンは「リセット」ボタンに置き換わり、「リセット」ボタンをクリックするとアニメーションが停止され、データは初期状態に戻される。さらに、「リセット」ボタンは「スタート」ボタンに置き換わり、再度「スタート」ボタンをクリックすれば、アニメーションを初期状態から再描画できる。「停止」ボタンをクリックすると、アニメーションが一時停止し、「停止」ボタンは「再開」ボタンに置き換わる。「再開」ボタンをクリックすると、アニメーションが再実行され、「再開」ボタンは「停止」ボタンに置き換わる。

「デモ」ボタンはデモンストレーションを開始するボタンであり、クリックするとデモンストレーションが実行される。デモンストレーションは、1つのプログラムのアニメーションを12秒間実行し、プログラムが複数ある場合には順番に切り替えながらアニメーションを行う。デモンストレーション実行後、「デモ」ボタンは「解除」ボタンに置き換わり、「解除」ボタンをクリックするとデモンストレーションは終了し、「解除」ボタンは「デモ」ボタンに置き換わる。

「ページ数」の右横に配置されている「<」ボタンと「>」ボタンは、プログラムのページが複数ある場合に実行するプログラムの切り替えを行うボタンである。「現在のページ」に今選択されているプログラムのページ数が表示され、そのプログラムがテキストボックス内に表示される。「ページ数」は読み込んだファイルの最大のページ数が表示される。

「<」ボタンをクリックすると、現在のページ数が1つ減少する（最小値は1）。

「>」ボタンをクリックすると、現在のページ数が1つ増加する（最大値は最大のページ数）。

プログラムのファイルから読み込んで使用する場合、そのファイルフォーマットは以下の通りである。

1行目	プログラムの数, 最大の行数, 最大の列数
2行目	1番目のプログラムの行数, 1番目のプログラムの列数
3行目	1番目のプログラムの名前 (空白でもよい)
4行目	コマンド
	:
m行目	2番目のプログラムの行数, 2番目のプログラムの列数
m+1行目	2番目のプログラムの名前
m+2行目	コマンド
	:
n行目	i番目のプログラムの行数, i番目のプログラムの列数
n+1行目	i番目のプログラムの名前
n+2行目	コマンド
	:

次に、プログラムで利用できる各コマンドについて少し詳しく説明する。[]で囲まれたパラメータは省略可能であり、パラメータ間の空白はあってもなくてもよい。

描画範囲変更

```
rangex xmin, xmax (rangey ymin, ymax)
```

```
range min, max
```

描画範囲を一時的に変更する。メニューの描画範囲を表すテキストボックスは変化しない。2番目のように、`range`だけであれば全ての軸の描画範囲を同一に変更する。

時間変数 (time = 0, 0.1, 0.2, …)

```
time
```

時間を表す予約変数で、0, 0.1, 0.2, … (秒) と増加していく、アニメーションの基本となる変数である。描画は0.1秒ごとなので、簡単な動画で描画が追いつけば、実時間となる。また、描画が追いつかずコマ落ちが発生した場合には、コマ落ちに合わせて時間変数も変化するようにしてあり、その時にもなるべく実時間に合うようにしてある。

定義

```
define 変数名 = 定義式
```

変数名を定義式で置き換えることができる。プログラムの実行の最初に置換処理を実行するので、描画スピードのロスは少ない。

実行の繰り返し

```
playback 時間 (秒)
```

実行メニューの「スタート」ボタンをクリックした際、またはデモンストレーションを実行している場合には、各プログラムの開始時の描画の実行時間を決める。その描画時間が終了したら再描画が繰り返される。

HTML5 および Javascript による 3次元幾何アニメーション表示

球体

`ball(x, y, z, r, color [, tr])`

座標 (x, y, z) を中心にした半径rの立体的な球を描画する。色を指定するcolorは&Hffaa80のように16進数で指定する。trは不透明度 (0~1) で与える。trは省略可能であり、デフォルトで1 (不透明) である。colorとtrについては、他でも同じように使用する。

長方形

`box(x, y, z, wx, wy, wz, color [, x_angle, y_angle, z_angle, tr])`

座標 (x, y, z) を中心にしたx軸方向の幅wx、y軸方向の高さwy、z軸方向の奥行wzとなる長方形をcolorで指定された色で描画する。また、x_angleでx軸周りの回転、y_angleでy軸周りの回転、z_angleでz軸周りの回転を与えることができるが省略することもできる。省略された場合には、その軸周りの回転は0となる。

円錐

`cone(x, y, z, r, height, color [, x_angle, y_angle, z_angle, tr])`

座標 (x, y, z) を中心にして、x軸方向に半径rとなる円の下底を持ち、y軸方向へ高さheightとなる円錐をcolorで指定された色で描画する。高さはどちらでも指定することができる。また、長方形と同様にxyz軸周りの回転や不透明度を指定することができる。

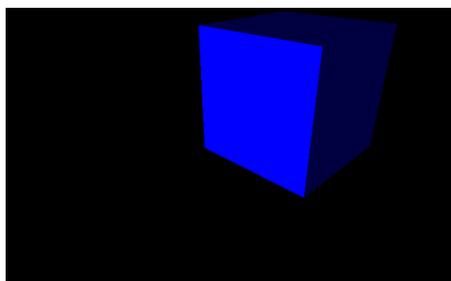


図 2.2 長方形の描画

コマンドを用いてプログラムを記述し、アニメーションを実行した結果を図2.2に示す。boxによりxyz方向にそれぞれ100となる長方形、すなわち立方体を青色で描画し、time関数を使ってz軸周りにxy方向に半径100となる回転運動を行わせている。また、視点は描画された長方形が分かりやすいように移動させている。

次に、図2.3では円錐の描画を行っている。円錐の高さをtime関数にsinを組み合わせることで、-200~200の範囲で変化させるアニメーションが描画される。



図 2.3 円錐の描画

最後に、図2.4では球を3個配置し、中心座標 (0, 0, 0) を軸としてtime関数と三角関数を組み合わせることで回転運動を描画している。

いずれのプログラムでも、直感的に分かりやすい簡単な記述によりアニメーションの描画を可能としており、プログラム初学者でも数学や物理がある程度理解できていれば実行が可能となっている。

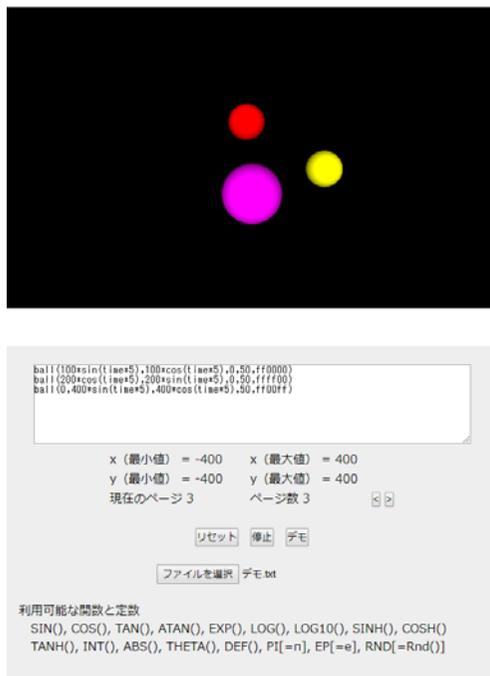


図 2.4 球の描画

3. おわりに

いくつかの例を用いて、幾何アニメーションのプログラムを説明してきたが、今回開発したプログラムでは、まだ基本的なコマンドしか使用することができない。これからもプログラムの開発を継続していくことで、今後の応用範囲は広がる。数学の関数や座標の概念の把握、1 次変換やベクトルについての学習、物理教材作り、簡単な物理シミュレーション、さらに将来的には本格的な物理シミュレーションと可能性は大きく広がっている。

今回は描画に WebGL を取り入れることにより、ブラウザ上での高速描画が可能となり、3 次元幾何アニメーションに対応させることができた。だが、2 次元幾何アニメーションのプログラムをできるだけ活用して 3 次元幾何アニメーションに対応させたため、十分に WebGL の機能が活用できていない。そのため、今後はプログラムの大幅な変更を行い、より高速な描画を可能としたい。これは、今後シミュレーション表示をさせるためには必要となってくる。WebGL の機能を十分に活用できれば、GPU などを演算に利用できるので、最新の環境では非常に高速化を図ることが可能となる。また、WebGL を取り入れたことで、多彩なライブラリを利用することが可能となった。例えば、VR のライブラリを組み込めば、3 次元幾何アニメーションを VR で鑑賞することが可能となる。VR に対応させることで、より直感的に 3 次元への理解が可能となるので、時間が許せば是非組み入れたい機能である。

現在、ファイル読み込みの機能を持たせ、プログラムを外部から読み込むことを可能とし、そのファイルフォーマットを College Analysis のものと共通とした。そのため、これまで作成してきた教材用のファイルが利用できるようになっただけでなく、これまで College Analysis を用いて講義を受けた学生にとって、簡単にプログラム環境の移行ができるようになった。

その一方で、タブレット端末やスマートフォンでの利用では、データファイルの読み込みについて課題が残っており、これについて早急に対応する必要がある。例えば、Dropbox や OneDrive などのクラウドサービスが利用できるように対応するなどを検討する必要がある。本学では、Office365 の利用環境が整っているため、OneDrive への対応が望ましいように思えるが、出来るだけ利用者の負担が少なくなるよう慎重に検討していきたい。

本プログラムでは、これまで仕様を主体に考えてきたため、デバッグ機能は作成していない。しかしこのままでは、実際に初心者が利用するにはマクロ作成が難しいであろうと思われる。そのため、まずは簡単に作成できることからデバッグ機能を追加し、実際に使用して利用者が間違えるところを見極め、順次整備していく必要がある。これについても今後の課題である。

今回、コマンド解析および描画については、Javascript のライブラリとして、本体とは別の JS ファイルとして作成した。そのため、他のプログラムからも呼び出して利用することができる。他にどのような利用方法があるかについても検討を行っていきたい。

参考文献

- [1] HTML5 および Javascript によるグラフィックスビューアとその応用、尾崎誠、福井正康、福山平成大学経営研究、第 11 号、(2015)、27-32
- [2] 社会システム分析のための統合化プログラム 2 2 —幾何アニメーション—、福井正康、福山平成大学経営研究、第 10 号、(2014)、79-104

3D geometric animation displays in HTML5 and Javascript

Makoto OZAKI

*Department of Business Administration, Faculty of Business Administration,
Fukuyama Heisei University*

Abstract: In the previous studies, I developed a program called 3D Graphics Viewer^[1] which runs on HTML5 in Javascript. And utilizing this program, I have made it possible to have 2 variable function graphs and 3D parameter functions in 3D displays on web browsers. I also was successful in increasing the speed of painting by utilizing WebGL. In the current study, I worked on the development of a 3D geometric animation display function, and successfully made the 3D geometric animation displays for basic objects. The program is run by a simple macro language and animates figures on 3D displays, thus make it possible to offer an appropriate environment for learning mathematical functions and coordinates. This program can also be used in the field of physics (e.g. simulation experiments). I plan to develop a simulation language based on this program in forthcoming studies.

Key Words: HTML5, Javascript, web browser, 3D geometric animation

尾崎 誠