

タイトル	タブレット端末での使用を考慮した3Dオブジェクト操作UIの開発
著者	菊地, 慶仁; Kikuchi, Yoshihito
引用	工学研究 : 北海学園大学大学院工学研究科紀要(12): 31-35
発行日	2012-09-28

タブレット端末での使用を考慮した 3D オブジェクト操作 UI の開発

菊地 慶仁*

Development of 3D object operation user interface for tablet terminal

Yoshihito Kikuchi*

要旨 (Abstract)

近年、ウェブや電子図書の閲覧用としてタブレット端末が広く用いられている。これらの端末で、従来のマウスによる操作を前提として開発されてきた3Dオブジェクトを読み込んで操作するとPC上での操作と挙動が一致しない問題が発生する。本報告では、PC上での従来のマウス操作及びタブレット端末上で操作並びに効果を共通化する方式について検討を行い、実際に開発を行った結果について報告する。

1. タブレット端末上での3Dオブジェクトの操作

近年ではWWW (World Wide Web) や電子書籍の閲覧を目的としてiPadに代表されるタブレット端末が広く用いられている。これらの端末は画像を表示する画面に接触センサが設けられたタッチインターフェースを備えている。従来のマウスを用いる操作の形態とは違い(1)画面にタッチする(タップ動作)、(2)1本ないし2本の指先で端末の表面を撫でる(スワイプ動作、ピンチイン/アウト動作)などの動作によって操作を行う。

これらの端末では専用のアプリケーションを用いることはもちろんであるが、ウェブを閲覧する用途も多く、そのコンテンツの一つにWebGL¹⁾がある。WebGLはコンピュータグラフィックスの分野で用いられている3Dオブジェクトを描画するためのJavaScriptプログラムである。WebGLは、専用アプリケーションでPCの画面上でハードウェアアクセラレートされた3Dオブジェクトの描画を行うライブラリのOpenGLと互換性がある。WebGLのメリットは、描画に関してはWebGLをサポートするブラウザに任せてハードウェアアクセラレートされた3Dオブジェクト描画を行うことができる点がある。このため

に3Dオブジェクト描画用の専用プログラムを必要とせずに3DCGを含むウェブコンテンツを配布することが可能となる。この機能を用いて、対象を様々な視点から描画して操作手順を教示する機械系の電子マニュアルなどの用途で利用されている。

WebGLコンテンツの操作はブラウザに読み込んで従来型のマウス操作を行うことを前提として開発されてきた。このためタブレット端末のブラウザでは、PCと同じような操作をしても結果が異なる場合があることが判明した。

本研究では、WebGLで表現された3Dオブジェクトを対象として、PC上のブラウザとタブレット端末上のブラウザで、操作及び実行結果が同一となるインターフェースを開発することを目的とする。

このことによって、異なる環境下でも同一のコンテンツの操作方法と効果を共通化させることが可能となり、また開発したプログラムをGUI用部品として共有化すること、などが期待できる。

* 北海学園大学大学院工学研究科電子情報工学専攻

Graduate School of Engineering (Electronics and Information Eng.), Hokkai-Gakuen University

2. WebGL の概要と本研究における課題

2.1 Web ブラウザによる 3D オブジェクト描画に関する必要性

これまで 3D オブジェクトを含むコンテンツをネットワーク経由で配信するためには

- 1) Java アプレットなどの専用の表示プログラムをウェブページに埋め込む形で提供し、表示用プログラムがデータを読み込んで表示する。
- 2) 3D オブジェクトの表示機能を持った PDF リーダー向けに PDF 形式で提供する。

の 2 つの方式があった。しかしながら 1) ではウェブブラウザに組み込むプラグインの限界があるためにハードウェアアクセラレートされた高速な表示は無理である。現在の 3D オブジェクト描画では、数千万以上の多数の 3 角形要素 (ポリゴン) に対して座標変換や模様パターンのマッピングを行った上で陰面を消去して一秒間に 30 フレーム以上の頻度でレンダリングを行う必要がある。したがってプラグインタイプでは、高度な表現は現実的には難しい。2) ではファイル単位での配布しか行えず、動的にコンテンツを生成することができないなどの難点があった。

そのために、1) ブラウザ自身にハードウェアアクセラレートされた 3D オブジェクト描画能力を持たせること、2) 3D オブジェクト描画プログラムをウェブコンテンツとして提供してブラウザが備えている描画能力で実行させること、が必要となってきた。この要求に対応するために、次節で述べるように HTML に組み込まれた JavaScript プログラムとして WebGL を実行させる枠組みが開発された。

2.2 HTML 5 によるウェブページ構成

図 1 に本研究の対象である WebGL を含む HTML 5 のドキュメント構造とブラウザの関係を示す。

ドキュメントは全体として HTML 5 規格として規定されている。WebGL オブジェクトはドキュメント中の <canvas> タグで指定された領域に描画される。

実行プログラムに相当する JavaScript は <script> タグで指定した領域中に格納され、WebGL の描画用スクリプトもこの領域に格納さ

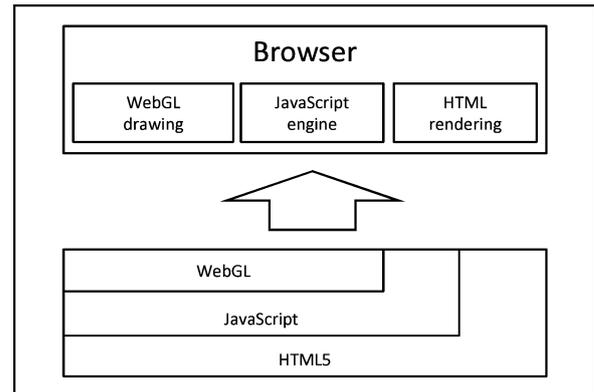


図 1 HTML 5 の構造とブラウザでの処理

れる。

このような HTML ファイルを読み込むブラウザには、HTML 5 の規格に準拠した <canvas> タグのレンダリング機能が必要となる。また通常のブラウザでは標準で JavaScript の実行機能が備えられているが、特に WebGL を読み込んでハードウェアアクセラレートした 3DCG 描画機能も必要となる。この描画機能では Web ページのレンダリングと、GPU による 3D オブジェクトの描画をスムーズに行う必要がある。

2.3 タブレット端末操作とマウス操作の違い

タブレット端末ではマウスを使わずにタッチパネルに指で触ることだけで操作を行うユーザーインターフェースが提供されている。ポインティングデバイスで画面上のカーソルを移動させる点は同じだが実際には次のような違いがある。

- 1) 従来の GUI では画面のマウスカーソルを見ながらマウスは見ずに操作をするが、タッチインターフェースでは画面上の該当するエリアに直接タッチすることで直感的に操作できる。
- 2) 従来のマウスを用いる GUI では画面上の特定の座標をポイントする際には比較的十分な分解能を持っていたが、逆にタッチインターフェースでは指の大きさに影響されて画面上でのタッチできる分解能が小さい。このためタッチ画面上にキーボードを表示して入力する際には困難が生じる。

2.4 WebGL 操作における問題と本研究での課題

本研究で対象としている問題は、WebGL を表

示している Web ページの操作を行う際に発生する。PC 用ブラウザ上でマウスカーソルを WebGL オブジェクト上におき左ボタンをクリックした状態で画面上でカーソルを移動させる動作を行う。PC ブラウザ上ではこの操作によって 3D オブジェクトの空間中での姿勢を変更することができる。

しかしながらタブレット端末のタッチインターフェース上で指をブラウザ上で移動させると、ブラウザのウィンドウが画面上で移動してしまい 3D オブジェクトが回転しない問題が発生する。

この問題は PC 用 Windows とタブレット端末用 OS とでのマウス/指タッチで入力されるイベントの処理方式が異なっているために発生する。Windows では、マウスクリックのイベントは OS からアプリケーションに伝達され、アプリケーションでは WebGL オブジェクトにタッチしスワイプ動作をしていると判断してイベント処理を行う。結果としてオブジェクトが回転する。タブレット端末では、OS 側がイベントを取得して処理をしてしまいブラウザにイベントが渡されない。従って画面上でブラウザ画面が移動してブラウザ画面中の 3D オブジェクトには変化は生じない。

上記の問題点に対して、本研究では PC 用のブラウザ及びタブレット端末上でのブラウザで同じ方法で操作可能となるユーザインターフェースを開発することによる解決を目的とする。

またこのような解決方法を実現することで、端末にとらわれない標準的なユーザインターフェース部品を提供することが可能となる。

3. 本研究での提案

実際に開発に入る前に予備的な実験を行い、どのジェスチャがアプリケーションに渡るか、また OS 側で取得されてアプリケーションに渡らないかについて確認を行った。

その結果、画面に指を接触させたままなぞるスワイプ操作は OS 側で取得されてアプリケーションには伝わらないことが判明した。

これに対して画面上を指で叩くタップ操作は、OS からブラウザにイベントが伝達され HTML 5 の canvas 要素で取得して対応する JavaScript を動作できることが確認できた。

そこで本研究では、ユーザのジェスチャを受け

取る canvas と 3D 画像を表示する canvas を同一位置にレンダリングすることで、見た目では画面上のオブジェクトの領域をタップさせ、その位置によって 3D オブジェクトの回転方向を制御するユーザインターフェースの開発を行う。

4. 実験

実験では図 2 に示す Creative 製 Android タブレット端末の ZiiO 10 を用いた。OS のバージョンは Android 2.1 で提供されているがファームウェアアップデートによって Android 2.2 として用いた。

ブラウザは Android 端末上で WebGL の動作が保証されている Firefox for Mobile をダウンロード/インストールして使用した。

試作したインターフェースでは、操作部分のアニメーションと WebGL の表示の両方で複数の canvas 要素を使用している。canvas 要素を表示させるためには onload という処理が必要になるが、onload は通常、1 つのウェブページに対して 1 度しか使えないため、そのままの状態では複数回使用すると最後に使われた onload の canvas しか表示されない。そのためここでは、onload を複数回使用可能にするライブラリ jQuery⁴⁾ を使用した。

試作したユーザインターフェースを表示したブラウザ画面を図 3 に、その構造を図 4 に示す。表示されている月面は空間中に浮かぶ球体の 3D オブジェクトに月面のテクスチャをマッピングしたものである。元となる WebGL 用の JavaScript や月面のテクスチャは WebGL 用のチュートリアルサイト³⁾において提供されているものを使用した。

本研究のユーザインターフェースでは WebGL



図 2 Android タブレット端末 ZiiO 10

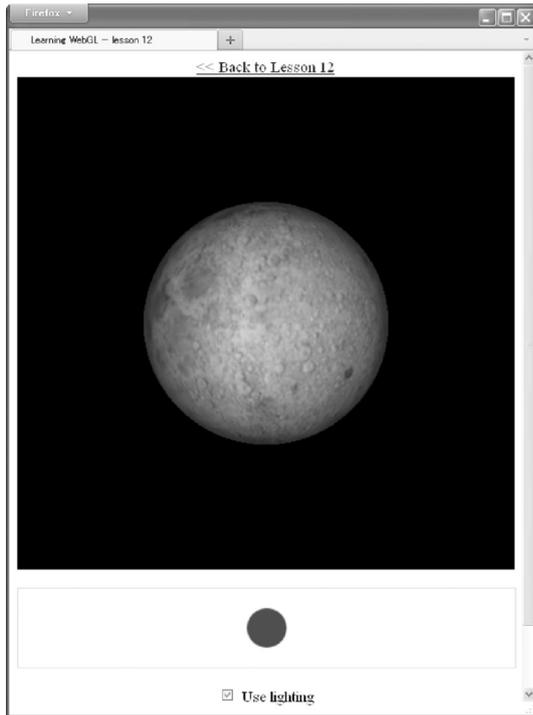


図3 試作ユーザインターフェースの表示

を表示する canvas の下に全く同じ大きさのコントローラとなる canvas が重なっている配置となっている。画面上をタップしてオブジェクトを回転させる際には、画面上の水平軸と垂直軸の両方で同時に回転角を指定させて回転させる必要がある。また回転量はタップ/クリックした点と中心点との差から求める。このために canvas 上のタップポイントの座標値から三平方の定理を利用して水平軸及び垂直軸上での回転角及び回転量を求めている。

WebGL の canvas ではタップ/クリック操作による x 軸及び y 軸の両方での回転，HTML 5 のアニメーション部分ではタップ/クリック操作

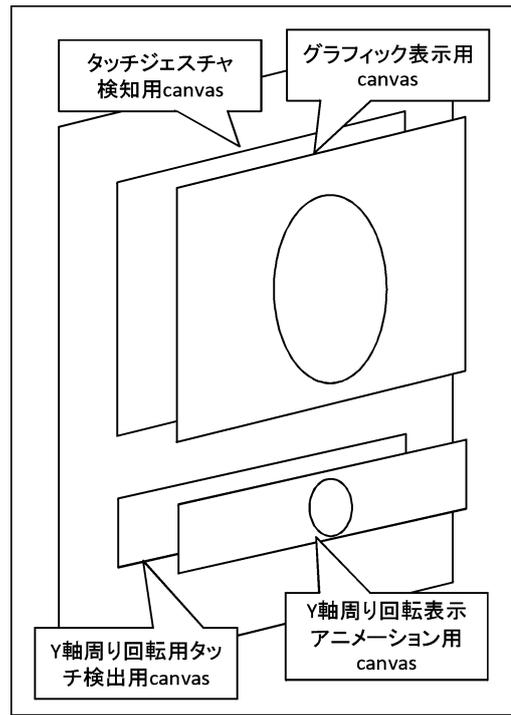


図4 試作ユーザインターフェースの構造

によって月面のモデルを Y 軸周りに回転をさせることができる。図3の画面下側にある赤い丸がHTML 5で作成したアニメーションである。こうすることで、画像を直接タッチして操作する方法と、オブジェクト操作のグラフィックユーザインターフェースを用いる形式の両方を用意できる。

図5及び図6にPCブラウザとTablet上のブラウザで同じWebページを表示している状況を示す。画面に表示された月面オブジェクトの周囲の同じ部分をそれぞれマウスクリック/タッチすることで同じように月面の画像がx軸及びy軸周りに回転することを確認した。

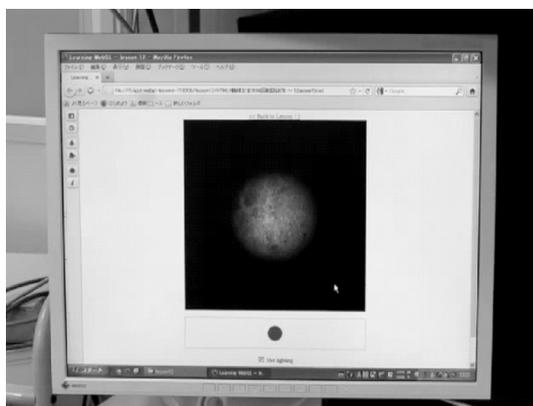


図5 PCブラウザでの試作ユーザインターフェースの操作



図6 タブレット上での試作ユーザインターフェースの操作

また下の赤い丸の左右をタップすることで赤丸が左右に移動し、これと同期して月面の画像も回転することを確認した。

5. 結論

本報告では以下の報告を行った。

1. タブレット端末でWebGLによる3Dオブジェクトを表示する際には従来のPC上でマウスを用いた操作とは結果が異なる。タップ操作は両方で共通に用いることができるが、スワイプ操作ではタブレット上のブラウザが移動してしまい、PC及びタブレットの両方でユーザインタフェースを共通にすることが出来ない問題点を指摘した。
2. 指摘した問題点に対して、タブレット端末とPCの両方で共通に用いることを目的として、PC/タブレットの両方で同じく用いることができるタップ操作を用いたユーザインタフェースの方式を提案した。
3. 提案した方式に基づいてウェブページに組み

込むユーザインタフェースの開発を行い、PCとタブレット端末の両方で同じ動作をすることを確認し、本論文での方式が有効であることが判明した。

WebGLには、セキュリティ上の問題が指摘され、本格的な運用には至っていない状況であるが、徐々に普及が進むと考えられる。JavaScriptは汎用的な言語であるので、機器やシステム構成に起因する問題を解消するために本研究で用いたユーザインタフェース用の部品をプログラムとして提供する方式を今後も進めたいと考えている。

【参考文献】

- 1) 松田晃一：WebGL+HTML5 3D CG プログラミング入門，CUTT 出版，p.1，2012.
- 2) 永野 直，林 秀彦：マルチタッチインタフェースの認知モデルと教育利用，鳴門教育大学情報教育ジャーナル No.6 pp.9-13，2009.
- 3) Learning WebGL，“<http://learningwebgl.com>”
- 4) 西畑一馬：Web制作の現場で使う jQuery デザイン入門，アスキー・メディアワークス，p.1，2010.