

# 3 DCGによる建造物を表わすソフトウェアの作成

Production of a Computer Software Showing the Structure of Building by 3DCG

田 縁 正 治

3 DCGと呼ばれる3次元コンピュータグラフィックスを利用して建造物を表わすソフトウェアの開発を行った。開発の対象は大学入学のためのセンター試験の試験会場をコンピュータ上でバーチャルに表現し、下見に来た受験生に提供するソフトウェアとした。これまでに開発したソフトウェアを基に受験生の席が規則的に並んでいない場合でも対応できるように改造した。また、廊下を移動する際によりリアルな感覚を与えられるように窓の外の景色を作成した。これはかなりの効果があり、好評であった。今後のソフトウェア作成の参考となると考えられる。今後の開発環境についても議論された。

キーワード：3次元コンピュータグラフィックス、ソフトウェア、C++、WindowsXP、バーチャル

## 目 次

- I 序 論
  - II 開発環境
  - III 開発の目的
  - IV 開発
    - 1 受験席の表示
    - 2 窓の作成
    - 3 窓の外の景色の作成
  - V 今後の開発環境
    - 1 ハードウェアとソフトウェア
    - 2 開発者の養成
  - VI 開発結果
-

## I 序論

コンピュータで立体的な物体を表現する方法は3Dコンピュータグラフィックスと呼ばれる。これは3次元で表わされる物体を2次元であるコンピュータのディスプレイにリアルに表現するための技術である。この技術で行う処理を簡単にまとめると、次の3段階に分けることができる。まず、現実の世界を構成する建物などの物体を測量することにより幅や高さといった数値データを入手する。これらのデータをコンピュータに入力する。次に、コンピュータのメモリの中に現実に存在する物体のデータを使用して仮想的な物体を作成する。物体と物体との距離や角度といったデータを利用して物体間の位置関係も正確に再現する。最後に、このようにして作成された3次元世界にカメラを持ち込む。このカメラに写る映像は2次元となる。このカメラに写る映像をディスプレイに表示する。

ディスプレイに表示する映像は単に3次元を2次元で表現するだけでなく、動きをつけて映像を変化させることで見る人によりリアルな感じを与えることができる。映像の変化は2種類の方法で行う。第1は、カメラの位置や向きを変えて映像を変える。第2はコンピュータの中に作られた3次元世界を表現するデータを変更することにより映像を変える。

このような3D技術を利用することにより建造物をリアルに再現することはさまざまな利点がある。今回はセンター試験を受験するために宮崎公立大学を訪れた受験生に受験会場までの順路をコンピュータ上でバーチャルに示すことを考えた。この試行はこれまでも行ってきており<sup>1,2</sup>、一定の成果を挙げてきた。今回、改めて開発を行うこととした理由はいくつかあるがそのうちの主なものは次の2点である。第1は、平成18年1月のセンター試験で初めて受験番号を机の順番に配置できない事情が生じたことによる。第2は、これまでより一層リアルな3Dとすることである。そのためにこれまで、建造物の中では建造物の外の世界とは切り離していたが、今回は建物の外を含めて3Dグラフィックスを作成するという新しい試みを行うことである。このような理由によりこれまでのソフトウェアを基にして新しいバージョンのソフトウェア開発を行うこととした。

## II 開発環境

開発環境はOSとしてWindowsXPを、グラフィックチップの制御のためにDirectX8.0を利用した。DirectXの最新版はDirectX9.0cである。しかし、これは利用しなかった。その理由は開発するコンピュータだけで動作するソフトウェアではなく、他のコンピュータでも動作するソフトウェアとしたかったからである。とはいえ、DirectXのバージョンが8.0より低いと機能が低過ぎることになるのでこれは止めることとした。

言語はC++を使用することとし、MicroSoft社のVisualC++6.0<sup>3~5</sup>を利用することとした。この理由はDirectXのバージョンと同じ理由で、最新のDotNetFrameworkを利用しないこととした。

### III 開発の目的

宮崎公立大学を3Dで表現することに関してはこれまでもソフトウェア開発を行い利用してきた。今回はこれまでに開発したソフトウェアをベースにしてこれまで以上にリアルに表現するための方法を模索することとした。今回の目標は次の通りである。

- A. 特別措置のための受験する席の変更
- B. 窓の追加と窓から見える外の景色を工夫する
- C. トイレなどの場所を特定する表示を増加する
- D. 場所を特定する表示がある場合でも図の場合は写真に変更する
- E. 同じ色の壁が隣接しているが向きが異なる場合に識別しやすくする
- F. その他の改良

### IV 開発

#### 1 受験席の表示

これまでは受験のための席は受験番号順に指定されており、ソフトウェア作成では繰り返しを使うことで単純に表現することができた。しかし、今回は状況が異なった。体調不良や精神的な不安から受験する席を他の人から離して欲しいという要求があったからである。体調不良はトイレが近いという理由であり、精神的な不安は腹の中にガスがたまりごろごろと音がする可能性があり、不安であるという理由である。今後、受験環境の多様化が求められる兆しを感じた。また、センター試験の試験監督を行っている際に気づいたのだが、机のボルトが緩んでいてぎしぎしと鳴る場合があった。他に机にでこぼこがあったりして受験用には不向きの場合も考えられる。受験の席が受験番号順になっていない場合でもおおむね順番になっていて一部の受験生の席が番号順になっていないと考えられる。しかも、順番から外れる受験生の受験番号は毎年変化する。このような場合の措置として、次の方法が考えられる。

- A. 直接必要な場所に繰り返しと条件判断を組み合わせて記述する。
- B. メソッドを用意しそこで繰り返しと条件判断を組み合わせて記述する。
- C. 受験番号と机の対応関係を表したファイルを用意し、それを読み込むことで対応する。

今回は人数が少なかったのでAの方法で対処することを試みた。ソフトウェアを実際に作成してみると、2、3名の席が規則的でない場合はこの方法で十分に対応できることが分かった。しかし、人数が増加するとこの方法では対処しにくくなる可能性がある。今後の課題として、BやCを採用する必要がある。受験する席の順番を変更するように要望する受験生の人数が現在より増加するようになればBやCの方法を検討することが必要になる。

## 2 窓の作成

今回の最大の特徴は、廊下に窓を設け窓の外に景色を配置したことである。前回は北側の廊下の壁は単純に一枚の四角形のオブジェクトを配置しただけだった。この部分はリアルさに欠けるという批判があった。そこで、窓と外の景色を追加してリアルにすることとした。初め、窓を単純に作成した。つまり、窓の上下左右の壁および窓のサッシをそれぞれオブジェクトとして作成して窓を作成した。この作業をそれぞれの窓について行うとコードが長くなり複雑化してしまった。そこで、この方法を止めて別な方法を模索することにした。目的は外の景色を見せることなので、壁にテクスチャを貼りガラスの部分は透明色に指定することとした。このために窓の写真撮影し、ガラスの部分をその写真では使われていないある色で塗りつぶすように画像を編集した。そしてその色を透明色と指定した。透明色指定はBMPやJPEGといったファイル形式では扱えないので、GIFとした。

次に使用したコードを示す。クラスのメンバとしてBUILDING型の配列

```
BUILDING      m_Kabe[40];
```

を用意した。BUILDING型は

```
struct BUILDING
{
    BUILDVERTEX v[4];
};
```

と定義された構造体とした。大きさが4の配列であるが、これは一つの四角形を表わすには4つの頂点が必要なので、4次元の配列を頂点データの格納用に用意した。この型を利用することで、立方体の建築物ならば、そのひとつの壁を表現することができる。テクスチャーを用いることで窓のある壁も表現することができる。立方体でない場合も四角形の壁を組み合わせることで表現できることが多い。この型の中のBUILDVERTEXは

```
struct BUILDVERTEX
{
    D3DXVECTOR3 p;      // Vertex position
    DWORD       color; // Vertex color
    FLOAT       tu, tv; // Vertex texture coordinates
};
```

### 3 DCGによる建造物を表わすソフトウェアの作成 (田縁正治)

と定義された構造体とした。この構造体はひとつの頂点を表わすことができ、頂点として3次元座標、色、テクスチャー座標を表現できるようにした。

実際に壁を作成する際は上記の構造体にデータを代入しなければならない。単純に代入すると、ひとつの頂点だけでBUILDVERTEXの中の D3DXVECTOR3で3個、その他に色とテクスチャー座標2個の合わせて6個のデータを代入しなければならない。この処理を4つの頂点で行うことになる。つまり、4角形の壁1枚につき24個のデータ入力が必要になる。これを壁の枚数分をコードとして記述すると、見づらいコードとなるので、別にデータ代入用のメンバ関数を用意した。この関数は地面に垂直に立つ壁のデータを入力するためのものである。

```
void CMyD3DApplication : MakeBuilding (BUILDING& m_Building, float x1, float x2,
float y1, float y2, float z1, float z2, float tustart, float tuend, float tvstart, float tvend)
{
    m_Building.v[0].p=D3DXVECTOR3(x1, y1, z1);
    m_Building.v[1].p=D3DXVECTOR3(x2, y1, z2);
    m_Building.v[2].p=D3DXVECTOR3(x2, y2, z2);
    m_Building.v[3].p=D3DXVECTOR3(x1, y2, z1);
    m_Building.v[0].tu=tustart;
    m_Building.v[0].tv=tvstart;
    m_Building.v[1].tu=tuend;
    m_Building.v[1].tv=tvstart;
    m_Building.v[2].tu=tuend;
    m_Building.v[2].tv=tvend;
    m_Building.v[3].tu=tustart;
    m_Building.v[3].tv=tvend;
}
```

ここで、CMyD3DApplicationはクラスの名前で、MakeBuildingはそのクラスのメンバである。第1引数はデータが代入されるBUILDING型の構造体である。値を変更できるように&をつけて参照型としている。第2引数から第7引数までは4角形の左上と右下の座標を指定する。第8引数から第11引数まではテクスチャー座標を指定する。

窓の作成ではこれまでに説明したことの他に柱を作成することでよりリアルにすることとした。通常の四画柱の柱は廊下から見るとその3つの面が見える。これらの面の作成は、上記の壁を3面作成することで実現した。

### 3 窓の外の景色の作成

廊下から窓外の景色が見えるように改良するときに検討したことは、大学の建物と同様の方法で外の景色を作成するべきかどうかであった。検討した結果次のような理由から窓の外の景色は写真データのみで作成することが望ましいと考えられた。

- A. 民家などのデータは測定不可能である。
- B. バーチャルな世界で、建物の窓の外に見える景色は遠くから見るだけでその中を移動することはない。
- C. ソフトウェアを複雑にしすぎるとパフォーマンスが落ちる。

ソフトウェアが複雑になり過ぎないようにするために外の景色をまともに3Dで作成するのではなく、簡便な方法を利用することとした。写真は2枚使用し、遠景と近景を表現することとした。具体的には大学の敷地内の建物を写した写真と敷地外の景色を写した写真を用意した。敷地内の建物は廊下の移動距離が少なくても見える位置が大きく変化するのに対し、大学の外の景色は距離が遠いので、廊下の移動距離が大きくてもあまり変化しない。別な表現をすると、大学内の建物は距離が近いので、廊下をまっすぐに進むとすぐに視界の外に出るのに対し、大学の外の景色は大きく変化しない。実際にはこの変化にさらに窓とカメラの位置関係があるので、さらに複雑になる。これらの複雑な要素はすべて窓の作成と大学外の景色を2種類に分けることで解決を図った。大学の敷地内の建物と大学の外の景色はカメラが移動することでその位置関係が変化する。この変化を分かりやすく表現するために大学内の建物の写真データは建物の端で写真をトリムすることで行った。

受験会場は1階から3階まで配置されており、この他に4階はトイレを利用可能としていた。北側の窓の外の景色は2階と3階と4階から見るようになる。このとき、2階から3階に移動したら景色が少し上から見たように変化する方がよりリアルと考えた。この感覚は次のような状況を作ることで実現可能と考えられる。窓から外を見ると窓の外の景色が窓の枠で切り取られて枠の中だけが見える。この切り取られた景色の中に民家や大学の建物があり、その他に空が見える。2階から窓の外を見ると、民家と大学の建物の割合が多く、2階から3階と上に移動するにしたがって、空の割合が増加するようにすると階を登っていった感覚がリアルになると考えた。実際に調査してみると、2階から4階とわずかな階の変化であるが上記の景色の変化が確認できた。この際に再検討する必要性が見出された。4階から窓の外を見ると空の面積が半分以上になり、単調な景色となることが分かった。単に事実としてこのままソフトウェア開発を続行しても良かったのだが、今回は写真の再検討を行った。真っ青な空が窓の外に見えていると、廊下を移動しても空の景色に変化がないので、バーチャルな世界の移動感覚が弱まるのである。そこで、雲の多い日に写真を撮ることとした。その雲も空一面に広がっているのではなく、青空が一部見えており、その中にぽっかりと雲が浮かんでいるような日の写真が適している。このような写真を利用することで4階からも変化のある景色を見ることができるようになった。

### 3 DCGによる建造物を表わすソフトウェアの作成 (田縁正治)

完成した窓と外の景色が見る人に与えるリアル感は、他の方法で作成したオブジェクトと比較することで確かめた。第1試験室と第2試験室は従来から窓を作成していた。しかし、この窓は実際には窓を写真撮影したデータを利用していた。つまり窓越しに見える外の景色を窓のフレームと共に撮影した写真データを利用していた。この写真をテクスチャとして利用していたが、この場合試験室の中を移動しても窓と外の景色の位置関係は変化しない。この方法でも窓の存在は認識できるので、全く窓がない状態よりははるかに分かりやすかった。利用者に感想を求めた結果、今回作成した窓の方がリアルであるとの結果を得た。しかし、従来からの方法でも大きな違和感がないことも分かった。

次に実際に利用したコードを示す。

```
LPDIRECT3DTEXTURE8    BackLandscape1;  
LPDIRECT3DTEXTURE8    BackLandscape2;
```

として、クラスのメンバとなるテクスチャー型変数を用意した。このテクスチャーは画像データとして用意しているファイルからデータを読み込み、テクスチャーを作成する。実際にファイルから読み込んでテクスチャーを作成する処理はDirectXが用意している関数D3DXCreateTextureFromFileExAを利用した。

```
D3DXCreateTextureFromFileExA (m_pd3dDevice, "..¥¥FigStage2¥¥窓 2 階外北.jpg",0,0,0,0,  
D3DFMT_A8R8G8B8, D3DPOOL_MANAGED, D3DX_FILTER_LINEAR, D3DX_FILTER_  
LINEAR, 0, NULL, NULL, &BackLandscape1);
```

ここで、第2引数はファイルを相対パスで指定している。最後の引数が上記の変数である。このテクスチャーを貼り付けるためのオブジェクトは前節で説明したBUILDING構造体を利用した。クラスのメンバとして次のように宣言した。

```
BUILDING    m_Landscape[20];
```

後の発展のために20の大きさの配列にしたが今回は2個利用している。このオブジェクトのデータも前節で説明した方法で指定している。

```
MakeBuilding(m_Landscape[0], 0.f, 40.f, 40.f, 0.f, 10.f, 10.f, 0.f, 1.f, 0.f, 1.f);  
MakeBuilding(m_Landscape[1], 50.f, 50.f, 40.f, 0.f, 30.f, 0.f, 0.f, 1.f, 0.f, 1.f);
```

実際の描画は次のコードで行っている。

```
m_pd3dDevice->SetTexture(0, BackLandscape1);
m_pd3dDevice->DrawPrimitiveUP(D3DPT_TRIANGLEFAN, 2, _Landscape[0].v, sizeof
(BUILDVERTEX));
m_pd3dDevice->SetTexture(0, BackLandscape2);
m_pd3dDevice->DrawPrimitiveUP(D3DPT_TRIANGLEFAN, 2, m_Landscape[1].v, size
of(BUILDVERTEX));
```

## V 今後の開発環境

### 1 ハードウェアとソフトウェア

Windowsの次期バージョンであるWindows Vistaでは、デスクトップのインターフェースが変更されることになっている<sup>6</sup>。これまではLunaと呼ばれるデザインが用いられていたが、これがWindows Aeroというデザインになる。この結果、ウインドウの検索時に3Dグラフィックでウインドウを一覧表示するフリップ3Dが利用されるなど、ユーザーインターフェースが大幅に変更される。このため、Windows Aeroを利用するためにハードウェアに求められる性能が変更される。これまでも提供されていたが、DirectX9を利用することとなる。また、ハードウェアとしてはWDDM(Windows Display Driver Model)をサポートしたグラフィックカードが必要となる。もう少し詳しく述べるとWindows Vistaが動作するという意味のWindows Vista Capable PCと呼ばれる仕様とWindows Vista が本来の性能を発揮して動作するWindows Vista Premium Ready という2つの仕様が公開されており次の表のようになっている。

表1 Windows Vista が求めるハードウェア性能

	Windows Vista Capable PC	Windows Vista Premium Ready
CPU	800 MHz	1 GHz
メモリ	512 MB	1 GB
グラフィック	DirectX9 対応	DirectX9 対応 (WDDM ドライバ、ピクセルシェーダ 2.0 対応)
ビデオメモリ	32 MB	128 MB
ハードディスク	20 GB (空容量 15 GB)	40 GB (空容量 15 GB)

明らかに要求されるハードウェア性能が現在のWindowsよりも向上している。このことは3Dを利用したソフトウェア開発という観点から見ると好ましい傾向にあるといえる。これまでは、最新の技術を利用して3Dソフトウェアを開発した場合に開発したコンピュータでは動作するこ



とは当然であるが、他のコンピュータにインストールしようとする、その動作が保障されなかった。このため最新の技術ではなく少し古い技術に限定して開発するようにしていた。具体的には、DirectXは最新版のバージョン9ではなくバージョン8を利用していた。また、OSのAPIではDot NetFrameWorkではなくWin32APIのみを利用していた。今後新しいOSが提供され、普及すれば、これまで以上に豊かな表現方法を利用することができ、機能の豊富なソフトウェア開発が可能になると考えられる。以上をまとめると、WindowsVistaになると3DがOSレベルで標準になり、ハードウェアとソフトウェアの両面で3Dが利用しやすい環境が入手できる予想される。

以上はハードウェアとOSを基準にしてみた開発予定だが、プログラミング環境からみた変化も予想される。これまでに3Dとして、VisualBasic<sup>7</sup>やJava<sup>8</sup>といった言語による開発を試みたことがあった。実際建造物を3Dで表現するために開発当初はVisual Basicを利用した。しかし、Visual Basicではハードウェアが持つ性能を十分に引き出すことができなかった。そのために多くの制限がある中での開発を行った結果、高い性能を持たないソフトウェアとなっていた。Javaによる開発も同様であった。JavaはWrite Once Run Anywhereという考え方があるようにプラットフォームを選ばないという利点がある。しかし、これは逆に完成したソフトウェアの性能を犠牲にしている。JavaVirtual Machineを通してOSに処理命令を出すことになり、そのOSがハードウェアに命令を出すので、間接的になり、とてもハードウェアの性能を引き出すことができない。その後開発のための言語をC++に移行した。このC++はコンピュータの性能を引き出す良い方法であり、市販されているソフトウェアや性能の良いフリーソフトウェアはC++で作成されることが多かった。このような事情から、C++は性能が高く、他の言語を利用する必要性はないように思われていた。しかし、新しくC#が登場し、最近ソフトウェア開発会社でもC#を利用することが増加している。JavaはC言語やC++といった言語を基にして開発されたので、これらの言語に良く似た仕様になっている。一方、ガーベージコレクタや多重継承の禁止など基にした言語とは異なった仕様を採用されている。これらの違いは容易でバグの少ないソフトウェア開発という点で大きな効果があった。しかし、3Dソフトウェア開発という点に関しては前述したようにJavaは不向きであった。このような理由からC#が登場した。C#の特徴はC++の欠点を改良し、Javaなどの他の言語の良いところを取り入れた言語仕様になっており開発されたソフトウェアの性能を落とさない工夫を行いながら開発者の負担を軽減するようになっている。今後はC#とDotNetFrameWork<sup>9</sup>を利用する方法を検討する必要がある。

## 2 開発者の養成

コンピュータソフトウェアの開発では、専門の訓練を受けたプログラマーが必要である。特に3Dグラフィックスを利用している場合は他のソフトウェア開発の場合と違って、三角関数やベクトルといった解析・代数の双方の数学的な素養が必要なこと、写真データを多用すること、3次元的な感覚が必要なことなど、特殊な状況にある。今後は開発者の育成が必要であろう。現在、

社会的に強く求められている人材の一つにソフトウェア開発者が挙げられている。平成18年6月に、宮崎県立北高校で高校生を対象に出前講座と呼ばれる授業を行った。その際に3Dに関するテーマを扱ったが、土曜日であるにも関わらず多くの高校生が参加した。当初予定していた人数より多くなり、会場に入りきれないくらいの参加者がいたことは3Dに関する高校生の関心が高いと感じさせてくれた。講座が終わった後でアンケート調査を行ったがそこでも好評であることが確認できた。このアンケートの中で分かったことであるが、高校生は3Dに関する知識はほとんど皆無ということである。3Dで物体を作成するには三角形を組み合わせるといった基本的な話に興味を持っていた。これからは、高校生などにもっとソフトウェア作成の情報を伝えることで人材育成に役立つような取り組みが必要であろう。

宮崎公立大学でも一部の学生には、卒業論文の作成のためにソフトウェア作成を要求している。過去の例では3Dによる卒業論文作成を選んだ学生は5名いた。これまでの学生は3Dを利用した場合は他の学生よりもより努力しないと開発ができないようで、かなりがんばって作成してきた。このうちの1人は現在4年生であるが、C++とDirectXという強力な組み合わせを利用しており、この組み合わせは初めての例である。ゼミ内で卒業論文の進捗状況を発表する機会を設けていると、3Dを利用していないゼミ生のソフトウェアと比較して見栄えが良いようで、たとえ3Dを利用することを選択しなかった学生でも3Dに関する関心は高いことが分かった。過去にJava 3Dを利用する試みを行ったが、スピードと学生の能力の両面から断念したこともあった。3Dでは、APIの読み方、数学の手ほどき、サンプルコードの読み方といったさまざまなサポートをしないと3Dプログラマーが育たない。卒業生の例では、グラフィックスを利用するためのソフトウェア作成に従事した人がいた。現在は2度転職して他の仕事についているが、今から考えるとグラフィックスの仕事が一番楽しかったそうだ。今後は単に3Dソフトウェアを開発するだけでなく開発者を養成することを同時に考えなければならないようだ。

## VI 開発結果

第Ⅲ章の「開発の目的」でAからGまでの7つの開発目的を挙げた。これらの開発結果をまとめてみた。まず、「A. 特別措置のための受験する席の変更」に関しては場合分けで実現した。「B. 窓の追加と窓から見える外の景色を工夫する。」に関してはかなりの成果が得られ、今後の開発の参考とすることができた。「C. トイレなどの場所を特定する表示を増加する」についてである。これに関しては第1試験室や第2試験室の近くのトイレを表示し分かりやすくした。次に「D. 場所を特定する表示がある場合でも図の場合は写真に変更する。」については階段の後ろの壁の階数表示を写真に変更したり、第3から第6試験室のドアを写真に変更した。「E. 同じ色の壁が隣接しているが向きが異なる場合に識別しやすくする」については壁ごとに異なったテクスチャーを貼ることで実現した。「F. その他の改良」に関してはバーチャルな世界で移動する

### 3 DCGによる建造物を表わすソフトウェアの作成 (田縁正治)

際に、回転する速度を0.05から0.1にスピードアップした。以上のような作業を行った結果、より改良されたソフトウェアが開発された。また、よりリアルにするために、窓の作成が有効であることも分かった。

これらの変更を施したソフトウェアを平成18年1月20日(金)に利用に供した。受験生の感想を求めた結果好評であったことが分かった。また、2局のテレビ局や3誌の新聞取材があった。

### 謝 辞

本研究において、宮崎学術振興財団より一部資金援助をいただいた。ここに謝意を表する。

### 参考文献

1. 田縁正治、2001年、「3 DCGによる大学案内ソフトウェアの作成」、宮崎公立大学人文学部紀要、第9巻、第1号、43-58ページ
2. 田縁正治、2002年、「建造物の3次元コンピュータグラフィックス」、宮崎公立大学人文学部紀要、第10巻、第1号、113-128ページ
3. 柴田望洋、1998年、「明快C言語入門偏」、SOFTBANK
4. 林晴比古、1999年、「新Visual C++6.0入門」、SOFTBANK
5. 瀬戸 遥、2000年、「Visual C++6.0入門教室」、翔泳社
6. PCJapan Vol 11 No 8 p95 2006
7. 林晴比古、1997年、「Visual Basic 5.0入門教室」、翔泳社
8. 柏原正三、2002年、「はじめてのJava完全入門」、技術評論社
9. 醍醐竜一、斉藤友男、2005年、「パーフェクトC#」、技術評論社