

3DCGによる大学案内ソフトウェアの作成

Production of a computer software showing the structure of a university building by 3DCG

田 縁 正 治

3次元コンピュータグラフィックスをパーソナルコンピュータで作成するための環境をハードウェアおよびソフトウェア両面から検討した。ハードウェア面では、必要な機能を持った製品が入手可能であり、それを利用するソフトウェアの作成手法が重要であることが分かった。ソフトウェア作成においては、言語はVisual Basicを利用し、更にハードウェアの性能を引き出すためにMicrosoft社製のDirect Xを利用することが簡単にソフトウェアを作成するための良い方法であることが分かった。以上の手法で複雑な建物内の構造を、3次元コンピュータグラフィックスを用いて案内するソフトウェアを作成した。このソフトウェアをテストした結果、さまざまな改良点が明らかになったのでソフトウェアを変更し、宮崎公立大学を案内するソフトウェアを作成し利用した結果、分かりやすいという評価が得られ3次元コンピュータグラフィックスの利用価値を示すことができた。

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、仮想空間、大学案内、センター試験

目 次

I 序 論	IV ソフトウェアの改善
II 利用したハードウェアとソフトウェア	1 ソフトウェアの操作方法を改善するための調査
1 リアルタイム3DCGを作成するためのハードウェア	2 目的地の位置を示すガイド
2 Direct X	3 試験室の作成
3 プログラミング言語	4 操作方法の改善
III ソフトウェアの作成	5 表示の改善
1 建物のデータ収集	6 現実と3DCGが大きく異なる場合
2 ウィンドウの構成	7 使用した際の評価
3 3次元空間とカメラ	
4 座標系と動き	
5 操作方法	

I 序 論

パーソナルコンピュータの性能はCPUのクロック、ハードディスクの容量、CD-ROMのスピード、インターフェースなどさまざまな面で向上し、もはやビジネスで利用するだけなら十分な性能に達して数年が経過したように思える。つまり、文字、写真、イラスト、グラフを含んだ文書の処理においてはこれ以上の性能は必要としない。これに対応してさまざまなソフトウェアが作成され利用されている。

これに反して、3DCGは全く状況が異なる。3DCGとは3次元コンピュータグラフィックスのことであるので、立体的な表現が可能なグラフィックスということになる。現在普通に入手可能なパーソナルコンピュータでは、ディスプレイは平面的であり、真の意味での立体的な表現は不可能である。そこで、コンピュータグラフィックスでは、さまざまな方法を用いて、ディスプレイ上に描く図形が立体的に見えるよう表現する。たとえば、遠くにある物体は手前にある物体より小さく描くいわゆる遠近法を利用する。また、ひとつの物体のある面は明るい色を使用し別の面では同系色の暗い色を使用して光源に対する向きを表す。円柱の場合は、側面をグラデーションを用いて表現する。もうひとつの方法は、前方にある物体が後方にある物体を隠して前後関係あるいは奥行きを表現することである。したがって、3DCGでは通常のビジネス用ソフトウェアと比較して処理内容が異なるので、それに適したプログラミング技術を必要とする。また、処理する情報量が非常に多いことから通常のソフトウェア作成技術だけでは困難である。単純計算を行うと、平面的な写真に奥行きを追加することになるので、情報量は写真の場合の情報量より更に1000倍程度増えることになる。

ここでは動きのある3DCGを作成することを目的としている。映画やテレビジョンでは、動きはコマと呼ばれるある一瞬の映像をいくつも用意してそれを連続的に交換して見せることにより表現する。コンピュータグラフィックスでも同じ方法で動きを表現する。ただし、作成する方法は、大きくわけて2通りある。ひとつは、コマに相当するグラフィックスをあらかじめ作成しておき、それを適当な方法でつなぎ合わせて動きを表現する方法である。この方法はコンピュータ上でビデオ映像を再生する方法と同じであり、この処理を行うソフトウェアは、CODECと呼ばれ、ファイル形式もAVI、モーションJPEG、MPEGなどさまざまな規格が利用されている。もうひとつの方法は、リアルタイムに、つまり必要になった時点でコマに相当するグラフィックスを作成する。この方法では、ひとつのコマから次のコマに移る短い時間の間に3DCGを作成しなければならないので、最も厳しい条件が課せられる。しかし、この方法を用いなければインタラクティブなソフトウェア、つまり操作した時点でリアルタイムに変化する3DCGを作成することはできない。ここではこの最も厳しい条件が課せられる3DCGを作成することとする。

現実世界は3次元であるので、3DCGを必要とするソフトウェアが多いと思われる。しかし、

3DCGの作成が容易でないことにより、これまでグラフィックスを効果的に利用したソフトウェアは、文字主体のソフトウェアより大幅に少なかった。実際、市販の入門を前面に出したプログラミングの書物では3DCGは全く扱われていない¹⁶⁾。応用編においてもビジネスに適した機能については記述があるが、リアルタイム3DCGを作成する手助けとなる機能に関してはあまり記述がない⁷⁸⁾。このため、グラフィックス処理を多用したソフトウェアの開発を推進するために、パーソナルコンピュータでのグラフィックス作成に必要なハードウェアの現状調査、新たなソフトウェアの開発・試用を行い、ソフトウェアの作成手順や効用や限界について公表することとした。その際、なるべく簡単にリアルタイム3DCGを作成することを目標とし、多くのソフトウェア作成者の参考になるよう努めることとした。

宮崎公立大学の建物を3DCGで作成し、平成13年1月のセンター試験を利用してリアルタイム3DCGの利用価値を調査した。宮崎公立大学は、センター試験の会場として約500名の受験机を用意し、試験室は6箇所とした。試験会場の管理上の問題から受験生は試験前日に建物の中に入ることを禁じられていた。したがって、受験会場を下見する方法を、試験前日に受験生に提供することは意味のあることである。一人ひとりの受験生に応じて受験会場のビデオ映像を前もって用意するという方法は採用しなかった。500名分のビデオ映像を用意することは現実的でないことと、あらかじめ撮影されるビデオ映像が受験生の興味を十分に反映したものとする保障がなかったからである。これに対し、リアルタイム3DCGならば、受験生が自分で操作するので、受験生の興味に応じてコンピュータのディスプレイに映像が表示されることになる。試験前日に試験会場の下見に訪れた受験生に、コンピュータの中の仮想空間に作成された試験会場を移動できるソフトウェアを提供した。受験生は自分の受験番号をコンピュータに入力すると、ディスプレイに仮想空間が表示されるので、試験会場まで案内するガイド表示にしたがってコンピュータを操作し試験会場の下見を行うことができた。リアルタイムに変化する3DCGにより受験生は自分が使用する机まで到達でき、その途中の階段の様子やトイレの位置を確認できて安心感が得られたようだった。

II 利用したハードウェアとソフトウェア

1 リアルタイム3DCGを作成するためのハードウェア

動きのある3DCGを表示することは従来から行われていた。その方法はコマに相当する図を多数用意しておき、その図を次々に表示することであった。つまり、ビデオ映像をあらかじめ作成しておき、それを表示することで動きのある3DCGを表示することと似た手法であった。したがって、多くの図を用意しておく必要があり、その図を次々に交換して表示するための処理が多く必要であった。この方法では、インタラクティブに動きを変更することはできないという致命的な欠点があった。ここでは、人間の操作に応じてリアルタイムに変化するようつまりイン

タラクティブな3DCGを作成することを目指しているのでこの方法では不可能である。

もうひとつの3DCGの作成方法として多数の三角形を組み合わせて立体的な物体を仮想的な3D空間に作成し、その投影図をディスプレイ上に表示する方法がある。この方法では、現実存在する3次元物体を人の目で見ることとほぼ同じ過程を経て作成されるので、インタラクティブな3DCGを作成することが可能になる。この手法をより詳しく述べると、次の4段階で3DCGを作成することになる。第1に、メモリ上に仮想的な3D空間を用意しその中に三角形を組み合わせて立体的な物体を作成・変形・移動する。この操作はTransformationと呼ばれる。三角形を組み合わせて作成された図形はポリゴンと呼ばれる。第2に、作成された立体的な物体に光をあて個々の三角形の明るさを決定する。これはLightingと呼ばれる。第3に、仮想的な3D空間はコンピュータのディスプレイに表示するために2Dに変換されなければならない。すなわち、ある一方から3D空間にある物体をカメラで写した時のグラフィックスに変換することが必要である。これはラスライズと呼ばれる。最後に、各ポリゴンは単色で塗ったのではリアルではないので、写真や図を貼り付けることでリアルにすることができる。この操作はテクスチャマッピングと呼ばれる。第1と第2の段階をまとめてジオメトリ演算、第3と第4の段階をまとめてレンダリングと呼ばれる⁹。

現在入手可能なパーソナルコンピュータ用の部品として3DCGを容易に描画するためのグラフィックスカードがある。グラフィックスカードを利用する利点は、CPUに全ての処理を集中させないということと、グラフィックスという特殊な処理はその目的に特化して開発されたICで処理することで処理速度を上げることである。グラフィックスカードはマザーボードのAGP expansion slotに挿入することで利用可能となる。グラフィックスカードの中にはグラフィックス処理を行うICであるグラフィックチップやディスプレイに表示するデータを記憶するための記憶装置としてVRAMを組み込んである。ディスプレイに文字やグラフィックスを表示するにはたとえその情報をCPUが処理しても最終的にCPUからVRAMへ転送しなければならないので、VRAMに近いところに配置されたグラフィックチップに大部分の処理を行わせることでデータ転送量を減少させることができる。現在調査した結果では、グラフィックカードの中には上記のレンダリングを容易に行うだけのものもあれば、ジオメトリ演算まで行うものも存在する。後者はハードウェアT&Lを行うと表示されているので、容易に区別できるはずである。TとLはTransformationとLightingの頭文字である。ここではハードウェアT&Lに対応したグラフィックチップとしてnVIDIA製のGeForce3を搭載したグラフィックカードを利用した。VRAMは64MB搭載されており、本稿での目的では十分な容量といえる。

2 Direct X

3DCGを利用したソフトウェア開発は通常のビジネス用ソフトウェアの場合とは大きく異なる。プログラマ自身が3次元空間の把握をしなければならないし、その処理に必要なプログラミ

3 DCGによる大学案内ソフトウェアの作成 (田 縁 正 治)

ングは前節で述べたように処理の手順が複雑で、その結果処理量が多くなるので効率的なプログラミングが求められる。しかし、より重要なことは、3 DCGを表示するためのハードウェアにある。グラフィックス表示にはグラフィックスカードが利用される。これらのハードウェアは発展途上にあり、さまざまなメーカーが開発・発表している。すなわち、ハードウェアメーカーはより効率が良く、より安価なハードウェアを作成しようとし、他のメーカーと差別化したハードウェアを市場に投入することがある。その結果、ハードウェアに対する命令は共通の命令ではなくハードウェアごとに異なってものとなる。通常ハードウェアにはデバイスドライバと呼ばれるソフトウェアが付属し、利用者の便宜を図っている。つまり、ハードウェアに対する命令が異なることによる個性はデバイスドライバが吸収する仕組みをつくっている。しかし、デバイスドライバが担当する部分はハードウェアに密着した部分であり、デバイスドライバのみを利用してプログラミングを行おうとすると、さまざまな低レベルの処理までプログラマが意識しなければならない。通常、職業としてプログラミングを行うプログラマは、自分が所有するコンピュータだけで動作するソフトウェアを作成するのではなく、できるだけ多くのコンピュータで利用可能なソフトウェアを作成したいと考えている。そこでハードウェアメーカーの違いによる命令体系の違いをプログラムに反映させなければならない。すると、ハードウェアの命令体系がメーカーにより異なることをプログラマが調査し、作成するソフトウェアの中でメーカーによる違いを場合分けして吸収しなければならないことである。しかし、これは大きな組織に属さないプログラマにとっては不可能に近いし、無駄が多い。したがって、ハードウェアの違いを吸収するための仕組みをOSあるいはOSのメーカーが提供するソフトに期待することになる。

パーソナルコンピュータで利用可能なOSとしてここではMicrosoft社が開発したWindowsを選択した。Windowsでは、さまざまな処理をOSが行っているがその処理はひとつのファイルの中に記述されているわけではない。必要になった時点で必要になったファイルをメモリに読みこんで利用するように分割されている。このように分割することによりOSの立ち上げ時間が短縮されることと、不必要なファイルをメモリに読みこむことによるメモリの無駄づかいを防ぐことができる。必要になった時点で読みこむように設計された機能はダイナミックリンクライブラリまたは略してDLLと呼ばれる。ライブラリを格納したファイルの拡張子がDLLとなっているので分かりやすい。また、OSが利用するこれらの機能はOS専用の機能ではなくC++などの言語でソフトウェアを開発する際にも利用可能である。ウインドウを開ける機能や、そのウインドウのタイトル部分に文字を表示する機能はOSが必要とする機能であるが、ソフトウェア開発者も必要とする機能だからである。これらはApplication Program InterfaceまたはAPIと呼ばれ、ソフトウェア作成で通常利用されるC++のような言語から簡単に呼び出すことができるよう呼び出しの際の引数の種類・個数・順序および戻り値が公開されている。

グラフィックス処理に関しては、WindowsにはグラフィックスデバイスインターフェースすなわちGDIと呼ばれるAPIが存在している。これはウインドウの中に文字や図形を描くためのA

PIで、少数の2次元グラフィックスを描画するには有効であった。しかし、高速に2次元画像を交換しながら表示したり、3DCGを作成するには十分な機能を持っていなかった。そこで、その処理をコンピュータ上で容易に実現するための機能として、WindowsではDirectXを用意している¹⁰。このDirectXを利用することにより、高度な描画機能や、複数の音を同時に利用することが可能となった。ここでは、1999年に発表されたDirectX7を利用することにする。DirectX7は、DirectDraw、Direct3D、DirectSound、DirectMusic、DirectInput、DirectPlay、DirectShow、DirectAnimationの9つのコンポーネントからなり、それぞれ2次元描画、3次元描画、WAVE関係、MIDI関係、入力関係、ネットワーク関係、セットアップ関係、動画などのストリーミング再生、総合アニメーション環境の機能を提供する。これらの機能はDLLファイルに格納されたAPI関数を呼ぶことによって利用できる。従って、最初にDirectXをインストールする必要がある。このインストーラはMicrosoft社のサイトから入手可能である。また、市販の書籍に付属するCD-ROMに入れられていることもある。このDirectXは現在、Version 8まで発表されている。一般にVersion番号が付されている場合は新しいVersionのものは古いVersionの機能を含んでいる。つまり、上位互換性があることが多い。このDirectXの場合も概ね上位互換性を持っているので、最新VersionであるDirectX8をインストールすれば良いように思われる。しかし、ソフトの内容は非常に複雑であるから、完全な互換性が保たれていることは期待しない方が安全である。

3 プログラミング言語

パーソナルコンピュータで利用可能な言語は、C++、Basic、Pascalなどさまざまである。ここでは、なるべく簡単で多くの人が利用可能なBasicを選択した。具体的には、Microsoft社が発売しているVisual BasicのVersion6.0を利用した。この言語は、ウインドウのデザインをビジュアルに設計できることや、イベントプロシージャの記述が容易であることなどから採用した。

Visual Basic からDirectXを利用する方法は、DirectXのVersion7から提供されるようになった。Version7以前は、比較的使用方法が複雑なCあるいはC++を利用する方法しか提供されていなかったため、今回の簡単にソフトを作成するという目的では、DirectX Version7以降を利用することが必要である。最新のVersion8を利用しなかつた理由のひとつはその普及度にある。あまりにも新しいVersionを利用すると作成したソフトウェアは限られたコンピュータでの動作のみ可能なソフトウェアとなってしまう。もうひとつの理由は、Version8は新しい機能を提供しており、現在のパーソナルコンピュータではハードウェアがその機能を実現できないことが多い。つまり、Version8では通常のパーソナルコンピュータが備えている機能をより簡単に効率的に引き出すことだけを目的にしているのではなく、将来のハードウェアに対するMicrosoft社の考えを提案しているように見える。現状で調査した結果、DirectXのVersion8の新しい機能を利用できる市販のグラフィックカードはほんのひとつかふたつである。この状況では、簡単で多

くのコンピュータで利用可能なソフトウェアの開発という本稿の目的には合わない。以上がDirectX、Version7を採用した理由である。なお、Microsoft社が提供するもうひとつのOSの流れの中でWindows NTでは、DirectXは全く利用できないが、Windows2000では利用可能である。

DirectXにはImmediate Mode あるいはIMモードと呼ばれる高度な命令群とRetained Mode あるいはRM Mode と呼ばれる簡単に3DCGを作成できるモードが用意されている。また、言語としてC++を利用する場合はMicrosoft社からのVisual C++ を選択することが可能である。したがって、DirectXを利用する場合は1. Visual Basic と RM mode、2. Visual Basic と IM Mode、3. Visual C++ と RM Mode、4. Visual C++ と IM Modeの4種類が考えられる。1が最も簡単で、4が最も困難な方法である。今回は1の最も簡単な方法を採用することとした。更にDirectXの命令を簡単に利用するために今回は川田氏が作成したDirectX利用支援クラスモジュールを利用した¹⁴⁾。

コードの記述に際してはなるべくDirectX以外の命令は通常のVisual Basic の命令のみで記述するよう心がけた。しかし、Visual Basicはイベントドリブン形の記述方法を採用しているのに対し、ここでの目的であるリアルタイム3DCGでは単にキーを押したときに行う処理以外に、キーを押し続けているときや複数のキーを押しつづけたときや、キーを押さないときに行う処理など、イベントドリブン形の記述になじまない処理がある。したがって、通常はVisual BasicからはDirectX以外のAPI関数は使用しなかったが、キー操作に関するGetAsyncKeyStateだけは使用した。このAPI関数を使用することによりキーを押していなくても必要な処理を続けキーを押した場合はリアルタイムで押されたキーに対応した処理へ移ることができた。

Ⅲ ソフトウェアの作成

1 建物のデータ収集

当然のことであるが、なるべくリアルなソフトウェアにするために実際の建物の長さを測ることとした。大学の建築物はその設計図が存在するので、最初は設計図を見たのだが、1つの建物の設計図が複数の紙にまたがっていたり、詳しくすぎたり、周りとの関連が複数の設計図を総合して考えなければ分からなかったりして、設計図を利用することはあまり現実的ではなかった。特に設計図の枚数が多いことが困難さを増していた。そこで、実際に建物の長さをメジャーで測定することとした。2階と3階はほとんど同じデータを用いるといった簡略化の手法を利用すれば、かなり手間が省けた。しかし、庭と建物の関係や円形の柱の位置・数などは実際に目で見てメジャーで測ることがソフトウェア作成者の頭の中で全体像を作ることに役立つように思えた。

2 ウィンドウの構成

ウィンドウは3つ表示することとした。3つのウィンドウは関連が深いので、最初に各ウイン

ドウの概略を説明しておき、後により詳しい説明を行うこととする。まず概略を述べると、第1のウィンドウは操作方法の説明や目的地の表示など文字主体のウィンドウとする。第2のウィンドウは、全体像を表示し、地図の役目を果たす。第3のウィンドウは最も重要なウィンドウで、ここでリアルタイムに変化する3DCGを表示する。このウィンドウを以後メインウィンドウと呼ぶことにする。次により詳しい説明を行う。

第1のウィンドウは、このソフトウェアの基本的な説明を行うことから始まる。最初このウィンドウのみが表示されることとし、ディスプレイの中央部に表示されることとした。このウィンドウの操作はENTERキーのみとした。ENTERキーを押していくことにより説明が進み、このソフトウェアの目的が把握できる。また、このウィンドウはディスプレイの左上端に移動し、ディスプレイを有効に利用することとした。説明が終了すると、このソフトウェアを操作して到達することを期待している目的地を示す。目的地は試験室、試験場本部、トイレ、控え室の4箇所とした。この程度の知識を受験生が前日に手に入れば、当日不安なく受験に専念できると考えたからである。試験室の番号は受験生に応じて変えなければならない。試験室の番号は受験生には分からないが、受験番号は分かっている。そこで、試験室をコンピュータが探すための材料として受験番号を入力することとした。この受験番号は、このソフトウェアを使用した受験生の記録を取るためにも利用した。また、目的地の表示は、初めはすべて緑色であるが、ある目的地に到達した後はその目的地を赤色表示とし、達成度を表示するために利用した。

第2のウィンドウは全体図を示すために使用することとした。これは第3のウィンドウであるメインウィンドウの助けをするために使用する。すなわち、大学全体が見渡せるような図とし、メインウィンドウのスタートの図と対応させることとした。したがって、このウィンドウに表示されるグラフィックスは始めから終わりまでほとんど変化しないことが望ましい。また、このウィンドウとメインウィンドウとの関連が分かりやすいことが望ましい。関連性に関しては、このウィンドウの最初の位置と表示されるグラフィックスをメインウィンドウの位置および最初に表示されるグラフィックスと同じにした。第2のウィンドウは表示された後に、最初の位置からディスプレイの中の左下端に移して邪魔にならないようにした。この移動と同時にウィンドウの中の表示を簡略化した図に変更した。3DCGに比べてリアルさに劣るが把握しやすいという利点を持つ。写真よりも簡略化した図の方が分かりやすいことがあるが、ここではその効果をねらっている。また、簡略化した図には説明となる文字を入れた。この図は始めから終わりまで変化しないという方針から、3DCGではなくBMPファイルで用意した。

第3のウィンドウはキー操作によって仮想的に建物の内外を移動するので、最も面積が大きくディスプレイの中央に位置することとした。ただし、通常のウィンドウと同じで、位置を自由に変更することができるし、大きさを変更することもできる。大きさを変更したときは、ウィンドウの中に表示される3DCGも変化する。

3 3次元空間とカメラ

建物を3次元的に表現するとしても、現在容易に入手可能なディスプレイは平面的である。すなわち、2次元上に3次元物体を表示することから、3次元的な表現は厳密には、不可能である。そこで、コンピュータグラフィックスでは、コンピュータのメモリ上に仮想的な3次元空間を作成し、その中に建物を作成することになる。そして、作成された建物をディスプレイに表示することになる。

ディスプレイに表示された2次元のグラフィックスが3次元的に見える理由は、人が建物を見て、それが3次元な物体と認識する方法を思い起こしてみることによって理解できる。人が認識する方法は、ある一瞬では目の前にある景色をほぼ2次元的に見ているので、コンピュータのディスプレイに表示されるグラフィックスと大差ない。厳密に言えば左右両目で見た像のわずかな違いを利用しているし、この効果は決して無視できないほど大きい。この点に関しては今回はこれ以上触れないこととし、将来の課題としておく。人が建物を見てそれが3次元物体であると認識するもうひとつの方法は、見る位置が移動したときの変化である。すなわち、見る位置が変わるとそれまで手前にある柱が隠していた後ろの壁が見えるようになるなど、3次元物体特有の変化を観察することによって、対象に奥行きという3次元的な特徴があることを知る。ここではキー操作によって視点が動くことに対応して、グラフィックスが変化することにより3次元的な特徴を出すこととする。

この動きは仮想的に作成した3次元空間の中をカメラが移動することで実現することとした。したがって、キー操作はカメラの位置を変えたり、カメラの向きを変えたり、カメラの高さを変えたりすることになる。カメラは建物の壁にはぶつかるが、ドアは突き抜けることができることとした。このウインドウに表示されるグラフィックスは、このウインドウを通して実際の景色を見たときの感覚に近づくように工夫した。具体的には、図1に示したように人の目は前方90度の角度の範囲を見ることとし、人はディスプレイの中だけではなく、その周りも見えると想定した。カメラが写してディスプレイに表示する像は、人と同じようにカメラの前方90度の範囲内とすると、カメラが写した像をディスプレイ一杯に表示するならば、カメラの位置は人より前になければならない。15インチ液晶ディスプレイでは、横幅はほぼ30cmであるから、カメラとディスプレイの距離は15cmとしなければならない。目からディスプレイまでの距離は45cmで使用すると想定した。ディスプレイより手前にある物体は描画しないこととした。

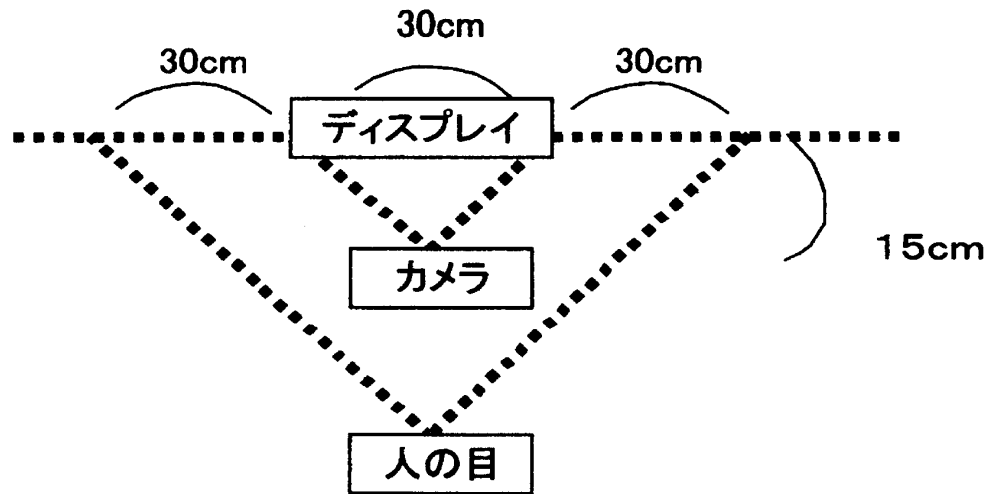


図1 カメラと物体と人の目の位置関係

4 座標系と動き

座標系はディスプレイの右方向をX軸の正の向き、上方向をY軸の正の向きとした。通常の座標軸はY軸方向をディスプレイの下向きにとることが多いが、感覚的には上向きの方がなじみやすいようである。3DCGでは3次元であるから、もうひとつZ軸が必要である。これは奥行きを表すので、感覚的には視線の向きを正と取ることが望ましい。したがって、Z軸はディスプレイの手前から裏に向かう向きを正とした。

この3次元空間内のカメラの位置を座標 (X, Y, Z) で表すことにする。また、右旋回と左旋回を取り入れることにすると、旋回角度を表す必要性がある。左右の向きを角度 ϕ で表すこととする。単位はラジアンである。Z軸方向を向いているときを0ラジアンとし、+X方向に向きを変えると ϕ が増加することとする。したがって、0.1ラジアンの角度だけ右旋回することは $\phi = \phi + 0.1$ と単純である。X、Y、Z座標は変化しない。まとめると、 $\Delta\phi$ ラジアンの旋回で、

$$\text{右旋回： } \phi = \phi + \Delta\phi$$

$$\text{左旋回： } \phi = \phi - \Delta\phi$$

となる。

移動は少し複雑である。右へ0.1の距離の移動は、 $X=X+0.1$ と書けばようさそうであるがそう単純ではない。カメラが常に+Z軸方向を向いているならばこれで良いのだが、3D空間内を自由に動き回れることを想定しているので、向きを変えられなければならない。つまり、右が常に+X方向ではないので ϕ の値が0でない値になってもうまく右に移動できるよう記述しなければならない。図2に示した実線の矢印はカメラの向きを表している。位置は原点においているが、他の場所でも同様に考えれば良い。破線は移動の向きをカメラの向きを基準にして右方向に取ったことを表している。分かりやすいように実線の矢の先端から描いた。大きさは Δ である。この

成分は点線で示した。この図から分かるように、 Δ の移動で次のようになる。

$$\text{右移動: } X = X + \Delta \cos \phi, \quad Z = Z - \Delta \sin \phi$$

$$\text{左移動: } X = X - \Delta \cos \phi, \quad Z = Z + \Delta \sin \phi$$

$$\text{前進: } X = X + \Delta \sin \phi, \quad Z = Z + \Delta \cos \phi$$

$$\text{後退: } X = X - \Delta \sin \phi, \quad Z = Z - \Delta \cos \phi$$

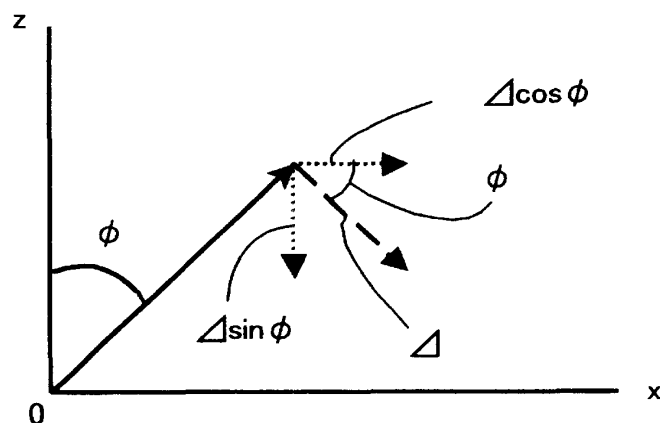


図2 右方向への移動ベクトルとその成分

5 操作方法

基本的な操作方法は、左・右・前・後・右旋回・左旋回の4つの動きを可能とすることにした。これは人の動きにほぼ対応している。前を見ながら右に動くといった動きも可能である。採用しなかったのは、見上げるや見下ろすといった動きである。これらの動きを取り入れるとコンピュータの操作方法が複雑になり、誤ってこれらの動きの操作を行ってしまった後に元に戻す方法が分からなくて困るという事態に陥ることを避けるためである。建物であるから当然上下の動きもあるわけだが、実際に人が動く場合は階段を利用するので、昇り階段に向かって進むと上に移動するというで上下の動きを実現した。また、センター試験では、エレベータの使用は禁止されているので、エレベータによる上下の動きは組み込まなかった。

操作をマウスではなくキーボードとしたことの大きな理由は、キーボードでは多くの情報を独立した操作で入力できることである。具体的には右移動を右矢印キー、左移動を左矢印キー、前への移動を上矢印キー、後ろへの移動を下矢印キーとし、右旋回をXキー、左旋回をZキーとした。このように配置すると、右手と左手にキー操作を分担させることになるので、慣れると移動の後に旋回といった異なった動きの入力がスムーズにできる。前後左右は簡単だが、旋回がXやZのキーというのはなじみにくいかも知れない。しかし、なじみにくい操作が2つなので、習得する時間があまり必要ないと期待できる。この配置のもうひとつの利点は、2つのキーの同時操作が可能であることである。旋回しながら移動することにより、ある1点を中心にした円運動を行うこともできる。

IV ソフトウェアの改善

1 ソフトウェアの操作方法を改善するための調査

ここで作成したソフトウェアの操作方法を改善するためにさまざまな人に使用してもらい感想を求めた。被験者は20歳代、30歳代、40歳代、60歳代であった。すべての人が操作方法の説明を傾聴し理解した。ただ、通常行わない操作を行うことになるので、戸惑いがあった。特に最近コンピュータの操作方法を熱心に練習した人ほど、戸惑いが大きかった。たとえば、ワープロ検定試験の練習をした人は、その操作に習熟したために新しい操作方法を理解することが困難であった。具体的には、カメラの操作キーの配置が反対であるように感じることもあった。通常カメラの位置を右に移動すると、カメラに移る像は左に移動する。したがって、3DCG上でカメラを右に移動するよう、右矢印キーを押すと、ディスプレイに表示される建物は左に移動するよう作成した。日常生活ではあたりまえのことだが、コンピュータのディスプレイ上ではなかなかなじめなかったようだ。ディスプレイの右端に位置するドアを中央に持ってくるには左矢印キーを押したくなるようだ。また、ディスプレイ上にあるオブジェクト、たとえば、グラフを左に移動するときは、マウスでそのオブジェクトを左にドラッグして移動するよう操作するので、マウスに手が伸びるようだ。キーボードで右矢印キーを押すべきときに、マウスで左にドラッグしたくなるようで、コンピュータの操作を熱心に練習したことがかえって新しい操作を習得することを妨げた例となった。

また、これらの被験者に聞いた結果、次の意見が出された。1. 目的地に向かう向きを示すガイドがあったほうが良い。2. 試験室の中を作成していないが、試験室の中を作成した方が良い。4. 階段の上下の操作は不要だ。3. 当初、各高校にコンピュータを持って行き、そこで実演することを想定していたが、試験前日に受験会場を下見に来た受験生を対象とすれば、前日は入れない建物にコンピュータの中の仮想的な建物で受験する机まで行けるように方針を転換すべきだ。これらの意見はすべてここで作成するソフトウェアのための参考とした。これらの意見の他にもさまざまな改良を加えて最終的なソフトウェアとした。その改良を以下に説明する。

2 目的地の位置を示すガイド

案内のためのガイドが必要だと感想が多かったので、目的地まで進むにはどちらの方向に進めば良いかを示すガイドを表示することにした。このガイドは、形状は矢印で、色は周囲の景色に溶け込まないよう明るい黄色とした。矢の向きは進むべき方向を表している。ただし単に目的地の向きを示すことにすると、役に立たない。3階が目的地で現在建物の外にいる場合、矢は3階を指すことになってしまう、つまり斜め上方を指すからである。空を飛んで3階に行く訳ではない。そこで、現在建物の外にカメラが位置しており、3階に向かう必要がある場合は、次のようにガイドの向きが変化するように工夫した。建物の外から建物に入るまでは建物の入り口を指す

3 DCGによる大学案内ソフトウェアの作成 (田 縁 正 治)

こととした。今回は建物の入り口は1箇所しかないなので、その入り口を指し示すこととした。もちろん、左旋回や右旋回といった向きの変化に対応してガイドの向きを変えることにした。また、カメラの位置が前方に移動することによって入口が横から後方へと移動した場合は、カメラの位置と建物の入口との位置関係が変化するので、それに応じてガイドの向きを変え結果的にガイドが建物の入り口を常に指しているよう工夫した。建物の中では階が異なっていれば階段の位置を指し、同じ階ならば試験室の入り口を指し、試験室に入ると受験する机を指すようにした。試験室内では、机の上に受験番号が貼られているので、この番号の上にガイドが来るとみづらくなる。このため、ガイドは表示・非表示をメニューから選択できることとした。つまり、非表示を選択すると、ガイドは見えなくなり、受験番号が邪魔されることなく見えるようになる。2人の受験生が続けてこのソフトウェアを使用する場合には、建物の中に入るまではすでに理解できているので、現在の試験室から次の試験室に移動する部分だけを理解したいはずである。このような目的で操作の途中で受験番号を変更したときはそれに対応するため、ガイドは最初試験室からの出口を指すように工夫した。

3 試験室の作成

試験室は、受験生が最も関心を持っている場所であるから、なるべく実際の部屋の様子が分かるように作成しなければならない。そこで、机の大きさ、位置、高さなどさまざまな長さを測ってなるべくリアルになるように努力した。特に注意したことは、机の天板の面積である。一人用の机、2人用の机、3人用の机とさまざまな机が利用されているので、それを忠実に再現した。また、机の個数は実際の試験室と同じ状態にして、前から数えても後ろから数えても受験する机の位置が把握できるようにした。また、受験する机の上には受験番号を貼ることとしたが、すべての机に受験番号を貼ると探すときにかえって分かりにくいと考えられるので、このソフトウェアを操作している受験生の受験番号のみを貼ることとした。

4 操作方法の改善

移動スピードは2段階設定した。コンピュータ上で仮想的に建物内を移動するときは、現実と同じ時間が必要だと機敏さに欠けるように感じられる。そこで、移動スピードを変えられることとした。建物の外で移動距離が長いところではスピードを速い方に限定し、建物が近づいた時点でスピードを遅くした。この場合は、速い方のスピードは選択肢とし、必要に応じて選ぶことができることとした。建物の中では常に遅い方を標準とし、速いほうはオプションとした。

自動的に移動する方法も一部組み込んだ。その方法は、階段を昇り降りするとき、その操作が意外と複雑なので、ひとつの階から別の階まで階段を利用して移動する操作を自動化することである。階段では、踊り場があるので、踊り場までの階段と踊り場内の移動ともうひとつの階での動きが必要になる。もう少し具体的に説明すると、下の階から上の階へ昇るときは、下の階

で階段に昇ろうとしているときの向きと上の階で昇りきったときの向きは逆になっている。全体で180度回転することになる。つまり、1階で、階段を昇ろうとしていたときに北を向いていたとすると、2階に到着した直後は南を向いていることになる。したがって、ひとつ上の階に行くための操作は、まず最初の階段をうまく壁に当たらないように注意しながら昇り、踊り場に到着したら直角に左旋回し、踊り場の左端や右端のような移動に不便な位置であるような状況ならば、左右に位置を修正し踊り場を直進する。次に再び左旋回し、2つ目の階段に正対する必要がある。最後に2つ目の階段を壁に当たらないように注意しながら昇りきれば終了する。操作に不慣れな状態でこれだけの操作を要求されると、注意の大部分が操作の方に向けられ本来の目的である建物の構造を理解する方には向けられないため、目的を達成することが困難となる。自動化した結果は好評で、操作が楽になり他の方面に注意が向けられるようになった。

階段を昇りきった後の向きも自動化した。最初は、前節で述べた方法で自動的に階段を昇り、昇りきった時点でそのまま停止し、人の操作を待つこととしていた。しかし、コンピュータが自動的に操作した直後は、現在どういう状況になっているのかを把握することが困難になっていることがある。自分で操作していない間は思考停止の状態になることがあるようだ。そこで、階段を昇りきった時点で、向きを少し変え、廊下が見えるようにした。こうすることで、目的地や、次の階段の位置を把握しやすくなった。

ある受験生がこのソフトウェアの操作を始めるときの位置、すなわち出発地は簡単に変更できるようにした。受験生が必ず正門から入って来るとは限らないので、このソフトウェアの、使用開始点を正門または東門とした。また、すでに大学構内を実際に散策し、建物の中に入れないことに気付いてから、このソフトウェアを使用することになった受験生には正門や東門から建物の入口までの移動は不要なので、建物の入口近くから出発する方法も用意した。また、ある受験生の操作が終わってから次の受験生へと操作する人が替わる時は、一旦出発地まで戻ることになるが、この操作は簡単な操作で行えることとした。

建物や試験室に入る条件は緩やかにし、現実にはありえないような動きでも建物や部屋の中に入れるようにした。すなわち、入口に斜めの角度から近づいたり、少々入口とずれた位置から入ろうとした場合でも、入口に入る操作を行ったものとして処理することとした。

トイレは階段の両側にあるので、男性用トイレに行った場合は女性用トイレの位置を知らせるメッセージウインドウを表示することとした。この結果、ソフトウェアを操作する受験生の男女の入力を不要とした。また、女性が男性用トイレに到達したとしても女性用のトイレの位置を知ることになるので、女性用トイレに到達する操作を省略することができる。したがって、男女の入力の省略と誤ったトイレに行った場合の、その後の操作の省略の2つの簡略化を行えた。

5 表示の改善

メインウインドウの表示は、コンピュータ内に作成された仮想的な空間をカメラが写した像で

あるが、これだけでは、現在カメラがどこにあるのか把握しにくいことがある。そこで、3DCGとは別のウインドウに全体図を表示することとしたが、この全体図の中にカメラの位置を表示するマークを挿入することにより全体図とメインウインドウとの対応関係を把握しやすくした。この全体図には目的地の位置を示すマークも挿入することとした。また、これらのマークの視認性を上げるためにマークを点滅させることとした。目的地も同様に点滅するマークで表示した。建物の中では2階と3階の作りが酷似しているので、このマークを見れば、現在何階にいるのかを容易に把握できる。

必要などころにはテクスチャを利用して分かりやすくした。たとえば建物の前面図には窓を表示、エレベータには受験生は利用できないという張り紙をすることでエレベータであることを明瞭にした。試験室の入り口は遠くからでも分かりやすいように明瞭な色のテクスチャを貼った。

6 現実と3DCGが大きく異なる場合

試験室の場所を理解するという目的からみて、不必要だと考えられるところはなるべく簡略化してソフトウェア利用者の注意が必要な方向に向くことを助けることとした。この目的のためにカメラの動きを束縛することもある。たとえば、不必要な向きにある程度進んだ場合、それ以上進めなくなるようにした。この簡略化は、試験室が準備されている建物の裏、大学の敷地の外、建物の中でも試験室が設定されていない上の方の階、該当しない試験室などで行った。該当しない試験室とは、このソフトウェアを操作している受験生が使用する試験室以外の試験室のことで、このような試験室に迷い込むことは時間の浪費になると考えたからである。

逆にカメラの動きを現実では不可能な動きまで許すことにした場合もある。中庭にある植木・花壇は省略し、カメラの動きを束縛しないこととした。試験室では机が並んでいるので本来なら机間のみ移動可能だが、目的とする机まで直進できることとした。

現実世界で2階から3階に向かうと階段の途中に2と3という2つの数字が表示されて、現在位置を分かりやすくしている。通常階段を昇るときは、やや顔を上向きにするので、この表示は上の方にあるようだ。しかし、ここで開発するソフトウェアでは、カメラを上向きにする操作方法は既に記述した理由で取り入れなかった。これに対応して階数の表示は現実のものより下に置き、分かりやすくした。昇ろうとした時点で現在の階と上の階が分かるようにした。降りるときは階段の前に立った時点で現在の階と下の階が分かるようにした。

建物の中を移動していると、壁がウインドウ全体を占め、動きが分からなくなることがある。今回採用した3DCG作成方法は簡単に作成できることをねらっているため、壁にはテクスチャを貼っていない。また、壁に当たる光のグラデーションも採用していない。したがって、何らかの方法で、動きを表現することとした。今回採用した方法は、壁の中に縦の線をいれ、この線が左右に移動することで動きを表現した。具体的には細い立方体のオブジェクトを壁の中に埋め込んだ。

7 使用した際の評価

平成13年1月20日、21日に行われたセンター試験では、約500名の受験生が宮崎公立大学を試験室として割り当てられた。試験室は第1試験室から第6試験室まで、建物の位置では、1階から3階まで試験室が準備された。したがって、受験生は自分の受験番号から試験室を探す際に、建物の中に入って階段を昇ることが必要な受験生がかなりの人数となった。

高校ですでに情報関連の授業を受けている生徒は自分でコンピュータの操作をしないと主張し、実際操作方法を簡単に習得した。一方、これまで、あまりコンピュータになじんでこなかった生徒は自分で操作することに躊躇したので、担当者が代わりに操作を行った。したがって、操作方法の習得に関して事前に検討したことは殆ど無意味であった。

ここで作成した下見用ソフトウェアを使用した受験生は17名で、全員好印象であった。3DCGがリアルタイムに動くこと、つまりインタラクティブであることや、目の前に見える現実の大学の建物がある程度のリアルさを持ってコンピュータ上に表現されていること、各自の試験で使用する机まで到達できることが好印象の原因であるように見受けられた。

謝 辞

ここで作成したソフトウェアはさまざまな段階でさまざまな方に試用して頂き、貴重な御意見を頂きました。また、実際にセンター試験前日に下見用ソフトとして受験生に提供するために多くの事務職員の方の支援を頂きました。ここに感謝の意を表します。

本研究において、宮崎学術振興財団より一部資金援助を頂いた。

参考文献

1. 新井利幸、Windowsオブジェクト指向プログラミング、1994年、メディア・テック社
2. 柴田望洋、明快C言語入門編、1998年、ソフトバンク株式会社
3. 林晴比古、新Visual C++6.0 入門、1999年、ソフトバンク株式会社
4. 瀬戸遥、Visual C++6.0 入門教室、2000年、翔泳社
5. 武田英雄、Delphi4 プログラミング入門、1999年、エーアイ出版株式会社
6. 瀬戸遥、Visual Basic5.0 入門教室、2000年、翔泳社
7. 川口輝久、河野勉、Visual Basic 6 応用編、1999年、技術評論社
8. 河西朝雄、Visual Basic 6.0 上級編、1999年、技術評論社
9. PCJapan, 190ページ、第6巻、第7号、2001年、SOFTBANK PC Japan
10. 藤田伸二、Direct X7 for Visual、1999年、翔泳社
11. 川田徹、Visual Basic による Direct X 活用術、2000年、工学社