

オリエンテーリング教育用ゲームの作成

高久雅生

1998 年 1 月

目次

1	はじめに	1
2	オリエンテーリング	2
3	VRML2.0	4
3.1	概要	4
3.2	特徴と利点	4
3.2.1	ノードとフィールド	4
3.2.2	再利用	5
3.2.3	アニメーション	5
3.2.4	イベント	5
3.2.5	マルチメディア	5
3.2.6	双方向性	5
3.2.7	プロトタイプ	6
3.3	スクリプト	6
3.3.1	Java 言語の概要と特徴	6
4	数値地図	8
4.1	国土地理院の数値地図	8
5	数値地図データの変換	10
5.1	VRML ファイルへの変換	10
5.2	Gnuplot データファイルへの変換	12
5.3	数値地図データ変換プログラム	14
6	システム	15
6.1	構成	15
6.2	地形表現	15
6.3	地図表現	17
6.4	コンパス	18
6.5	コントロールフラッグ	18
6.6	実行例	19
7	考察	29
7.1	数値地図の選択について	29
7.2	開発手法について	29
7.3	今後の課題	30
	謝辞	31
	参考文献	32
	付録	33

A	VRML ファイル	33
A.1	terrain.wrl	33
A.2	control.wrl	43
B	Java プログラム	46
B.1	CompassScript.java	46
B.2	CompassCanvass.java	46
C	数値地図の変換プログラム (mem2wrl.c)	49

図一覧

1	等高線と地形の対応	2
2	オリエンテーリング用地図の例	3
3	数値地図のファイルフォーマット	8
4	変換の流れ	10
5	Gnuplot による等高線出力例	14
6	地形表現の一例	16
7	地図表現の一例	17
8	コンパスの表現	18
9	コントロールフラッグの表現	19
10	実行例：北向き（初期状態）	20
11	実行例：東向き	21
12	実行例：西向き	22
13	実行例：南向き	23
14	実行例：下りのコントロールフラッグ	24
15	実行例：下りのコントロールフラッグ 2	25
16	実行例：登りのコントロールフラッグ	26
17	実行例：登りのコントロールフラッグ 2	27
18	実行例：登りのコントロールフラッグ 3	28

1 はじめに

オリエンテーリング競技は、いくつかの要素を基にした技術を必要とし、上達の難しいスポーツである [1]。その上達には何よりも経験の積み重ねが欠かせないと言われる。特に基礎的な技術の一つには、地図の読みとり能力があげられる。この技術は、地図の情報を基に実際の現地の地形の特徴を読みとる能力である。地図の情報とは、具体的には等高線などのことである。この地図の読みとり能力の向上には、実際に地形の特徴と地図を見比べながら学習することが必要である。そのため、この地図と地形の照合による競技能力の学習は、実際に現地に出向いて行なわなければならない。ところが、オリエンテーリングは自然の地形を利用した山や林など人里はなれた場所で行なわれるので、実際に競技を行なう場所に出向くには大変手間がかかる。この地形と地図の見比べ・読みとりのオリエンテーリング技術の学習の手間を軽減するために、本研究ではオリエンテーリング競技の疑似体験ができるシステムの開発を行なった。

このようなオリエンテーリング競技の疑似体験をコンピュータ上で行なうシステムには、他にも Navigate! [2] や Win OL [3] などのオリエンテーリングゲームがあるが、これらの既存のゲームは Windows95 パソコン上の独自のゲームとして作成されているため、機種・OS が違うと利用できない点や地図データの形式が異なる点といった問題がある。本システムは、WWW 上の 3 次元表現のための標準言語 VRML2.0 を中心として構成した。そのため、ネットワークを通じて機種や OS が異なってもシステムを利用できるようにした。つまり、WWW を利用できるシステム上では、ネットワークを介して、このシステムを利用することができる。また、地図データには数値地図データを利用し、国土地理院発行の DEM データ [4] や、米国全土の標高データが公開されている USGS (U.S. Geological Survey) の DEM データ [5] を変換して利用することができる。

本論文の構成は以下の通りである。2 章でオリエンテーリング競技の特徴などについて述べる。3 章では地形を表現する VRML2.0 について述べる。4 章では地形データとして用いた数値地図について述べる。5 章では開発に際して行なった数値地図データの形式の変換について述べる。6 章では開発したシステムの構成や機能について述べる。7 章では本システムについて考察し、まとめを行なう。

2 オリエンテーリング [1]

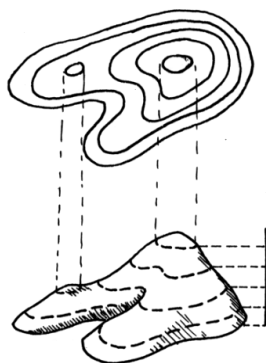


図 1: 等高線と地形の対応 [1, p.10]

オリエンテーリング競技は、山や林などの地形を生かした場所で実施されるスポーツである。競技者は、コントロールフラッグと呼ばれる決められたポイントを通り、スタートからのゴールまでの所要時間を競う。競技者は地図とコンパスだけで持ち、その情報を元に自分の進むべきルートを決めなければならない。地図情報の読みとりは重要な技術である。図 1 は地図と地形の対応の内、等高線によるものを表した概略である。等高線は地形の形状を理解するための最低限の要素である。

競技者は自分の進むルートを決めるために、地図の情報を頼りに実際の地形の様子をできるだけ精密にイメージする必要がある。さらに、移動するにしたがって、現れる地形がイメージした通りかどうか確認し、自分の進んでいるルートが、自分の選んだルート通りかを確認しなければならない。また、自分がどこにいるかわからなくなった場合には、まわりの地形を地図の特徴と見比べて、地図上で自分のいる地点を再確認しなければならないこともしばしば起こる。このように、オリエンテーリングの競技者は、地図の情報を元に実際の地形の様子をイメージし、実際の地形の様子と地図を同時に参照し、また、地形を元に地図上での居場所を確認するなど、地図と実際の地形の样子の照合をできるだけ正確に、素早く行なわなければならない。この基礎的な能力がなければ、オリエンテーリング技術は向上しない。

オリエンテーリングで用いられる地図は、図 2 のようなものである。一般にオリエンテーリングで用いられる地図は縮尺 15,000 分の 1 または 10,000 分の 1、等高線間隔 5m である。

地図上に表現される情報の種類には、次のようなものがある。地形（等高線）、植生（植生界）、道、水系・溝、人家、ほこら・墓などの人工特徴物などである。このような地形やその他の特徴物は、地図上に色と線や記号などで書き表される。

地形の様子を表す等高線はとりわけ重要である。なぜなら、競技中の地形の起伏はレース全体のスピードに大きく関係してくるからである。進路上の登距離が増せば、その分体力を消耗し、その後のレースに影響を及ぼす。また、その疲労から精神的な余裕がなくなり、ミスを起こしやすくなる。レースのルート上ではなるべく登距離を減らすルートを考えて進まなければならない。



図 2: オリエンテーリング用地図の例 [6]

3 VRML2.0 [7][8][9][10]

3.1 概要

VRML2.0 は、分散型情報システムとして広く普及した WWW 上で 3 次元表現を行なうための記述言語で ISO 標準として採択されている。VRML とは、Virtual Reality Modeling Language の略である。VRML は、WWW 上のハイパーリンクされた仮想世界を記述するための言語で、最終的な目標は、多人数参加のインタラクティブなシミュレーションを行う、インターネット上の標準的な言語となることである。

歴史的には、1994 年春の第 1 回 WWW 会議で WWW 上での 3 次元データの標準的な利用法について議論され、メーリングリストを中心として仕様の提案・検討が行われた。その結果、1995 年 5 月に最終的に VRML1.0 の仕様が決定した。VRML1.0 では、ネットワーク言語としての特性を考慮して以下の点を満たすように設計された。

- プラットフォームに独立
- 拡張性
- バンド幅の狭いコネクションでの処理能力の高さ

VRML1.0 によって WWW 上で 3 次元の仮想的な世界を記述できるようになったが、VRML1.0 は 3 次元世界の空間における物体の動作の記述と、ユーザとの対話的・双方向的やりとりが記述できなかったため、いわば、静的な 3 次元空間の記述しかできなかった。そこで、VRML1.0 を元によりインタラクティブで動的なものを記述できる VRML2.0 の仕様が提案され、策定された。

3.2 特徴と利点

VRML2.0 の特徴と利点について述べる。

3.2.1 ノードとフィールド

VRML の基本単位はノード (Node) と呼ばれるものである。ノードは 3 次元のオブジェクトの標準的な集合であり、例えば立方体、球、テクスチャ、幾何変換などがある。VRML2.0 ではノードは全部で 53 と多様なノードを用意しており、目的にあわせていくつかのノードを組み合わせることにより、容易に 3 次元の仮想世界が記述できる。

ノードにはフィールドと呼ばれるパラメータをあたえることができる。フィールドのパラメータ値を変えることで、同じノードであってもさまざまなオブジェクトが表現できる。例えば、球を表す Sphere ノードでは、球の半径を radius フィールドで与える。このことにより、さまざまな大きさを持った球を記述できる。また、特に記述されないフィールドの値は標準の値となる。

ノード間の関係は、階層的に記述でき、階層構造を与えるためのノードは、グループ化ノード (Grouping nodes) と呼ばれる。グループ化ノードは他のノードを子ノードとすることができ、子ノードは親のノードが持つ属性の影響を受ける。例えば、物体の位置を変更するための移動や回転の幾何変換用の Transform ノードを用いて、幾何変換の及ぶオブジェクトと及ばないものを選ぶことができる。また、物体の材質を規定する Material ノードの記述は同じ階層内でのみ

有効となる。また、このようなノード間の関係は累積的に蓄積され、先に記述した内容は後から記述したノードに影響を与える。このように、一部のノードの記述はお互いに関係しあい、ノードによる仮想世界の記述はノード間の階層関係、累積関係を記述することになる。

3.2.2 再利用

VRML2.0 ではオブジェクトに名前をつけて再利用することができる。全く同じパラメータを持つノードはこの名前を使うだけで再利用できる。名前付けは DEF 文で行ない、再利用は USE 文で行なう。

3.2.3 アニメーション

VRML2.0 は物体をあらかじめ決めた通りに動かすことができる。このことで、雲や太陽、鳥など自然に動く物体を記述できる。VRML2.0 ではインターポレイタ (Interpolator) ノードを用いて、ある時間内でのアニメーションを容易に記述できる。

3.2.4 イベント

VRML2.0 のイベントの概念とその仕組みについて説明する。イベントはノード間のパラメータの受渡しで行なわれる。パラメータを他のノードに渡すノードをセンサノードと呼ぶ。センサノードはその種類に応じて、決まったイベントを発生させ、自身のノードの持つパラメータを他のノードに渡す。渡されたノードでは、そのパラメータはノードのフィールド値となり、オブジェクトの属性を変化させたり、また別のノードに渡されたりする。ノードの持つフィールドには、field, exposedField, eventIn, eventOut という種類があり、このフィールドの種類がイベントの受け渡しの際の与え方を決定する。field は、単に固定的なパラメータである。eventIn, eventOut は、イベントで受け渡されるパラメータである。exposedField は、field, eventIn, eventOut を全て持つフィールドタイプである。つまり、デフォルト値を与えながら、イベントの受け渡しもおこなえるパラメータになっている。ノード間のイベントの受け渡しは、全て ROUTE 文を使って定義される。ROUTE 文は、eventIn と eventOut を結びつける。あるノードで発生したイベントで、eventOut が送られると、ROUTE で結ばれたノードの eventIn にその値が、送られる。例えば、ある viewpoint とあるオブジェクトの位置を結び付けたいければ、

```
ROUTE Object1.translation_changed TO Viewpoint1.set_position
```

とする。Object1 の translation フィールドの値が変化を起こすと (イベントが発生)、Viewpoint1 の position フィールドの値にその値がセットされることを表す。

3.2.5 マルチメディア

VRML2.0 は画像、音声、動画などが取り扱える。VRML2.0 はこれらのマルチメディア化したコンテンツの作成に Texture ノード、MovieTexture ノード、Sound ノードなどを用いる。

3.2.6 双方向性

VRML2.0 ではユーザーの行動に応答するような世界を記述できる。センサノードは、ユーザーのクリックなどでイベントを発生させ、そのイベントを他のオブジェクトに渡していくことに

よって、変化する世界を記述できる。また、時間の経過などにあわせて、イベントを発生させることもできる。

3.2.7 プロトタイプ

プロトタイプは、VRML ノードタイプの拡張のために用意されており、複数のノードを組み合わせて、新たなノードを自分で定義できる。プロトタイプ化することで、オブジェクトにパラメータを付けて、カプセル化する事ができる。また、いったん作成したプロトタイプノードはネットワークを通じて、何度でも利用できる。これは、オブジェクトのカプセル化・パラメータ化を実現する。プロトタイプの定義は、PROTO, EXTERNPROTO 文で行なわれる。

3.3 スクリプト [11]

VRML2.0 におけるアニメーションやユーザとの応答動作は、ノード間のパラメータの受渡しによるものなので、VRML の仮想世界を自由に制御することは難しい。例えば、センサとインターボレイタによる単純な物体の動作の記述だけでは、条件判断を元にした記述などができない。そのため、VRML2.0 だけでは記述できない動作・インタフェースを実現するために VRML2.0 以外の外部のプログラムを仮想世界に連係させることができる。このような機能の一つである VRML2.0 の仮想世界に Java 言語のプログラムを組み入れる JSAI(Java Scripting Authoring Interface) について解説する。

VRML は、Script ノードを用いて外部のプログラムと連係した表現が可能となっている。Script ノードは、外部プログラムにどのようなイベントが渡され、どのように値が返ってくるかを規定する。また、外部プログラムの種類は指定されていないので、どのような言語を使用することもブラウザの開発者に任されている。仕様においては、ネットワーク言語として急速に普及している Java と JavaScript の 2 種類の特定のプログラミング言語との API が付録として記載されている。今回 Java 言語を使用した理由は、ネットワーク上でのプログラミング言語としての普及と大半のブラウザが採用していることによるものである。

Java と VRML の Script ノードを連係させるためには、`vrml`, `vrml.node`, `vrml.field` の各クラス・パッケージを利用する。このクラスパッケージでは、VRML のノード自体とそれに対する参照・変更などのメソッドが記述されている。このクラスパッケージを用いて VRML の 3 次元中におけるプログラムの動作を記述する。

- まず、VRML ファイル中の Script ノードの url フィールドに Java のバイトコード (クラス) ファイルの URL を記述する。
- このクラスファイル中に記述されたクラスは、`vrml.node.Script` クラスのサブクラスでなければならない。
- 指定されたファイルの名前は、このスクリプトのクラス名と一致しなければならない。

3.3.1 Java 言語の概要と特徴 [12]

Java 言語は汎用的なオブジェクト指向型プログラミング言語であり、機種に依存しないプログラムを作成できることが一番の特徴である。Java は機種に依存しないバイトコードにコンパイルされ、各機種に依存した処理は Java VM (Virtual Machine) で行なう。この中間ファイルであるバイトコード命令は、プラットフォームに依存しないので、ネットワーク上でプログラ

ムの交換に適している。Java の基本単位はクラス (Class) である。このクラスはプログラムで使用するデータとそのデータに対する処理を記述したメソッド (Method) からなっている。クラスにすることで、データに対する処理・アクセスの方法を規定・制限することができる。クラスは、それらの集まりとしてパッケージ (Package) にできる。

4 数値地図

数値地図とは、地図（地形図）の情報をデジタル情報に変えたものである。現在、種々の地図データのデジタル化が行なわれている。例えば、都市空間情報、地形情報、土地利用情報などである。

標高データの数値地図には、いくつかあるが、特に数値標高モデル (DEM : Digital Elevation Model) と呼ばれるモデルの数値地図が主要なものである。DEM は地図平面を格子状にモデル化し、等高線情報を元に格子の中心点の標高値を記述したものである。DEM による数値地図は、USGS(U.S.Geological Survey) が無償で提供しているものが有名で米国全土の 100,000 分の 1 の地形図を元にした約 100m メッシュの DEM モデルのデータが公開されている [5]。また、日本では国土地理院が 25,000 分の 1 の地形図を元にした約 50m 間隔のメッシュの DEM データを公開している。

4.1 国土地理院の数値地図 [4]

この数値地図のデータは、国土地理院が刊行している 25,000 分の 1 地形図に描かれている等高線から求めた数値標高モデル (DEM: Digital Elevation Model) データである。地形図をを経度方向および緯度方向に 200 等分して得られる各区画 (25,000 分の 1 地形図上で約 2°) の中心点の標高値が記録されている。標高点間隔は実距離で約 50m である。

この形式の数値地図の実際のファイルフォーマットについて説明する。

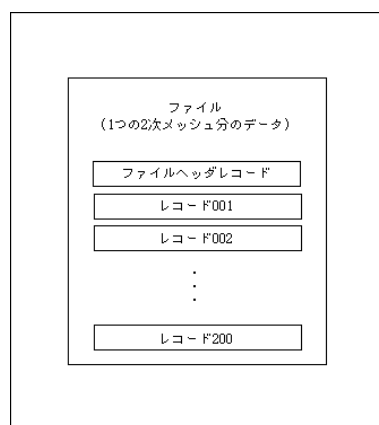


図 3: 数値地図のファイルフォーマット

[illegible]

標高データ部

54402000100220002200022000200002000021000210002100021000210002100021000210002200
022000220002200022000220002200022000220002200022000220002200022000280003600
035000350003400035000350003500035000350003500035000350003500035000350003500
0360003600036000360003500036000360003500034000330003300034000330003200
027000250002500025000240002500024000240002400024000240002400024000240002400
0240002400025000250002600026000260002600025000250002500025000260002600
026000260002600026000260002600026000260002600026000260002600026000260002600
026000260002600025000250002500025000250002500025000250002500025000270002600
026000280002800029000290002900030000310003400036000360003600036000360003500
0350003600038000390003900040000470006100074000830008300084000870010300
1220012300111001070010100091000770006000061000770008000092000980011200
0990010700109001180011900117001210012000125001300014600161001760019200
2100023200253002720028200289002750026400250002400022000190001620017100
1550016100152001900017500184001720014800156001710019200215002150024800
25800258002970032300302003110
...省略...

9

5 数値地図データの変換

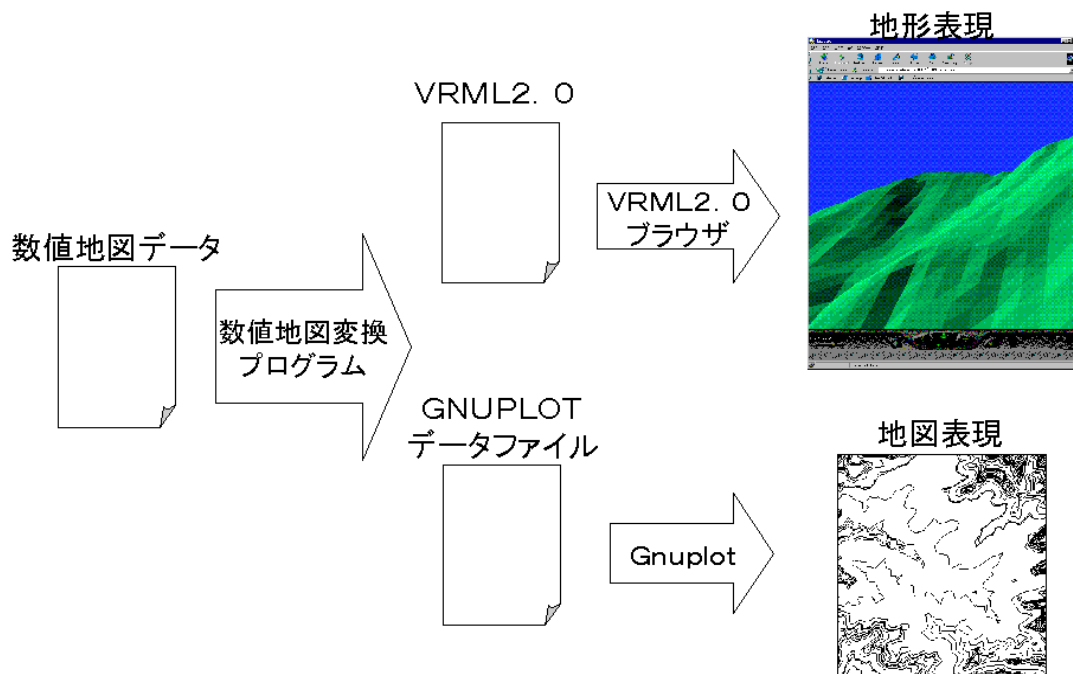


図 4: 変換の流れ

本研究では、数値地図データの形式を変換し利用した。数値地図データの変換について解説する。変換の概要は図 4 の通りである。

数値地図のデータは、VRML2.0 ファイル、Gnuplot データファイルに変換した。VRML2.0 は 3 次元の地形表現を行なう。また、Gnuplot データファイルはグラフ描画ツールである Gnuplot を使用して地図画像とするためのものである。

変換には、自作の C 言語によるプログラム（付録 C）を使用した。このプログラムでは、数値地図のヘッダ情報を除き、地形データをそれぞれ適切なフォーマットに変換した。以下の節ではそれらの変換とファイルフォーマットについて解説する。

5.1 VRML ファイルへの変換

まず、VRML2.0 形式への変換について説明する。

ElevationGrid ノード 地形データの変換に直接関係する VRML2.0 のノードは ElevationGrid ノードである。このノードは凹凸のある地形などを表現するためのノードで、その記述は格子状のデータモデルを持っている。標高データそのものは、height フィールドに記述する。格子の

全体の大きさは、xDimension フィールド、zDimension フィールドが決定する。また格子点間の距離は、xSpacing フィールド、zSpacing フィールドの値で与えられる。xSpacing、zSpacing フィールドの値は 50(m) とした。height フィールドのデータ値は、標高値の並びになる。

標高データを記述する ElevationGrid ノード以外に付加したノードとその働きについて述べる。

DirectionalLight ノード VRML2.0 における光源を表現するノードであり、本システムでは自然に存在する太陽光に見立てて、direction フィールドで地形を上から照らす光源を設定した。

BackGround ノード VRML2.0 の仮想世界における物体の背景となる空などのパノラマの情報を表現するノードであり、skyColor フィールドで自然の空の青色に近い色を設定した。

NavigationInfo ノード VRML2.0 の仮想世界をユーザが移動する動作に関わる記述をするノードである。avatarSize フィールドは移動を行なうユーザの大きさを記述するためのフィールドで、3つの要素からなる。その値の意味はそれぞれ、物体とユーザがぶつかる距離、ユーザの視点の高さ、ユーザが乗り越えられる物体の高さである。本システムではこのフィールドに以下のように値を設定した。物体とユーザがぶつかる距離は 0.25(m) であり、これは標準の値である。ユーザの視点の高さは 1.5(m) と設定した。ユーザが乗り越えられる最大の大きさは 5.0(m) として、少々急峻な地形であってもユーザが地形に登ことができるように設定した。type フィールドは、VRML2.0 の仮想世界の中でユーザが物体を眺めたり移動する方法を設定するフィールドで、このフィールドに設定できる値は、

- 移動を行なわない "NONE" タイプ
- 物体を回しながら形状を調べることができる "EXAMINE" タイプ
- 仮想世界の中を歩き回る "WALK" タイプ
- 仮想世界の中を地面を離れて飛びながら調べる "FLY" タイプ

の計 4 タイプである。地形中は歩いて移動することを前提とするので、"WALK" タイプを設定した。

WorldInfo ノード 現在表示している VRML の仮想世界に関する情報を記述する。title フィールドは VRML 世界のタイトルを記述するものであり、本システムでは地形の数値地図のヘッダからとった地図の名前を設定した。また、info フィールドにデータの由来と変換プログラムにかわる情報を付加した。

Viewpoint ノード VRML ブラウザを通じてユーザに視点を与えるためのノードであり、ユーザが地形をいくつかの方向から眺められるよう複数の視点を与えた。description フィールドはその視点を表す表記であり、"Start"、"In the Center"、"Over Head" など視点位置にふさわしい記述にした。position フィールドはどの座標にユーザの視点を設定するかを表し、それぞれスタート地点、地形の中央、地形を鳥瞰する地点の座標を全標高点の情報から記述する。orientation フィールドは、ユーザがどの方向を向くかを示す値である。それぞれその視点に適切な方向を設定した。

Material ノード 物体の材質を記述するためのノードであり、地形の地面の色を記述するために用いた。diffuseColor フィールドは物体の持つ色を設定する。ここではやや緑色に近い値を設定した。

Appearance ノード 物体の持つ特性を設定するノードであり、ここでは Material ノードによる材質の設定を行なった。

Shape ノード 物体の形状と特性を同時に表現するためのノードであり、ここでは実際の地形の情報を与える ElevationGrid ノードとその色を表現するための Appearance ノードを記述した。物体の形状は appearance フィールドで、物体の形状は geometry フィールドで記述した。

Transform ノード ノードのグループ化と座標系の分離を行なうノードであり、本システムでは地形に関わるノードをまとめて記述し、他の幾何変換から独立させるために用いた。children フィールドはグループ化されたノードの子ノードを記述するフィールドであり、このフィールド中に地形を表現するノードと地形上で行動するユーザに関するノードをまとめた。ここでは、まず全オブジェクトを一つの座標系で表現するために全ノードをまとめる Transform ノードを記述し、また実際の地形である Shape ノードは他と区別できるようこれをさらに Transform ノードでまとめた。

5.2 Gnuplot データファイルへの変換 [14]

地図画像は、VRML2.0 ファイルと同様に数値地図の標高データを抽出し、そのデータを元に等高線画像を描いた。等高線の間隔は、通常のオリエンテーリング地図と同じく 5m としたが、地図の縮尺は一般のオリエンテーリング地図で使われている 15,000 分の 1 ではない。等高線の描画は対話的グラフ描画ツールである Gnuplot を用い、等高線画像を抽出した。

等高線の描画は、数値地図のデータを取りだし、Gnuplot のデータファイルフォーマットに変換し、Gnuplot を使ってデータファイルを等高線として描写することで行なった。Gnuplot とそのデータファイルについて解説する。Gnuplot は、対話的関数グラフ描画プログラムであり、本研究では、等高線画像の生成のために使用した。Gnuplot データファイルは先に述べた数値地図の変換は VRML2.0 形式へ変換したプログラムから出力される。Gnuplot のデータファイルフォーマットは、一行に一つの標高値を記述し、格子（メッシュ）状のデータに関しては、空行がそのデータ列の終了を示す。例えば、5×5 の格子状のデータ要素のデータファイルフォーマット例とその等高線の出力例である。

369	383	412	435	403
375	366	400	434	415
376	357	393	430	419
362	357	394	415	386
341	362	397	411	373

表 1: 数値地図メッシュのデータ例

まず、表 1 のような 5×5 の格子状の標高データは、変換プログラムによって次のような Gnuplot データファイルとして出力される。

```
341
362
397
411
373

362
357
394
415
386

376
357
393
430
419

375
366
400
434
415

369
383
412
435
403
```

さらに、このデータファイルを対話的グラフ描画ツール Gnuplot で描画すると図 5 のような等高線画像を得ることができる。実際に画像の出力を得るには、次のような Gnuplot コマンドを使用した。

```
set contour base
set nosurface
set view 0,0
set size 0.721,1.0
set noxtics
set noytics
set data style lines
set nokey
set cntrparam bspline
set cntrparam levels 500
set cntrparam levels incremental 0,5
set terminal postscript portrait solid
set output "contour.ps"
splot "map.dat"
```

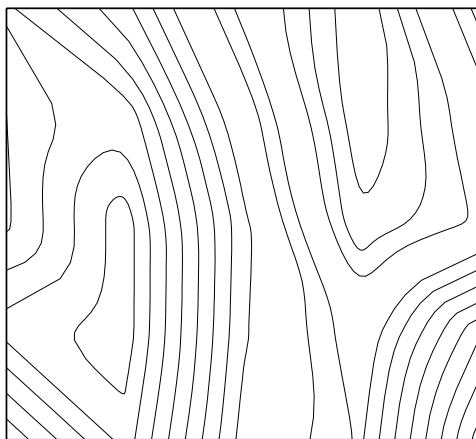


図 5: Gnuplot による等高線出力例

5.3 数値地図データ変換プログラム

数値地図データを VRML2.0、Gnuplot データファイルの形式に変換するプログラムを C 言語で作成した（付録 C）。プログラムは VRML 形式もしくは Gnuplot データファイル形式の出力を行なう。また、変換は数値地図のデータ全てを出力するだけでなく、変換する格子の大きさを指定して出力させることもできる。

プログラムの実行方法は、

```
% mem2wrl -size 60 -o out.wrl 538480.MEM
```

とすることで、数値地図の 60×60 のデータを VRML2.0 形式に変換する。

また、Gnuplot データファイル形式にするには、

```
% mem2wrl -gnuplot -size 60 -o out.dat 538480.MEM
```

とする。

6 システム

6.1 構成

本システムは、3次元地形、地図、コンパスの3つのウィンドウから構成される。ユーザは、VRML2.0で記述された3次元の地形の中で、地形に対応した地図とJavaプログラムにより実現されたコンパスを見ながら、オリエンテーリングを行なう。コンパスはVRML2.0のスク립トの機能を用いて実現し、地形中でのユーザの動作に連動したJavaプログラムにより、表現されている。ユーザはVRML2.0により記述された3次元地形の仮想空間の中を自由に動きまわることができる。

本システムは、WWW上のコンテンツとして利用できるように、ユーザはWWWシステムを通じて利用することもできる。その場合には、ユーザはWWW環境とVRMLブラウザを利用すれば、機種に依存せず本システムを利用できる。例えば、ユーザはUNIXのワークステーションから、Windows95パソコン上からと機種に依存せず、本システムにアクセスし、利用できる。

ユーザは次のように本システムを利用する。

1. ネットワーク上のWWWシステムを通じて、VRML2.0ファイルを取得し、
2. VRMLブラウザが表現されている3次元地形を表示し、
3. VRMLブラウザはJavaプログラムを取得し、コンパスや地図の表示を行なう。
4. ユーザはVRML2.0の地形に連動したコンパスと地図を見ながら、オリエンテーリングを行なう。

開発は図4のように行なった。本システムを構成するファイルは

- VRML2.0 ファイル
おもに地形表現を行なう。
- Java プログラム
コンパスを表現するために用い、コンパスウィンドウを生成し、VRML2.0による地形表現とスク립ト機能で連係している。
- 等高線画像ファイル
VRML2.0の地形データに対応した等高線間隔5mの地図である。

の3種類である。

6.2 地形表現

地形表現はVRML2.0で行なう。まず、数値地図の変換プログラムによるVRML2.0ファイルの出力を元にして、

- コンパスの表現のためのScriptノード

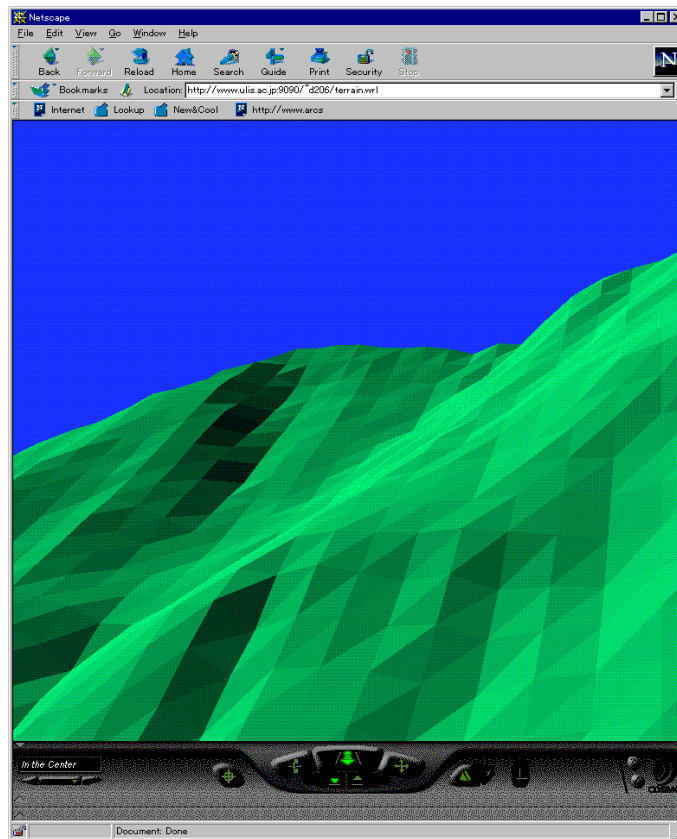


図 6: 地形表現の一例

```

DEF USER ProximitySensor {
    size 10000 10000 10000
}
DEF WINDOW Script {
    url "CompassScript.class"

    eventIn SFRotation user_orientation
    eventIn SFVec3f      user_position
}
ROUTE USER.orientation_changed TO WINDOW.user_orientation
ROUTE USER.position_changed TO WINDOW.user_position

```

- コントロールフラッグのための EXTERNPROTO 文と ControlFlag ノード

```

EXTERNPROTO ControlFlag [
    field SFCOLOR poleColor
    field SFCOLOR punchColor
    exposedField SFVec3f position
] "control.wrl"

ControlFlag {
    position 1400 1203 1500
}
ControlFlag {
    position 1500 1143 1400
}
ControlFlag {
    position 1500 1181 1450
}
...省略...

```

を追加すると地形表現の VRML2.0 ファイルが完成する。

6.3 地図表現

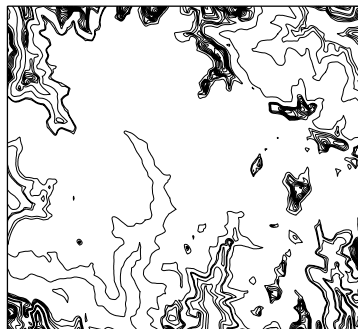


図 7: 地図表現の一例

地図は数値地図のデータを 5m 間隔の等高線に直した地図画像ファイルである。5.2節で述べたように、この地図画像は、数値地図変換プログラムから得た Gnuplot データファイルを使って、Gnuplot で等高線を描かせたものである。ユーザはこの等高線地図を見ながら、地形中を動き回ることができる。なお、この地図は北が上を向くよう置かれている。

6.4 コンパス

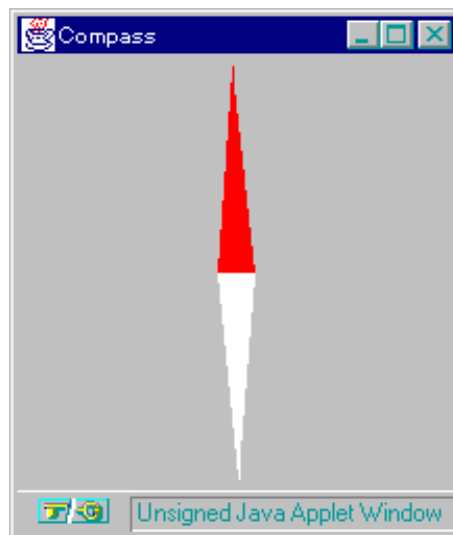


図 8: コンパスの表現

コンパスとは、本システムではユーザの向いている方向に応じて地形表現中の北を赤い針が指すウィンドウである。このコンパスは地形表現に完全に対応しており、ユーザが北を向いた時、画面上で上を指し、南を向けば、下を指す。同様に東には右が、西には左が対応して指される。

このコンパスは地形表現の VRML ファイル中の Script ノードで指定された Java プログラムで表現した。つまり、スクリプトとして呼び出した Java プログラム中でウィンドウの生成などを行ない、コンパスの針を VRML ファイル中の ProximitySensor ノードのフィールド値に対応させ、リアルタイムで動かすようにした。ProximitySensor ノードは VRML2.0 の世界におけるユーザの向いている方向、居場所、といったイベントを扱うためのセンサノードである。

Java のプログラムは CompassScript.java, CompassCanvas.java の 2 つである（付録 B）。CompassScript クラスは、vrmf.node.Script クラスのサブクラスであり、VRML ファイル中で指定される Java プログラムファイルとなるスクリプトクラスである。CompassCanvas クラスは、実際にウィンドウとしてコンパスを描くためのクラスである。CompassCanvas クラスでは、実際にコンパスを描くためのデータやメソッドを定義した。CompassScript クラス中で、CompassCanvas クラスのデータやメソッドを利用して、コンパスを描いた。

6.5 コントロールフラッグ

図 9 のように、オリエンテーリング競技で通過すべき地点を表すコントロールフラッグを VRML2.0 で表現した。コントロールフラッグは約 2m の大きさがあり、ユーザはこれを実際の競技におけ

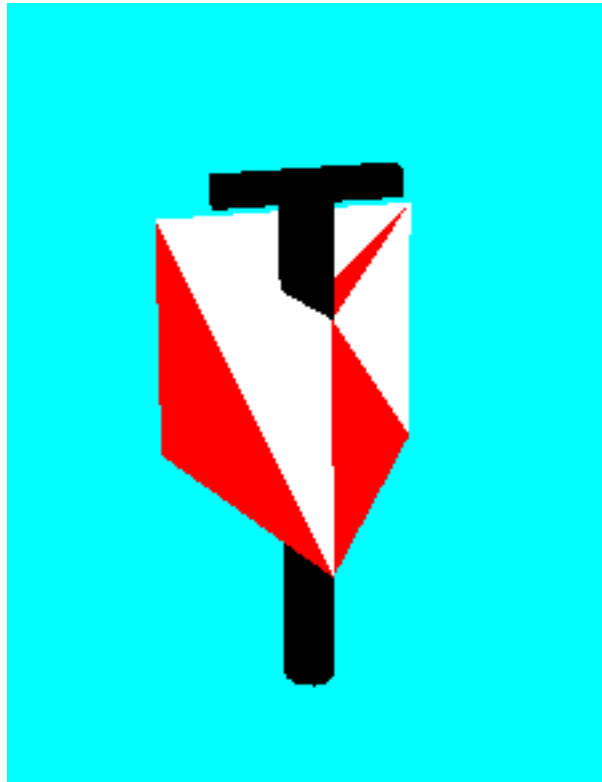


図 9: コントロールフラッグの表現

るコントロールフラッグに見立てて近付いたりすることができる。コントロールフラッグの表現は VRML2.0 のプロトタイプの機能を使って、実現した。このことで地形表現とは別にコントロールフラッグの表現がされているので、別の数値地図データから地形表現を行なった際にも新しくコントロールフラッグ自体の表現を行なう必要がない。そのため、新しい地形表現中にコントロールフラッグを置く際には、その地点の位置を指定するだけで良くなった。

まず、コントロールフラッグ単体で PROTO 文で表現し、実際の地形表現の VRML2.0 ファイル中で表現する際には、EXTERNPROTO 文で表現した。

6.6 実行例

ユーザの利用例を図 10 に示す。まず、ユーザは WWW によるネットワークを通じて、地形表現を行なっている VRML2.0 ファイルを取得し、VRML ブラウザによって表示を行なう。この時、VRML ブラウザはコントロールフラッグ表現のための VRML2.0 ファイルも取得し、コントロールフラッグも地形中で表現する。さらに、コンパス表現を行なっている Java プログラムを取得し、コンパスの表現を行なう。更に、用意された等高線画像を表示すると、地形表現、地図表現、コンパスのそれぞれのウィンドウが揃うこととなり、システムが利用できるようになる。図 10 は、この初期状態を Windows95 パソコン上の VRML2.0 ブラウザで示したものである。

初期状態からの行動に沿って説明を行なう。まず、初期状態ではユーザは地形上の中心にいて、北を向いている（図 10）。ここで視界を東（図 11）、西（図 12）、南（図 13）と向けた

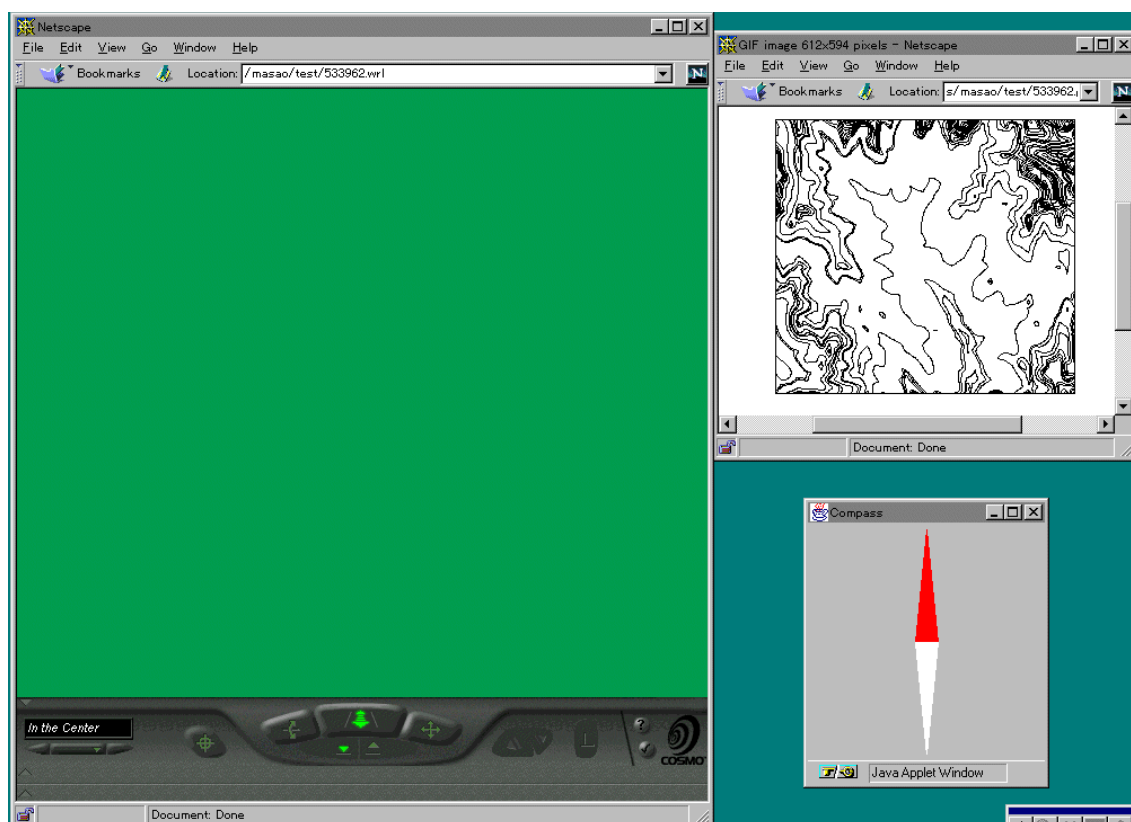


図 10: 実行例：北向き（初期状態）

様子も紹介する。ユーザの位置は変わらず、地図上の中央である。特に南を向いた時には、西の方に下る谷が地図上で確認できるのと同様に地形上でも確認できる（図 13）。ちなみに、それぞれ東西南北には確認のためコントロールフラッグを配置している。コンパスの針も指す方角を変え、VRML による地形の様子が変わっているのがうかがえる。

さらに、動き回った様子についてもそれぞれ例を挙げながら、説明する。コントロールフラッグに向かって進む様子について、図 14、15 では下りながら近寄る様子、図 16、17、18 ではもう少し遠めのコントロールフラッグに登りで近寄る様子を示す。

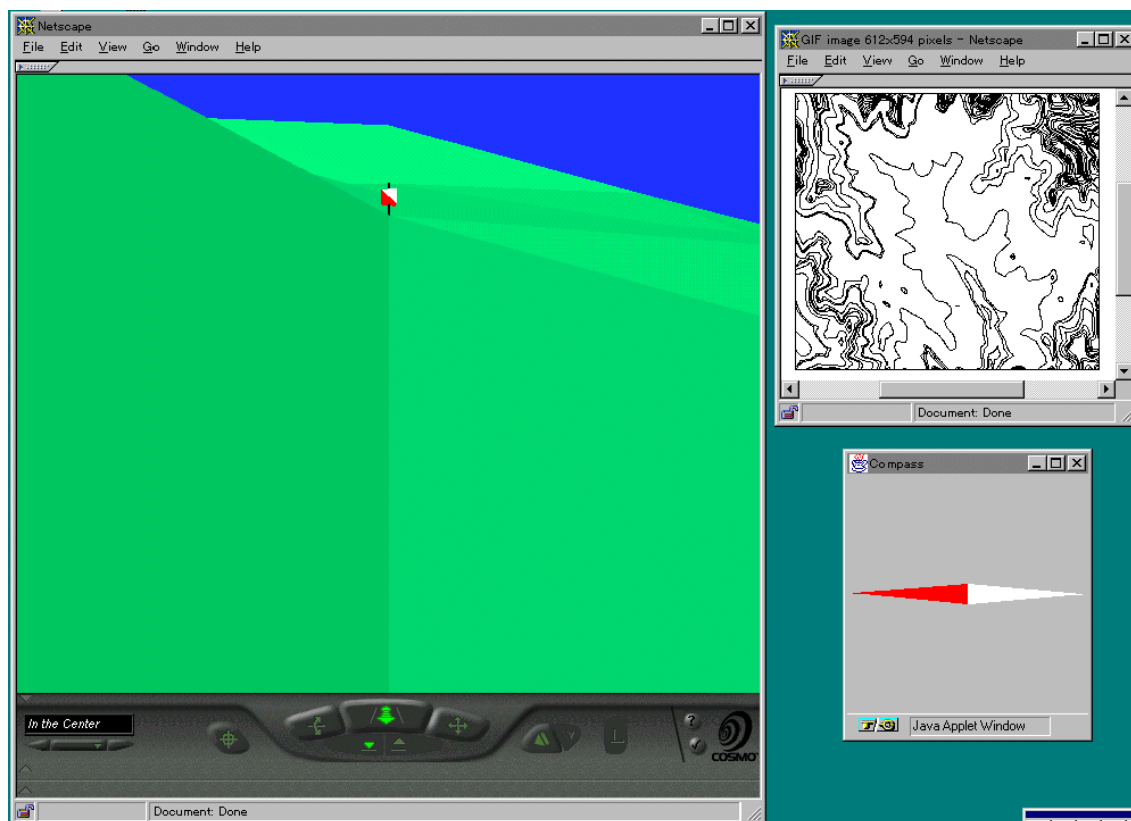


図 11: 実行例：東向き

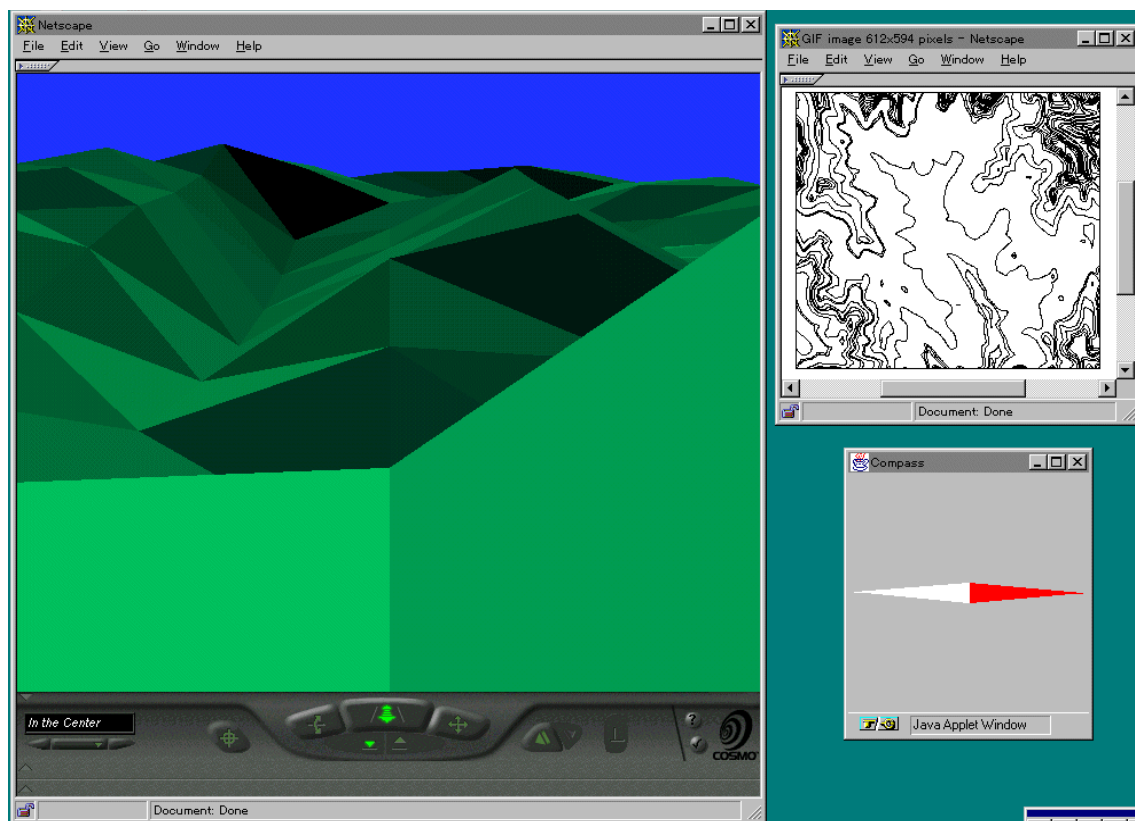


図 12: 実行例：西向き

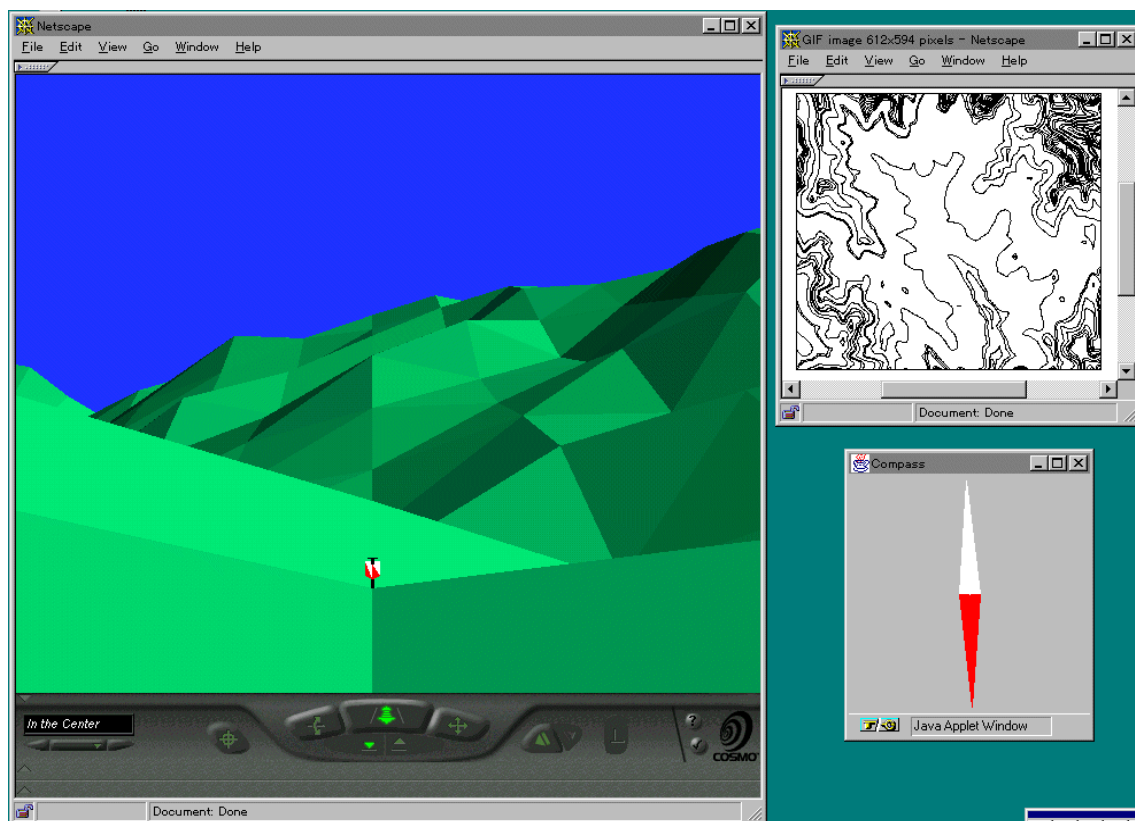


図 13: 実行例：南向き

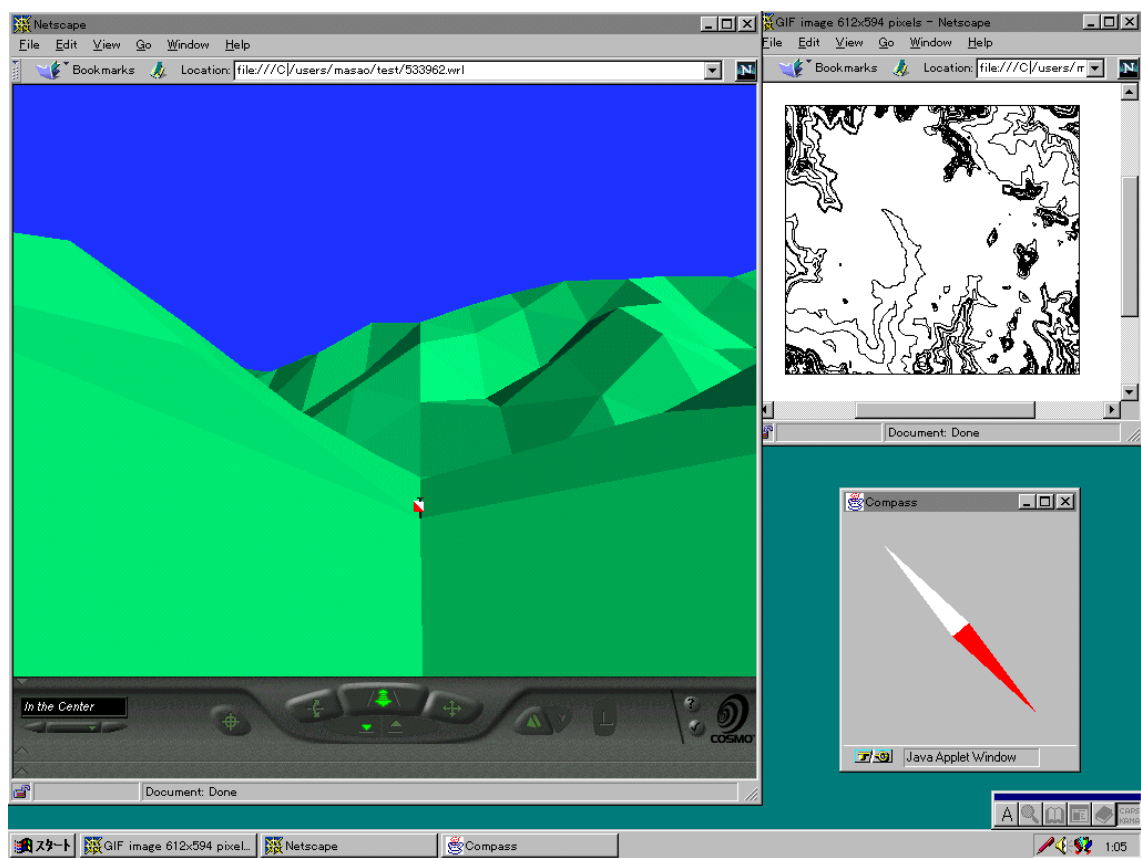


図 14: 実行例：下りのコントロールフラッグ

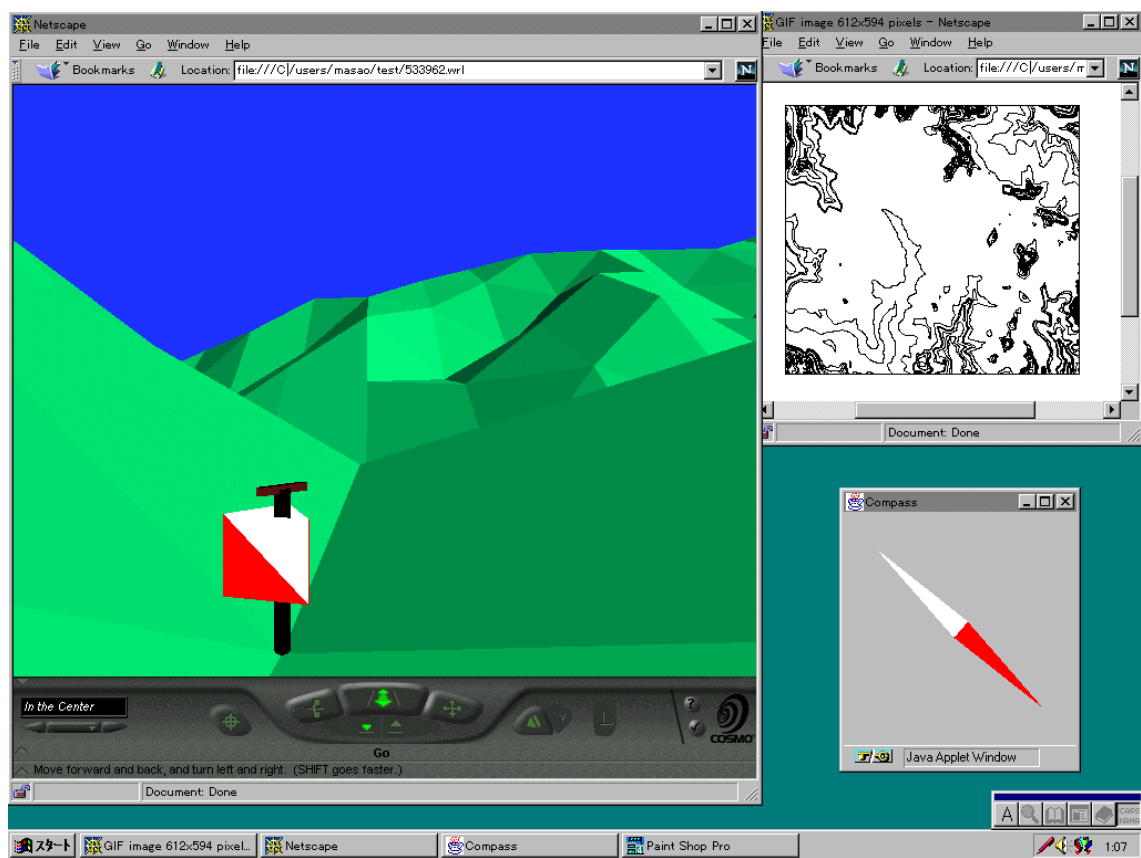


図 15: 実行例：下りのコントロールフラッグ 2

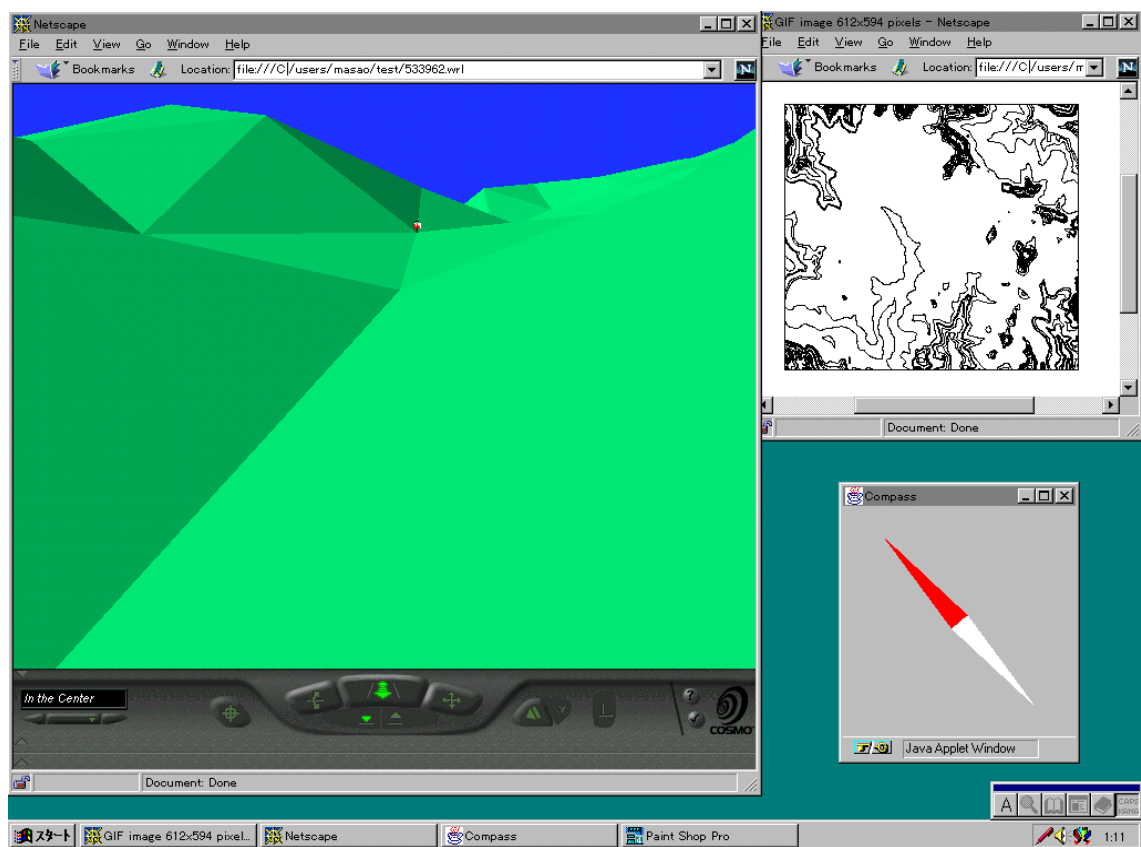


図 16: 実行例：登りのコントロールフラッグ

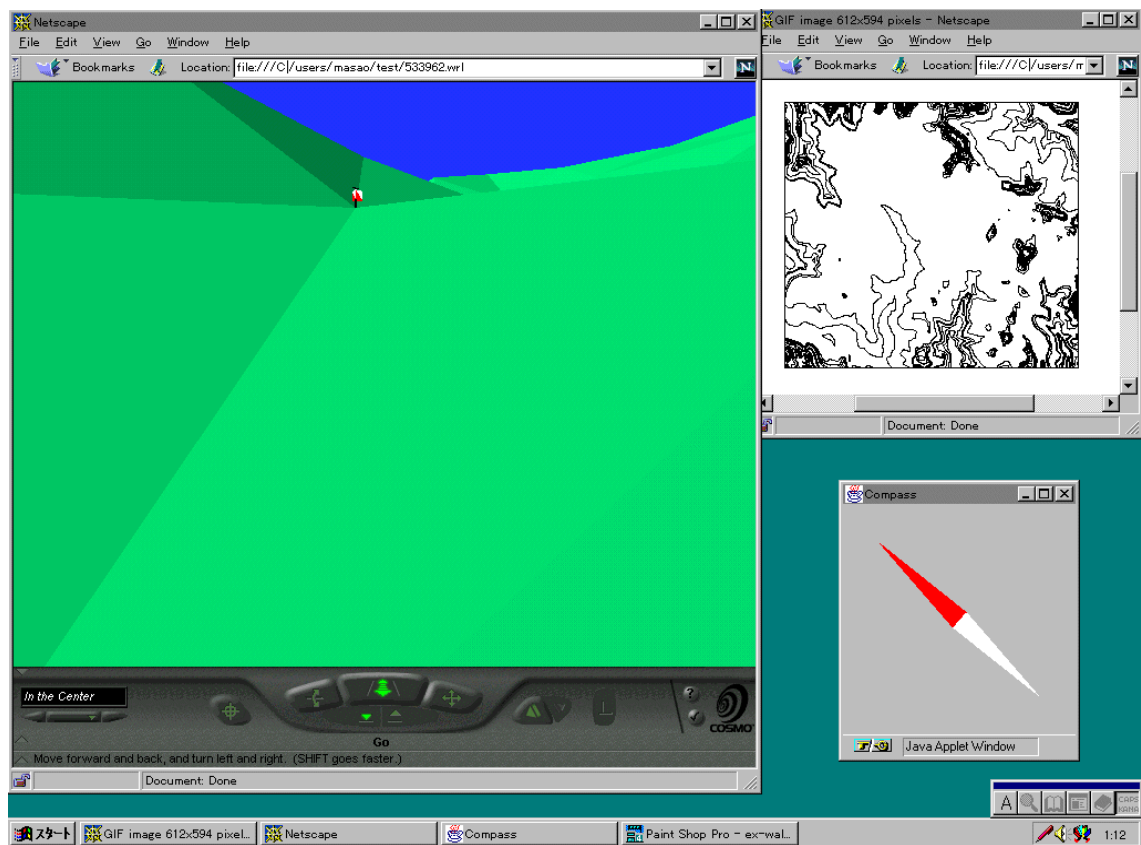


図 17: 実行例：登りのコントロールフラッグ 2

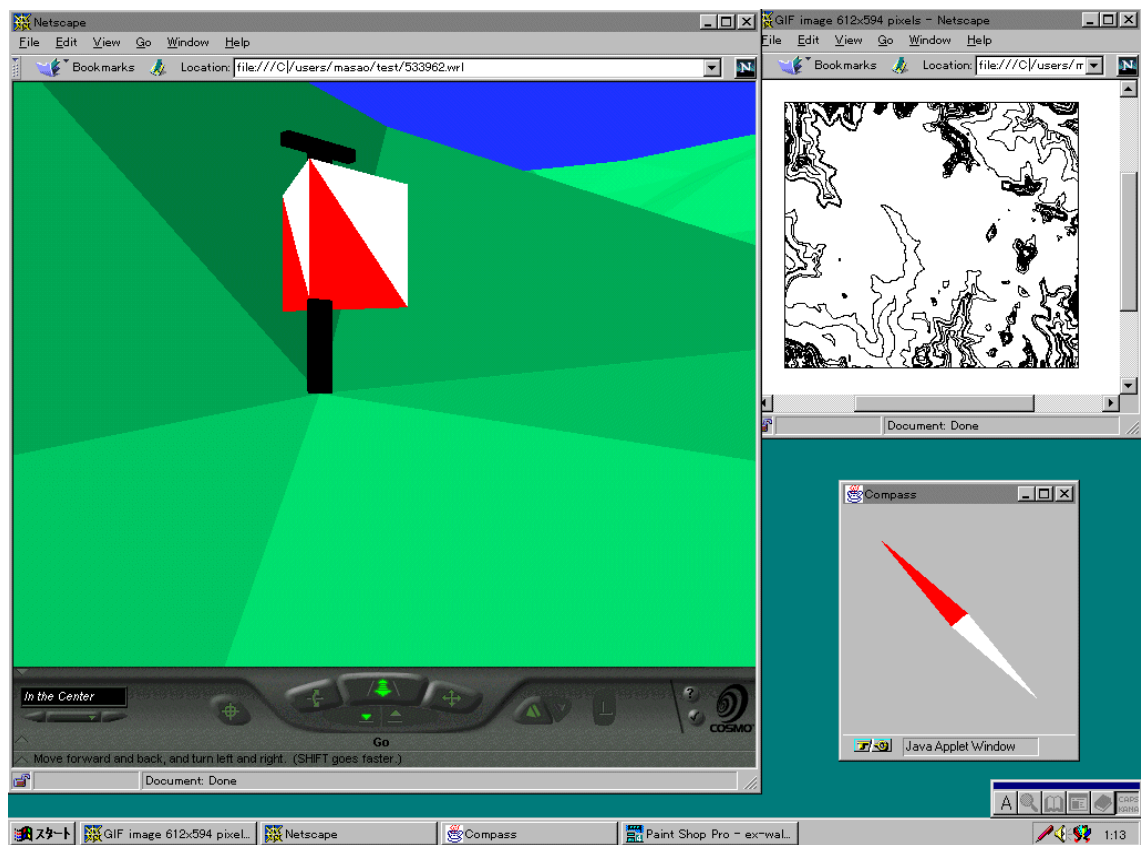


図 18: 実行例：登りのコントロールフラッグ 3

7 考察

7.1 数値地図の選択について

本研究で国土地理院の数値地図（50m メッシュ標高データ）を選んだ理由は、

- オリエンテーリングにおける地形の起伏の違いによって、競技の性質が大きく変化する点
例えば、起伏の激しい日本のような地形では、地図を読み良いルートを選択する技術が重要だが、起伏の少ない地形では直進・スピードコントロールの技術が重要となる。
- 類似のシステムが欧米の地形（テライン）では既にいくつか開発されている [2][3] が、日本の地形を元にしたシステムは確認できなかった点

の2点である。実際に USGS の DEM データを用いても地形の表現を行なった [13] が、日本の数値地図データとは大きく異なり、地形の起伏がほとんどなく、地形と等高線による地図の対照とその読みとり技術のレベルが学習という目的を適切に果たせないと判断したので、日本の地形を表現できる国土地理院の数値地図 50m メッシュを採用した。

7.2 開発手法について

VRML2.0 と Java を関係させたシステムを開発したことの利点について、他の手法に対する検討を交えて考察する。

本システムの開発の方針は、地形と地図という3次元表現、2次元表現の対応をとることができ、多くの人が計算機の環境に左右されずに利用できるシステムを念頭に置いた。このことから、ネットワーク（特に WWW）を通じた利用を実現する手法とすることとした。さらに、システム全体に拡張性を持つことを目指した。

これらの方針に沿う手法として、VRML2.0 の利点は、

- 物体の記述のみ（プログラム自体は不要 – 開発が容易）
- スクリプトを併用すればインタラクティブな記述が可能
- 3次元表現の標準言語
- ネットワーク上での利用が容易
- 非機種依存
- 拡張性

である。

一方、システムを開発する VRML2.0 以外の手法として、Java、Open GL などを検討したが、これらの手法を採用しなかった理由について解説する。

まず、Java 言語は、汎用的なオブジェクト志向プログラミング言語でネットワーク言語として、非機種依存といった特徴を備えてはいるが、3次元表現に関して標準的なサポートがない状態で本研究では現時点での使用は見合わせた。

Open GL は、ある程度の互換性を持った3次元表現が強力に行なえるが、ネットワーク上でやりとりを行なえるほどの互換性を持っていないこと、実行する環境がある程度制限されることから選択しなかった。

7.3 今後の課題

今後の課題としては、システムの機能の充実が挙げられる。特に最も基本的な地形と地図の構成のうち、現在は地形とそれに対応した地図があるが、これらが真に統合されたシステムにはなっていないので、これを統合したシステムにすることが必要と思われる。他の具体的な機能としては、オリエンテーリング競技を疑似体験するための機能として、時間を計測する機能、自分の進んだ距離を測る歩測機能、コントロールフラッグの番号付けやコース設定を行なう機能、などが考えられる。

また、3次元表現の地形と2次元表現の地図を見比べることによるオリエンテーリング競技の学習は、単に見比べるだけではなく、実際に感じとれるようになることが重要と考える。そのため、現在の等高線だけの地図表現では、地図だけを見て地形の様子を完全に認識することはなかなか難しい。この問題の解決策としては、オリエンテーリングでは地図を自分のいる地形の向きと一致させることを「正置」と呼び、最も基本的な技術と考えられているため、地図の表現を眺めるだけでなく、手にとっているかのように回して、別の角度から見ることもできることも重要な要素だと考える。また、等高線だけで判別できない地形の高低は、水系や植生などといった、現在は表現していない要素を加えることにより、認識が容易になると思われる。

本研究で開発したシステムの発展形としては、多人数参加型のオリエンテーリング競技をネットワーク上で行なうことが考えられる。実際に VRML2.0 による多人数参加型の共有的な仮想世界の作成はすでにいくつか試みられている。

謝辞

本研究全般にわたり、常に力強い励ましとご指導をいただきました石塚英弘先生、宇陀則彦先生に感謝致します。本研究のアイデアについて、オリエンティアの視点から鋭いアドバイスをいただいた江草由佳さん、斉藤ひとみさんに感謝します。また、石塚研究室の皆様をはじめとして、研究を通じてお世話になった全ての方に感謝致します。

参考文献

- [1] 村越真. 村越真の実践オリエンテーリング講座. 東京, 日本オリエンテーリング協会, 1994, 131p.
- [2] Lawford,G. Navigate! Orienteering Simulation Software. 1998-01-26.
<URL: <http://www.ozemail.com.au/~glawford/navigate.html>>
- [3] Melin Software. Win OL. Sweden. 1997-11-14.
<URL: <http://home1.swipnet.se/~w-18592/winol/>>
- [4] 国土地理院. 数値地図 50m メッシュ (標高) 日本 -II. つくば, 1997. [CD-ROM]
- [5] U.S. Geological Survey. USGS DEM Format. 参照 1998-01-29.
<URL: http://edcwww.cr.usgs.gov/glis/hyper/guide/usgs_dem>
- [6] 日本学生オリエンテーリング連盟. 巴川源流. 愛知, 日本学生オリエンテーリング連盟, 1987.
[オリエンテーリング地図]
- [7] VRML Consortium. The VRML Consortium. 1998-01-26.
<URL: <http://www.vrml.org/>>
- [8] 安藤幸央. VRML Repository in Japan. 参照 1998-01-29.
<URL: <http://www.webcity.co.jp/info/andoh/vrml/>>
- [9] 三浦健二郎. VRML2.0 : 3D サイバースペース構築言語. 東京, 朝倉書店, 1996, 272p.
- [10] Hassinger, S. ; Erwin, M. (安藤幸央監訳). VRML 60 分ガイド. 東京, ソフトバンク, 1996, 174p.
- [11] Lea,R. 松下晃一, 宮下健. Java+VRML : Java と VRML2.0 で作るインタラクティブ 3D ワールド. 東京, プレンティスホール出版, 1997, 354p.
- [12] Flanagan,D. (永松健司訳, 日本サンマイクロシステムズ監訳). JAVA クイックリファレンス. 東京, オライリー・ジャパン, 1996, 567p.
- [13] Larimer,J. dem2wrl.c. V1.2, 1997-8-24. [プログラム]
<URL: <http://hiwaay.net/crispen/src/dem2wrl.zip>>
- [14] Williams,T. Kelley,C.(升谷保博訳). gnuplot-3.5 Manual [日本語訳]. 1993, 38p.

付録

A VRML ファイル

A.1 terrain.wrl

#VRML V2.0 utf8

```
EXTERNPROTO ControlFlag [  
    field SFCOLOR poleColor  
    field SFCOLOR punchColor  
    exposedField SFVec3f position  
] "control.wrl"
```

```
ControlFlag {  
    position 1400 1203 1500  
}
```

```
ControlFlag {  
    position 1500 1143 1400  
}
```

```
ControlFlag {  
    position 1500 1181 1450  
}
```

```
ControlFlag {  
    position 1450 1228 1500  
}
```

```
ControlFlag {  
    position 1550 1246 1500  
}
```

```
ControlFlag {  
    position 1500 1278 1550  
}
```

```
ControlFlag {  
    position 1550 1278 1550  
}
```

```
ControlFlag {  
    position 1450 1178 1450  
}
```

```
ControlFlag {  
    position 0 871 50  
}
```

```
ControlFlag {  
    position 50 925 0
```

```

}
ControlFlag {
    position 50 887 50
}

Transform {
    children [
        WorldInfo {
            title "
        },
        DirectionalLight {
            direction 1 -1 0
        },
        Background {
            skyColor 0.1 0.2 1
        },
        Viewpoint {
            description "In the Center"
            position 1500 1235 1500
        },
        Viewpoint {
            description "Start"
            orientation 0 -1 0 2.3561945
            position 0 872 0
        },
        Viewpoint {
            description "Over Head"
            orientation -1 0 0 1.5707963
            position 1500 1889 1500
        },
        NavigationInfo {
            avatarSize [0.25, 1.5, 5.0]
            type "WALK"
            speed 1.0
        },
        Transform {
            children [
                Shape {
                    appearance Appearance {
                        material Material {
                            diffuseColor 0 1 0.5
                        }
                    }
                },
            ],
        }
    ]
}

```

```

geometry ElevationGrid {
  xDimension 60
  zDimension 60
  xSpacing 50
  zSpacing 50
  height [
    871, 925, 963, 993, 1041, 1087, 1096, 1132, 1179, 1226,
    1276, 1316, 1348, 1382, 1406, 1419, 1418, 1423, 1422, 1414,
    1410, 1411, 1404, 1392, 1411, 1406, 1426, 1481, 1527, 1541,
    1551, 1575, 1560, 1522, 1518, 1555, 1571, 1612, 1614, 1644,
    1658, 1697, 1729, 1752, 1757, 1748, 1749, 1794, 1809, 1838,
    1867, 1863, 1865, 1866, 1835, 1800, 1769, 1735, 1694, 1646,
    871, 887, 922, 979, 1026, 1055, 1097, 1151, 1183, 1217,
    1253, 1290, 1333, 1356, 1393, 1380, 1377, 1381, 1384, 1382,
    1371, 1369, 1369, 1365, 1374, 1393, 1446, 1489, 1512, 1508,
    1535, 1558, 1577, 1568, 1562, 1595, 1602, 1632, 1642, 1678,
    1694, 1713, 1728, 1740, 1758, 1779, 1785, 1811, 1848, 1856,
    1850, 1830, 1825, 1836, 1834, 1792, 1746, 1719, 1684, 1647,
    926, 930, 960, 1024, 1037, 1056, 1096, 1134, 1165, 1199,
    1235, 1277, 1312, 1352, 1382, 1352, 1329, 1329, 1338, 1342,
    1334, 1329, 1327, 1336, 1334, 1384, 1421, 1456, 1476, 1479,
    1505, 1536, 1556, 1585, 1605, 1617, 1634, 1654, 1669, 1672,
    1656, 1681, 1673, 1682, 1721, 1747, 1796, 1824, 1824, 1806,
    1803, 1788, 1782, 1814, 1824, 1783, 1746, 1718, 1701, 1694,
    972, 973, 1010, 1057, 1085, 1092, 1102, 1133, 1156, 1186,
    1225, 1264, 1313, 1363, 1358, 1327, 1296, 1297, 1303, 1306,
    1299, 1298, 1293, 1303, 1330, 1363, 1392, 1431, 1436, 1453,
    1484, 1509, 1531, 1566, 1584, 1590, 1614, 1637, 1627, 1623,
    1619, 1614, 1608, 1640, 1677, 1723, 1794, 1788, 1785, 1761,
    1764, 1754, 1745, 1779, 1808, 1805, 1783, 1750, 1733, 1729,
    1007, 1016, 1040, 1075, 1115, 1136, 1144, 1161, 1190, 1215,
    1246, 1279, 1319, 1343, 1341, 1307, 1273, 1266, 1274, 1280,
    1271, 1269, 1263, 1287, 1318, 1340, 1374, 1395, 1404, 1443,
    1465, 1479, 1507, 1540, 1550, 1550, 1565, 1593, 1570, 1564,
    1564, 1562, 1577, 1616, 1671, 1719, 1772, 1747, 1746, 1726,
    1734, 1723, 1723, 1748, 1792, 1810, 1806, 1786, 1769, 1762,
    1030, 1046, 1063, 1100, 1143, 1169, 1180, 1195, 1218, 1258,
    1288, 1311, 1319, 1319, 1318, 1288, 1249, 1236, 1242, 1248,
    1243, 1237, 1236, 1268, 1297, 1326, 1348, 1363, 1382, 1423,
    1436, 1456, 1484, 1510, 1516, 1502, 1515, 1499, 1517, 1525,
    1513, 1527, 1555, 1604, 1661, 1712, 1736, 1721, 1699, 1691,
    1700, 1694, 1704, 1739, 1765, 1783, 1797, 1805, 1804, 1781,
    1049, 1076, 1090, 1122, 1161, 1192, 1211, 1226, 1241, 1252,

```

1268, 1287, 1290, 1289, 1288, 1272, 1229, 1208, 1215, 1218,
 1213, 1211, 1225, 1247, 1275, 1302, 1325, 1336, 1365, 1395,
 1416, 1439, 1462, 1482, 1466, 1470, 1452, 1448, 1489, 1487,
 1476, 1504, 1542, 1594, 1649, 1681, 1696, 1682, 1656, 1650,
 1673, 1661, 1688, 1719, 1741, 1756, 1777, 1790, 1810, 1799,
 1061, 1087, 1115, 1126, 1147, 1172, 1189, 1203, 1211, 1225,
 1243, 1258, 1262, 1258, 1257, 1244, 1207, 1178, 1188, 1190,
 1185, 1187, 1211, 1231, 1254, 1283, 1294, 1316, 1343, 1368,
 1394, 1417, 1443, 1448, 1430, 1449, 1408, 1424, 1458, 1440,
 1443, 1490, 1541, 1588, 1613, 1644, 1666, 1650, 1619, 1620,
 1635, 1622, 1659, 1694, 1724, 1742, 1762, 1776, 1793, 1816,
 1063, 1085, 1096, 1093, 1118, 1139, 1154, 1171, 1180, 1198,
 1221, 1230, 1226, 1222, 1225, 1213, 1186, 1147, 1157, 1159,
 1157, 1169, 1187, 1219, 1237, 1256, 1269, 1300, 1328, 1353,
 1377, 1401, 1423, 1420, 1393, 1408, 1378, 1397, 1412, 1395,
 1442, 1489, 1541, 1580, 1577, 1596, 1605, 1615, 1589, 1582,
 1604, 1594, 1640, 1680, 1706, 1731, 1746, 1763, 1782, 1805,
 1058, 1078, 1081, 1070, 1087, 1110, 1119, 1138, 1148, 1177,
 1199, 1207, 1194, 1187, 1193, 1180, 1156, 1127, 1125, 1128,
 1129, 1146, 1175, 1196, 1224, 1231, 1254, 1276, 1300, 1332,
 1351, 1374, 1391, 1386, 1363, 1363, 1351, 1349, 1366, 1389,
 1435, 1474, 1505, 1545, 1541, 1555, 1556, 1555, 1560, 1547,
 1565, 1571, 1623, 1648, 1686, 1711, 1740, 1760, 1775, 1794,
 1043, 1054, 1061, 1045, 1058, 1076, 1088, 1104, 1124, 1155,
 1181, 1175, 1165, 1160, 1161, 1152, 1133, 1105, 1099, 1105,
 1114, 1127, 1158, 1186, 1208, 1215, 1230, 1248, 1271, 1291,
 1306, 1321, 1340, 1340, 1319, 1312, 1310, 1299, 1321, 1367,
 1411, 1444, 1483, 1519, 1507, 1519, 1506, 1515, 1528, 1518,
 1525, 1564, 1600, 1632, 1663, 1694, 1731, 1757, 1773, 1792,
 1019, 1027, 1028, 1021, 1034, 1047, 1067, 1076, 1098, 1139,
 1157, 1144, 1131, 1125, 1132, 1119, 1103, 1083, 1074, 1080,
 1093, 1110, 1144, 1171, 1182, 1196, 1202, 1221, 1241, 1253,
 1256, 1266, 1300, 1294, 1288, 1268, 1267, 1269, 1304, 1360,
 1385, 1416, 1464, 1492, 1468, 1475, 1470, 1486, 1499, 1489,
 1512, 1548, 1587, 1621, 1646, 1686, 1726, 1754, 1771, 1786,
 991, 992, 994, 995, 1012, 1027, 1039, 1053, 1077, 1115,
 1131, 1114, 1099, 1100, 1108, 1086, 1068, 1059, 1054, 1064,
 1068, 1104, 1139, 1160, 1167, 1175, 1180, 1197, 1204, 1205,
 1216, 1233, 1254, 1257, 1252, 1238, 1237, 1248, 1293, 1334,
 1357, 1395, 1429, 1465, 1440, 1429, 1438, 1446, 1472, 1458,
 1509, 1545, 1572, 1602, 1640, 1675, 1716, 1742, 1765, 1783,
 964, 963, 960, 968, 987, 1010, 1013, 1037, 1061, 1093,
 1101, 1084, 1069, 1071, 1080, 1062, 1042, 1036, 1038, 1045,

1060, 1102, 1127, 1139, 1146, 1152, 1156, 1164, 1169, 1166,
 1181, 1202, 1217, 1224, 1217, 1213, 1204, 1234, 1269, 1297,
 1327, 1368, 1406, 1427, 1407, 1408, 1414, 1402, 1427, 1462,
 1497, 1527, 1566, 1599, 1631, 1664, 1700, 1735, 1763, 1787,
 940, 936, 933, 954, 972, 978, 997, 1024, 1048, 1078,
 1083, 1059, 1045, 1045, 1053, 1038, 1026, 1018, 1019, 1024,
 1064, 1091, 1110, 1119, 1120, 1122, 1130, 1129, 1128, 1125,
 1141, 1167, 1184, 1186, 1179, 1178, 1183, 1215, 1235, 1261,
 1306, 1349, 1384, 1390, 1375, 1373, 1366, 1387, 1434, 1477,
 1500, 1536, 1576, 1618, 1648, 1679, 1712, 1730, 1755, 1794,
 915, 917, 906, 931, 956, 951, 983, 1007, 1035, 1062,
 1060, 1034, 1028, 1023, 1026, 1019, 1011, 994, 997, 1023,
 1054, 1076, 1073, 1088, 1083, 1077, 1089, 1076, 1084, 1087,
 1110, 1135, 1147, 1154, 1151, 1152, 1161, 1175, 1200, 1244,
 1284, 1335, 1357, 1353, 1341, 1335, 1385, 1436, 1448, 1498,
 1528, 1545, 1574, 1625, 1666, 1702, 1726, 1733, 1745, 1790,
 899, 893, 886, 911, 931, 932, 967, 991, 1022, 1047,
 1039, 1017, 1010, 998, 1003, 993, 975, 949, 981, 1012,
 1040, 1051, 1022, 1043, 1047, 1042, 1049, 1040, 1053, 1060,
 1085, 1099, 1112, 1118, 1123, 1124, 1133, 1153, 1187, 1230,
 1270, 1319, 1326, 1301, 1311, 1353, 1413, 1451, 1485, 1505,
 1533, 1570, 1595, 1631, 1673, 1711, 1738, 1757, 1765, 1791,
 873, 869, 866, 885, 896, 925, 950, 984, 1013, 1032,
 1024, 998, 985, 982, 975, 964, 938, 930, 963, 987,
 1009, 1012, 988, 1003, 1010, 1012, 1007, 1002, 1024, 1035,
 1053, 1067, 1079, 1084, 1093, 1094, 1101, 1133, 1176, 1219,
 1260, 1299, 1295, 1287, 1333, 1354, 1397, 1430, 1486, 1529,
 1548, 1575, 1609, 1639, 1683, 1717, 1738, 1766, 1786, 1809,
 843, 841, 840, 847, 877, 909, 943, 970, 999, 1013,
 1012, 978, 963, 958, 961, 931, 917, 902, 937, 955,
 970, 968, 947, 973, 977, 980, 969, 977, 992, 1006,
 1025, 1034, 1044, 1046, 1064, 1071, 1084, 1118, 1160, 1207,
 1245, 1264, 1260, 1284, 1343, 1392, 1414, 1436, 1463, 1517,
 1549, 1588, 1627, 1652, 1696, 1730, 1746, 1764, 1788, 1813,
 803, 800, 810, 836, 868, 900, 930, 953, 976, 990,
 993, 958, 941, 939, 945, 926, 880, 878, 898, 919,
 927, 932, 920, 940, 944, 945, 929, 946, 956, 982,
 994, 992, 1006, 1023, 1037, 1046, 1065, 1101, 1146, 1192,
 1199, 1220, 1233, 1285, 1334, 1395, 1453, 1476, 1498, 1524,
 1546, 1590, 1642, 1679, 1714, 1742, 1767, 1786, 1802, 1830,
 770, 763, 791, 825, 858, 881, 908, 940, 959, 967,
 963, 941, 921, 922, 929, 909, 850, 846, 867, 883,
 894, 897, 892, 911, 912, 904, 894, 909, 927, 943,

946, 960, 982, 1006, 1029, 1061, 1092, 1104, 1121, 1146,
 1172, 1197, 1252, 1302, 1340, 1397, 1447, 1495, 1543, 1572,
 1593, 1598, 1644, 1689, 1729, 1764, 1790, 1810, 1833, 1847,
 736, 738, 782, 815, 846, 876, 907, 924, 939, 938,
 933, 912, 894, 901, 902, 879, 822, 817, 833, 849,
 859, 862, 864, 869, 876, 864, 866, 883, 898, 918,
 954, 977, 1002, 1021, 1047, 1089, 1131, 1151, 1162, 1206,
 1222, 1243, 1261, 1324, 1368, 1397, 1424, 1482, 1531, 1581,
 1618, 1641, 1643, 1703, 1747, 1787, 1817, 1843, 1864, 1884,
 702, 736, 773, 805, 836, 865, 884, 904, 916, 897,
 893, 879, 872, 861, 848, 825, 794, 792, 806, 823,
 822, 831, 831, 839, 845, 849, 876, 909, 924, 947,
 975, 1001, 1024, 1039, 1069, 1107, 1148, 1186, 1196, 1232,
 1272, 1304, 1295, 1321, 1370, 1423, 1466, 1500, 1531, 1564,
 1609, 1670, 1670, 1709, 1750, 1803, 1845, 1871, 1889, 1855,
 680, 725, 764, 793, 824, 845, 869, 885, 881, 857,
 860, 844, 835, 817, 802, 789, 759, 771, 778, 791,
 792, 799, 807, 815, 824, 861, 891, 927, 949, 971,
 1002, 1026, 1045, 1066, 1090, 1141, 1179, 1212, 1223, 1257,
 1301, 1337, 1339, 1339, 1367, 1422, 1478, 1532, 1560, 1595,
 1640, 1690, 1719, 1730, 1767, 1816, 1855, 1886, 1868, 1819,
 662, 714, 751, 780, 807, 833, 852, 847, 840, 818,
 805, 798, 784, 770, 758, 748, 737, 739, 747, 758,
 764, 776, 794, 805, 841, 880, 914, 955, 982, 1003,
 1030, 1050, 1078, 1098, 1110, 1150, 1200, 1231, 1253, 1293,
 1331, 1360, 1375, 1377, 1391, 1424, 1481, 1534, 1580, 1618,
 1666, 1713, 1754, 1765, 1791, 1839, 1875, 1882, 1850, 1797,
 653, 700, 739, 766, 796, 821, 824, 808, 797, 775,
 756, 747, 745, 734, 721, 715, 713, 722, 738, 764,
 788, 795, 810, 845, 878, 902, 941, 990, 1019, 1031,
 1054, 1078, 1115, 1135, 1146, 1167, 1206, 1252, 1301, 1324,
 1352, 1381, 1406, 1417, 1417, 1447, 1485, 1539, 1588, 1620,
 1675, 1717, 1768, 1804, 1817, 1852, 1870, 1854, 1833, 1789,
 644, 688, 732, 765, 787, 808, 801, 767, 749, 739,
 721, 704, 704, 692, 683, 696, 726, 750, 765, 791,
 819, 836, 866, 894, 911, 950, 1001, 1032, 1054, 1061,
 1082, 1117, 1141, 1166, 1177, 1195, 1230, 1248, 1310, 1353,
 1378, 1400, 1433, 1463, 1455, 1486, 1504, 1546, 1595, 1644,
 1692, 1741, 1781, 1821, 1840, 1845, 1837, 1818, 1802, 1761,
 634, 677, 721, 766, 776, 785, 763, 729, 714, 703,
 683, 671, 675, 665, 693, 722, 752, 775, 804, 832,
 840, 883, 920, 941, 957, 1005, 1044, 1069, 1088, 1088,
 1114, 1150, 1185, 1205, 1229, 1230, 1271, 1289, 1315, 1368,

1397, 1421, 1451, 1492, 1497, 1513, 1555, 1572, 1595, 1646,
 1701, 1751, 1799, 1818, 1823, 1826, 1802, 1781, 1772, 1748,
 628, 669, 711, 750, 741, 748, 730, 698, 683, 668,
 656, 648, 652, 676, 709, 740, 772, 811, 835, 872,
 893, 892, 951, 991, 1005, 1032, 1081, 1115, 1142, 1134,
 1143, 1173, 1215, 1238, 1271, 1285, 1306, 1337, 1327, 1385,
 1419, 1447, 1476, 1515, 1543, 1540, 1587, 1618, 1631, 1663,
 1708, 1757, 1790, 1793, 1797, 1807, 1784, 1742, 1731, 1729,
 628, 673, 721, 716, 702, 705, 691, 668, 655, 641,
 630, 636, 666, 693, 731, 767, 796, 834, 870, 907,
 939, 937, 955, 1004, 1044, 1068, 1106, 1152, 1177, 1178,
 1181, 1217, 1242, 1272, 1304, 1331, 1345, 1378, 1363, 1404,
 1453, 1481, 1498, 1538, 1580, 1588, 1609, 1651, 1675, 1701,
 1722, 1747, 1761, 1764, 1769, 1778, 1766, 1721, 1688, 1692,
 636, 688, 709, 676, 658, 659, 656, 641, 617, 609,
 622, 649, 692, 719, 763, 809, 815, 853, 898, 940,
 973, 979, 1003, 1019, 1061, 1094, 1140, 1174, 1203, 1228,
 1234, 1246, 1293, 1319, 1339, 1363, 1378, 1403, 1398, 1413,
 1473, 1508, 1538, 1566, 1611, 1633, 1645, 1669, 1695, 1693,
 1699, 1724, 1729, 1731, 1736, 1740, 1739, 1712, 1681, 1650,
 650, 693, 681, 648, 617, 608, 611, 611, 592, 608,
 645, 696, 728, 752, 791, 833, 863, 875, 913, 961,
 999, 1020, 1042, 1053, 1076, 1117, 1161, 1190, 1222, 1261,
 1278, 1278, 1317, 1350, 1369, 1395, 1411, 1440, 1453, 1453,
 1494, 1526, 1570, 1599, 1621, 1642, 1661, 1660, 1666, 1665,
 1671, 1693, 1699, 1701, 1708, 1716, 1706, 1678, 1659, 1599,
 654, 684, 660, 618, 586, 579, 585, 584, 597, 631,
 685, 732, 770, 810, 818, 847, 890, 917, 937, 976,
 1013, 1038, 1075, 1098, 1111, 1138, 1172, 1207, 1242, 1277,
 1304, 1314, 1350, 1376, 1401, 1415, 1437, 1462, 1479, 1495,
 1512, 1536, 1560, 1581, 1597, 1622, 1634, 1632, 1634, 1637,
 1643, 1662, 1668, 1678, 1683, 1697, 1684, 1646, 1642, 1587,
 647, 668, 634, 591, 569, 565, 572, 596, 623, 656,
 695, 745, 784, 818, 857, 874, 903, 940, 967, 996,
 1023, 1048, 1090, 1121, 1150, 1172, 1199, 1228, 1264, 1279,
 1310, 1331, 1351, 1367, 1377, 1395, 1413, 1444, 1463, 1468,
 1491, 1513, 1530, 1557, 1584, 1595, 1598, 1600, 1603, 1610,
 1617, 1632, 1644, 1653, 1667, 1682, 1651, 1616, 1595, 1562,
 643, 647, 598, 565, 554, 566, 604, 623, 653, 682,
 720, 760, 800, 830, 863, 900, 922, 957, 985, 1017,
 1037, 1058, 1098, 1131, 1164, 1187, 1212, 1229, 1245, 1261,
 1283, 1308, 1326, 1339, 1349, 1370, 1393, 1422, 1437, 1438,
 1463, 1483, 1504, 1539, 1563, 1564, 1566, 1569, 1576, 1583,

1591, 1606, 1625, 1645, 1661, 1650, 1642, 1593, 1561, 1541,
 631, 604, 568, 545, 551, 575, 617, 658, 682, 712,
 736, 773, 805, 840, 872, 905, 947, 970, 997, 1025,
 1051, 1078, 1106, 1131, 1155, 1174, 1191, 1200, 1211, 1230,
 1252, 1279, 1295, 1306, 1321, 1346, 1377, 1398, 1406, 1405,
 1436, 1450, 1487, 1519, 1531, 1531, 1536, 1541, 1547, 1561,
 1576, 1599, 1629, 1630, 1626, 1629, 1600, 1586, 1536, 1515,
 602, 565, 539, 543, 557, 583, 620, 673, 711, 728,
 758, 790, 826, 855, 892, 919, 957, 981, 1009, 1033,
 1050, 1067, 1089, 1109, 1131, 1156, 1163, 1172, 1183, 1198,
 1232, 1256, 1272, 1283, 1296, 1325, 1358, 1370, 1372, 1381,
 1402, 1428, 1466, 1498, 1500, 1501, 1506, 1516, 1533, 1557,
 1588, 1610, 1602, 1598, 1593, 1600, 1554, 1544, 1527, 1490,
 577, 538, 531, 547, 566, 595, 636, 687, 723, 751,
 776, 814, 852, 880, 908, 942, 963, 971, 992, 1004,
 1024, 1042, 1061, 1083, 1104, 1126, 1137, 1144, 1157, 1177,
 1204, 1228, 1255, 1263, 1279, 1308, 1333, 1334, 1335, 1350,
 1376, 1409, 1453, 1465, 1470, 1478, 1495, 1524, 1542, 1567,
 1582, 1578, 1577, 1568, 1566, 1582, 1532, 1492, 1494, 1481,
 553, 523, 534, 547, 572, 608, 640, 681, 719, 755,
 792, 831, 868, 902, 924, 942, 936, 935, 957, 968,
 995, 1020, 1039, 1055, 1080, 1099, 1110, 1122, 1133, 1160,
 1185, 1208, 1234, 1239, 1262, 1291, 1309, 1305, 1303, 1324,
 1352, 1396, 1430, 1438, 1447, 1482, 1504, 1511, 1526, 1543,
 1549, 1548, 1546, 1539, 1547, 1560, 1539, 1490, 1450, 1445,
 537, 514, 527, 546, 577, 614, 648, 684, 719, 757,
 797, 847, 881, 903, 913, 919, 898, 902, 922, 931,
 959, 995, 1013, 1032, 1055, 1068, 1078, 1092, 1121, 1154,
 1161, 1182, 1211, 1218, 1241, 1268, 1279, 1273, 1282, 1320,
 1353, 1374, 1407, 1436, 1451, 1467, 1477, 1477, 1490, 1512,
 1521, 1519, 1516, 1509, 1532, 1549, 1533, 1478, 1432, 1406,
 533, 508, 519, 539, 575, 616, 653, 687, 721, 764,
 809, 836, 860, 881, 880, 878, 863, 871, 879, 896,
 920, 957, 977, 1000, 1023, 1035, 1044, 1073, 1111, 1129,
 1128, 1162, 1185, 1192, 1224, 1242, 1251, 1242, 1288, 1328,
 1367, 1393, 1413, 1417, 1419, 1435, 1444, 1444, 1459, 1481,
 1495, 1492, 1488, 1490, 1518, 1547, 1516, 1477, 1429, 1372,
 533, 507, 507, 534, 580, 619, 665, 704, 738, 773,
 794, 808, 823, 852, 841, 835, 827, 839, 844, 863,
 886, 921, 950, 964, 984, 995, 1016, 1060, 1085, 1094,
 1114, 1137, 1155, 1164, 1190, 1201, 1213, 1244, 1284, 1329,
 1361, 1377, 1386, 1379, 1385, 1405, 1407, 1409, 1426, 1450,
 1466, 1467, 1466, 1494, 1532, 1542, 1498, 1467, 1432, 1385,

523, 510, 491, 537, 578, 622, 671, 712, 736, 745,
 753, 771, 797, 820, 803, 793, 795, 801, 810, 824,
 852, 884, 907, 923, 947, 957, 1002, 1030, 1050, 1064,
 1093, 1111, 1121, 1134, 1148, 1156, 1206, 1256, 1296, 1328,
 1340, 1347, 1354, 1347, 1352, 1368, 1366, 1402, 1414, 1424,
 1435, 1453, 1476, 1512, 1537, 1526, 1484, 1439, 1402, 1377,
 496, 486, 501, 542, 586, 628, 672, 692, 701, 704,
 704, 728, 765, 784, 767, 752, 764, 762, 773, 789,
 812, 845, 874, 884, 906, 927, 975, 1000, 1018, 1046,
 1073, 1083, 1092, 1105, 1142, 1184, 1226, 1257, 1289, 1303,
 1311, 1318, 1318, 1312, 1324, 1350, 1382, 1408, 1423, 1440,
 1454, 1458, 1477, 1502, 1516, 1516, 1475, 1448, 1415, 1406,
 473, 479, 520, 558, 603, 638, 658, 662, 656, 661,
 663, 687, 728, 741, 730, 710, 719, 729, 741, 755,
 774, 807, 830, 844, 871, 892, 935, 968, 1003, 1031,
 1045, 1051, 1055, 1086, 1144, 1186, 1204, 1224, 1247, 1264,
 1280, 1284, 1284, 1297, 1344, 1354, 1364, 1379, 1393, 1412,
 1428, 1442, 1450, 1468, 1488, 1510, 1503, 1457, 1433, 1396,
 483, 506, 546, 588, 604, 618, 628, 634, 615, 624,
 626, 658, 684, 696, 701, 667, 674, 693, 706, 723,
 746, 764, 792, 801, 835, 848, 886, 944, 992, 1006,
 1010, 1020, 1038, 1094, 1131, 1156, 1164, 1188, 1213, 1233,
 1242, 1247, 1257, 1302, 1315, 1318, 1332, 1344, 1364, 1382,
 1392, 1399, 1430, 1429, 1452, 1504, 1511, 1478, 1442, 1411,
 502, 532, 563, 556, 560, 578, 595, 590, 570, 575,
 595, 625, 641, 654, 648, 635, 646, 658, 686, 701,
 713, 726, 750, 770, 791, 820, 873, 932, 964, 972,
 976, 1014, 1048, 1072, 1100, 1113, 1126, 1153, 1177, 1198,
 1224, 1261, 1278, 1281, 1279, 1282, 1296, 1306, 1325, 1347,
 1361, 1385, 1386, 1397, 1423, 1470, 1505, 1504, 1463, 1445,
 506, 541, 549, 511, 505, 534, 563, 561, 554, 565,
 569, 589, 610, 614, 641, 704, 731, 738, 759, 779,
 788, 777, 783, 788, 783, 804, 857, 911, 928, 934,
 980, 1008, 1016, 1038, 1064, 1090, 1131, 1165, 1192, 1219,
 1241, 1250, 1249, 1247, 1247, 1248, 1257, 1269, 1293, 1314,
 1334, 1339, 1344, 1370, 1421, 1435, 1489, 1522, 1499, 1462,
 493, 506, 507, 489, 521, 530, 534, 539, 568, 620,
 611, 601, 597, 617, 656, 714, 755, 781, 811, 847,
 836, 818, 825, 830, 817, 787, 832, 861, 901, 942,
 971, 972, 1011, 1046, 1072, 1109, 1148, 1173, 1194, 1202,
 1200, 1208, 1211, 1202, 1212, 1208, 1209, 1240, 1276, 1278,
 1286, 1275, 1316, 1347, 1384, 1424, 1470, 1518, 1516, 1485,
 468, 465, 480, 501, 547, 584, 544, 573, 621, 649,

670, 657, 641, 648, 661, 713, 759, 805, 845, 874,
 878, 865, 864, 872, 857, 829, 808, 838, 896, 930,
 941, 979, 1022, 1063, 1092, 1115, 1126, 1149, 1161, 1167,
 1167, 1173, 1175, 1167, 1167, 1159, 1206, 1236, 1234, 1215,
 1224, 1284, 1329, 1362, 1402, 1429, 1466, 1511, 1541, 1493,
 496, 499, 512, 533, 570, 613, 545, 602, 646, 696,
 719, 716, 694, 695, 709, 710, 759, 810, 859, 897,
 920, 910, 908, 910, 900, 882, 847, 840, 897, 923,
 955, 990, 1034, 1064, 1079, 1099, 1102, 1114, 1126, 1136,
 1132, 1137, 1139, 1134, 1132, 1158, 1172, 1181, 1178, 1196,
 1264, 1303, 1345, 1364, 1404, 1453, 1501, 1549, 1552, 1513,
 539, 542, 547, 573, 603, 634, 590, 603, 652, 704,
 741, 760, 742, 737, 754, 738, 757, 799, 854, 899,
 945, 950, 956, 954, 941, 915, 855, 851, 907, 937,
 965, 999, 1034, 1049, 1065, 1075, 1079, 1086, 1095, 1099,
 1095, 1095, 1092, 1093, 1094, 1117, 1134, 1153, 1186, 1217,
 1242, 1306, 1363, 1402, 1430, 1469, 1521, 1570, 1566, 1551,
 579, 591, 580, 606, 647, 656, 613, 612, 658, 711,
 754, 792, 785, 778, 791, 773, 791, 832, 856, 904,
 951, 986, 1001, 997, 971, 922, 867, 842, 885, 921,
 959, 990, 1009, 1023, 1041, 1052, 1053, 1053, 1054, 1062,
 1061, 1057, 1054, 1069, 1115, 1151, 1162, 1187, 1217, 1260,
 1285, 1289, 1341, 1387, 1448, 1499, 1549, 1581, 1558, 1542,
 621, 628, 627, 649, 683, 676, 638, 625, 679, 728,
 764, 804, 827, 826, 831, 805, 798, 844, 891, 912,
 953, 1002, 1031, 1024, 996, 960, 900, 846, 863, 887,
 924, 958, 975, 991, 1005, 1019, 1022, 1017, 1022, 1028,
 1030, 1032, 1050, 1086, 1117, 1156, 1200, 1214, 1246, 1287,
 1323, 1333, 1354, 1391, 1449, 1503, 1563, 1580, 1545, 1510,
 650, 669, 666, 675, 713, 692, 660, 629, 687, 734,
 772, 809, 841, 861, 862, 842, 813, 853, 898, 935,
 972, 1016, 1044, 1052, 1007, 965, 935, 903, 876, 858,
 892, 932, 942, 958, 971, 981, 987, 989, 992, 998,
 1004, 1033, 1072, 1098, 1145, 1175, 1197, 1253, 1278, 1301,
 1347, 1382, 1416, 1448, 1458, 1508, 1570, 1575, 1540, 1501,
 673, 711, 713, 716, 742, 708, 674, 633, 687, 730,
 769, 811, 842, 874, 893, 873, 842, 856, 893, 942,
 982, 1027, 1057, 1062, 1023, 1001, 969, 949, 927, 888,
 868, 888, 905, 928, 948, 956, 959, 963, 967, 987,
 1014, 1049, 1091, 1126, 1167, 1214, 1232, 1263, 1308, 1343,
 1371, 1415, 1445, 1477, 1498, 1525, 1580, 1572, 1533, 1493,
 709, 732, 756, 761, 766, 730, 694, 645, 687, 729,
 770, 803, 844, 879, 916, 905, 879, 878, 896, 945,

```

987, 1034, 1072, 1081, 1051, 1023, 1005, 979, 950, 900,
880, 871, 881, 913, 927, 934, 938, 957, 990, 1012,
1039, 1070, 1100, 1148, 1186, 1226, 1261, 1279, 1307, 1335,
1379, 1430, 1465, 1503, 1534, 1549, 1571, 1573, 1537, 1493,
738, 776, 800, 799, 791, 752, 707, 665, 694, 726,
763, 806, 851, 892, 930, 933, 912, 906, 934, 946,
993, 1040, 1081, 1107, 1079, 1053, 1029, 1007, 957, 945,
910, 887, 880, 891, 912, 922, 937, 971, 1014, 1042,
1061, 1083, 1110, 1141, 1177, 1203, 1227, 1255, 1282, 1326,
1376, 1430, 1461, 1470, 1505, 1515, 1517, 1563, 1545, 1503,
765, 807, 830, 834, 811, 758, 714, 672, 702, 747,
776, 815, 858, 904, 944, 963, 947, 931, 958, 986,
1009, 1046, 1089, 1129, 1111, 1084, 1056, 1030, 1004, 964,
950, 913, 887, 885, 903, 923, 958, 1002, 1018, 1031,
1051, 1052, 1080, 1116, 1144, 1173, 1203, 1244, 1289, 1336,
1372, 1414, 1433, 1447, 1465, 1463, 1494, 1532, 1547, 1520,
801, 834, 866, 861, 821, 767, 721, 685, 711, 757,
803, 840, 873, 915, 952, 991, 986, 969, 980, 1014,
1043, 1056, 1104, 1144, 1149, 1114, 1087, 1068, 1036, 1005,
981, 942, 919, 891, 894, 926, 967, 987, 989, 1000,
1017, 1034, 1059, 1100, 1143, 1195, 1234, 1270, 1303, 1316,
1341, 1380, 1392, 1407, 1434, 1428, 1459, 1497, 1520, 1524,
]
}
}
]
}
]
}
}

```

A.2 control.wrl

```

#VRML V2.0 utf8
# オリエンテーリングのポストを表現する VRML。
PROTO ControlFlag [
    field SFCOLOR poleColor 0 0 0
    field SFCOLOR punchColor .5 .1 .1
    exposedField SFVec3f position 0 0 0
] {
    Transform {
        translation IS position
        children [
            Transform {

```

```

translation 0 1 0
children [
  Shape {
    appearance Appearance {
      material Material {
        diffuseColor IS poleColor
      }
    }
    geometry Cylinder {
      height 2.0
      radius 0.1
    }
  }
]
},
Transform {
  translation 0 2 0
  children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor IS punchColor
        }
      }
      geometry Box {
        size 0.7 0.1 0.1
      }
    }
  ]
},
Transform {
  translation 0 1.2 0
  children [
    Shape {
      appearance Appearance {
      }
      geometry IndexedFaceSet {
        solid FALSE
        coord Coordinate {
          point [
            -0.5 0.5 0.5,
            0.5 0.5 0.5,
            0 0.5 -0.5,

```



```

        -0.5 -0.5  0.5,
        0.5 -0.5  0.5,
        0 -0.5 -0.5
    ]
}
coordIndex [
    0 3 4 -1
    0 4 1 -1
    1 4 5 -1
    1 5 2 -1
    2 5 3 -1
    2 3 0
]
colorPerVertex FALSE
color Color {
    color [
        1 0 0, # 赤
        1 1 1 # 白
    ]
}
colorIndex [ 0 1 0 1 0 1 ]
}
}
]
}
}
}
Background {
    skyColor .2 .3 .4
}
ControlFlag{
    poleColor 1 1 1
}

```

B Java プログラム

B.1 CompassScript.java

```
/* CompassScript.java :
 * コンパスをウインドウに表示。
 */
import java.awt.*;
import vrml.*;
import vrml.node.*;
import vrml.field.*;

public class CompassScript extends Script {
    private Frame window;
    private CompassCanvas compass;

    public void initialize() {
        window = new Frame("Compass");
        compass = new CompassCanvas();
        window.setLayout(new FlowLayout());
        window.add(compass);
        window.pack();
        window.show();
    }
    public void shutdown() {
        window.dispose();
    }
    public void processEvent(vrml.Event e) {
        if (e.getName().equals("user_orientation")) {
            float orientations[] = new float[4];
            ((ConstSFRotation) e.getValue()).getValue(orientations);
            compass.setNeedle(orientations);
        } else if (e.getName().equals("user_position")) {
            //System.out.println("User's position is "+ e.getValue());
        }
    }
}
```

B.2 CompassCanvass.java

```
/* CompassCanvas.java :
 * CompassScript から利用されるキャンバスオブジェクト。
 */
import java.awt.*;
```

```

import java.lang.*;

public class CompassCanvas extends Canvas {
    public final int size = 200;
    public final int length = 200;
    public final Point center = new Point(length/2,length/2);
    public Point north;
    public Point south;

    private float theta;

    public CompassCanvas() {
        north = new Point(100, 0);
        south = new Point(100, 200);
        theta = 0;
    }

    public void paint(Graphics g) {
        g.setColor(Color.red);
        //      System.out.println("center: "+center+", North: "+north+"South: "+south);
        Polygon p = new Polygon();
        p.addPoint(north.x, north.y);
        p.addPoint(center.x+(int)(Math.cos(theta)*10),center.y-(int)(Math.sin(theta)*10));
        p.addPoint(center.x-(int)(Math.cos(theta)*10),center.y+(int)(Math.sin(theta)*10));
        g.fillPolygon(p);
        g.drawLine(north.x, north.y, center.x, center.y);

        g.setColor(Color.white);
        p = new Polygon();
        p.addPoint(south.x, south.y);
        p.addPoint(center.x+(int)(Math.cos(theta)*10),center.y-(int)(Math.sin(theta)*10));
        p.addPoint(center.x-(int)(Math.cos(theta)*10),center.y+(int)(Math.sin(theta)*10));
        g.fillPolygon(p);
        g.drawLine(center.x, center.y, south.x, south.y);
    }

    public void setNeedle(float rotations[]) {
        if (rotations.length != 4) {
            System.out.println("Invalid Orientation data!");
            return;
        }
        theta = rotations[3];
        if (rotations[1]>0) {
            theta = -1 * theta;
        }
    }
}

```

```

    north.move((int)(center.x-Math.sin(theta)*100),(int)(center.y-Math.cos(theta)*100));
    south.move((int)(center.x+Math.sin(theta)*100),(int)(center.y+Math.cos(theta)*100));
    Graphics g = getGraphics();
    g.clearRect(0, 0, size, size);
    g.setColor(Color.red);
    Polygon p = new Polygon();
    p.addPoint(north.x, north.y);
    p.addPoint(center.x+(int)(Math.cos(theta)*10),center.y-(int)(Math.sin(theta)*10));
    p.addPoint(center.x-(int)(Math.cos(theta)*10),center.y+(int)(Math.sin(theta)*10));
    g.fillPolygon(p);
    g.drawLine(north.x, north.y, center.x, center.y);

    g.setColor(Color.white);
    p = new Polygon();
    p.addPoint(south.x, south.y);
    p.addPoint(center.x+(int)(Math.cos(theta)*10),center.y-(int)(Math.sin(theta)*10));
    p.addPoint(center.x-(int)(Math.cos(theta)*10),center.y+(int)(Math.sin(theta)*10));
    g.fillPolygon(p);
    g.drawLine(center.x, center.y, south.x, south.y);
}

public Dimension minimumSize() {
    return new Dimension(size, size);
}

public Dimension preferredSize() {
    return new Dimension(size, size);
}
}

```

C 数値地図の変換プログラム (mem2wrl.c)

```
/*
mem2wrl.c :
国土地理院発行の MEM ファイル (50m メッシュの標高データ) を
VRML ファイルに変換するプログラム。
プログラム作成にあたり、dem2wrl.c を参考にさせていただきました。
参照 URL: http://hiwaay.net/~crispen/src/\[index.html,dem2wrl.zip\]
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define ON 1
#define OFF 0
#define MAX 200
#define LINELIMIT 10000

void usage(void);
void initialize(int, char**);
void read_header(void);
void read_body(void);
void write_header(void);
void write_body(void);
void write_tailer(void);
void write_gnuplot_data(void);
int atoin(char*, int);

FILE *memfile;
FILE *wrlfile;
int data[MAX][MAX];
int gnuplot = OFF;
char buf[LINELIMIT];
char mapname[21];
int size = MAX;

main(int argc, char **argv)
{
    initialize(argc, argv);

    read_header();
    read_body();
}
```

```

write_header();
write_body();
write_tailer();

fprintf(stderr,"control Position is %d %d %d.\n",
        (size/2-2)*50,data[size/2][size/2-2],size/2*50);
fprintf(stderr,"control Position is %d %d %d.\n",
        size/2*50,data[size/2-2][size/2],(size/2-2)*50);

fprintf(stderr,"control Position is %d %d %d.\n",
        (size/2-1)*50,data[size/2][size/2-1],size/2*50);
fprintf(stderr,"control Position is %d %d %d.\n",
        size/2*50,data[size/2-1][size/2],(size/2-1)*50);
fprintf(stderr,"control Position is %d %d %d.\n",
        (size/2+1)*50,data[size/2][size/2+1],size/2*50);
fprintf(stderr,"control Position is %d %d %d.\n",
        size/2*50,data[size/2+1][size/2],(size/2+1)*50);
fprintf(stderr,"control Position is %d %d %d.\n",
        (size/2-1)*50,data[size/2-1][size/2-1],(size/2-1)*50);
fprintf(stderr,"control Position is %d %d %d.\n",
        (size/2+1)*50,data[size/2+1][size/2+1],(size/2+1)*50);

fprintf(stderr,"control Position is %d %d %d.\n",1*50,data[1][1],1*50);
fprintf(stderr,"control Position is %d %d %d.\n",0*50,data[1][0],1*50);
fprintf(stderr,"control Position is %d %d %d.\n",1*50,data[0][1],0*50);
}

void usage(void)
{
    fprintf(stderr, "Usage:      mem2wrl [ options ] memfile\n");
    fprintf(stderr, "Options:\n");
    fprintf(stderr, "\t-o wrldatafile  Specify VRML 3D World filename\n");
    fprintf(stderr, "\t-size          Specify the converted Terrain Size\n");
    fprintf(stderr, "\t-gnuplot       Convert to GNUPLLOT data file format.\n");
}

void initialize(int argc, char **argv)
{
    int n;
    char infile[100], outfile[100];
    memfile = stdin;
    wrldatafile = stdout;

```

```

for (n = 1; n < argc && argv[n][0]!='-'; n++) {
    if (!strcmp(argv[n],"-gnuplot")) {
        gnuplot = ON;
    }
    else if (!strcmp(argv[n],"-o")) {
        if ((wrlfile=fopen(argv[++n],"w")) == NULL) {
            fprintf(stderr,"Error: Can't open WRL file.\n");
            exit(-1);
        }
    }
    else if (!strcmp(argv[n],"-size")) {
        size = atoi(argv[++n]);
    }
    else {
        fprintf(stderr,"Error: Unknown option \"%s\"\n",argv[n]);
        usage();
        exit(-1);
    }
}

if (n == argc-1) {
    if ((memfile=fopen(argv[n],"r")) == NULL) {
        fprintf(stderr,"Error: Can't open MEM file. \"%s\"\n",argv[n]);
        exit(-1);
    }
}

fprintf(stderr,"Size is %d.\n",size);
fprintf(stderr,"Converting to %s.\n",
        gnuplot==ON?"Gnuplot data-file" : "VRML 3D geometry");
}

void read_header(void)
{
    int i;

    fgets(buf, LINELIMIT, memfile);

    fprintf(stderr,"メッシュコードナンバ : %d\n",atoin(buf+0,6));

    strncpy(mapname, buf+58, 20);
    fprintf(stderr,"図名 : %s\n", mapname);

    if (atoin(buf+57,1) != 1) {

```

```

        fprintf(stderr, "Error : This file isn't Simple map\n");
    }
    for (i = 0; i < MAX; i++) {
        if (*(buf+i+225) != '1') {
            fprintf(stderr, "Error : MEM File isn't Regular file\n");
            exit(-1);
        }
    }
}

int atoin(char *string, int num)
{
    int val = 0;

    while (--num >= 0)
    {
        if (isdigit(*string))
            val = val*10 + (*string - '0');
        string++;
    }
    return val;
}

void read_body(void)
{
    int i,j;

    for (i = 0; i < size; i++) {
        fgets(buf, LINELIMIT, memfile);
        for (j = 0; j < size; j++) {
            data[i][j] = atoin(buf+9+j*5, 5) / 10;
        }
    }
}

void write_header(void)
{
    int i, j;
    double min, max, sum, average;

    if (gnuplot == ON) {
        return;
    }
}

```



```

min = 9999;
max = -999;
sum = 0;
for (i = 0; i < size; i++) {
    for (j = 0; j < size; j++) {
        if (data[i][j] > max) max = data[i][j];
        if (data[i][j] < min) min = data[i][j];
        sum += data[i][j];
    }
}
average = sum / (size*size);
fprintf(stderr,"Maximum Height: %f\n",max);
fprintf(stderr,"Minimum Height: %f\n",min);
fprintf(stderr,"Average Height: %f\n",average);

fprintf(wrlfile,"#VRML V2.0 utf8\n");
fprintf(wrlfile,"Transform {\n");
fprintf(wrlfile,"    children [\n");
fprintf(wrlfile,"        WorldInfo {\n");
fprintf(wrlfile,"            title \"%s\"\n",mapname);
fprintf(wrlfile,"            info [\n");
fprintf(wrlfile,"                \"Data by GSI(Geological Survey Institute)\"\n");
fprintf(wrlfile,"                \"Converted by mem2wrl - M. Takaku\"\n");
fprintf(wrlfile,"            ]\n");
fprintf(wrlfile,"        },\n");
fprintf(wrlfile,"        DirectionalLight {\n");
fprintf(wrlfile,"            direction 1 -1 0\n");
fprintf(wrlfile,"        },\n");
fprintf(wrlfile,"        Background {\n");
fprintf(wrlfile,"            skyColor 0.1 0.2 1\n");
fprintf(wrlfile,"        },\n");
fprintf(wrlfile,"        Viewpoint {\n");
fprintf(wrlfile,"            description \"In the Center\"\n");
fprintf(wrlfile,"            position %d %d %d\n",
                (size/2-1)*50,data[size/2-1][size/2-1]+1,(size/2-1)*50);
fprintf(wrlfile,"        },\n");
fprintf(wrlfile,"        Viewpoint {\n");
fprintf(wrlfile,"            description \"Start\"\n");
fprintf(wrlfile,"            orientation 0 -1 0 2.3561945\n");/*135 度の回転*/
fprintf(wrlfile,"            position 0 %d 0\n", data[0][0]+1);
fprintf(wrlfile,"        },\n");
fprintf(wrlfile,"        Viewpoint {\n");

```

```

fprintf(wrlfile,"      description \"Over Head\\\"\\n");
fprintf(wrlfile,"      orientation -1 0 0 1.5707963\\n");
fprintf(wrlfile,"      position %d %d %d\\n",
      (size/2-1)*50, (int) max * 3, (size/2-1)*50);
fprintf(wrlfile,"    },\\n");
fprintf(wrlfile,"    Viewpoint {\\n");
fprintf(wrlfile,"      description \"North Edge\\\"\\n");
fprintf(wrlfile,"      orientation 0 -1 0 3.1415927\\n");/*180 度の回転*/
fprintf(wrlfile,"      position %d %d 0\\n",(size/2-1)*50, data[0][size/2-1]+1);
fprintf(wrlfile,"    },\\n");
fprintf(wrlfile,"    Viewpoint {\\n");
fprintf(wrlfile,"      description \"East Edge\\\"\\n");
fprintf(wrlfile,"      orientation 0 -1 0 4.712389\\n");/*270 度の回転*/
fprintf(wrlfile,"      position %d %d %d\\n",
      (size-1)*50, data[size/2-1][size-1]+1, (size/2-1)*50);
fprintf(wrlfile,"    },\\n");
fprintf(wrlfile,"    Viewpoint {\\n");
fprintf(wrlfile,"      description \"South Edge\\\"\\n");
fprintf(wrlfile,"      position %d %d %d\\n",
      (size/2-1)*50, data[size-1][size/2-1]+1, (size-1)*50);
fprintf(wrlfile,"    },\\n");
fprintf(wrlfile,"    Viewpoint {\\n");
fprintf(wrlfile,"      description \"West Edge\\\"\\n");
fprintf(wrlfile,"      orientation 0 -1 0 1.5707963\\n");/*90 度の回転*/
fprintf(wrlfile,"      position 0 %d %d\\n", data[size/2-1][0]+1, (size/2-1)*50);
fprintf(wrlfile,"    },\\n");
fprintf(wrlfile,"    NavigationInfo {\\n");
fprintf(wrlfile,"      avatarSize [0.25, 1.5, 5.0]\\n");
fprintf(wrlfile,"      type \"WALK\\\"\\n");
fprintf(wrlfile,"      speed 1.0\\n");
fprintf(wrlfile,"    },\\n");
fprintf(wrlfile,"    Transform {\\n");
fprintf(wrlfile,"      children [\\n");
fprintf(wrlfile,"        Shape {\\n");
fprintf(wrlfile,"          appearance Appearance {\\n");
fprintf(wrlfile,"            material Material {\\n");
fprintf(wrlfile,"              diffuseColor 0 1 0.5\\n");
fprintf(wrlfile,"            }\\n");
fprintf(wrlfile,"          },\\n");
fprintf(wrlfile,"        geometry ElevationGrid {\\n");
fprintf(wrlfile,"          solid TRUE\\n");
fprintf(wrlfile,"          xSpacing 50\\n");
fprintf(wrlfile,"          zSpacing 50\\n");

```

```

        fprintf(wrlfile,"                xDimension %d\n", size);
        fprintf(wrlfile,"                zDimension %d\n", size);
        fprintf(wrlfile,"                height [\n");
        fprintf(wrlfile,"                ");
    }

void write_body(void)
{
    int i, j;

    if (gnuplot == ON) {
        write_gnuplot_data();
        return;
    }
    for (i = 0; i < size; i++) {
        for (j = 0; j < size; j++) {
            fprintf(wrlfile, "%3d, ",data[i][j]);
            if ((j+1) % 10 == 0) {
                fprintf(wrlfile, "\n                ");
            }
        }
    }
}

void write_tailer(void)
{
    if (gnuplot == ON) {
        return;
    }
    fprintf(wrlfile,"]\n");
    fprintf(wrlfile,"        }\n");
    fprintf(wrlfile,"    }\n");
    fprintf(wrlfile,"]\n");
    fprintf(wrlfile,"    }\n");
    fprintf(wrlfile,"]\n");
    fprintf(wrlfile,"}\n");
}

void write_gnuplot_data(void)
{
    int i, j;

    for (i = size-1; i >= 0; i--) {

```

```
    for (j = 0; j < size; j++) {  
        fprintf(wrlfile, "%d\n",data[i][j]);  
    }  
    fprintf(wrlfile, "\n");  
}  
}
```