# 2006 年度 メディア情報学実験 2 バーチャルリアリティ

# 1. 目的

本実験では、人工現実感 (VR) 、複合現実感 (MR) と呼ばれるメディア処理技術、ヒューマンインタフェース技術のエッセンスを、自ら課題を設定し、VR 作品、MR 作品を制作し体験することによって学ぶことを目的としている.

本実験では、コンピュータ内に構築した VR 空間やそれを現実世界と合成した MR 空間の視覚的な側面だけを取り扱う。その意味では、CG プログラミングの発展形であるが、実時間対話型での実行を前提としたリアルタイム CG プログラミングが大きな特徴である。

本実験では、眼鏡なし立体ディスプレイや頭部装着型ディスプレイといった普段余り見かけない特殊な機器に触れ、それらを使いこなすことも目標としている.

本実験では多様な使用機材や新しいプログラミング環境,ツール群が登場するので、その習得にかなりの時間を要するかと思われる。これは、学生時代の必須科目で、なるべく多くの環境やツールに接することが、「メディア情報学科」の卒業生として社会に出てから大きな財産となり、より新しい環境に適応できる糧となるとの配慮である。

# 2. 基本概念

# • VR (Virtual Reality)

人工現実感 (Virtual Reality; VR) とは、コンピュータ 内に構築された仮想空間・仮想環境を、我々人間が諸 感覚 (五感など) を通じて体験する技術の総称である。 即ち、VR システムとは、コンピュータに我々人間と対 話できる入出力機器を接続し、現実を模して作った電 子的な仮想世界をあたかも実在するかのように疑似体 験できるようにしたシステムをいう。

また, VR 技術は、現実世界をなるべく忠実に再現するためだけでなく、現実には絶対に存在しえない世界、あるいは現実には存在するが人間にとってミクロやマ

クロであったり、時間的または場所的な隔たりがあって体験できない世界を、効率良く手軽に体験できるようにしたい場合にも威力を発揮する.

五感を通じて仮想世界を体験するといっても,人工的な嗅覚や味覚の研究はまだまだ未開拓であり,実際に用いられるのは,視覚・聴覚・触覚(力覚)の提示である.中でも,人間とっても最も重要な視覚的な刺激の提示が進んでいるので,本実験でも「視覚的な VR」のみを取り扱う.

本格的な視覚的没入感を与えるには、体験者の視野の大半を覆う頭部装着型ディスプレイや広視野没入型ディスプレイを用いることが望ましいが、学生実験に用いるには高価で現実的ではないため、本実験では、コンピュータ内の仮想空間を構築して実時間対話する体験や、眼鏡なし立体ディスプレイで立体感を体験することのみを行う.

# • MR (Mixed Reality)

複合現実感 (Mixed Reality; MR) とは、目の前の現実世界とコンピュータ内に構築した仮想世界を実時間で融合・表示する技術である. 拡張現実感 (Augmented Reality; AR) と呼ばれることもあるが、ほとんど同義であり、本実験では MR を用いる.

従来の人工現実感がコンピュータで生成したフル CGの映像で描くのに対して、複合現実感では目の前の 現実の光景を借景として、仮想物体や仮想現象を重畳 表示する. VRの限界を超える新感覚の体験が可能であ り、現実世界に立脚した新しい情報提示技術としても 期待されている.

# 3. 実験の流れとスケジュール

本実験は、大きく前半と後半に分かれて 2 つの実験 (作品制作)を行う。前半の第  $1\sim2$  週目は、Java と Java3Dを用いて VR 空間の構築とその空間とのインタラクションを実現し、後半の第  $3\sim4$  週目は、C 言語と

ARToolKit を用いて MR 空間の構築とその空間とのインタラクション機能を実現する. 第5週目に、その両方の成果を発表し、評価を受ける.

本実験は、全期間を通じて、予め定められたグループ単位で課題を実行する。各グループ内では、メンバーの適性を考慮して役割分担を決め、互いに助け合って課題を達成する。

# 第1週目 VR 空間の構築1

冒頭に本実験全体の目的・概要・手順・課題等の説明がある。その後、各グループ内での役割分担を決め、全員で使用機器と資料・事例を確認し、めざす作品の目標・シナリオを考える。

残る時間で、役割に応じて実験(VR作品制作)の作業にとりかかる.

グループ内の役割分担一覧と目標とする作品の概要を記したものを、(グループで1部)実験時間終了時もしくは第2週目の開始時に提出する.

### 第2週目 VR 空間の構築2

第1週目に自分たちで立てた目標に向かって、作品 制作の続きを行う。

第2週目から第4週目までの各実験時間後と、その他2日間の計5回、各1.5時間のフォローアップ時間が設けられている。実験時間内に目標を達成できないグループは、その時間に自発的に作品制作を続けてもよい。

目標が達成しない場合でも、第2週目終了時までの 各自の作業内容・達成したことを、各人がレポートと してまとめ、所定の期日までに提出する.

# 第3週目 MR 空間の構築1

冒頭に後半の MR 実験に関する簡単な説明・注意がある. 第2週目までの目標 (VR 作品制作) に到達していないグループは、 MR 作品制作に向かわずそのまま前週までの課題を続けるか、VR 作品制作は未完成のまま中断して MR 作品制作に向かうかを、教員あるいは TA に相談して決めること.

その後は、第1週と同様、グループでの目標を立て、 各自の役割に応じて、実験(MR 作品制作)にとりか かる.ここで、VR 作品制作とは、グループ内での役割 分担を変更しても差し支えない.

VR 作品制作と同様, 実験時間終了時もしくは次回の 開始時に<u>グループ内の役割分担一覧と目標とする作品</u> <u>の概要を記したものを提出する.</u>

# 第4週目 MR 空間の構築2

第3週目の実験の続きを行う.

作業が早く終わったグループは、当初の目標以上に 機能を追加したり、各作品の質を向上させる努力をす ること、さらに時間があれば、翌週の発表準備やレポ ート執筆にとりかかっても良い。

# 第5週目 VR・MR 作品の発表・体験会

冒頭の30分間で、2つの作品の仕上げや発表準備を 行う.

各グループが、約5分間で自分たちが作り上げた作品のポイントを全員の前で口頭発表する(なるべく PowerPoint を使用して、分かりやすくアピールする).

発表終了後、教員及び TA が各グループの作品を体験して回って出来映えを評価する.

<u>5 週間の実験期間終了後に、各人が最終レポートを提出する.</u>

# 4. グループ内での役割分担

本実験は,6~7人でチームを編成し,各グループ内で下記の担当を決め,自分たちが立てた目標に対して,全員協力して作業を行う.

### [プログラマ(インタラクション担当):2人]

作品制作作業のうち,主として VR/MR 空間とのインタラクション機能を実装する. 必要に応じて,インタラクション部分以外のプログラミングも担当する.

## 「プログラマ (アニメーション担当):2人]

作品制作作業のうち、主に CG アニメーション部分のプログラミングを担当する. 必要に応じて、他の部分のプログラミングも受け持つ.

# [モデラ:2人]

CG オブジェクトのモデリングを行う. 作業の進行状況によっては、プログラミング作業の応援に回る.

# [ディレクタ:1人]

グループの進行状況を確認したり、メンバーに必要な指示を出したりする監督業務を行う. 目標設定・シナリオ作成も積極的に行う.

なお, 6 人班の場合は、モデラまたはプログラマの うちの1人がディレクタを兼任する. 作業の際には2人で1台のPCを使用することになる. 1人がPCを使っている間には、もう1人は資料を読んで必要な情報を探し出すなど工夫し、効率よく作業できるようにすること.

# 5. プログラミング環境とツール

### • Java3D

Java3D とは、Sun Microsystems 社が開発した Java の標準的な 3 次元コンピュータグラフィックス API である。 VR 用 API(Application Program Interface)として設計されているため、VR 用入力 I/O デバイスと相性が良い。 また、シーングラフと呼ばれるグラフィックス用データベースを用いることによって VR 空間のシーン構造を容易に管理することが可能となっている.

Java3D は描画部分を OpenGL や DirectX に依存しているものの,ユーザ (プログラマ) はその存在を意識せずに Java3D のみで 3D 空間を構築することが可能である. 既に開講されている講義で OpenGL を用いた 3D 空間を構築している学生も少なくないと思われるが,OpenGL と Java は上述のシーングラフなど,扱う上でいくつか異なる点がある. Java3D に関する詳細は,各グループに配布している資料を参照すること.

また、本実験では Java3D の上位層に位置する「彩画3D」というグラフィックス API も利用する.「彩画3D」

の詳細もJava3D同様各グループに配布している資料を参照すること.

### eclipse

eclipse とは、Java 及び Java3D のプログラミング環境として、本実験で用いるソフトウェアである。eclipse は Java プログラムの構造をビジュアル的にわかりやすく扱うことができるため、プロジェクトの管理が容易となる。詳しい使用方法については各グループの配布資料を参照することに加え、後の作業手順に書かれている事柄をよく読んでおくこと。

# シーングラフ

シーングラフとは、オブジェクトの関係を木構造で表したものである。ここで述べるオブジェクトとは、球や立方体などの形状のみでなくその色や質感などの属性、移動・回転などの座標変換、さらに時間的変化などの動的要素も含んでいる。

本実験の第1~2週で扱うことになる Java3D で用いられるシーングラフは、これらの要素に加えてビューモデルをもオブジェクトとして扱い、オブジェクトが配置される座標系から独立した視点移動を可能としている。

Java3D で用いられるシーングラフの基本形と各要素の持つ意味を図1、表1に示す.

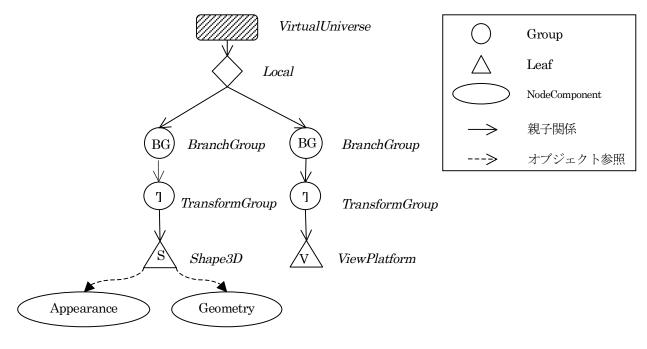


図1 シーングラフの基本形表1 シーングラフの各要素

Node(要素)	意味	
VirtualUniverse	仮想空間,すなわち VR 空間全体を示す.	
Local	仮想空間上の座標を示す.	
BranchGroup	グループノード.	
	1つまたは複数のオブジェクトを1個のグループとして扱う.	
TransformGroup	グループノード.	
	それ以下のオブジェクトまたはグループの位置や姿勢を制御する.	
Shape3D	3D オブジェクト.	
	Geometry (頂点座標などの物理的形状), Appearance (色などの物体の見え方に関する属性)	
	を保持する.	

### • VRML

Virtual Reality Modeling Language (VRML)とは、WWW 上で VR 空間を体験するために開発された記述言語である. VRML の最初のバージョン (通称 VRML 1.0) は 1994 年 11 月に制定された. このバージョンでは SGI 社により開発されていた Open Inventor と よばれるツールのファイルフォーマットに良く似た仕様として制定された. その後、インタラクティブな動きなどの新しい機能を追加した VRML 97 (ISO/IEC DIS 14772-1、通称 VRML 2.0) 仕様が策定された. 現在では、VRML と呼ぶ場合にはこの VRML 2.0 を指すことが多い.

当初は VRML ビューワが無償提供されていたが、現在は本来の目的では使用されず、3次元 CG データのファイルフォーマットとしてよく用いられている.

ファイル形式はテキストファイルであり、ヘッダ、コメント、ノード、プロトタイプ、ルートの5つの要素から構成される.3Dポリゴンの頂点および線の座標、ポリゴンや色や画像によるテクスチャ、光源による明るさなどを指定できる。また、URL指定によってインターネット上の別の場所にある画像やVRMLファイルを指定できる。アニメーションや光源、視点の設定などといったインタラクティブな効果も設定でき、一種の仮想空間を構築できる。さらに、Scriptノードを使って、Java・JavaScript などのプログラミング言語と連携させた動作を行うことも可能である。

VRML ファイルは「ワールド」とも呼ばれ, ".wrl" という拡張子が付く. VRML ファイル自体はテキスト 形式だが, 座標値などの 3D データを多く含み, ファイル容量が大きくなるため, gzip を使った圧縮が行われる場合も多い. たいていの 3 次元モデリングツールには, VRML 形式での保存機能が付いている.

# ARToolKit

ARToolKit とは、AR/MR 機能を簡便に実行するためのツールとして公開されているライブラリ群であり、現実空間と仮想空間の幾何学的な整合を取るために利用される.

ARToolKit は Linux を主プラットフォームとして C 言語で実装されている. 移植性が高く, Windows, IRIX, Mac OS X などの種々のプラットフォームにも対応している. 本実験では, Windows 版を使用している.

ARToolKit は図2に示すような黒色正方形マーカを使用する。現実空間内にマーカを配置し、カメラから得られた入力画像からマーカとカメラの位置姿勢関係を計算して求め、実時間で現実空間と仮想空間との位置合わせを実現している。各々のマーカの内側には非点対称となる固有のパターンを描いており、複数の正方形マーカが存在する場合でも、各マーカの位置姿勢を独立して求めることが可能である。図3にARToolKitによる位置検出の流れを示す。

ARToolKit の詳細は各グループに配布している資料を参照すること.

### • C++

C++とは、オブジェクト指向型プログラミングのために生まれた C 言語であり、1979 年に Bjarne Stroustrup によって開発された. 拡張子は「.cpp」を用いる. C 言語を拡張した言語で、C 言語との互換性を保っているために、基本的には C 言語の関数やキーワードを用いることができる. ただし、コンパイラによっては一部正常に動作しない関数があるので、使用する際には注意が必要である.

C 言語に比べてファイルサイズが大きくなり、パフォーマンスの低下が見られることにあるものの、オブ

ジェクト指向型言語であるために高い保守性を誇り、大規模システムの開発に向いているといえる.

C++は C 言語に「クラス」という概念を適用してオブジェクト指向型言語となっている. クラスとは, C 言語に備わっていた構造体に関数機能を持たせたものといえる. 以下に C 言語で用いられる構造体, C++で用いられるクラスの宣言を示す.

```
struct testStruct {
   int a;
   char b[16];
};
```

## 構造体

```
class testClass {
  public:
     int a;
     char b[16];
};
```

# クラス

構造体では"struct 構造体名"と宣言していたのに対し、クラスでは"class クラス名"と宣言する.

先述の通り、クラスに関数機能を追加するには、右段上部に示した 2 例のように記述する. なお、この 2 例は同じ意味を持つ.

クラス内で定義された関数は右段下のように使用することができる.

また,クラスも構造体同様ポインタや配列として変数を宣言することが可能である。実際にどのように使用されているかは、サンプルプログラムを参照して確認すること。

# • Visual Studio.Net & Visual C/C++

Visual Studio.Net とは、C言語のプログラミング環境として、本実験で用いるソフトウェアである. Visual Studio.Net は C 言語等のプログラムの構造をビジュアル的にわかりやすく扱うことができるため、プロジェクトの管理が容易となる.

Visual C/C++は Microsoft 社の, C 言語/C++言語によるソフトウェア開発のための統合開発環境であり, Visual Studio.Net に含まれている. 使用方法については、実験手順の項目を確認すること.

```
class testClass {
public:
    int a;
    char b[16];
    void setA(int n) {
        a = n;
    };
    void printB() {
        cout << b;
    };
};</pre>
```

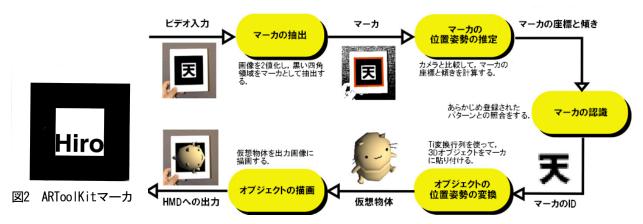


図3 ARToolKitによる位置検出とCGの描画

# 6. 使用機材

以下では、本実験で用いる各機器(図 4) について 説明する。全て揃っているかどうか確認すること。 また、高価な機材もあるので、その取り扱いには十 分注意すること。

# • Dell Precision 380

PC 本体以外にも同時に起動すべき機器が同時に搭載されているので、全て電源が入っているか、また正常に動作しているかをチェックする. チェック項目は以下の通りである.

- (1) PC 本体の電源が ON
- (2) PC が収まっている台の中に設置されたセンサス イッチ2つがON (右図)
- (3) 同台の中にある電源タップのスイッチが ON
- (4) 画面解像度が 1600×600 (第3~4 週目のみ)



# ● SHARP PC-AL3DH (裸眼立体表示ディスプレイ付き PC)

3D 表示ディスプレイは、偏光メガネのような PC 以外の特別な機器を必要としない、いわゆる裸眼立体視することのできるディスプレイである.

本実験で用いる SHARP PC-AL3DH のディスプレイには、一般的な TFT 液晶に加えてスイッチ液晶と呼ばれる液晶が使用されている。このスイッチ液晶で視差バリアとなるものを設けてその透過性を制御し、光の進路を調節して左右の目に異なる映像が映るようにしている (図 5).

3D 空間を表示するプログラムを全画面表示させる と、自動的に立体表示されるようになっているので、 実際にどのように表示されるかは全画面表示サンプル プログラムで確認すること.

# • Dell Precision M20

CG 物体のモデリングに用いる PC である.

モデリング用ソフトとして Metasequoia (シェアウェア版) を搭載している. 既に実験 1 で使用しているはずであるが、作成したモデルの保存方法が実験 1 とは異なるので、課題を参考して作業すること.

# HMD (Head Mounted Display)

頭部装着型ディスプレイ HMD は、VR や MR の実現において非常に重要な役割をなす装置であり、VR の象徴的存在である。 本実験の VR 実験には HMD は使用せず、MR 実験にのみ用いる。

MR 実験で用いるのは、ビデオシースル一型 HMD

と呼ばれるものである. HMD に内蔵された一対の小型 テレビカメラが、左右の目に代わりに目の前の光景を デジタル化して映像をコンピュータに取り込む. コン ピュータ内で CG 映像と合成した結果、右目用と左目 用の映像が HMD 内蔵の小型液晶ディスプレイに表示 される. この結果、現実と仮想が融合された MR 空間 の映像を両眼立体視することができる(図6参照)、

実験開始前に、HMD に接続されている各ケーブルが 図7に示すように全てコントロールボックス (緑のスイッチのついた黒い機械) にしっかりささっているこ

とを確認する.外れている場合は近くの TA に報告し、 対処してもらうこと.

また、使用時には、カメラに付けられている小さなカバーを外すことになるが、<u>絶対にこのカバーを無く</u>さないように、しっかりと管理する.

HMD でマーカを撮影した際にピントが合わない場合は、カメラのレンズ周りの黒いリングを回すことによって調節できる。左右のカメラのピントは独立しているので、左右それぞれを調節すること。



[Dell Precision 380] ・プログラミング用 PC



[SHARP PC-AL3DH]

- ・プログラミング用 PC
- ・3D 表示ディスプレイ



[Dell Precision M20]

- ・モデリング用 PC
- · Metasequoia 搭載



「ビデオシースルーHMD]

- · MR 空間視認用
- ・Dell Precision 380 で ARToolKit を用いる際 に使用



「USB カメラ〕

- ・現実世界キャプチャ用
- ・SHARP PC- AL3DH で ARToolKit を用いる 際に使用



[USB ゲームパッド]

- ・Joystick を用いたイン タラクションに使用
- ・プログラミング用 PC に接続する

図4 実験で用いる機器

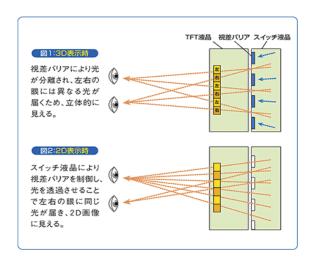


図 6 HMD の処理の流れ

図5 3D表示ディスプレイの仕組み



ケーブルに書かれた名称と接続されて いる部分が対応しているか確認する

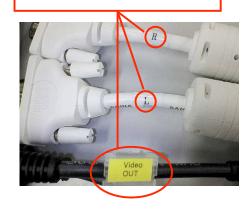


図 7 コントロールボックスとケーブル

# ● USB カメラ

USB カメラは SHARP PC-AL3DH で MR 実験を行う際に用いるビデオカメラである. プログラムを実行する前に必ずカメラが PC の USB 端子に接続されていることを確認すること.

また、マーカを撮影した際にピントが合っていないようであればレンズ周りの黒いリングを回すことで調節可能である.

● USB ゲームパッド

ゲームパッドはプログラミング用 PC に接続して用いる.

PC を起動する前に接続すると PC が起動中にフリーズする場合があるので,  $\underline{\textit{vi PC}}$  が起動した後に接続すること.

このゲームパッドは、家庭用ゲーム機のゲームパッドと同様な感覚で使用する PC 用の機器で、本実験では自ら設計した VR 空間との対話操作のために用いる. 各ボタン/ジョイスティックの振舞いは、既に Java3D のプログラムとして用意されているので、後に与えら れている課題を参考にし、操作方法を確認すること.

### ● 各機器の接続

各機器はLANケーブルによって図8のように接続されている. したがって、共有フォルダを使用することによってPC間のデータの受け渡しが可能である.

各 PC が正しく接続されていることと、各 PC の電源 が入っていることを確認した上で「マイ ネットワーク」を開き、他の 2 端末の共有フォルダが見えること を確認する.



図8 各機器の接続関係

# 7. 第1週目 VR 空間の構築1

第1週目の作業の流れは次の通りである.

- step 1 グループ内での作業分担決定
- step 2 使用機器のチェック
- step 3 実験の実行環境の準備
- step 4 既存プログラムを用いた VR 空間の確認
- step 5 グループ作品のシナリオ作成
- step 6 作業担当ごとの準備課題の実行
- step 7 グループ作品の制作(第2週目にも継続)

step1 は「5. グループ内での役割分担」, step2 は「6. 使用機器」をよく読み, 進めること.

step3 以降の作業について、詳しい手順を説明する.

# 7.1. 実験の実行環境の準備

(1) プログラミング用 PC を使ってエクスプローラを 開き,「D:¥jikken2¥グループ名<sup>(注)</sup>¥VR」フォル ダを作成する.

注)グループ名:実験全体のグループ( $C1\sim C2$ )\_ 班番号

- 例: C1の1班 → c1\_01
- (2) 「D:¥VR¥VRSet」フォルダを1で作成したフォ

- ルダ内にコピーする.
- (3) eclipse を起動し Workspace Launcher ウィンド ウが開いたら、ワークスペースを(2)で作成した「D:¥jikken2¥グループ名¥VR¥VRSet」フォル ダにする.
- (4) 画面左の「パッケージ・エクスプローラ」ウィンドウ内が空白でなければ、「7.2. 既存プログラムを用いた VR 空間確認」へと移る. パッケージ・エクスプローラウィンドウが空白の場合のみ、そのウィンドウ内で「右クリック→Import」を選択する.



eclipse のアイコン





(5) Import 画面が開いたら「Existing Projects into Workspace」を選択する.



(6) ルートディレクトリを「D:¥jikken2¥グループ名 ¥VR¥VRSet」とする. テキストボックスが空白 の場合は、右にある「Browse」ボタンをクリック し、上記のフォルダを選択する.



(7) プロジェクトの一覧が読み込まれた後,全てのチェックボックスがONになっていることを確認し,「終了」をクリックしてプロジェクトを読み込む. 読み込まれたプロジェクトは,「パッケージ・エクスプローラ」ウィンドウ内にその名称が表示されるので,確認する.



# 7.2. 既存プログラムを用いた VR 空間確認

既に用意されたプログラムを実行し、VR 空間とはどのようなものかを感覚的に理解する.

参考までに用意されている事例プログラムの種類と 内容は**表2**の通りである。

表 2 事例プログラムの種類

スニー 1777 ・ フェーロース				
種類	内容			
	プロジェクト名称: Ex00_〇〇~Ex96〇〇			
	Java3D の基本的な機能を確認するための			
基本機能	プログラム群である.			
操作事例	プロジェクトの数が膨大なので、各プロジ			
	ェクトの名称を確認し、必要なものを選別			
	して実行してみると良い.			
動作事例	プロジェクト名称:tank			
	基本機能を複数組み合わせて実装されたプ			
	ログラムである.			
	作品を作るのに参考となる.			
作品事例	プロジェクト名称: VRTest_UFO, Slime			
	動作事例を基に、作品として作られたプロ			
	グラム.			
	この例が本実験で求めるレベルの作品なの			
	で、必ず両方実行すること.			

# 《実行方法》

- (1) 各既存プロジェクトのフォルダから「デフォルト・パッケージ→○○.java」を開き、プログラムを画面右に表示する.
  - ○○.java は、各プロジェクトの実行ファイルである. 1 つのプロジェクト内に複数のファイルがある場合は、main メソッドのあるファイルを開く.



- (2) メニューから「実行→実行→Java アプリケーション」を選択し、実行する.
- 注) main メソッドのあるファイルが画面右に表示されていないと、選択できないので注意する.
- (3) 各サンプルはキーボードやマウス,ゲームパッド でインタラクション可能となっているので,付録 Aを参考にして,どのように変化するか確認する. 班員全員で VR 空間とはどのようなものか確認すること.





図 9 作品事例(上:Slime 下: VRTest UF0)

# 7.3. グループ作品のシナリオ作成

第 1~2 週を使って、グループで 1 つのオリジナル VR 空間を構築する.

作品事例プログラムを参考に、どのような作品を作るのか、またそれに必要な機能をできるだけ詳しくグループ内で議論する. なお、以下に挙げる基本項目は必ず含むこと. 余裕があれば更に発展項目に挑戦する(発展項目の達成は、成績評価で加点する).

決定した目標・機能は、第2週目の始めに、別途配布した用紙に記入して提出する。実験時間内には、詳細なシナリオの作成に多くの時間を費やすよりも、プログラミング及びモデリングに取りかかることが望ましい。与えられた時間内でシナリオの細部が詰め切れなかった班は、第2週目実験開始までの時間外に相談して完成させること。

# 《基本項目》

- ・VRML ファイルの描画
- 背景作成
- ·CG オブジェクトのアニメーション
- ジョイスティックを使ったインタラクション

# 《発展項目》

- ・キー操作によるウォークスルー(空間視点移動)
- · 立体視 (裸眼立体視)

### 7.4. 準備課題

7.4.1. プログラマ用準備課題

下記の準備課題を順次実行し、VR 空間がどのように 構築されているかの参考にする. この課題を行うため に用いるプロジェクトは「Slime」である.

グループで作品を完成させるためには必要となるポイントなので、課題1~4は全て行うことが望ましいが、準備課題に時間をかけすぎると作品を作る時間がなくなる、各課題に設定された所要時間を目安として常に時間に注意しながら進めることが肝要である.この準備課題を手早く終えて、グループ作品のプログラミング作業へと進むこと.

課題 5 は発展項目用課題なので、作品に発展項目を 取り入れない場合は実行する必要はない.

# 課題1 座標系の確認(所要時間目安:10分)

「Slime→デフォルト・パッケージ→○○.java」に記述されている移動・回転・拡大に関するメソッドのパラメータ変更や記述順の変更などを行い、どのように変化するか確認する.使用されるメソッドは以下の通りなので、記述箇所は各自で探すこと.

# 平行移動メソッド:

setTranslation( Vector3f(x, y, z))

### 回転メソッド:

setRotation( AxisAngle4f( *x-axis*, *y-axis*, *z-axis*, angle))

拡大・縮小メソッド:

# 課題2 表示オブジェクトの変更

(所要時間目安:10分)

読み込み・表示するオブジェクトを変更する. オブジェクトは VRSet¥metaSamples 内にある VRML(.wrl)ファイルを用いることができる.

オブジェクトを読み込むメソッドは以下の通りなので、記述箇所は各自で探すこと.

# VRML 読み込みメソッド:

myVrmlLoader(String file name)

# 課題3 背景の作成(所要時間目安:10分)

プロジェクト「Slime」内では、背景が平面で描かれている. 背景を描く位置、背景の形状 (球など)、背景の画像などを変更する.

画像は VRSet¥metaSamples 内にある png ファイルを 用いることができる. 背景(平面)生成メソッドは以 下の通りなので,記述箇所は各自で探すこと.

# 背景(平面)生成メソッド:

MyPlaneBackground( String *file name*)

# 課題4 インタラクションの実装

(所要時間目安:20分)

既に記述されているインタラクションの他に、新たなインタラクションを実装する。手順としては、まずジョイスティックからの入力で実行されるようになっているメソッド「processStimulus」を確認する。次にメソッド内のパラメータを変更し、実際にどのように反映されるかを確認する。

確認できたら、他の既存プログラムを参照してサン プルとは異なったインタラクションを実装し、正しく 動作するかを確認する.目的に応じて更に機能を追加 する.

Joystick に関するメソッドはプロジェクト「utils」の MyJoystick.java に書かれているので、必ずどのようなメソッドがあるのかを確認しながら進めること.

### 課題 5 全画面表示

# (発展項目用. 所要時間目安: 20分)

基本機能操作事例の Ex85\_FullScreen を参考に、実行結果を全画面表示する.

まず手順としては、ウィンドウを表示しているメソッドを確認する. ウィンドウを表示するメソッドは以下のものなので、記述箇所は各自で確認すること.

次に、Ex85\_FullScreen のソースと Slime のソースを 比較し、どのように異なっているかを確認する. 具体 的には、使用するクラスやメソッド、あるいはそれら の位置(どのメソッドから呼ばれているか、など)を 確認する.

なお, ウィンドウサイズを画面サイズに指定しただけのものは全画面表示とは言えないことに注意する.

### ウィンドウ表示メソッド:

MainFrame (Applet applet, int width, int height)

MainFrame (Applet applet, int width, int height, String title)

MyMainFrame (Applet applet)

# ● 参考(プログラマ用):新規プロジェクトの作成

(1) メニューバーから「File→New→プロジェクト」 を選択し、実行する.



(2) 「Java プロジェクト」を選択する.

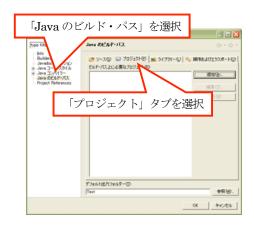


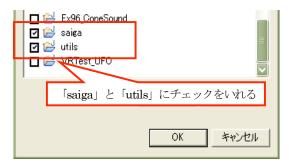
- (3) プロジェクト名を入力し、「終了」をクリックすると、パッケージ・エクスプローラ内に作成したプロジェクトが表示される.
- (4) パッケージ・エクスプローラに表示されたプロジェクト名の上で右クリックし、「Properties」を選択する.



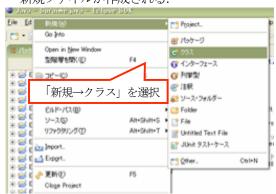


- (5) 「Java のビルド・パス」から「プロジェクト」 タブを選択し、「追加」ボタンをクリックする.
- (6) 「saiga」と「utils」にチェックを入れ,「OK」 ボタンをクリックする.

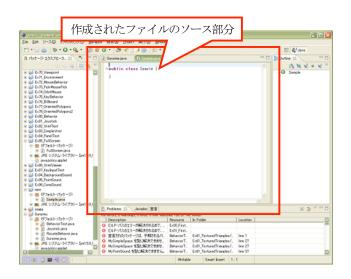




- (7) 新たにクラスを作成するには,「プロジェクト名 の上で右クリック→新規→クラス」を選択する.
- (8) クラス名を入力し、「終了」をクリックすると、 新規ファイルが作成される.



- (9) 確認のために、作成するクラス名を「Joystick」 とし、「VRTest\_UFO→デフォルト・パッケージ →Joystick.java」のソースをコピーする.
- (10) 7.2 を参考に、作成したプロジェクトが正しく動作するかを確認する.



# 7.4.2. モデラ用準備課題

本実験で用いる PC には予め Metasequoia のシェアウェア版を搭載しているが、フリーウェア版は http://www.metaseq.net/からダウンロードすることができる. 実験時間内に作業し切れない場合は、自宅の PC などにインストールし、実験時間外にも作業を進めることができる. フリーウェア版では VRML97 形式で出力することはできないが、実験室で変換可能であるので、Metasequoia ファイルで保存して実験時間に持参し変換すればよい.

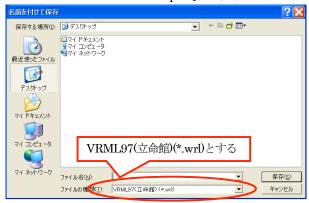
下記の準備課題が終わり次第,グループ作品のモデリング作業へと進むこと.

# 課題1 VRML 形式での保存

(所要時間目安:15分)

- (1) Metasequoia を起動し、立方体や球といった簡単な形状を作成する。ただし、以降の課題を円滑に進めるため、最低2つの形状を使用して3軸方向全てに非線/点対称の形状とすること。
- (2) メニューバーから「ファイル→名前をつけて保存」を選択する.
- (3) ファイルの形式を「VRML97 (立命館) (.wrl)」 として保存する. なお, VRML97 形式にすると ファイルの変更ができなくなるので, モデリング を変更する可能性の有無に関わらず, Metasequoia ファイル (拡張子 ".mqo") も同時 に保存しておくこと.
- (4) 「保存」をクリックした後に表示されるウィンド ウでは、設定は変更せずに全て「OK」を選択す る.

(5) プログラミング用 PC にデータを移す際には、移動先のフォルダを「D:¥jikken2¥グループ名 ¥VR¥VRSet¥metaSamples」とする.



# 課題 2 Java3D でモデルを表示 (所要時間目安:10分)

- (1) 課題 1 で作成したモデルをプログラミング用 PC に転送し、プログラムの必要箇所を書き換え (プログラマ担当) て、どのように表示されるか確認する.
- (2) 作成したモデルと表示されたオブジェクトがどのように異なるかを確認する.

# 8. 第2週目 VR 空間の構築2

第2週目は、第1週目の作業の続きを行う.

作品が完成といえるレベルまでできたら近くの TA にチェックをしてもらい、その後の指示を仰ぐこと.

また、VR 空間の構築に関しては原則としてこの週で終了する。第2週目から第4週目までは、実験終了後にフォローアップの時間として 1.5 時間実験室を使用可能とするので、実験時間内に作業が終わらなかったグループは活用すること。

第3週目までの課題として,この週までに達成したことをレポート(中間レポート)にして提出する.「12.レポートの書き方」に沿って,VR作品についてのレポートを作成し、期限までに指定の場所へ提出する.

# 9. 第3週目 MR空間の構築1

第3週目の作業の流れは次の通りである.

step 1 グループ内での作業分担

step 2 実験の実行環境の準備

step 3 既存プログラムを用いた MR 空間の確認

- step 4 グループ作品のシナリオ作成
- step 5 作業担当ごとの準備課題の実行
- step 6 グループ作品の制作(第4週目にも継続)

# 9.1 実行環境の準備

- (1) プログラミング用 PC でエクスプローラを開き, 「D:\(\mathbb{D}\) フォルダがあることを確認する.
- (2) プログラムを実行する前に、HMD (Dell Precision 380) および USB カメラ (SHARP PC-AL3DH) が接続されていることを確認する.

# 9.2 サンプルプログラムの確認

予め用意されているサンプルプログラムを実行し、 MR 空間がどのようなものかを確認する.

サンプルファイルの種類と内容はグループごとに配布している「MR サンプルプログラム解説」を参照すること。

# 《実行方法》

実行するサンプルプログラムとして、複数の機能を 実装した「allFunctions」を用いる。その他のサンプル についても実行方法は同様なので、必要に応じて各自 実行すること。特に Dell Precision 380 を用いて実行す る場合、プロジェクト名に「HMD」とついてあるサン プルを必ず最低 1 つ実行し、どのように見えるのかを 確認すること。ただし、これらのサンプルは SHARP PC-AL3DH では実行できない。

- (2) 「allFunctions (VC++ プロジェクト)」をダブ ルクリックし、起動する.



(3) メニューバーから「デバッグ→開始」を選択, 実 行する.



(4) グループごとに配布されている ARToolKit 用マ

一カを HMD またはカメラで撮影し、マーカに応じた CG オブジェクトが表示されることを確認する. このサンプルは 2 つのマーカを使うことによってインタラクションを行うことができる. 操作方法については付録 B を参照すること.



図 10 all Functions の実行画面

# 9.3 シナリオの作成

第 3~4 週を使って, グループで 1 つのオリジナル MR 空間を構築する.

事例を参考に、どのような作品を作るのか、またそれに必要な機能をできるだけ詳しくグループ内で議論する. なお、以下に挙げる基本項目は必ず含むこと. 余裕があれば更に発展項目に挑戦する.

決定した目標・機能は、第4週目の始めに、別途配布した用紙に記入して提出する。実験時間内には、詳細なシナリオの作成に多くの時間を費やすよりも、プログラミング及びモデリングに取りかかることが望ましい。与えられた時間内でシナリオの細部が詰め切れなかった班は、第4週目実験開始までの時間外に相談して完成させること。

# 《基本項目》

- ・VRML ファイルの描画
- ·CG オブジェクトのアニメーション
- マーカを利用したインタラクション

# 《発展項目》

・立体視 (HMD による両眼立体視. 視差を計算した画像を2画面で表示する)

# 9.4 準備課題

# 9.4.1 プログラマ用準備課題

下記の準備課題を順次実行し、MR 空間がどのよう

に構築されているかを考える. 課題 1,2 を行うために用いるプロジェクトは「allFunctions」, 課題 3 を行うために用いるプロジェクトは「simpleLite」とする.

グループで作品を完成させるためには必要となるポイントなので、課題1~2を全て行うことが望ましいが、準備課題に時間をかけすぎると作品を作る時間がなくなる。各課題に設定された所要時間を目安として常に時間に注意しながら進めるようにする。

準備課題が終わり次第,グループ作品のプログラミング作業へと進む.課題3は発展項目用課題なので,作品に発展項目を取り入れない場合は行う必要はない.

# 課題1 オブジェクトの読み込み部分の確認 (所要時間目安:10分)

allFunctions プロジェクトで読み込まれるオブジェクトは,「D:¥MR¥arset¥data¥wrl¥doubleVRML」に記述されている(フォルダではなくファイルである). ワードパッドなどで doubleVRML を開き,書かれた内容を確認する.

特に次の点について、その記述方法をよく確認する - レ

- ・VRML ファイル: #no.1 model から始まる記述.
- ・マーカデータ: marker name, size, center から始まる記述.



# 課題2 オブジェクトの変更(所要時間目安:10分)

課題1で確認した内容を踏まえ、allFunctionsで表示するオブジェクトを変更する.

表示するオブジェクトは、既に data¥wrl フォルダに 用意されているものでも良いが、可能であれば各グル ープで作成したオブジェクトの使用に挑戦すること.

# 課題3 立体視への拡張方法の確認

(発展項目用課題. 所要時間目安:10分)

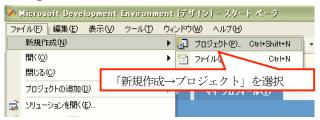
「simpleLite」と「simpleLiteHMD」を比較する.

HMD で立体視するということは、あるシーンに視差情報を追加したものを 2 画面で表示することであることをふまえ、どのように立体表示が行われているのかを確認する. 特に次のような点に注意すること.

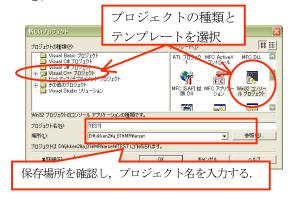
- ・カメラのセットアップ
- ・2 画面表示のために用意された変数
- ・1 フレームごとの描画処理

# ● 参考:新規プロジェクトの作成

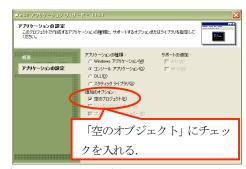
(1) Visual Studio.Net を起動し、メニューバーから 「ファイル→新規作成→プロジェクト」を選択す る.



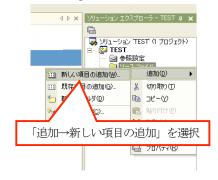
- (2) プロジェクトの種類から「Visual C++ プロジェクト」, テンプレートから「Win32 コンソールプロジェクト」を選択する.
- (3) 保存場所が「D:¥jikken2¥グループ名 ¥MR¥arset¥src」となっていることを確認し、 プロジェクト名を設定して「OK」をクリックする。



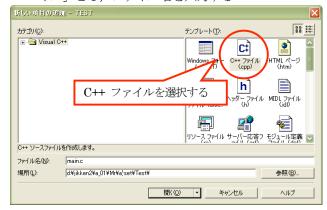
(4) 「Win32 アプリケーションウィザード」ウィンドウの「アプリケーションの設定→追加のオプション」から「空のオブジェクト」にチェックを入れる.

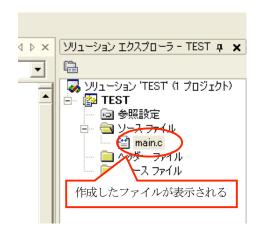


(5) 新たにファイルを追加するには、画面右に表示されたソリューションエクスプローラ内の「ソースファイル」フォルダ上で「右クリック→追加→新しい項目の追加」を選択する.



(6) 開いたウィンドウでテンプレートを「C++ ファ イル」とし、ファイル名を入力する





(7) 「simpleLite」などの単純なサンプルプログラム のソースを作成したファイルにコピーし、実行し て正しく動作することを確認する.

### 9.4.2 モデラ用準備課題

モデリングには引き続き Metasequoia を用いる. VR 実験と表示環境の違いを確認するために短時間で課題を終わらせ、順次グループ作品で用いるオブジェクトの作成に取り掛かること.

# 課題 1 簡単な VRML オブジェクトの作成 (所要時間目安:10分)

- (1) Metasequoia を起動し、立方体や球といった簡単な形状を作成する. ただし、以降の課題を円滑に進めるため、最低2つの形状を使用して3軸方向全てに非線/点対称の形状とする.
- (2) VR 実験の課題と同様、オブジェクトを VRML 形式で保存する.

# 課題 2 作成したオブジェクトを表示させる (所要時間目安: 15 分)

- (1) 課題 1 で作成したモデルをプログラミング用 PC に転送し、プログラムの必要箇所を書き換え (プログラマ担当) て、どのように表示されるか確認する. (保存場所はプログラマ用課題 1 で示した「double VRML」ファイルのあるフォルダ.)
- (2) 作成したモデルと表示されたオブジェクトがど のように異なるかを確認する.

# 10. 第4週目 MR空間の構築2

第4週目は、第3週目の作業の続きを行う. 作品が完成といえるレベルまでできたら近くの TA にチェックをしてもらい、その後の指示を仰ぐこと、

この週も実験終了後にフォローアップの時間として 1.5 時間実験室を使用可能とするので、実験時間内に作業が終わらなかったグループは活用すること.

第5週目に行う VR・MR 作品の発表と体験のための 準備として、各作品について全体の概要・機能・操作 説明などを記述したレジュメを各グループで 1 枚作成 しておくこと.

# 11. 第5週目 VR・MR 作品の発表・体験

第5週目は、VR・MR作品の発表と体験を行う. 実験時間の冒頭に約30分仕上げと発表準備時間を与える.

各グループ約5分程度,全員の前で口頭にて作品を紹介する (PowerPoint を用いても良い). 発表順は TA が決めるので,必ずしも班の番号順ではない.

その後、教員と TA が各グループを回り、作品を見て評価する. VR 作品、MR 作品それぞれで優秀作品を発表する.

全 5 週間の実験期間終了後に、指定日時までに指定の場所に各自最終レポートを提出する.

# 12. レポート課題

レポートは第2週目終了後及び第5週目終了後に, それぞれ中間レポート,最終レポートとして提出する. 以下に指定した項目に沿って作成すること. なお,最 終レポートは中間レポートを包含する形で作成し,提 出する.

本実験におけるレポートの形式と注意点を,以下に示す.これは必要最小限の内容であるので,他に特に述べたいことが別途ある場合は,各自自由に追加して記述しても良い.また,レポートの雛形はhttp://www.rm.is.ritsumei.ac.jp/lecture/jikken2/report.docに用意しているので,必要に応じて各自でダウンロードし、参考すること.

# 【レポートについての注意点】

- 1. 表紙には次のものを必ず全て記述する.
- ・タイトル: メディア情報学実験 2 バーチャルリアリティ
- 提出日
- 学籍番号
- ・クラス及び作業グループ番号

- 氏名
- ・共同作業者名と担当作業 (グループ内の全員について記述する)
- 2. レポートは、左上1箇所に斜めにホチキス止めすること (3箇所で止めない).
- 3. Microsoft Word や LaTex などのワープロソフト を用いて作成する.
- 4. 表紙を除くページの下部中央にページ番号を入れる.
- 5. レポートは必ず「ですます」調でなく、「である」 調で記述する. ただし、「感想」についてはこの限りで はない.
- 6. 報告すべき箇所は論理的に、読む側が理解しやすい文章を心がける.
- 7. 可能な限り図を用いて、見やすいレポートの構成を心がける.
- 8. 自分が担当した個人の作業と、他人の作業及びグループ全体の作業を明確に区別して記述すること.
- 9. (特に新たに知った) 言葉の意味は、しっかり把握してから使用する.
  - 10. フォントスタイル及びフォントサイズを揃える.
  - ・見出しはゴシック体, 本文は明朝体で  $10\sim10.5$ pt.
  - ・見出しのフォントサイズはレベルに応じて変更.
- 11. 誤字脱字は減点対象なので、チェックしておくこと. 提出期限遅れも、勿論減点の大きな要因となる.

# 【レポートの形式】

- 1. 実験の目的・課題
- 2. VR 空間の構築
- 2.1 構築した VR 空間(作品)の概要
- 2.2 VR 空間構築のために必要とした知識・機能とその達成方法
  - 2.3 構築した VR 空間のシステムフロー

シーングラフも含め、図示する.

全てを図示できない場合も、可能な限り図を用いて述べる.

# 2.4 担当作業内容

作業内容. グループで構築した VR 空間と各自の作業内容の関係を明確にする.

[ディレクタの場合]

グループで VR 空間を構築する作業内で発生した問題およびその解決方法,スムーズに作業をするためにどのような指示をしたかを記述する.

[プログラマの場合]

実装した機能及びプログラムコードの記述する

参考にしたファイルがあればそのファイル名と変更 箇所を記述すること

実験テキストに書かれている実装すべき基本項目及 び発展項目に対応した記述も書く

### [モデラの場合]

作成したモデルの名称、その画像とポリゴン数 テクスチャを用いている場合はその画像も載せる 作成上の工夫も記述すること

### 2.5 VR 実験に関する考察

計画を立てたが、実装する予定でできなかった機能 や表示する予定でできなかったオブジェクトに関して 記し、なぜ達成できなかったのか原因を分析する.

また、自らの工夫で達成できた点で、特に記しておきたい技術的分析があればここに記しても良い.

### 3. MR 空間の構築

VR 空間の場合と同様の形式で、以下を記す.

- 3.1 構築した MR 空間 (作品) の概要
- 3.2 MR 空間構築のために必要とした知識・機能とその達成方法
  - 3.3 構築した MR 空間のシステムフロー
  - 3.4 担当作業内容
  - 3.5 MR 実験に関する考察
  - 4. 感想

本実験全体を通しての感想. 自由記述で良い.

### 5. 参考文献

本実験に際して配られたもの以外で、自ら見つけて参考にしたものがあれば記す.

# 付録 A 作品事例の操作方法

# (1) 作品事例 (プロジェクト名: Slime)

操作エリア	操作ボタン ・方向	動作
Joystick (左)	上	Slime が上に移動
	下	Slime が下に移動
	左	Slime が左に移動
	右	Slime が右に移動
十字キー	上	Slime が上に移動
	下	Slime が下に移動
	左	Slime が左に移動
	右	Slime が右に移動
ボタン	9	プログラム終了
	10	Slime が初期位置
		に戻る
マウス	ドラッグ	オブジェクトの回
192		転

# (2) 作品事例 (プロジェクト名: VRTest\_UFO)

操作エリア	操作ボタン ・方向	動作
Joystick	上	UFO が奥に移動
(右)	下	UFO が手前に移動
Joystick (左)	上	UFO が上に移動
	下	UFO が下に移動
	左	UFO が左に移動
	右	UFO が右に移動
ボタン	1	UFO が奥に移動
	3	UFO が手前に移動
	9	プログラム終了
	10	UFO が初期位置に戻る

このプログラムでは、A から F までの文字が描かれたマルチマーカと Hiro と描かれたシングルマーカを利用する.

- (1) マルチマーカを水平な台の上に置き、文字が正しく読める方向からカメラで撮影する. この際、マーカの一部がカメラの撮影範囲内に入っていない場合でも CG は描画されるが、インタラクションの確認のために作業終了までマーカが全て撮影されている状態を保つ.
- (2) シングルマーカを文字が正しく読める向きに保って、カメラの撮影範囲内に入れる.
- (3) シングルマーカを傾けることにより、その傾きに 応じてマルチマーカ上に描画された CG が 1 つ回 転する.

前後の傾き	左右の傾き	アクションを行う CG の 位置(マーカに描かれたア ルファベット)
奥	左	A
	傾き無し	В
	右	C
手前	左	D
	傾き無し	E
	右	F

# 付録 B allFunctions の操作方法