

双方向型ネットワーク対応仮想空間共同構築システム

3D-NeWS (3D-Network World Simulator) Project

和田 健之介¹⁾、和田 佳子¹⁾、中口 孝雄¹⁾、金子 勇²⁾、石井 卓良³⁾、星 和明

1) 有限会社アントラッド

〒630-8101 奈良市青山4丁目4番地2-1 E-mail: kwada@m3.kcn.ne.jp

2) エクス・ツールズ株式会社

3) デジタルファッション株式会社

ABSTRACT. Our 3D-NeWS system is designed for children and the general public to utilize the 3D network environment, create their own original 3D world with congenial friends, and publish them on WEB.

It is the extremely attractive and challenging work for us to develop the system weaving our original new technologies such as smooth animation technique based on quaternion math, super fast physical simulation, and so on into 3D network system.

This Japanese little story is the record of our long and hard struggle...

1. プロジェクトの目的

本システムの名前の双方向型ネットワーク対応仮想空間共同構築システムは、ユーザー自らが創造発信の場となって、その輪を広げて行くためのコミュニティ・システムを表現しています。気の合ったユーザー同士がネット上で集まって、皆で相談しながらコミュニティを作っていければ、今までのゲーム開発者達が考えてきたお仕着せの世界観とは全く異なったオリジナルのワールドが生まれてくるかもしれません。このような作業過程は、ある意味ではシナリオが徐々にできて行くユーザー参加型の映画のようにも捉えられるし、また多人数参加型の箱庭の作成過程だと考えることもできます。

近年よく言われていることですが、最近の子供達は予め出来上がったゲームなどを短期間で効率よくこなす事にはとても長けていても、作文や絵画、作曲など自分の考えを表現する能力が減少してきているように感じられます。また、ゲームの中には過激なほど戦闘的であったり、道徳的に問題のあるシナリオが多く、さまざまな社会的弊害が生じてきています。ロールプレイングゲームでは、よその家を訪ねたときには、勝手に筆筒や宝箱などに隠してあるアイテムを探しだして、それを一方的に集めていくと得をするような仕組みになっています。また格闘技ゲームはプロトタイプ的になってしまっていて、必殺技を繰り出すための複雑なボタン操作を覚えるだけのゲームになってしまっています。このような仮想の世界にのめり込んでしまうと、自閉的なゲームオタクになってしまったり、ホームレスの人達にゲームの必殺技を試してみたり、といった問題が生じることがあります。このような問題が起きるたびに、マスコミはコンピュータ・ゲームの弊害を画一的な一般論で論じたりしています。

しかし、ゲームのシナリオや方向性が問題なので

あって、コンピュータ・ゲームという手段が否定されるべきではないと思います。子供達が持つさまざまな潜在能力を引き出すための活用の場として、ゲームという手段は極めて有効に活用することができます。従来型のエデュテイメント・ソフトは、現場の教育の延長としての補助教材であったり、音や動画を使ったマルチメディア教材がほとんどで、あらかじめ達成目標やシナリオが与えられているので、使いきりの個人的な補助教材として利用されています。

一方で、インターネットの普及によって自分達の意見を交換する場がかつてとは比べ物にならないくらいの規模で提供されています。遠く離れたユーザー同士がリアルタイムで情報を交換することのできる利便性と、コンピュータ・ゲームというエンターテインメント性をうまく融合すれば、殺伐としたネットワーク型対戦ゲームなどとは異なる、全く新しいジャンルを開拓できるのではないかと考えています。特に現在のエデュテイメント・ソフトには欠けている共同作業の大切さに焦点をあてることによって、アイデアの交換や意見の調節などといったプロセスを取り込むことができます。このようなプロセスは、従来から理科の実験をするときのグループ学習などで取り入れられてきましたが、これをネットワークのパワーをうまく利用することによって、もっと大きな規模でかつ動的なコミュニティの中で活用することができます。

このように、現在のエデュテイメント・ソフトには欠けている共同作業の大切さに焦点をあてることによって、アイデアの交換や意見の調節などといったプロセスを取り込むことができるだけでなく、ユーザーの潜在的なアイデアを引き出すことのできるアイデア・エンハンサーとして活用できるように設計されています。

このような機能を使って、ユーザーどうして3D

データを交換するショップや不思議の国の体験館、ボートの浮かぶ池や遊園地などの3Dワールドを作成し、そのコミュニティの中のユーザー同士で夢を膨らませながら、徐々に自分達の理想とする世界のシナリオを描いていくことができます。自分達で作上げていくだけでなく、他のコミュニティが作ったワールドを時々訪問することも一つの楽しみにもなるし、自分達のワールドを創造する時のヒントになるかもしれません。このときに、初めて訪れたユーザーに対しては、コミュニティのオブジェクトは見たたり触ったりするだけで、編集ができない等のオーサリング・レベルの制御も、私達システム開発者が上から決め付けるようなものではなく、コミュニティの同意によって決められるようにすれば、主体はあくまでも開発者ではなくユーザーであり、あらゆる決定権をコミュニティに委ねることができます。

また、3Dデータのリサイクル・システムを活用して、3Dコンテンツ・クリエイターの為に作品発表の場を提供したり、3Dカタログのポータルサイトとして機能させたり、ユーザーのニーズを取り入れた双方向型形状デザイン・システムなどに発展して行くことが期待されます。

2. プロジェクトの戦略

この未踏ソフトウェア創造事業のプロジェクトを進めるにあたって、私達はまず最初に、全く性格の異なった二つの道のどちらを選択するのか、という大きな分岐点に立っていました。一つはプロジェクトの提案者である和田健之介・和田佳子・中口孝雄がATRで長い間研究を行ってきたバーチャコベットのプロジェクトを出発点として、これをさらに改良・発展させて、より安定なシステムに仕上げるという安全かつ着実な戦略です。もう一つは、基本的な理念のみを残して、描画システム、基本機能そのものも全く新しいものを一から設計し直すという、短期間プロジェクトにしては極めてリスクの高いチャレンジャーな選択肢です。

未踏プロジェクトの契約説明会が10月11日に開かれ、その後、新メンバーとの交渉や、プロジェクト管理会社の設立などプロジェクトを軌道に乗せるまでの手続きに忙殺され、第一回のプロジェクト会議は10月の末日でした。この時点では、まずバーチャコベットの弱点でもあった3Dコンテンツを高品質なものに差し替えるということが決まりました。しかし、メンバー全員が揃った11月11日からの第2回目のプロジェクト会議では、各メンバーの技術要素や、それぞれがチャレンジしてみたい課題などについて、多くの角度から分析を行った結果、描画システムを今までのDirectXからOpenGLベースに変更して、より高度な3Dの機能を実装することに決定しました。このときに必須要素として、高速描画システム、極めて少ないキーフレームによるアニメーション補間システム、3Dワールドの物理シミュレーション、アバターの高速アクティブ・センサー、3Dモデラーのワールドへの埋め込み、物理シミュレーションを管理する通信制御システム、サウンドシステムなどの仕様が詳細に検討されました。これらの全てを2月末日までという短期間で開発できるのか、それぞれの課題をクリアするためにどのような問題があるのか、について

何度も議論を重ねた結果、期日までに間に合うのかどうか、ということより、このような目標に向かってとにかく突き進んでみたい、というメンバー全員の願望がとて強くなりました。最終的にはプロジェクト・マネージャーの竹内先生にデモを行って、私達が持っている技術要素やプロジェクトへの熱い思いをご理解いただいて、第2の選択肢を選ぶことになりました。

これらの課題を着実にこなすために、月に1,2回は4~6日の集中合宿を行い、各メンバーに与えられた課題のクリア、各技術要素の融合などを計り、月に一回は竹内先生にデモを行って進捗状況をご報告してご指導いただきました。あまりにも過激なスケジュールであったため、何度かつまずきそうにはなりましたが、結果的には2月25日の最終デモまでに、上記の目標を全てクリアすることができました。

私達メンバー全員に共通していたことは、いくら理念やモデルがしっかりしていても、『動いてなんぼのもの』という実践主義が徹底していたことです。このためいくら優れたアルゴリズムでも、高速化が不可能なものは容赦なく切り捨てられました。最終的にはそれぞれの機能モジュールが全体の何%を占めているのかチェックすることにより、CPUを多く占有するモジュールの担当者はモジュールの再設計を行い、再び最適化を計るといったプロセスを何度も繰り返すことによって劇的に高速化が計られました。モデルデータも高速化や物理演算との兼ね合いのために何度も作り直しを余儀なくされました。

このように徹底したプロ意識を持った優秀なプログラマーといっしょにプロジェクトに取り組むことができたのは、正に未踏ソフトウェア創造事業に採択していただいたおかげであり、竹内先生の激励とご指導のおかげだと感謝しております。

このようなまたとない機会に恵まれた幸運を大事にして、次のステップに繋げていきたいと思っています。

3. プロジェクトの成果の概要

本システムは、さまざまな先端的技術要素を組み合わせた総合的なシステムで、基本システムのネットワーク対応型仮想空間システムに、次のようなオリジナルの技術を付加することによって、従来のWEB3Dやネットワークゲームにはなかった斬新な機能を持つように構成されています。

- クォータニオンを使った滑らかなアニメーション補間機構
- アクティブ・センサーを搭載した自律型アバター
- オリジナルの高速演算方式によるワールドの物理シミュレーション
- アバターによるワールドオブジェクトのカット機構
- 物理演算を含むワールド・イベントの同期保証を行う通信制御システム
- ドラッグ&ドロップによるワールドオブジェクトの動的追加機能
- オリジナルのワールドを作成するための

編集機能

これらの技術要素の概略を、実際の開発ステップの順に従って、解説を行っていきたいと思います。



本システムは3Dのネットワーク・システムであるため、文章やスクリーンショットではなかなかの確にシステムの特徴を伝えることは難しいので、実際に Windows マシンで動作させながら解説を読んでもいただければ幸いです。

システムの推奨環境は、

OS : Windows98, Windows98 Second Edition, Windows ME, Windows2000

CPU : PentiumII 500MHz 以上

メモリ : 128MB 以上

HD : 空き 100MB 以上

グラフィックス:3D アクセラレータ (OpenGL ドライバ)

サウンド : DirectSound 対応サウンドボード

(DirectX7 以上のドライバ)

ですが、グラフィックス・アクセラレータのないマシンでも、それなりのフレームレートで動作します。

● クォータニオンを使った滑らかなアニメーション補間機構

STEP1 フォルダーの中の ANGEL.EXE を起動すると、次の図のような画面が現れて、ANGEL が歩く動作を繰り返します。通常このような滑らかなアニメーションはモーション・キャプチャーなどを行って、実際に人が歩いたときのデータを多数のセンサーを使って取り込みます。その後、取得に失敗したデータを手作業で補う等、多くの手間をかけて編集してから、アニメーション・データとして再生できるようになります。しかし、この ANGEL の歩行のモーションは、3D クリエーター (プロジェクト・メンバーの星氏) のつけたわずかなキーフレームのみで、その間のアニメーションは全て、数学的なアルゴリズムを使って、リアルタイムで自動的に補間されています。

驚くべきことに、この『色っぽさ』さえそこはかと感じさせるアニメーションは、下図に示すように、僅か4つのキーフレーム (静止ポーズ) のみから構成されています。アニメーションの補間は各関節の回転角に対して行われ、クォータニオンと呼ばれる4元数 (ハミルトンによって考案された4次元版の超複素数) の数学が使われています。その単位4元数が作る超球面上で、キーフレームを表す代表点が滑らかに繋がるように数学的制約を課して、接続しているために、極めて少ないキーフレームから滑らかなアニメーションを自動的に生成できるようになっています。

本システムは、その接続方式を複数実装していて、『Interpolation』メニューから、そのいくつかを選んで、補間方式の違いを実際にアニメーションで確認することができるようになっています。

未踏プロジェクト以前に開発をしていたバーチャコ

ペットでは、DirectX のライブラリに組み込みの補間方式を使っていたので、アニメーションがぎこちなくカクカクとした動作をしていました。『Interpolation』メ



ニューから『Q-Slerp』を選べると、その方式でのアニメーションに切り替わります。

未踏プロジェクトでは、基本描画ライブラリを OpenGL にしたために、アニメーションの基本ライブラリも全て自前で開発しなければならないため、クォータニオンの基本演算を始め、単位クォータニオン上での指数・対数関数、微分係数、各種補間方式のアルゴリズム、等を定理や数式のチェックから始めて、プログラムへの実装、動作チェック等を行うことになりました。このおかげで、さまざまな特徴を持った補間方式を実装することができるようになりました。

『Interpolation』メニューから『Q-Bspline2』を選ぶと、『Q-CatmullRom』より滑らかさが増します。

『Q-Bspline3』を選ぶとさらにアニメーションが滑らかになります。しかし、ANGEL の歩行動作の持つ、いわゆる『色っぽさ』は『Q-CatmullRom』のときが一番際立っていて、足の内股のひねり動作や手先の滑らかなそり返し動作が表現されています。これは、数学的には補間を無理に行ったための『オーバーシュート』の副作用です。つまり、車がカーブを曲がるときに、スピードを出しすぎていて、センターラインから外れて外側に大きく回りこむときの現象と似ています。できるだけ滑らかに繋ぐには、『Q-Bspline2,3』が優れていますが、本来の制



御点を通るとは限らないため、アニメーターがつけた元々のポーズが再現されません。一方、

『Q-CatmulIRom』方式では、本来の制御点を確実に通過するため、キーフレームではアニメーターのつけたポーズをとりますが、各制御点を滑らかに繋ごうとするために『オーバーシュート』が生じてしまいます。一長一短ですが、本システムでは、各関節毎に、また各キーフレーム毎に補間方式を適宜切り替えることができます。

このアニメーション・システムを使って、キャラクターを3Dフィールド上で操作できるようにしたものが、STEP2 フォルダの Walker.exe です。

キーボードの矢印キーを押し続けることにより、前(↑)、左(←)、右(→)、手前(↓)にキャラクターが歩きます。また、手早くキーを2度続けて押すと走ることができます。視点はシフトキーを押しながら、矢印キーを押すと変更することができます。

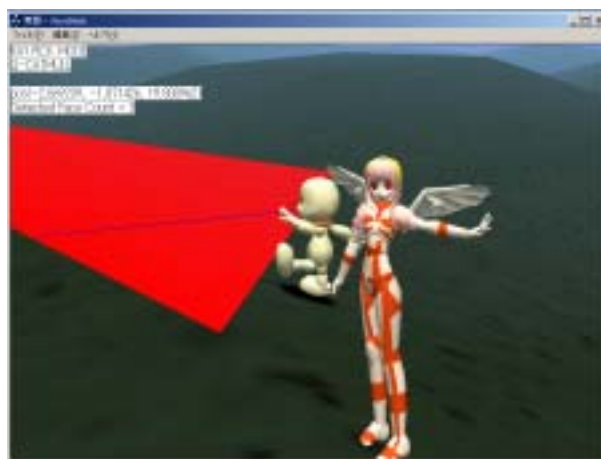
このようなアニメーション再生方式は、モーション・キャプチャー方式のデータに比べて圧倒的にデータサイズが小さいので、3Dのモーションデータをネットワーク上で瞬時に送信したり、その場で新たにオリジナルのモーションデータを生成したりするためには必須の技術要素です。さらにこの方式のデータを詳しく調べてみると、『歩く』、『走る』、『拾う』、『ジャンプする』、等のそれぞれのモーションに含まれるキーフレームのいくつかに極めて似ているものがあることに気がきます。この観察をさらに進めて、基本となるポーズはどのようにしたら得られるのか、いくつぐらい必要なのか、またその基本ポーズをどのように組み合わせたら各アニメーションになるのか、といった問題を突き詰めると、ある興味深いシステムが浮かび上がってきますが、これは項目を改めて紹介することにいたします。

● アクティブ・センサーを搭載した自律型アバター

アバターが3Dの世界の中で動き回るときに、ワールド・オブジェクトを認識するための機構にはいくつかの方法がありますが、その代表的な手法にマップ方式があります。これは3Dのオブジェクトのデータとは別にアバターが歩き回れるかどうかを示す『マップデータ』を別に作成しておく手法です。マップデータには、各格子点での地面の高さやテーブルの高さが書き込まれているので、アバターの位置から、どの格子点に相当するか計算して、マップデータを読み取れば、高さの情報を得ることができます。家の壁のように高さが身長より十分高ければ衝突したものと見なして、壁へ潜り込まない様になります。車道から歩道に上がるときや、階段のように小さな段差であれば、そのマップの高さを採用することにより、自動的に登ることができます。マップ方式では、マップデータを読み取るだけなので極めて高速ですが、本システムのようにネットワーク上で3Dオブジェクトが新たに作られたり、壊されたりするような動的な世界では、予めマップデータを作成することができません。また、2階建ての家やビルのような場合は、マップデータが複数必要になります。このため、本システムでは、アバターにアクティブ・センサーを実装することにより、アバターがリアルタイムで3Dの環境情報を取得して、その情報を使って衝突を検出するようにしています。

未踏プロジェクト以前のバーチャコペットでも、自律センサー方式を採用していましたが、センサーの実現に DirectX の RayPick 関数(本来レイトレーシング用の関数)を応用していました。本システムは描画ライブラ

りを OpenGL に変更したため、視覚・接触センサー用にオリジナルの基本ライブラリを開発しました。



STEP3 のフォルダの中の世界Walk.exe を起動すると、BOUSE と ANGEL の2体がアニメーションを繰り返します。このアニメーションは前項で説明したアニメーション補間をチェックするためのもので、本項目にはあまり関係がありません。矢印キーボードを押すと BOUSE を動かすことができます。このとき、ファンクションキーの F12 を押すとデバッグ表示モードに切り替わり、BOUSE の口からレーザーのような青い光線が出ていくことがわかります。この光線はセンサーのチェック用の半無限の直線で、この光線と交わった地面のポリゴンが近い順にソートされて、赤から青に順に塗りつぶされるようになっています。

ファンクションキーの【F11】を押すと、地面の高さの検出方式がマップ方式からセンサー方式に切り替わり、地面の高さは BOUSE が歩くたびに自ら検出するようになります。ファンクションキーの【F1】を押す度に、前項で説明したようにアニメーションの補間方式が切り



替わります。

このサンプルテストの段階では、一本しかセンサーを実装していませんが、最終仕様では、左図のように、下方方向に3本、前進方向に3本、そしてジャンプするときは、ジャンプ方向に1本のセンサーを使用しています。また、それぞれのセンサーは計算の効率化のため、必要に応じ

てアクティブになるため、無駄なセンシングはしないように工夫されています。下方向に3本ものセンサーを使っているのは、階段などを登る場合に、段と段との繋ぎ目に僅かな隙間があっても他の2本のセンサーですり抜けないようにするためと、どちらかの足が引っかかっていれば、崖のふちのぎりぎりまで歩けるようにするためです。しかし、前進方向には頭と両肘にしかセンサーがないため、それより下や上にあるでっぱりには反応することができず、すり抜けてしまいます。より精密な衝突判定を行うためには、より多くのセンサーが必要になりますが、それだけ計算コストを必要とします。また、4段階のフィルターによって、予め調べる必要のないオブジェクトをはじく事により計算の効率化を計っています。

● オリジナルの高速演算方式によるワールドの物理シミュレーション

未踏プロジェクト以前のバーチャコペットにはなかった、本システムの大きな特徴の一つに、ワールドの物理シミュレーションがあります

まずは STEP4 のフォルダーの中での VCTest.exe を起動してみてください。カラスの上にいるいろいろな物が振ってき



て、ぶつかって跳ね返ります。スペースキーを押すたびに、落ちてくるものの種類やぶつかってくる角度が次々



と変わります。矢印キーでカメラアングルを変更することもできます。

特に図のように椅子が落ちてくるときに、椅子の脚が微妙な衝突を行って跳ね返る様子を観察してください。従来であれば、このような複雑な物理シミュレーションは、スーパーコンピュータを長時間走らせた後で、短時間のムービーによって確認することくらいしかできませんでした。物理学に忠実に従って、撃力ベースの計算を行えば、確実に計算コストは膨大なものとなり、また多くの場合、発散の問題をかかえてしまいます。本システムでは、ペナルティ法にオリジナルの改良を加えた衝突判定アルゴリズムを使っているため、高速な衝突シミュレーションを可能にしています。このため、通常の物理学のように、力 ⇒ 加速度 ⇒ 速度 ⇒ 位置 というプロセスは踏まずに、予測位置 ⇒ ポリトープ k-dop 方式によるバウンディング・ボリユームの計算 ⇒ めり込み



量（ペナルティ）の計算 ⇒ 反発速度の計算 というサイクルを繰り返します。また、センサーと同様、多段階のフィルターによって無駄な計算を極力避けて計算の効率を上げています。また、独特の解釈によって、角加速度や動摩擦に相当する物理量を導入することによって、外見上は通常の物理学と極めて酷似した超高速な衝突シミュレーションが可能となっています。

STEP5 のフォルダーの中での WorldWalk.exe を起動すると、キャラクターのキーボード・コントロール、アニメーション補間、物理シミュレーションが融合したサンプルを試すことができます。前のサンプルと同様、矢印キーボードでキャラクターは歩き、2度押しで走ります。うまく操作すれば、サッカーゴールにボールやリングなどをシュートすることができます。また、サッカーゴールを引きずって移動させることもできます。

このシステムがネットワークで繋がれば、サッカーや缶蹴り鬼、ダルマ落としなどの遊びや、自分たちでルールを考えたオリジナルの遊びなど、いくらでも応用ができます。

STEP6 のフォルダーの中での NetWalker.exe では、オブジェクトの近くまで行ってから、キーボードの【C】を押すとオブジェクトを持ち上げることができます。また、キーボードの【X】を押すと、持っているオブ

ジェクトを投げることができます。

ボールを拾って、階段を登り、屋上からボールを放り投げてみてください。

このように、3Dワールドにリアルタイムで動作可能な物理シミュレーションを実装することによって、3Dワールドの可能性が限りなく広がっていきます。

● アバターによるワールドオブジェクトのカット機構

前項の物理シミュレーションによって、3Dワールドの可能性が大きく拡張しましたが、本システムは、さらにワールドの自由度を増すために3Dオブジェクトを自在にカットする機構を導入しました。

未踏プロジェクト以前のバーチャコベットでは、ネットワーク対応の3Dワールド・シミュレーターとオリジナルの3Dモデラーの2つのシステムからなり、モデラーで作成した3Dデータをドラッグ&ドロップによってワールド・シミュレーターに持っていったり、逆にワールド・シミュレーターのデータをモデラーに取り込んだりすることができました。この機構によって、ネットワーク上で3Dデータをコミュニティで共有化したり、他のユーザーが作ったデータを取り込んで改良し、再びネット上に戻すことによって、3Dデータのリサイクルを行うことができました。しかし、一般のユーザーにとっては、モデラーで3Dモデルを作ること自体に負担を感じる場合があり、また2つのシステムを行き来する煩わしさもありました。

そこで、本システムでは3Dのモデリング操作の中でも最も重要なカットの機能を3Dワールドの中で行えるようにしました。当然ながら、現行のゲームシステムなどのように、予め決められた場所にカット済みのオブジェクトを仕込んでおくようなものではなく、任意のオブジェクトがカットの対象となり、カット位置も自由に選ぶことができます。

ただし、殺伐としたワールドにならないように、キャラクターはカットの対象からはずしてあります。また、ユーザーがワールドを構成するときに、カット不可能なオブジェクトを作成することもできるようにしてあります。

左の図は、最終的なプログラムに実装されたカット機



能を使って、3Dワールドの中の竹を様々な方法で切っ

ている様子です。斜めに切った竹を持ち上げて、さらにそれをジャンプしながら縦割りにしている瞬間です。実際にカットする方法は、次章の操作説明を参考にして下さい。

カット可能なオブジェクトとして建物をワールドに持ち込めば、建物全体をカットすることも可能です。

● 物理演算を含むワールド・イベントの同期保証を行う通信制御システム

今まではスタンドアローンでの機能説明でしたが、これらの機能をネットワーク上に拡張することにより、複数のユーザーが3Dワールドを共有することが可能になります。未踏プロジェクト以前のバーチャコベットでも、ネットワーク上でのイベントの同期保障や、セッション・ホストのマイグレート機能により、セッション・ホストが故障しても、瞬時に代理セッション・ホストが残りのクライアントから選択されて、最後の一台になるまでセッションが継続する機能を持っていました。しかし、物理シミュレーションが実装されていなかったため、ワールドのイベントには、キャラクターの移動以外には、ドアの開閉、ライトの点灯、日照角度などシンボル化できるイベントが多く、通信コストも大きくはありませんでした。しかし、本システムでは、物理シミュレーションを実装しているために、キャラクターが移動していても、キャラクターと衝突したオブジェクトは飛び続けていたり、オブジェクトどうしが衝突したりしているので、移動しているオブジェクトの位置や角度の情報を通信しなければなりません。

ローカルに物理シミュレーションを行うと、計算誤差やマシンの性能差によるループ回数の違いにより、衝突時の物体の位置が各クライアントでほんのわずかなだけ異なる可能性があります。オブジェクトが角どうして衝突する場合などは、わずかな位置のずれのせいで、その後の物体の軌跡が大きく異なることがあります。このように物理シミュレーションとしては決定論的なシステム構成ではあっても、ネットワーク上で総体として捉えると、本質的にカオティックな要素が存在し、各クライアントがパラレル・ワールド化して、異なった世界をシミュレートしてしまう危険があります。

このため、ネットワークへの実装の第一段階として、完全な非対称のサーバー・クライアントの構成にして、各クライアントのキャラクターの位置の更新などは、予めクライアントの中でセンサー情報を使って各クライアントの中で計算を済ませ、更新されたキャラクターの位置をサーバーに逐次送信します。一方、可動オブジェクトの位置や角度の情報、オブジェクト間、およびオブジェクト・キャラクター間の衝突シミュレーションはサーバーが管理しています。サーバーが行ったワールドの物理シミュレーションの結果を各クライアントに送信することによって、パラレル・ワールド化を防いでいます。

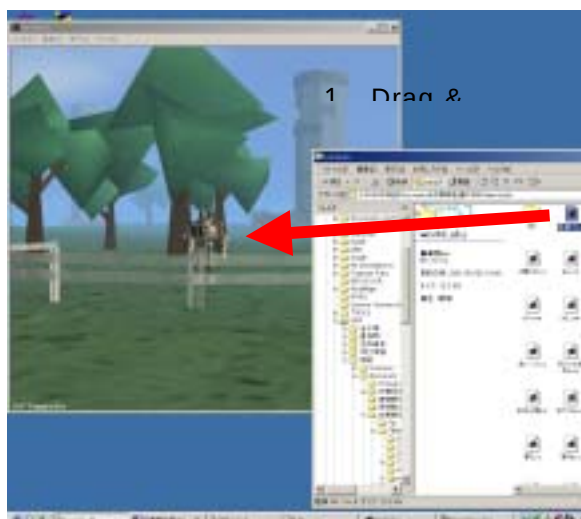
上図では、ネットワーク接続によって、多くのユーザーがボーリング大会を開いているところです。ちょうどロボットが二投目を投げた直後で、他のユーザーは思い思いのボール？を持って自分の順番を待っています。物理シミュレーションが実装されているおかげで、特別にボーリング用の衝突プログラムを用意する必要はありません。このように、ネットワーク上のユーザー自らが『遊び』を創り出せることが、本システムの大きな特徴です。

しかし、このようなシステムの構成では、サーバーへ

の計算負荷が大きくなるので、物理計算の分散負荷制御が次の課題となりますが、この問題については、最後の章でもう一度議論しようと思います。

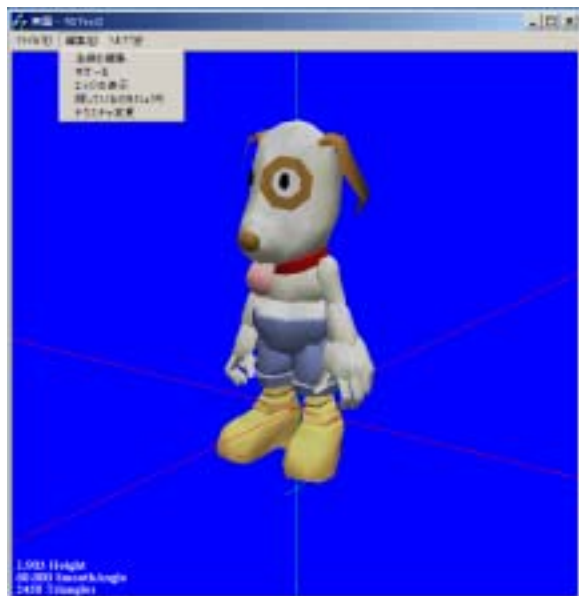
● ドラッグ&ドロップによるワールドオブジェクトの動的追加機能

本システムは、3Dのデータファイル(独自形式)をワールド・シミュレーターにドラッグ&ドロップするだけで、リアルタイムで新規オブジェクトとして追加することができます。この機能はネットワーク化されているので、ネットワーク上のユーザーは新規オブジェクトを共有資産として活用することができます。また、いつでも必要なオブジェクトをオンラインで追加できるので、ユーザー同士が相談しながら、徐々に自分たちのワールドを築いていくことが可能です。



実際に試してみましょう。FINAL フォルダの中 NetWalker.exe を起動してキャラクターを選択してから、FINAL フォルダの中の world_obj フォルダの中にある、適当な *.rs ファイルをワールド・シミュレーターにドラッグ&ドロップしてみてください。ドロップした瞬間に自分のアバターがそのオブジェクトを持っていることがわかります。

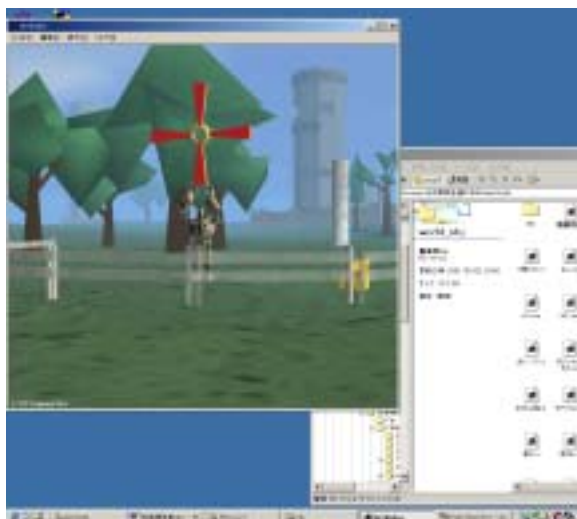
下の左図は、風車の羽のファイルをワールド・シミュレーターの上に、マウスでドラッグ&ドロップしようとしているところです。右図はドラッグ&ドロップした直後で、自分のアバターであるロボットが風車の羽を掲げています。



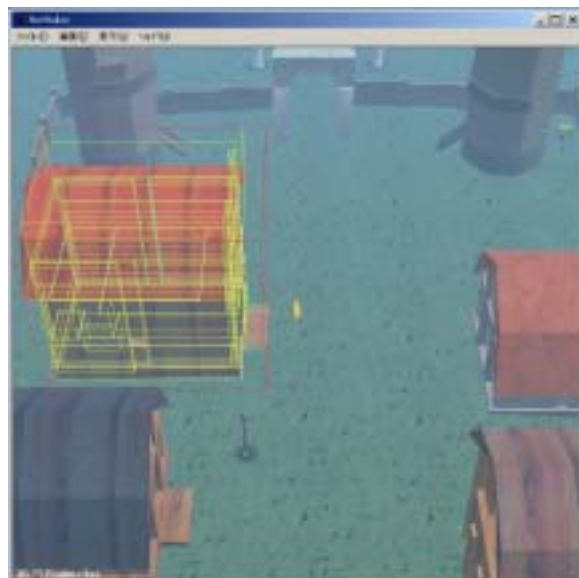
ネットワークに繋いでいれば、他のクライアント上では、自分のアバターがオブジェクトを持ち上げている様子が観察されます。後は好きな所に持って行って、置くなりカットするなりしてみてください。

● オリジナルのワールドを作成するための編集機能

本システムはオリジナルの3Dファイル形式を使用していますが、一般のユーザーが入手可能なフリーソフトウェアのファイル形式を元にしています。3Dの形状データは『六角大王』、アニメーション・データは『HuManMDL』を使っています。六角大王のデータは法線情報を持たないため、まず六角大王で作成した *.rok ファイルを、オリジナルのツールを使って法線情報を修正して独自の *.rs ファイルとして保存します。



このツールは、MeshConverter フォルダの中の MeshConverter.exe です。このツールを起動して、ファイル・メニューの『開く』で六角大王の形状データを読み込んだ後、編集メニューから『法線の編集』でスムーズに表示させる臨界角度を設定します。マウスの右ボタンで回転したり、中ボタンでズームしたりして、法線が反転しているメッシュ・グループを探して、そのメッシュ・グループをマウスの左ボタンでクリックすると法線が反転します。必要に応じて、スケールを変えてサイズを変更したり、テクスチャを変更したりした後、ファイル・メニューから『名前を付けて保存』を選択して、*.rs ファイル形式で保存します。この形式に変換すると、ワ



ールド・シミュレーターにドラッグ&ドロップしたり、編集モードで取り込んだりすることができます。また、個々のオブジェクトの配置情報は、NetWalker.exe のファイル・メニューの『Worldを開く』や『Worldの保存』によって、ワールド・ファイルの読み込みや保存ができて、FINAL フォルダの中 World フォルダの中に *.rsw というファイル形式で存在しています。本システムは、ワールド・シミュレーターが編集機能も同時に兼ね備えていて、ファンクションキーの【F12】を押すと編集モードに変わります。トグルになっているので、もう一度押せば、ワールド・シミュレーター・モードに戻ります。ネットワークに接続しているときは、編集モードに移行できません。



編集モードでの操作の詳細は、次章の操作説明を参考にしてください。

4. ワールド・シミュレーターの操作説明

FINAL というフォルダの中 WorldWalker.exe ファイルをダブルクリックすると、ワールド・シミュレーターが起動して、最初に下図のようなキャラクター選択画面が出できます。ここで、自分が操作したいキャラクターの絵をクリックして選択し、【OK】ボタンをクリックします。もし、

ここで、キャラクターを選択しないで【OK】ボタンをクリックすると、左上の天使が自動的に選択されます。

【OK】ボタンがクリックされると、ワールド・データが読み込まれて、登場音楽とともに選択されたキャラクターがその中に現れます。

キャラクターを動かすにはキーボードの矢印キーを使います。はじめの状態では、上矢印キー【↑】を押し続けると、画面の向こう側に歩いていきます。下矢印キー【↓】を押すとこちらを向いて手前に歩いてきます。

右矢印キー【→】を押すと右側を向いて右側に、左矢印キー【←】を押すと左側を向いて左側に歩きます。

同じ矢印キーをポン、ポンと2度続けて早く押す(2度目は押し続ける)と、それぞれの向きに走ります。

ファンクションキーの【F1】キーは、キャラクターの移動操作の切り替えを行います。【F1】キーを押すと、キャラクターの後方からキャラクターの進む方向を見た視点になり、上矢印キーで前進し、右矢印キーで右方向に少し回転し、左矢印キーで左方向に少し回転します。また、下矢印キーを押すと、向こうを向いたまま、画面の手前側に後ずさりします。このモードでは、例えば上矢印キーを押して前進したまま、左矢印キーを押すと、左斜め前方に歩きます。本システムでは、この移動モードを『アナログ・モード』、元々の90単位で移動する移動モードを『デジタル・モード』と呼んでいます。

その後、【F1】キーを押すたびに、キャラクターの移動操作のモードが交互に切り替わります。

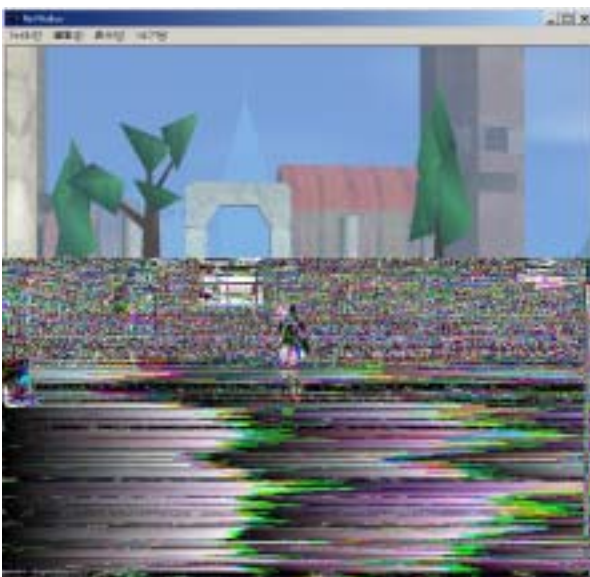
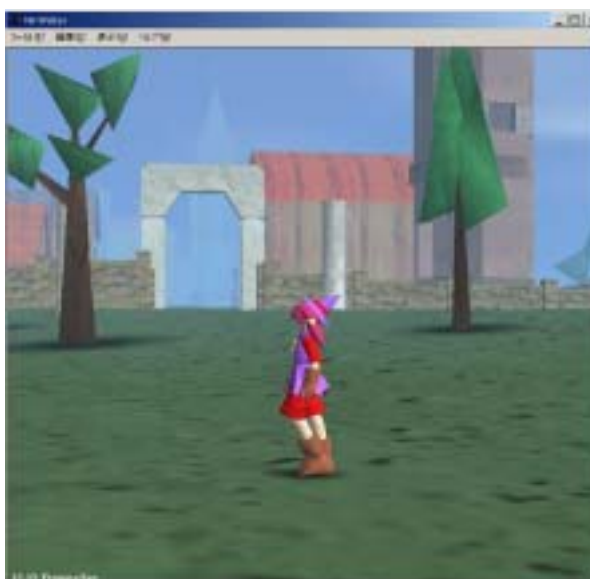
【Z】キーを押すとジャンプします。ただし、羽の生えた、天使や鳥などのキャラクターの場合には【Z】キーを押し続けることで飛ぶことができます。

ワールド内のオブジェクトに近づくと、自動的に赤や黄色のボックス型の選択マークが付いて選択対象に選ばれていることを示します。このとき、【C】キーを押すと選択されたオブジェクトを持ち上げることができます。

【X】キーを押すと持っているものを投げます。ただし、持っている状態で別のオブジェクトに選択マークがついている場合は、持っているものをその上に置くことができます。高度な技としては、持っているオブジェクトをジャンプしながら高い物の上に置くこともできます。

持ち上げているものを【A】、【S】、【D】の3種類のキーでカットすることができます。また、キャラクターが何も持ち上げていないときには、選択マークのついたものを【A】、【S】、【D】の3種類のキーでカットすることができます。

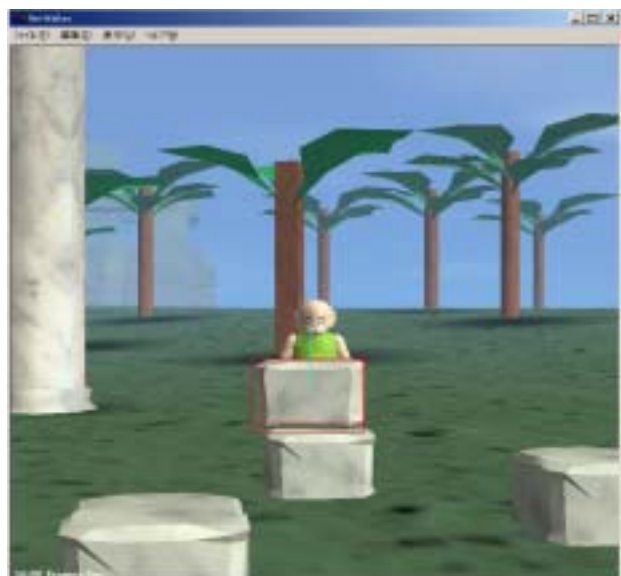
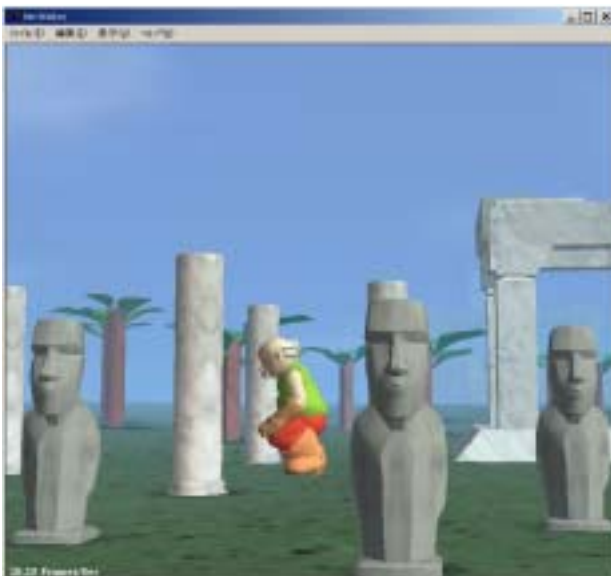
階段の狭い隙間などに入り込んでしまって、どうしても抜け出せなくなったときには、ファンクションキーの【F3】キーを押すと、上方向と前進方向のセンサーが解除されて、壁などを通り抜けられる緊急脱出モードに切り替わります。もう一度【F3】キーを押すと、元のモードに戻ります。



ファンクションキーの【F4】キーを押すと、背景に流れている音楽が切り替わります。何回か押すと音楽の無いモードになり、さらに押すと最初の曲に戻ります。

【Shift】キーを押しながら左または右の矢印キーを押

すと、カメラの回転と共にキャラクターの向きが変わります。【Shift】キーを押しながら上または下の矢印キーを押すと、キャラクターの位置はそのまま、カメラの位置を上下に移動できます。下の右図のようにカメラを下に向けすぎたときは、自動的に地面が半透明で表示さ



れます。また、カメラとキャラクターの間にオブジェクトがあった場合も、遮っているオブジェクトを半透明で表示します。

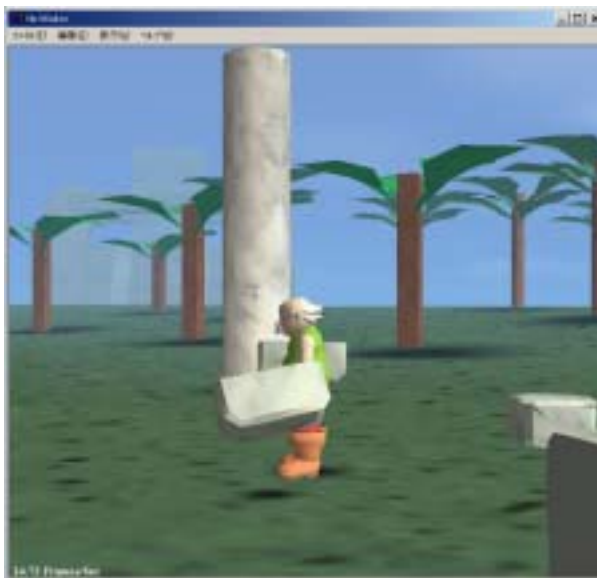
ファンクションキーの【F2】キーを押すと、デバッグ表示になり、センサーの状態やアクティブなセンサーの方向等を観察することができます。もう一度【F2】キーを押すと、元の状態に戻ります。

既に前章の機能概要で説明したように、ワールド・シミュレーターに3Dオブジェクトのファイルをドラッグ&ドロップするだけでネットワーク上に新規オブジェクトを追加できます。

ファンクションキーの【F12】キーを押すとワールドの編集モードに移ります。もう一度押すと、元のワールド・シミュレーター・モードに戻ります。ただし、ネットワーク接続時は編集モードに移行することはできま



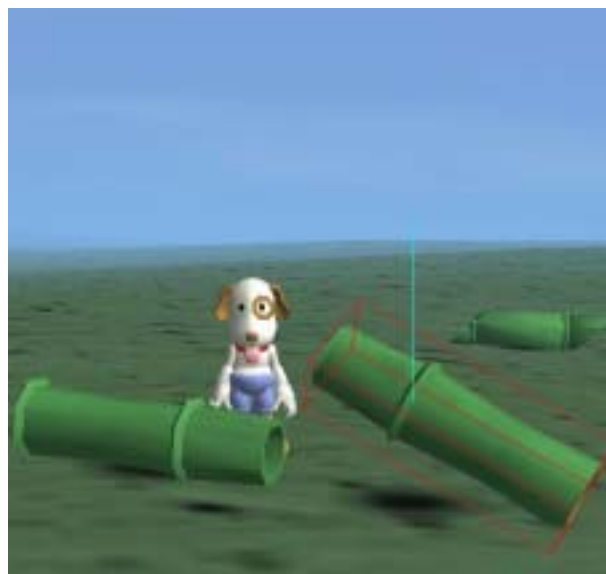
キーを使って読み込むと、家などのように、移動もカットもできない建物オブジェクトとして追加されます。一方、ファンクションキーの【F10】キーを使って読み込むと、植物や柱のように、キャラクターが衝突しても移動はしないけれど、引っっこ抜いて移動したり、カットしたりすることのできる固定オブジェクトとして追加されます。最後に、ファンクションキーの【F11】キーを使って読み込むと、椅子やリングのように、キャラクターが衝突したら飛んでいくし、持ち上げて移動したり、



せん。

編集モードでは、オブジェクトをマウスの左ボタンでクリックしてから、【Delete】キーを押すと、オブジェクトを削除できます。また、ワールドに新しいオブジェクトを追加するには、前章の機能概要で説明したように、あらかじめ追加したい3Dオブジェクトを

MeshConverter.exe を使って *.rs ファイルに変換して、FINAL フォルダの中 world_obj フォルダの中にファイルを移して置きます。追加する前に、追加オブジェクトに対して、キャラクターが衝突した際に跳ね返るのかどうか、持ち運び可能かどうか、カット可能か、等の属性を決めておきます。ファンクションキーの【F9】



カットしたりすることのできる可動オブジェクトとして追加されます。上図のように、これらの操作は、ファンクションキーを直接押す代わりに、【編集】メニューから選ぶこともできます。

また、追加後のオブジェクトの配置は、現在のキャラクターの手前になります。このオブジェクトを好きな位置に移動するためには、マウスの左ボタンでクリックして選択し、【Ctrl】キーを押しながら矢印キーを押して、前後左右、好きな方向に移動できます。また【Ctrl】キーに加えて、さらに【Shift】キーも押しながら矢印キーを押すと、上下に移動したり、その場で回転させることができます。

画面の外にあるオブジェクトを編集したいときは、キャラクターを矢印キーを使って移動させます。このとき【Shift】キーを押しながら矢印キーを押すと、速く移動できます。また、編集集中にカメラの視点を切り替えたいときには、ファンクションキーの【F5】で通常の視点、ファンクションキーの【F6】で真上から見た視点、ファンクションキーの【F7】で斜め上から見た視点、ファンクションキーの【F8】で正面から見た視点、に切り替わります。これらの視点切り替えの操作は、【表示】メニューから選択することもできます。

編集が完了したら、ファイル・メニューから【Worldの保存】を選択して、作成した3Dワールドを、FINALフォルダーの中のWorldフォルダーの中に*.rswファイルとして名前を付けて保存します。このようにして保存したワールド・ファイルは、ファイル・メニューの【Worldを開く】を選択することによって、読み込むことができます。

では、いよいよネットワークで接続してみましょう。サーバーには物理シミュレーションの管理など、他のクライアントに比べてCPUパワーを必要とするので、どのマシンをサーバーにするのか、をあらかじめ決めて、IPアドレスをユーザーに知らせておきます。まずサーバーになるマシンがNetWalker.exeを立ち上げて、ファイルのメニューの中から、【接続】メニューを選ぶか、ファンクションキーの【F5】を押してください。接続用のウィンドウが現れるので、サーバーの場合は、【接続先アドレス】に何も入力せずに、【待ち受ける】をチェックしてから、【OK】ボタンを押してください。

その他のクライアントは、同じように【接続】メニューを選ぶか、ファンクションキーの【F5】を押して、接続用のウィンドウを開いて、予め控えておいたサーバーのIPアドレスを【接続先アドレス】に入力してから、【OK】ボタンを押します。すると、ネットワーク上に参加している他のユーザーのアバターが現れます。

最初の練習として、その辺のオブジェクトを【C】ボタンで拾って、【X】ボタンで投げた事で、キャッチボールをしてみましょう。相手が投げたものが近くに飛んできたら、タイミングよく【C】ボタンを押してキャッチしてください。

遠く離れたユーザーどうしが何か相談したくなったときは、チャットウィンドウを使って下さい。もし、チャットウィンドウが閉じていたら、【表示】メニューから【チャットウィンドウ】を選んで、チャットウィンドウを表示して下さい。【Message】ボックスにメッセージを入力してから【送信】ボタンを押せば、メッセージが全員に送信されて、下のウィンドウに送信者のアバターの種類とともに内容が表示されます。

チャットをうまく活用することによって、集合場所を知らせたり、新しいオブジェクトを追加してもよいかどうかを問い合わせたりして、コミュニティをうまく運用して下さい。

接続を切るときは、ファイルメニューから【切断】を選んでください。ネットワークで接続していた他のユーザーのアバターが消えてなくなります。

5. プロジェクトの今後の課題とビジネス展開

本プロジェクトは最初の章でも述べたように、敢えて安全策を取らずに、全く新しいシステムを最初から設計し直すという道を選んだため、チャレンジの連続でした。

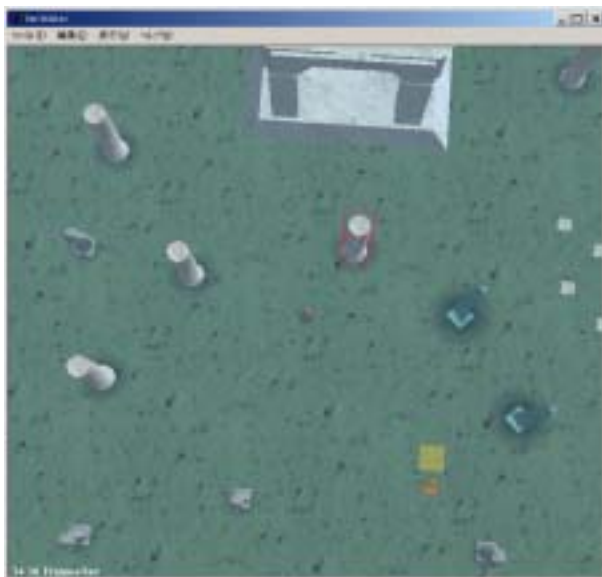
しかしこのおかげで、未踏プロジェクト以前のバーチャコベットのシステムにはなかった、高次アニメーション補間システム、高速センサー、物理シミュレーション、3Dワールドの中でのカット操作、などの斬新な機能を開発することができました。しかし、極めて短期間であったため、各メンバーがそれぞれ考案していたけれど、未実装になっている機能があります。

そのうちの中で、このプロジェクトを大きく発展させる可能性を秘めたテーマをいくつか説明いたします。

まず、ネットワークの分散負荷制御です。機能の概要でも述べたように、現在はパラレル・ワールドにならないために、物理シミュレーションの管理は全てサーバーが行っています。この方式ではサーバーの負荷が大きいため、イベントの遅延が生じてしまいます。キャラクターがボールなどのオブジェクトを蹴る場合について考えれば、まずクライアントからサーバーにキャラクターの移動が通知され、サーバーはその位置情報と全オブジェクトの位置・角度・情報から、キャラクター・オブジェクト間とオブジェクト・オブジェクト間の衝突を検出して、物理シミュレーションの結果を、全クライアントに通知します。したがって、通信帯域が狭い場合には、キャラクターがボールをすり抜けた後で、ボールが飛んでいく奇妙な様子が見えてしまいます。この問題を解消するひとつの方法が『オーナー申告システム』です。オブジェクト・オブジェクト間の衝突はサーバーが全て管理しますが、キャラクター・オブジェクト間は、ローカルで衝突検出を行い、キャラクターがオブジェクトとの衝突を検出した段階で、サーバーにそのオブジェクトの物理シミュレーションの『オーナー』になりたい旨をサーバーに申告します。サーバーは、最初に申告してきたクライアントに、次の衝突が生じるまでの間、物理シミュレーションを任せます。もしくは、ある決められた短い時間内に複数の申告があったときには、フレームレートの大きい高速マシンに物理シミュレーションを任せます。さらに次の段階としては、クライアントのCPUの負荷に応じて、オブジェクト・オブジェクト間の物理シミュレーションも分散負荷制御の対象とします。従来、このようなネットワークの分散負荷制御の問題は、トイ・プロブレムで試されることが多く、本システムのように3Dワールドの中でキャラクターがアクティブ・センサーによって自律的に形状を判断し、かつ本格的な物理シミュレーションが内在されているような極めて複雑な3D環境の中で研究されることは皆無でした。今の段階では、予測もできないような多くの困難な問題が発生するものと思われますが、極めて魅力的でチャレンジしがいのあるテーマであることは間違いのないと思います。

次に、既に機能概要でも少し触れた、アニメーション・データの最適化の問題です。クォータニオンを用いたアニメーション補間システムは、極めて少ないキーフレームのみから、滑らかなアニメーションを再生できます。このようなアニメーション再生方式は、アニメーション・データの高度な圧縮技術に応用できます。しかし、最初に用意すべき数フレームのポーズをいかに選ぶかは、アニメーターの類まれな『瞬間を読み取る』感性に任されています。しかし、『歩く』、『走る』、『拾う』、『ジャンプする』、等のそれぞれのモーションを詳しく調べると、それらのキーフレームのいくつかは極めて似ていることに気がきます。この観察をさらに進めて、基本となるポーズはどのようにしたら得られるのか、いくつぐらい必要なのか、またその基本ポーズをどのように

組み合わせたら各アニメーションになるのか、といった疑問が生じます。また、この問題はある意味で、2時間も3時間もある舞踊を人間はどのようにして記憶して



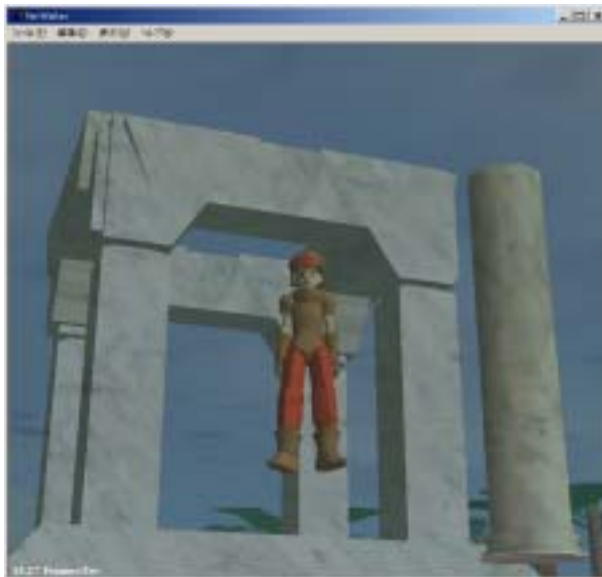
いるのか、といった素朴な疑問にも関連が深いような気がしています。

ひとつのブレークスルーを与えてくれるかも知れない、重要なヒントが意外なところから見つかりました。日本舞踊などの古典芸能の記法です。日本舞踊は舞(ま



い)・踊(おどり)・科(しぐさ)の三大基本要素から構成されています。舞には主に能のような静止ポーズが、踊には躍動的な動き、科には歌舞伎のようなシンボル化された所作が取り入れられていて、これらの要素が統合されて、長い年月を経て古典芸能として洗練されてきました。既に400年以上経過しているにもかかわらず、驚くべきことに踊りの記法にはほとんど流派の違いは見られません。また、一見棒人形のように見える所作を表す各記号に、極めて合理的な記法によって、多くの情報が埋め込まれています。また、踊りのパリエーションの豊富さに比べると、基本所作が極めて少ないのも大きな特徴です。長い年月を経て、記法に無駄がないように自然と淘汰を受けてきたものと思われます。もちろん、踊りに含まれる感情表現は踊り手の感性に大きく依存しますが、この記法のおかげで、現代に伝承されていない踊りでも、記録が残ってさえいれば、ほぼ忠実に再現

することが可能となっています。このモーションの記録は、音楽の楽譜のように時間的変化をシンボル化したものです。



本システムにとって必要な共通必須キーフレームや、個々のアニメーションに固有なキーフレームの抽出にも、何らかの淘汰が必要だと感じています。ネットワーク上で多くのユーザーが参加して、動的にオブジェクトを追加することを考えると、3Dの形状データに無制限に多くのポリゴンやテクスチャーを使えるわけではなく、何らかの制限が必要になります。これは3Dの形状データのみならず、アニメーション・データにとっても同様で、基本ポーズの総数や、各ポーズを表わすシンボルをシーケンスとして並べたアニメーションの総数などにも制限が必要になります。このような抽出作業を個人の研究ベースで行うには途方も無く地道なデータ解析作業が必要になりますが、視点を変えてネットワーク上での共有リソースの動的な更新手続きと捉えなおせば、新たな切り口が見えてきます。つまり、本システムで使える基本ポーズとアニメーション・シーケンスの長さや種類に一定の制限を設けることによって、ユーザーは与えられた制限の中で個性的かつ魅力的なアニメーションを工夫するようになります。また、ユーザー同士がアニメーション・データを交換したり、サーバーを通して共通リソースとしてデータを使うことができるようにすれば、データの利用頻度に応じて、ポーズやアニメーション・シーケンスを、より利用価値の高いものに入れ替えることができます。このような仕組みを使うと、ネットワーク上のユーザーの『評価基準』によって、データセットを『淘汰』することにより、共有リソースを絶えず『進化』させることが可能となります。

このようなサイクルを回し続けるためには、ユーザーにとって魅力のあるシステムやコンテンツが重要となります。一つの試みとして、それぞれのアニメーション・シーケンスをキーボードの各キーや音声認識システムのキーワードと対応させることで、ネットワーク上で音楽に合わせて、リアルタイムでダンスのセッションを組むことのできるシステムが考えられます。それをネットワーク上で放映するためのマルチカメラ・コントローラーや、バリライト・コントローラーをシステムに実装することにより、ネットワーク上でダンスグループ、カメラクルー、ライトクルーなどが共同して一つの作品を放映することが可能となります。サーバーで定期的に開かれるダンス・コンテストに先立って、それぞれのチー

ムは衣装を工夫したり、タイミングを合わせたり、リーダーの指示によってユニゾンで踊ってみたりと、さまざまな練習をネットワーク上で行うことができます。

このようなシステムは、一見、単なる通信のできるダンスゲームのようにも見えますが、先ほど述べた基本アニメーションの抽出の問題や、通信遅延の問題など、多くの課題を含んでいます。

また、アニメーションの再生方式も、クォータニオンを使った数学的な補間システムの他に、骨格や筋構造、運



動制御機構等を組み込んだキャラクターを設計して、物理シミュレーションと神経システムを密接にリンクさせることによって、モーションを生成する手法も興味深いテーマだと思います。

また、今回のプロジェクトでは、ワールド・シミュレーターに埋め込んだ3Dの操作機能はカット機能だけでしたが、これにドリル（ブーリアン演算）や接着の機能を追加することにより、オブジェクトをさまざまな方

法で加工して、それらを組み合わせれば、新たなオブジェクトを生成できます。木を栽培して、木材に加工して販売したり、それらの材料を元にして、家具や庭などを作る工房を開いたり、と新たな構造物を自由に構築して、さまざまなバーチャル・ショップが開けるようなフレキシブル・システムに発展させていきたいと考えています。

本プロジェクトでは時間的に間に合いませんでしたが、キャラクターに視覚センサーなど各種の入力センサーを実装し、さらに空腹度や感情などを表わした内部状態を記憶できるようにして、入力情報と内部状態の対から次の行動を決めることのできる状態遷移システムや、内部モデル規範適応制御システム等を実装しようと思っています。このような制御工学や人工知能技術の応用によって、キャラクターはユーザーからの指示が無い場合でも、複雑なワールドの中で自律的に動き回れるようになります。

また、キャラクターだけでなく、ワールドの中のきのこや木などの植物や魚などにも、遺伝子データを組み込むことにより、突然変異や交配などの進化機構を取り入れようと思っています。植物がうまく生育するように面倒を見てあげれば、植物は成長して花や実をつけます。もし、鳥がその実を食べて、種を遠く離れた場所に落としたときには、その地の生育環境さえ整っていれば、植物が繁殖できます。このような連鎖により、3Dワールドの中のオブジェクト間の相互作用が飛躍的に増えて、複雑さが増します。そして、この複雑な環境がキャラクターの行動基準の進化にも大きく影響するものと考えられます。

また、本システムでは、遠方のユーザーとは従来型のチャットができるだけですが、CPUのパワーが許せば、現在テスト中のリアルボイスでのチャットシステムを組み込んで、よりリアルな通信環境を実現したいと思っています。

また、今後のビジネス展開としては、未踏ソフトウェア創造事業以前のバーチャコペット・プロジェクトと、本プロジェクトで開発したそれぞれの要素技術を生かして、商用のシステムに仕上げていきたいと考えています。本プロジェクトの進行中にも、（株）デジタルマジックを始め数社から、ビジネスのお話をいただきましたが、それぞれの方が全く違った観点からシステムをご覧いただくため、興味を持っていただいた要素技術の組み合わせが全く異なっています。ある意味で、ビジネスとして要素技術の『切り売り』も必要なことですが、ライセンス上の問題などから、総体としてのシステムをさらに発展させるための妨げにならないように、十分考慮しながらビジネスを進めていきたいと考えています。

システム全体をバランスよく発展させていくという意味でも、また上で述べたような先端的なテーマに取り組むという意味でも、是非未踏ソフトウェア創造事業には継続申請をお願いしようと思っております。また、未踏のプロジェクトとして進める場合には、プロジェクトで得られた技術的成果を、単なるパッケージに製品化するための技術として捕らえるのではなく、コア・アルゴリズムを書籍やホームページで紹介したり、また基本ライブラリをオープンソース化することにより、より多くの人達に使ってもらって、使いやすさや、さらなる要望などを吸収しながら、より良いシステムに仕上げていきたいと考えています。