TR-H-294

仮想世界共同構築システム バーチャコペット

和田健之介、中口孝雄、和田佳子

2000.3.30

ATR人間情報通信研究所

〒619-0288　京都府相楽郡精華町光台2-2-2　TEL: 0774-95-1011

ATR Human Information Processing Research Laboratories

2-2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan

Telephone: +81-774-95-1011

Fax : +81-774-95-1008

Virtual World Cooperative Environment Tool Kit

仮想世界共同構築システム

和田健之介, 中口孝雄, 和田佳子

【はじめに】

●不思議ワールドを探検してみよう

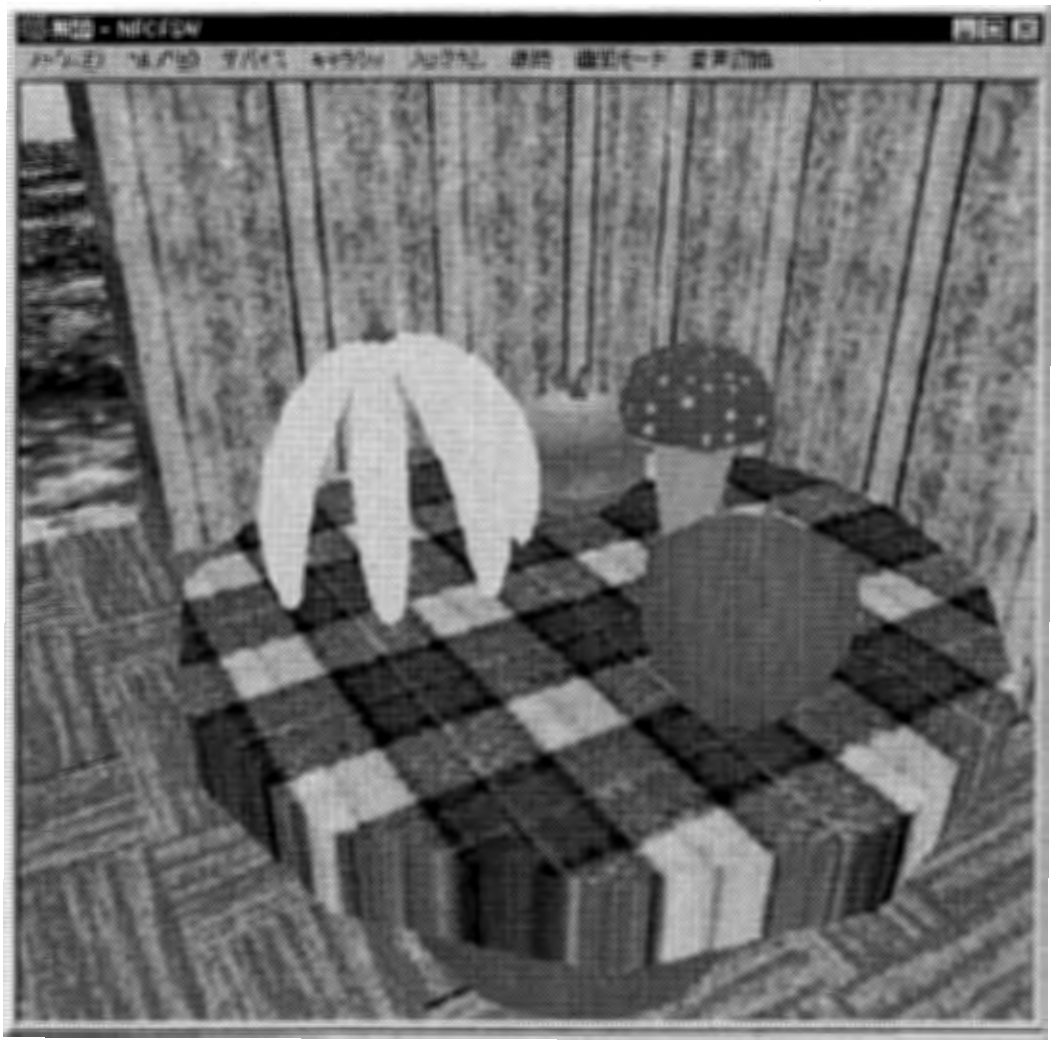
このシステムをネットワークの上で使うと，気の合った仲間どうしで仮想世界を共有することができます．それぞれのユーザはワールドの中で自分の分身となるキャラクタを選択して，そのキャラクタの視線で仮想世界を探索できます．フルーツを採ったり，キノコ狩りをしたり，釣りを楽しんだり，ときにはダンジョンの探索ツアーでモンスタと戦って貴重なアイテムを手に入れたり，円盤に乗って大気圏外の旅行に出たり，とさまざまな体験をすることができます．

また各ユーザのモニタにはワールドの日照やアイテムの位置，ドアの開閉などの状態が共通になっているので，いろいろなイベントを同時刻で楽しむことができます．例えばAさ

んが魚つりに夢中になっているうちに，真っ暗になって家の方角が分からなくなったとしましょう．こんなときでも，家にいるBさんに連絡して家のライトをつけてもらえば，方角が分かり夜道でも安心です．このときAさんのモニタには足元の海岸や遠くにぼんやりと明かりのついた家が見えていますが，Bさんのモニタには家の中のテーブルや，もしかしたら窓から家路へと急ぐBさんのキャラクタが小さく見えているかもしれません．しばらくすればBさんは，ドアを開けて入ってくるAさんのキャラクタを見ることになります．

●ワールドの中の食生活

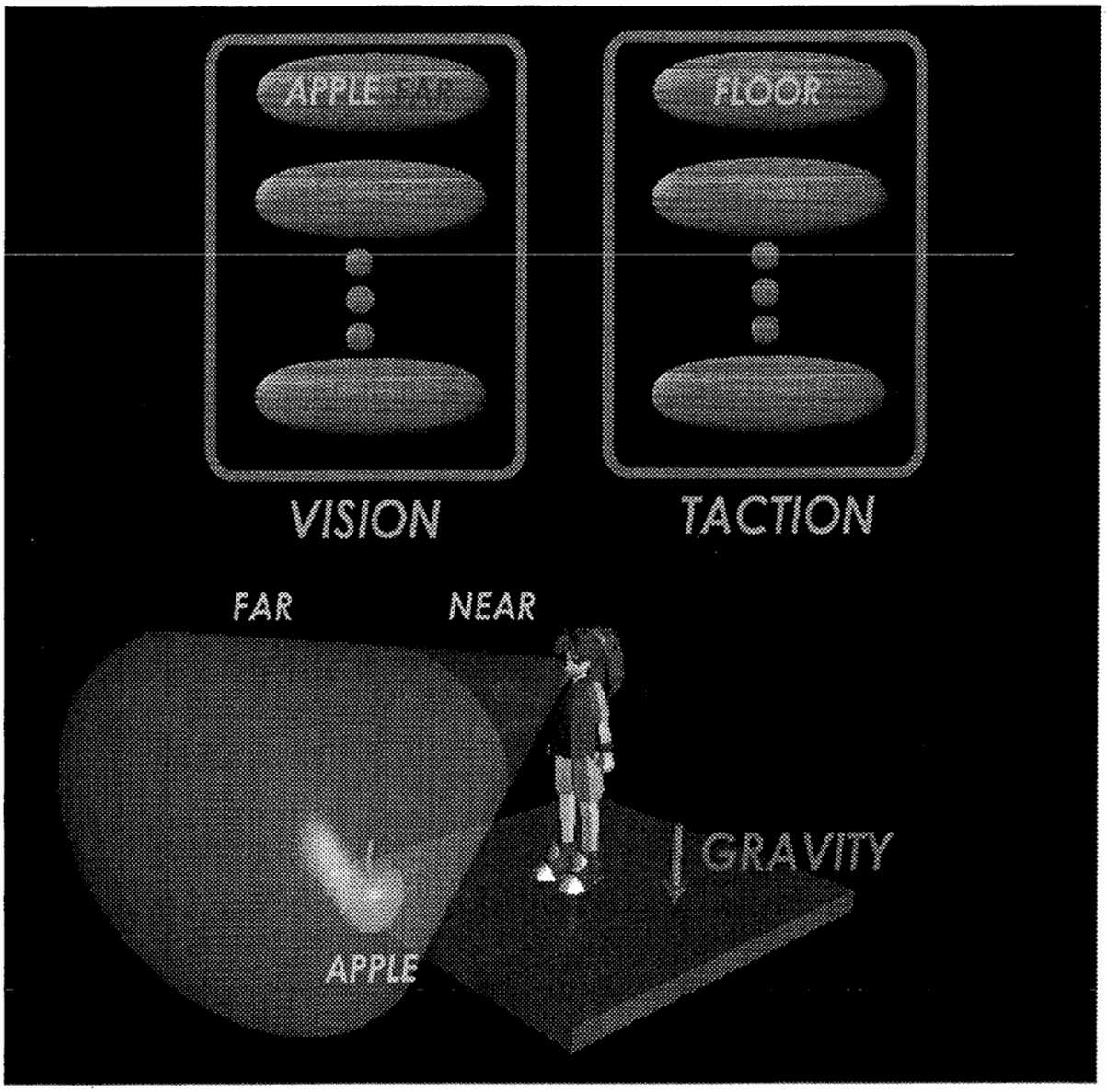
ワールドの中にはフルーツ，キノコ，魚などさまざまな食べ物があります．後で順を追って説明していきますが，キャラクタはいろいろな遺伝子を親から受け継ぎます．食べ物の消化能力もその一つです．この栄養消化に関係した遺伝子は家系によって異なるので，あるキャラクタにとって紫色のキノコがとても栄養価に富んだ食べ物であっても，他のキャラクタには猛毒かもしれません．それはキノコも遺伝子を持っていて，その遺伝子によって決まる栄養とキャラクタの消化能力の組合せで栄養価や毒性が変わってしまうからです．またキノコもキャラクタに食べられることによって，より広まることができるのなら，結果的には世代と共にそのキャラクタにとって，より栄養価の高い特性を持つように進化していくことになるでしょう．この食べ物の遺伝子は突然変異などを制御する進化エンジン（ワールドの進化を促進するための機構）によって変化していきますが，ユーザが遺伝子を直接操作してフィールドで栽培することもできます．



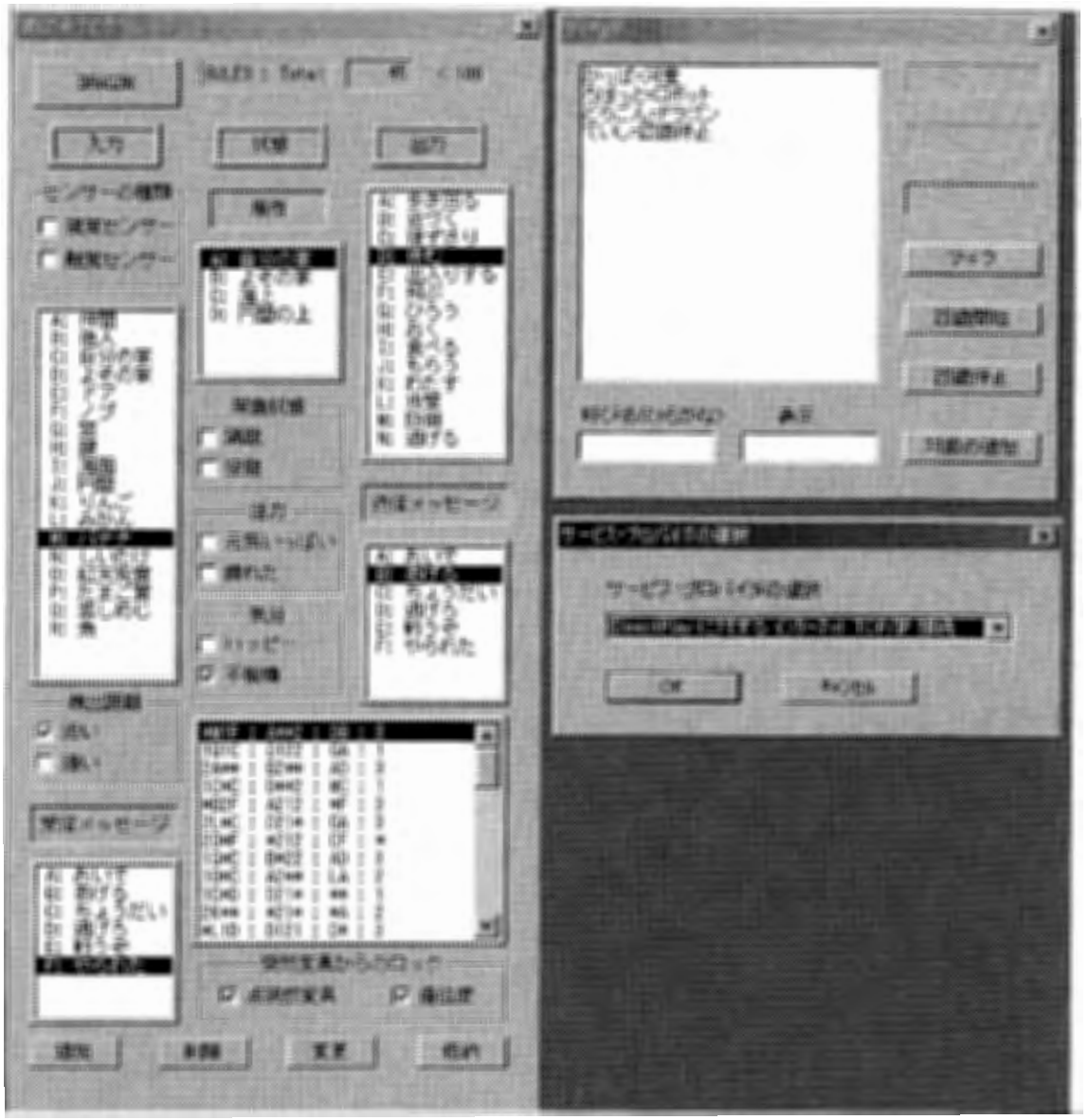
●賢い子供を造ってみよう

ここまでの解説では，各ユーザが自分の分身となるキャラクタを１つずつ決めて，ワールドを共有していました．しかしキャラクタが一つだけだと，家を建てたり，キノコを集めたり，魚をとったり，ときには怪獣退治に遠征したり，とやることだらけで大忙しになってしまいます．こんなときに，賢い子供，もしくは子分かペットがいて，ちゃんと教えさえすれば，せっせとキノコを集めてきてくれたり，怪獣にやられそうなときは加勢してくれたりすれば，大助かりです．言い方を変えれば，自分で状況を判断して，与えられた仕事をちゃんとこなすことのできる人工知能搭載のキャラクタを仲間に加えることができれば，ワールドの中でより楽しく過ごすことができます．このようなキャラクタは，今風の呼び方をするなら，人工生命とか自律エージェントなどと言ったりします．私達のシステムでは，このキャラクタを単にプログラムのできるロボットとしてではなく，ワールドの環境を自分で認識して，自ら判断することのできる自立した個体として捉えています．よくある3Dのゲームシステムでは，キャラクタが3Dのポリゴンでできた綺麗な家や橋などを歩き回るときに，壁や橋の中にめり込んだりといった不都合がおきないように，ある工夫が凝らしてあります．その工夫とは，3Dの形状データとは別に，“歩いてもよい場所”を記した地図のデータを参照して，キャラクタの動きをコントロールすることです．しかし，前もって作りこまれた世界をただ覗くだけなら，このような方法でスピード感のあるキャラクタの移動を行うことができますが，私達のワールドのようにリアルタイムで複数のユーザが共同作業を行ってワールドを構築しているような場合には，絶えず地形が変化したり，物が移動したりしています．このためオブジェクトが作られたり壊されたり，また移動したりするたびに，そのオブジェクトのデータと共に，“歩いてもよい場所”をマーキングした地図データも送信しなければなりません．またこのような方式はワールドをプログラムした開発者から見れば自然な発想かもしれませんが，キャラクタ側から見れば，自分の判断で世界を認識しているのではなくて，理由もわからずに， システムから“ここは行っちゃダメ，あそこもダメ”と指図ばかりされていることになり，とても自立した１個の知的個体と見なすことはできません．そこで，多少計算時間はかかりますが，私達のシステムでは，キャラクタ自身に世界の認識をさせることにしています．キャラクタには基本的には上下前後左右の方向に触覚センサがついていて，自分が今どんなオブジェクトと接触しているのかを，絶えず感じています．

ワールドにはライトや日照などの明るさの他に，重力や空気抵抗，水中での抵抗などがあります．キャラクタのセンサの入力情報を元に，キャラクタの移動が決まります．例えば，キャラクタの前に壁があれば，それ以上は前に進めないし，足元に何もなければ自分が空中にいることになり，重力と空気抵抗が働いて落下していきます．このときの空気や水に対する抵抗はキャラクタによって異なるので，飛んだり泳いだりするときのスピードがそれぞれ違ってきて，自分の能力にあった環境の中ではとても動きやすくなります．また遺伝的に受け継いだ視力も備わっています．草食動物のように広い視野角を持つキャラクタもいれば，肉食動物のように狭い視野角を持つキャラクタもいます．また，



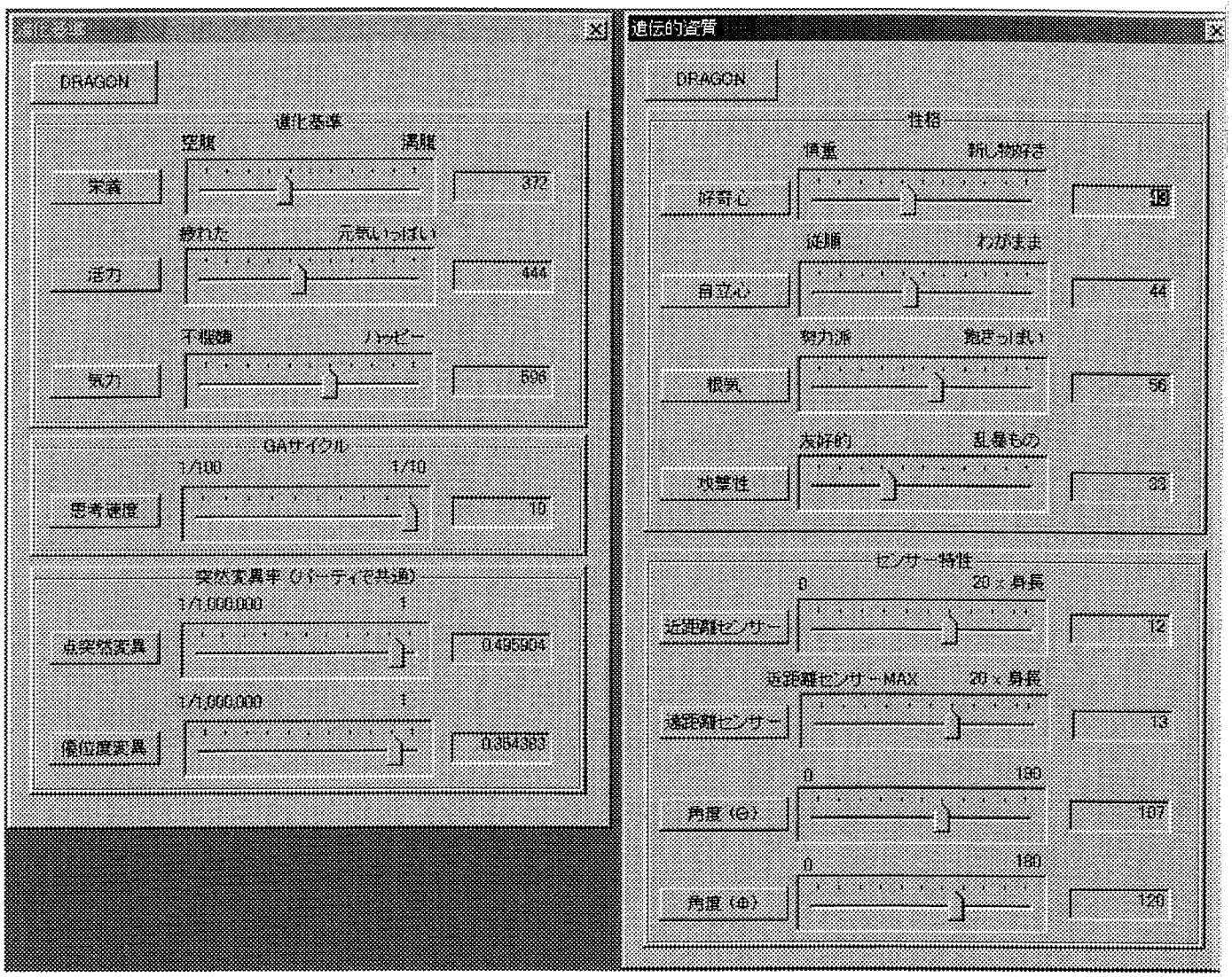
各キャラクタには，左右上下の視野角の他に，近距離・遠距離センサの到達範囲を決める遺伝子が内蔵されているので，見えている対象が近距離にあるのか，それとも遠距離にあるのかも判断することができます．また視野に入ったオブジェクトに対しては，自分とそのオブジェクトの間に障害物があるのかどうかも考慮しているので，例えば近くにいてもドアの向こうに隠れたりしていれば，キャラクタには見えていません．このようにキャラクタはかなり現実に近い認識システムを持っているので，ワールドの認識という点では，ユーザの分身であるキャラクタと自律型のキャラクタは対等な立場にあります．
自律型のキャラクタは，この認識情報と自分の現在の状態を把握してから，ユーザによってインプットされたプログラムを参照します．プログラムは，“こういう状態のときに，こう



いう場面に出くわしたら，こうやって行動しなさい” という単純な命令セットの集まりでできています．もし，その命令セットの中に，自分のおかれた状態と場面にぴったりのものがあったら，その命令の指示に従います．一つも一致するものがなければ，そのキャラクタがもともと持っていた行動癖に従って適当な行動が選ばれます．ここで，自分の”状態”と“場面”の違いは何かというと，場面とは言いかえれば，各種の感覚センサからの情報を集めたもの，つまり外部からの入力情報のことで，状態とは，お腹が空いているとか，疲れている，怒っている，などキャラクタ内部の状態のことを意味しています．
このように考えると，プログラムの中に書かれた個々の命令文は，入力情報と内部状態の組合せから，次に取るべき行動を支持したもの，と捉えることができます．ここで内部状態が入っていることが大事です．“りんごを見たら，食べる” というように，入力と行動

が直結した命令文よりも，“お腹が空いているときに，りんごを見たら，食べる” という命令文の方が複雑な状況に対応することができます．実際のプログラムでは，各命令書が重要度の高い上位ランクのものから順に並べてあって，キャラクタの入力情報と内部状態とのセットと，各命令書に書かれている入力情報と内部状態とのセットとを，照合していきます．

このマッチングを取るときに，完全主義を貫くか，寛容主義を認めるか，によってもシステムの特性が違ってきます．つまり完全主義を選んだときは，それぞれのセットの内容が一つでも違えば，それはマッチングがとれていないものと見なします．反対に寛容主義を選んだときは，セットどうしの類似性を見ます．完全主義だとYES/NOのデジタルなシステムになり，寛容主義だと“7割2分は合っている”という具合にアナログのシステムになります．このアナログ値をそれぞれの命令書が持つ主張の度合いと捉えて，主張の強い命令書ほど選ばれる頻度が高いと考えれば，確率論的なシステムになります．デジタルなシステムでは状況が同じなら，いつでもおなじ行動を取る決定論的なシステムになります．一方今お話した確率論的なシステムは，主張の強い命令書ほど選ばれる確率は高いけれども，それはあくまでも確率であって，ときにはかなりランクの低い命令が選ばれるかもしれません．決定論的なシステムでは，信頼性はありますが，ときには同じ行動のループにはまってしまって抜け出せなくなることもあります．確率論的なシステムでは，あまり信頼性はありませんが，ときには妙な行動にでるので，ジレンマから抜け出る策を思いつく可能性もでてきます．私達のシステムでは，このようなシステムの特性もキャラクタの遺伝的な資質の一つとして考えています．



これとよく似たキャラクタの遺伝的資質に“素直さ”があります．キャラクタの受け取る入力情報は視覚・触覚の他に3Dの聴覚もあり，他のキャラクタが発したメッセージを受け取ることができます．例えば“おいで”というメッセージが左後ろから聞こえてきたときに，このメッセージ通りの行動を取るかどうかは，キャラクタの“素直さ”によって異なります．また疲労感や空腹感の感じやすさ，頭の回転の良さ，なども遺伝的に親から受け継ぎます．

●ワールドの住民の進化と世代交代

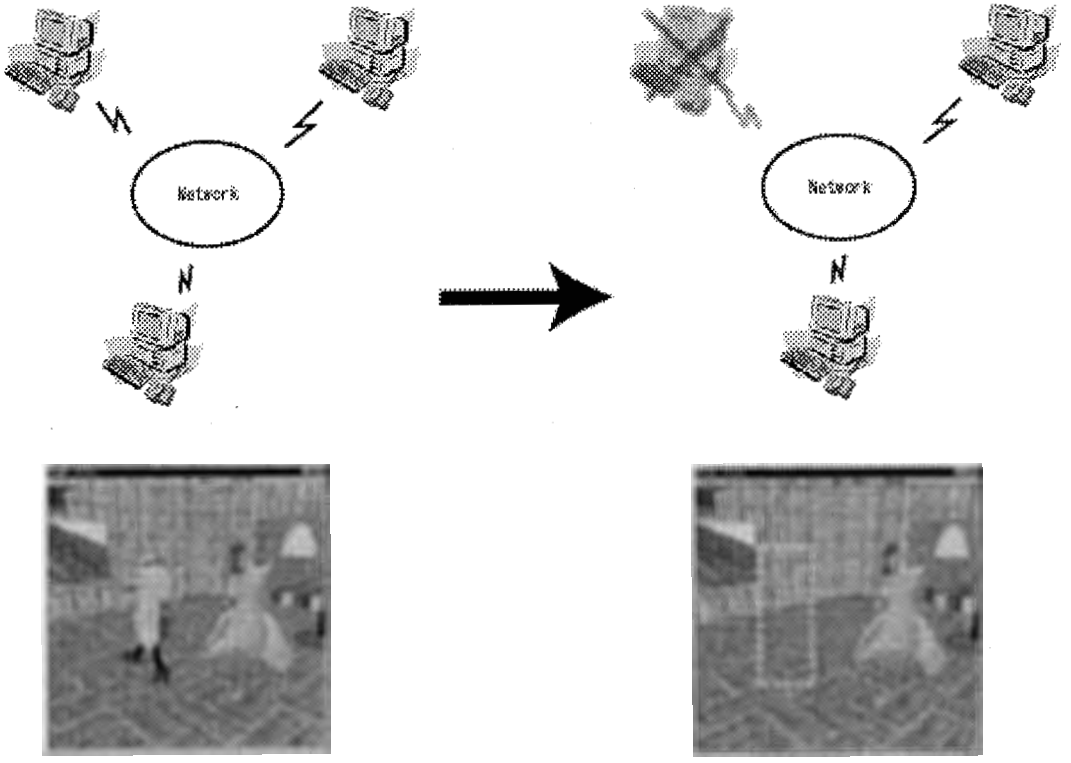
このように人工的に造られたキャラクタは親からの形質を受け継ぎつつ，ある程度の突然変異によって，少し親とは異なった特徴を持っています．今までの進化システム論を使ったシュミレータでは，与えられた課題をよりよく解けた個体を選んで増やしてきました．このような評価・選択の過程を何世代も繰り返すことによって，次第に課題をよりうまく解く個体が現れることを期待するのです．しかしこの発想は，キャラクタの認識方法のところでもお話しましたが，あくまでも主体はシステムもしくは設計者の方であり，個々の個体はただ造られて，選ばれたり捨てられたりするだけの存在です．私達はできるだけワールドの住民の人権？を尊重してあげたいので，世代交代のタイミングや結婚するときの結婚相手の条件も全てユーザ側にまかせることにしました．しかしコンピュータの性能には限りがあるので，無限に子供を増やすことはできません．自然界でも食べ物などのリソースに限りがあるときには個体数も飽和してしまいます．残念ながら，ワールド全体の個体数に限りがあるために，子孫の誕生と同時に両親の寿命も尽きるものと考えました．

●子供に話しかけてみよう

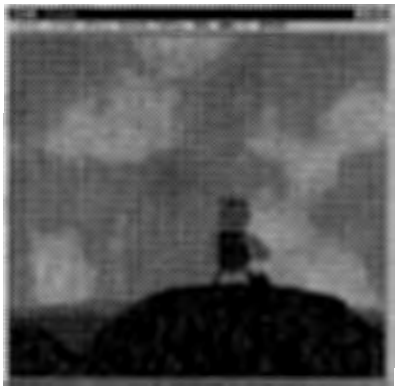
私達のシステムには，簡単な音声認識エンジンが組み込まれています．予め登録しておいた単語を認識できることはもちろんですが，必要に応じて認識できる単語を増やしたり減らしたりすることができます．このために子供が生まれて始めてリンゴを見つけたときに，“リンゴ”という単語を登録してやれば，それが“リンゴ”だということを音声で教えてあげることができます．キャラクタの頭の中でリンゴの形（というよりオブジェクトのID番号）と単語の対応がちゃんととれてしまえば，その後は“リンゴ”と呼びかけるだけで，その音声が“リンゴ”を意味しているのだと理解できるようになります．しかし，リンゴを探せと言っているのか，リンゴを食べろと言っているのかは，分かりませんから，“探せ”や“食べろ”という単語も命令セットの行動一覧の“探せ”や“食べろ”と対応させる必要があります．このような音声認識システムによるユーザーインターフェースを導入すると，よりキャラクタに感情移入しやすくなるし，ちょこまかと動き回るキャラクタに機敏な指示を与えることができるようになります．

●柔軟な仮想世界処理システム

私達はこのゲームのプラットフォームとして考えているOSは，ウィンドウズです．ウィンドウズ98やウィンドウズ2000であれば，最初からゲームライブラリであるDirectXがインストールされています．私達のシステムは特別なライブラリやハードウェアを必要としないように設計されています．3Dのグラフィックス・ボードがあれば，よりスムーズな描画能力が得られますが，基本的には普通のウィンドウズ・マシンで動作するようになっています．このウィンドウズは，突然フリーズすることでも有名です．ユーザの使い方にもよりますが，何もしていないのにリセットがかかることもあり得ます．ネットワークを使って複数のコンピュータで通信するシステムにとって，コンピュータがフリーズしてしまうことは致命的です．送られてくるはずのデータが送られて来なくなり，フリーズしたコンピュータが一台だけであっても，他の全てのコンピュータに故障が波及して，ゲームを続けられなくなることもあります．このようなシステム構成の場合は，故障を起きてはならないものと考えるフォールト・アボイダンス（故障回避）の発想から，故障は避けられないものであり，その被害をも予め見越してシステムを設計するフォールト・トレランス（故障寛容）の設計思想に切り替えなければなりません．私達は，ネットワーク上の故障の波及を回避するために，ネットワークシステムに障害が起きた際の動的処理継続機構を考えて，これをシステムに実装しました．



コンピュータの一台に障害が発生した際には，そのコンピュータをネットワークシステムから安全に取り除き，そのコンピュータが処理していたキャラクタを仮想世界から安全に取り除くことができます．他のユーザは今まで通りゲームを継続することができます．
また，複数のコンピュータで処理されている仮想世界間に矛盾が生じないよう，仮想世界の同期処理を行っています．これは，仮想世界の状態同期処理と，イベント同期処理に分けられます．時間の経過やユーザが起こすアクションと共に，仮想世界の状態は常に変化しています．時間と共に日照が変わり，ユーザのアクションによってドアや窓が開閉したり，ライトの on/off，きのこの数やフルーツの位置，そしてユーザの位置も変化します．あるユーザが参加する時点でのこれらの状態を既に参加しているユーザが教えることにより，全員が同じ世界を共有することができます．これにより，途中からゲームに参加したユーザも，それまでに参加していたユーザと同じ世界を共有することができます．
仮想世界内であるユーザが起こしたイベントは，システムに参加しているユーザ全てに送信され，常に仮想世界の同期がとられます．



日照(夕焼け)

巣物

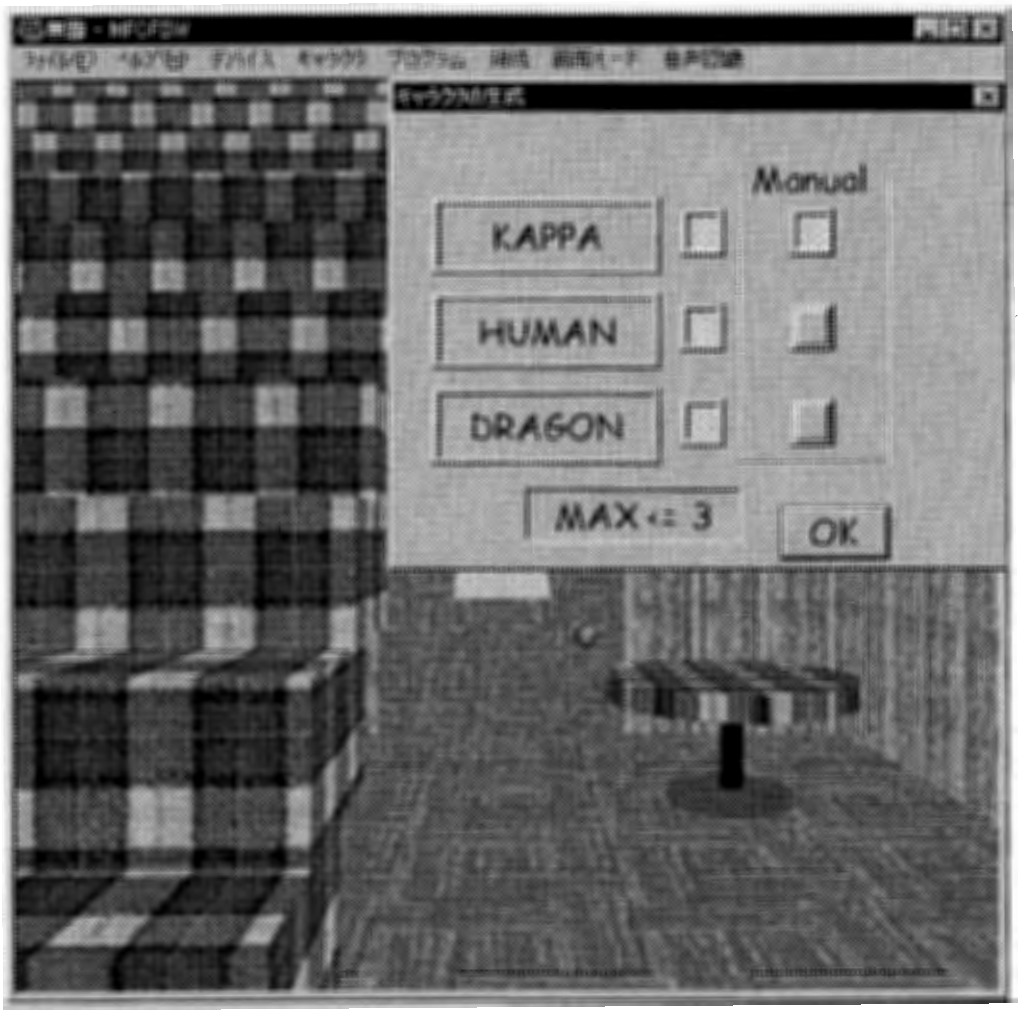
ライト

これらの処理により，システムに参加するコンピュータやその数が静的に決まっているのではなく，ユーザはいつでも仮想世界に参加し，いつでも仮想世界から抜けることができ，かつ全てのコンピュータで同じ仮想世界を共有できる動的なシステムを実現しています．

【操作方法】

●プログラムの起動とキャラクタの選択

まず Evo という名前のアイコンをダブルクリックして，プログラムを起動します．起動直後は図のように建物だけでキャラクタのいないワールドが見えています．そこで左から4番目の【キャラクタ】メニューをクリックして【作成】を選んでください．新しいダイアログが表れるので，そのなかから好きなキャラクタを選択して，キャラクタ名のすぐ右のボタンをクリックしてください．河童・人間（女の子）・ドラゴンの3種類を用意しました．複数のキャラクタを選んだ場合は，自分でコントロールしたいキャラクタの【Manual】ボタンをクリックしてください．他のキャラクタは Auto モード，つまりキャラクタ自身の知能システムを使って，自らの判断で自律的に行動します．この【Manual】モードと【Auto】モードはキャラクタの



作成後でも変更することができます．キャラクタが見えていれば，キャラクタの体をクリックするだけで，そのキャラクタが【Manual】モードに切り替わり，いままでユーザがコントロールしていたキャラクタは【Auto】モードで動作するようになります．スピーカをつければ，キャラクタが壁や階段にぶつかって‘コチッ，コチッ’と小さな音をたてているのが聞こえてくるでしょう．もし誰も見当たらなくても心配しないでください．カメラからはずれたところにいるだけですから．

●カメラの移動と視点の切り替え

ではカメラの移動の仕方を説明します．カメラアングルを変えるには，数字や四則演算のキーをまとめたテンキーを使います．テンキーの【4】と【6】はカメラの左右の移動です．押したままにすれば，スムーズにカメラが移動します．【2】と【8】はカメラの前後の移動です．扉に近づいたり遠ざかったりしてみてください．天上を見上げたり，床を見下ろしたりしたければ，【7】と【9】を押してみてください．また，カメラを左右に回転してみたければ，【1】と【3】を使ってください．カメラを上に持ち上げたり下げたりするには，【－】と【＋】を使います．キーボードを見なくてもいいように，キーが上に配置されている【－】はカメラの上への移動を，その下にある【＋】は下への移動をするようになっています．では，しばらくハリウッドのカメラマンになったつもりで，カメラクレーンを前後上下左右にぐりんぐりんと移動してみてください．それぞれのキーを組み合わせることができるので，例えば【1】と【6】と【8】を同時に押すと，右に大きく回り込みながらズームしていきます．どうです？結構慣れてきましたか？それともムチャしすぎてカメラが斜めに寝転んで変なアングルになってしまいましたか？もし最初の位置に戻したければ【．】を押してください．
今までは第三者の視点でカメラを動かしてきましたが，今度はあなたが【Manual】モードに指定したキャラクタの視点に切り替えてみましょう．それには【＊】を押してください．突然景色が変わってしまったと思いますが，この映像は今現在キャラクタが見ている映像です．カメラはキャラクタが移動したり回転すれば同じように動きます．このカメラをその場でキャラクタから切り離すには【5】を押してください．
これまでのことをまとめてみましょう．

【＊】=キャラクタ視点 【－】=カメラの上移動
【7】=カメラの上回転 【8】=カメラの前移動 【9】=カメラの下回転 【＋】=カメラの下移動
【4】=カメラの左移動 【5】=カメラの切り離し【6】=カメラの右移動
【1】=カメラの左回転 【2】=カメラの後移動 【3】=カメラの右回転
【．】=カメラのリセット

この他にも，カメラのパース変更などが【F5】【F6】【F7】【F8】に割り当ててあって，【F9】を押せばそれらの値のリセットを行うことができます．

●キャラクタの移動と行動

次に【Manual】モードで指定したキャラクタの操作方法を説明しましょう．まずカメラをキャラクタの視点に移動してみましょう．【＊】を押してみてください．キャラクタになったつもりであちこちと移動してみましょう．前に移動するには矢印キーの上【↑】，後ずさりするには矢印キーの下【↓】，左移動は左矢印【←】，右移動は右矢印【→】です．衝突センサが何かにぶつかったことを検知すると‘コチッ，コチッ’と音がしますが，気にせずに散策してみてください．慣れてきたら階段のそばに行ってみましょう．もし，とっぷりと日が暮れてしまって真っ暗けになっていたら，ちょっと辛抱して明るくなるまで待っていてください．階段の前にきたら，階段を見上げてみましょう．首を上に向けるにはテンキーの近くにある【PageUp】を押します．2階の様子が見えてきたと思います．

では，2階に上がってみましょう．キャラクタにはそれぞれの身長に応じて，越えられる段差があります．その範囲内の段差なら前に進めば自動的に登ってくれるので，

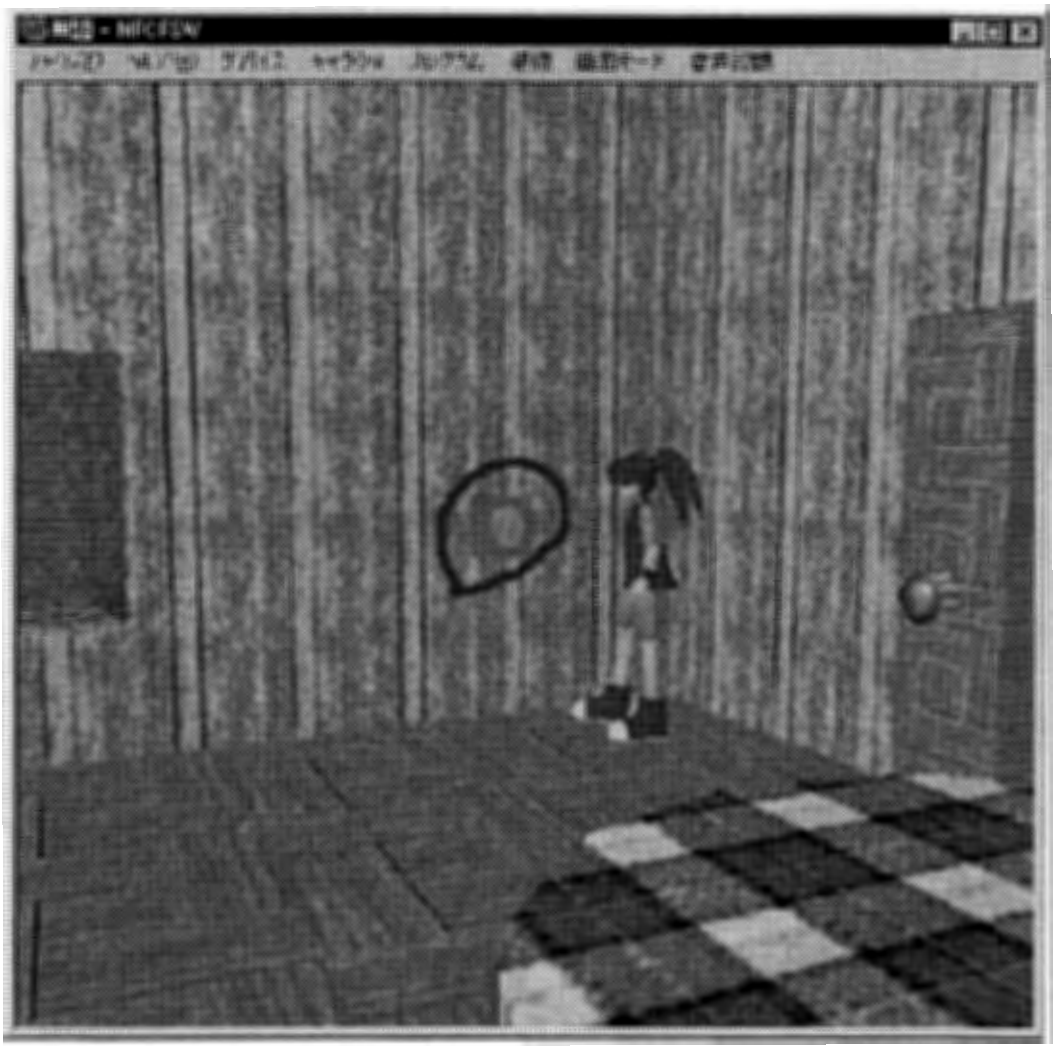
これくらいの階段なら【↑】を押しつづけるだけで，どんどん階段を上がっていきます．２階に着いたら上を向いていた首を戻しましょう．【PageDown】を押し続ければどんどん足元の方に首が傾きます．適当な角度になったら2階の部屋を見回してみましょう．菖蒲の写真が飾ってあったり，煙突があったりするのでそばへ寄って眺めてみてください．2階の窓から外の景色を眺めれば，遠くに桜が咲いていたり，海が広がっているのがわかります．

では，１階へ降りてきてください．【PageDown】を押して足元をよく見て降りてきてください．踏み外すと死んでしまうので，後の説明が続きません．一からやり直しになってしまわないように，くれぐれも慎重に！

●家の設備を利用しよう

無事に1階に下りてこれましたか？もしまた真っ暗けになっていたらライトを点けて明るくしましょう．えっ？真っ暗でどれがライトか分からないんですか？そんな場合はもうちょっと夜が明けてくるまでじっとしてましょう．出入り口の左に，ピンク色のライトスイッチがあるはずです．このスイッチをマウスでクリックするたびに‘パチッ’という音とともに1階の天井ライトが点いたり消えたりします．次に家の両側にある水色の透明の窓ガラスをマウスでクリックしてみましょう．‘シューウィッ！’っという軽快な音と共に開いたり閉まったりします．

では，最後に出入り口のドアのノブをクリックして開けてみてください．開いたらいよいよ外の世界に出かけてみましょう．



●ホームセキュリティはちゃんとしてますか？

おっと，その前に戸締りはちゃんとしておきましょう．両側の窓ガラス，ドアをちゃんと閉めておいてください．戸締りは防犯の基本です．ちゃんと閉めましたか？

念には念を入れてもう一度閉まっている出入り口のドアのノブをマウスでクリックしてみてください．‘ブブー’という不快な音だけがしてドアは開かないはずです．この家にはホームセキュリティが施されているので，家の内側からはドアを自由に開け閉めできますが，外からだと閉めることはできても，ちゃんとした暗証番号を入れない限り開けることはできません．ここでは，こっそりと私がかけた暗証番号をお教えします．【5】【6】【2】を押してからドアのノブをクリックすればOKです．暗証番号はいつでも変えられますが，方法はナイショです．暗証番号パネルを見て考えてみてください．

このように家にカギをかけることができるので，自分の取ってきた食べ物や仲間が持ってきたアイテム，もらったメイルなどを安全に家に保管しておくことができます．

でも，時間がなかったので，窓は開け閉めし放題です．窓から侵入するドロボーさんよけのセンサも今度作るつもりですので，必要な方は今後を期待していてください．

●走れ！飛べ！泳げ！潜れ！飛び降りろ！？

今まではとろとろと歩くだけでしたが，一度外の世界を知ると，これでは遅すぎてイライラしてしまいます．そんなときは【左シフトキー】で2倍のスピード，【右シフトキー】で5倍のスピードがでます．【右シフトキー】を押さえながら【↑】を押して，ビューンと走り回ってみてください．結構爽快になれると思います．そしたら今度は【Home】キーを押してジャンプもしてみましょう．押しつづければ，どこまでも高く飛んでいきます．このときに，先ほど覚えた【PageDown】キーを押して下界を眺めることを忘れないでください．空の散歩を楽しんだら海の近くの浜辺に下りてきてください．海に入ると流れが速いのであっという間に流されてしまいます．そこで【End】キーを押しつづけてみてください．海の中に潜水することができるはずです．運がよければ，魚の群れをみつけることができます． これはスーパーのチラシをスキャナで撮ったテクスチャが貼ってあるので，かなりリアルに見えます．次に海から上がって元の家に戻ってきてください．今までの復習をしてみましょう．飛び上がって家の屋根の上に降りてみてください．飛んだり視点を変えたりと，今まで覚えてきた事柄を組み合わせて上手に着地してください．ちゃんと降りられたら，今度は屋根からそのまま前進して飛び降りちゃってください．大丈夫です．飛び降りても死にませんから．さっきの階段を降りるときの忠告はなかったことにしてください．

かなり慣れてきたら，サンタクロースみたいに2階の煙突から出入りしたりしてみてください．

このレベルまで到達したら，もっと高度な技をお教えしましょう．【Insert】キーを押すと，飛んでいる最中にホバリングして空中静止ができるようになります．ホバリングを解除するには【End】キーをチョンと押してください．

今までの行動キーの操作をまとめると次のようになります．

【Insert】=空中静止	【Home】=ジャンプ	【PageUp】=見上げる
	【End】=潜水，落下	【PageDown】=見下ろす
	【↑】=前進	
【←】=左回転	【↓】=後退	【→】=右回転

●　選手交代して気分を変えよう

そろそろワールドの中を散歩することに大分慣れてきたころだと思います．もし，複数のキャラクタを選択していたなら，現在【Auto】モードで行動しているキャラクタを【Manual】モードに変更して気分を変えてみましょう．それにはただ【Manual】モードにしてみたいキャラクタをクリックするだけです．とは言っても，勝手に歩き回っているので，今は近くにいないかも知れません．そんなときは，

【F1】=河童をマニュアルモードに変更
【F2】=人間（女の子）をマニュアルモードに変更
【F3】=ドラゴンをマニュアルモードに変更
【F4】=全員が【Auto】モードで行動

を押して選手交代してください．突然別のキャラクタの視点に移るので，移った瞬間はそこがどこかわからないかもしれません．グルグルと辺りを見回したり，空から見下ろしたりして確認してください．
河童は泳ぎが得意で，ドラゴンは飛ぶのが得意です．人間は中間的な運動能力を持っていますが，走るのは他のキャラクタよりは得意です．このようにキャラクタ毎に水の抵抗や，空気抵抗などが異なっているので，それぞれのキャラクタによって動きやすい環境が違います．

●やっぱしゲームはジョイスティック＆フルスクリーンが常識さ！

ジョイスティックを持っているなら，【デバイス】メニューから【ジョイスティック】を選択することで，入力デバイスをキーボードからジョイスティックに変更できます．もっとリッチでフォース・フィードバック・ジョイスティックを持っているなら，壁にぶつかったときに振動を感じることができるので，より直感的に操作することができます．振動の効果を付けるか付けないかを決めて，【デバイス】メニューの【フォースフィードバック】から【振動あり】か【振動なし】を選択してください．ジャンプするときには，スティックを握ったときに人差し指の所にあるボタンを押してください．首を上下に振るにはジョイスティックの左に付いているスライダーを回してください．

ウィンドウモードのときには，ウィンドウのインターフェースは通常のウィンドウズ・アプリケーションとほぼ同じです．角をドラッグすればウィンドウのサイズを自由に変更できます．右上のアイコン化ボタン，最大化ボタン，終了ボタンも通常と同じです．キーボードの一番左上の【Esc】キーか【F12】キーを押せば，フルスクリーン・モードに切り替わり，大画面で3Dワールドを楽しむことができます．ウィンドウ・モードへ戻るときも同じキーを押してください．

●私のマシンはどれくらい速いのかな？

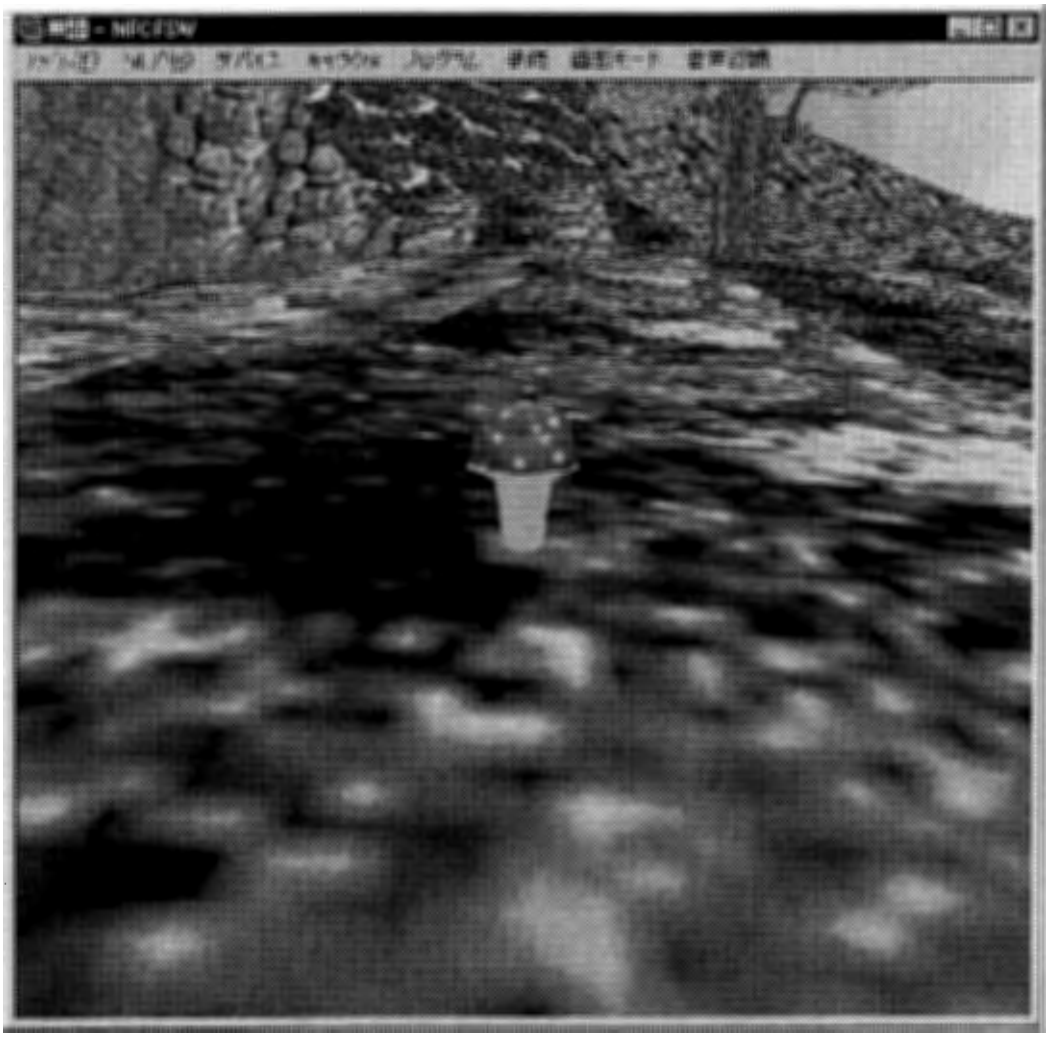
自分のグラフィックスボードがどれくらい速いのか知りたければ，【F11】キーを押してみてください．

画面の上の行に白い帯が現れて，その真中あたりに 28．5 FPS などと書かれた数字があるはずです．これは画面を一秒間に何回書き換えているかを示しています．この値が大きければ大きいほど，キャラクタが滑らかに動きます．見えている景色が違えば，この値はかなり変わってしまいます．だいたいの目安で言えば，プログラム起動直後でキャラクタを作成する前なら，30 FPSを越えていれば，全く問題無し，というより理想の環境です．最近は50FPSを越えてしまうモンスター・マシンもあります．20台であれば，それなりに滑らかに動きます．10台だと，ちょっとガクガクした感じがしてしまいます．1桁台だと，紙芝居を見ているようなぎこちなさがあり，操作性も今一つ悪くなり，ちょっと辛いかもしれません．この描画性能はCPUやグラフィックスボードの3Dサポートの有無など，さまざまな条件が関係するので一概には言えませんが，一番大きな影響はグラフィックスボードの3D性能なので，あまりにも遅い場合は，新しいボードを試してみるのも，一つの手かもしれません．最近はとても高性能なボードでも，あっという間に新製品に押されて値段が急落してしまいますので，1万円以内でもすごく速いものがたくさんあります．

●キノコ，リンゴ，みかん，バナナ，魚

仮想世界の中には，いろんな食べ物があります．それらの食べ物は，木になっていたり，地面に生えていたり海の中にいたりします．キャラクターを操作して，いろんなものを集めてきて見ましょう．

まず，一番簡単に取れるのが，キノコです．家を出て斜め左を見ると，木が生えています．その木の手前の地面には，キノコが生えています．



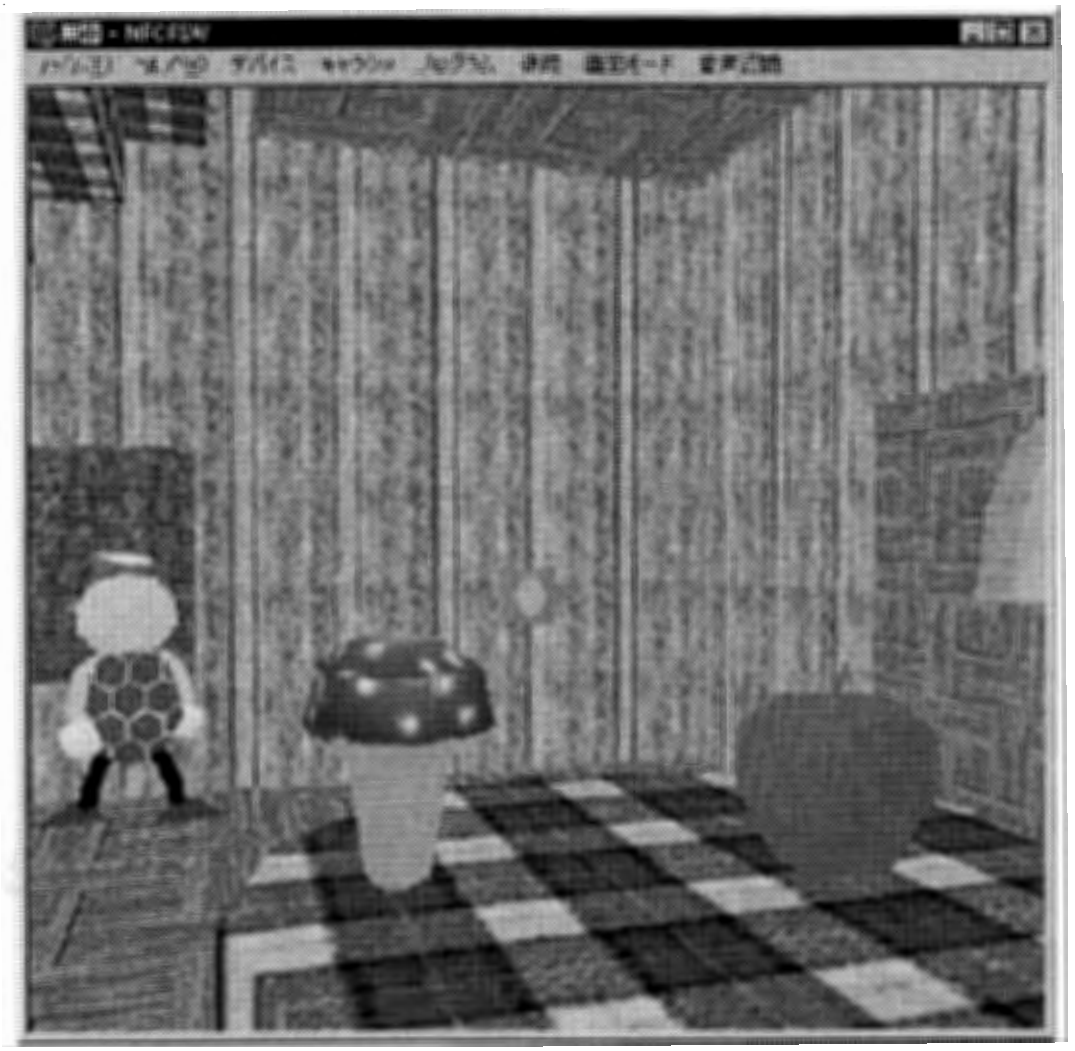
カメラをキャラクタから切り離し，そばによって行ってください．ちょうどキノコがキャラクタの足元に来るぐらいの位置で，'P'キーを押しながら前に進んでみてください．うまく行けば，キノコが採れます．

では次に，木になっているリンゴを採ってみましょう．近くにある滝の上のほうに，木があります．その木には，リンゴがなっています．

木になっている食べ物を採るのは，かなりのテクニックが必要です．一回では採れないかも知れませんが，あきらめずに何回でもチャレンジしてください．
まず，HOME キーでジャンプして，キャラクタの目線を食べ物の高さに合わせます．キャラクタは足元付近にある物を取ろうとするので食べ物が画面の下になるくらいがいいでしょう．こまめに HOME キーと INS キーを使って，ちょうどいい高さで停止してください．

次に視点をキャラクタから切り離し，リンゴと重なるくらいまで前に進めます．そして先ほどの要領で，'P'を押しながら前に進めてみましょう．うまく行くと，リンゴを採ることができます．

キャラクタは，最大 2 個の食べ物を持つことができます．持っている食べ物は，もちろん置くこともできます．'U'キーを押せば，キャラクタは足元に持っている物を置きます．家に帰って，机の上に置いて見ましょう．



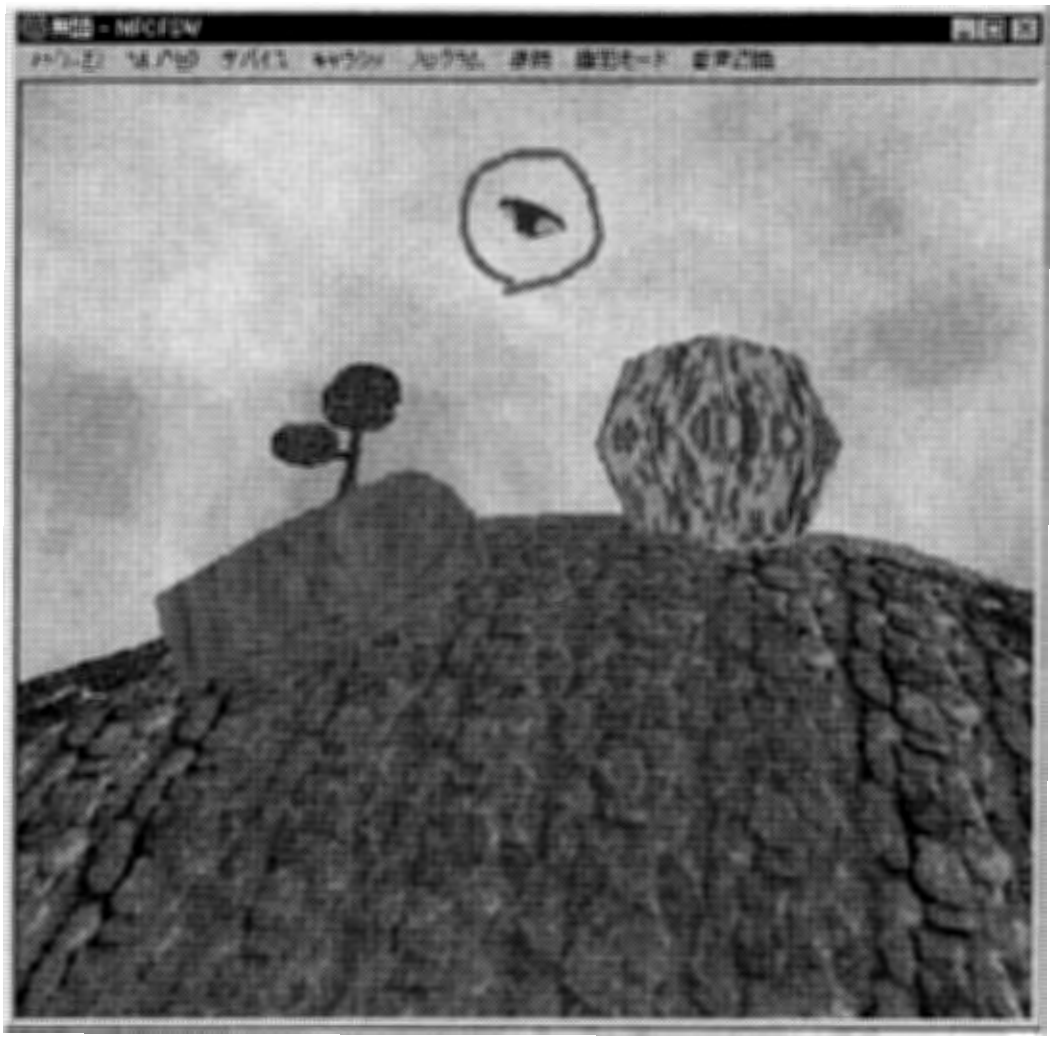
いろんな食べ物を採って他のキャラクタにあげたり，逆に鍵をかけた家に集めて独り占めすることもできます．

キー操作をまとめると次のようになります．

【P】=足元の食べ物を取る　【U】=持っている食べ物を置く

● 円盤

では次に，山の上でくるくる回っている円盤に乗ってみましょう．

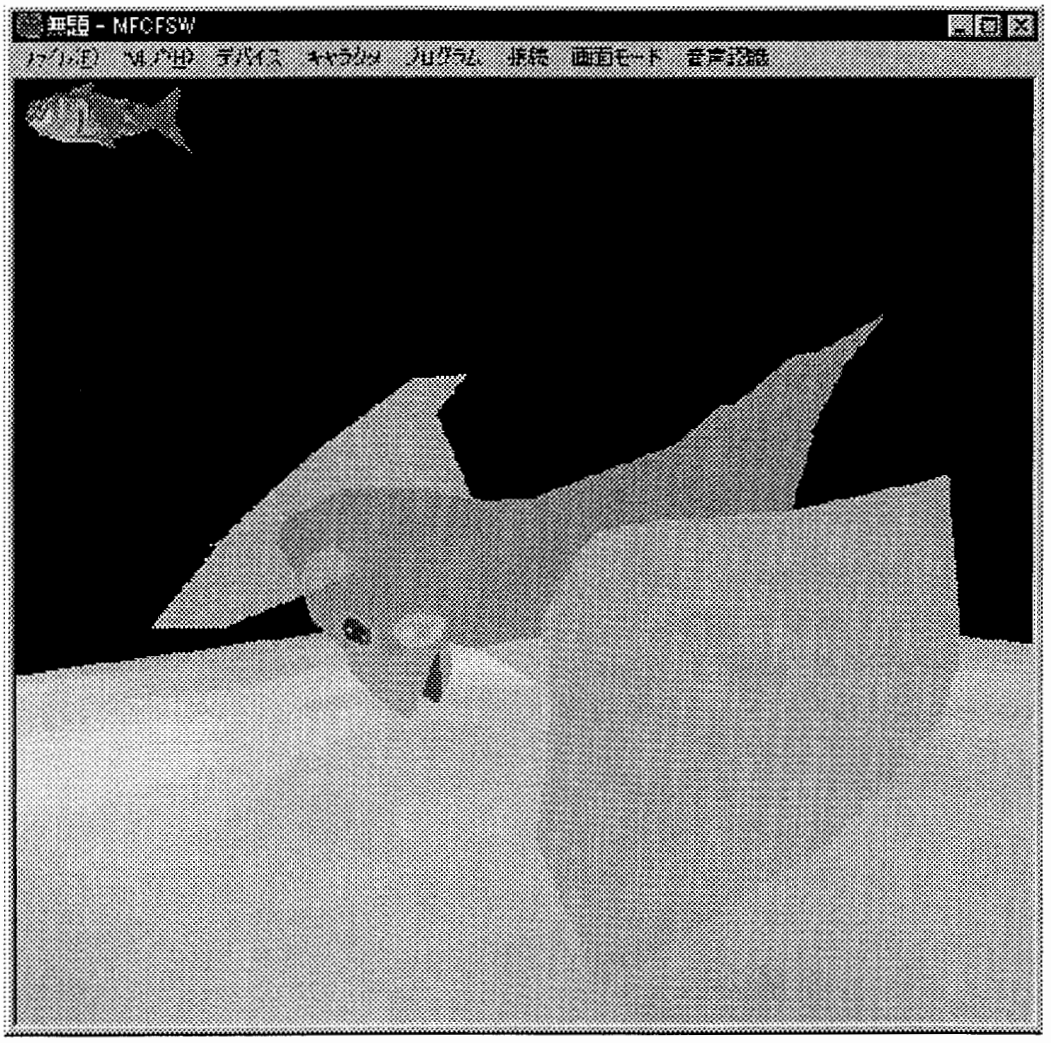


家を出て左を見ると，山肌に大きな滝があり，その上には円盤がぐるぐると回っています．まずは近づいてみます．HOME キーで飛び上がり，ある程度の高さまで行ったら INS キーを使ってホバリングしてください．後は矢印キーを操作して移動すれば，簡単に近づくことができます．

遠くからだと小さく見えましたが，近づいてみると意外に大きいことがわかります．いったい誰が何のためにこの円盤をこんなところに放置しているのでしょう？
ここで視点をキャラクタから切り離します．細かい操作をする場合は，この方が操作しやすいためです．

高度や距離を見ながら，少しずつキャラクタを近づけていってください．円盤はかなり大きいので，思い切って近づいてください．見事円盤に接触することができると…

このように，ぐるぐる回る円盤に乗ることができました．見ようによっては謎の円盤に捕らわれたようにも見えます．
ここで，'*'キーを押して，キャラクタ視点に切り替えてみましょう．すると，ものすごい勢いで周囲が回っていて，酔いそうなくらいです．じつは円盤は大きく回りながら，それ自身も回っていたのです．
では次にこの状態のまま END キーを押してみましょう．すると…



一気にはるか上空，宇宙にまでワープしてしまいました．どうやらあの円盤は，宇宙へのワープ装置だったようです．

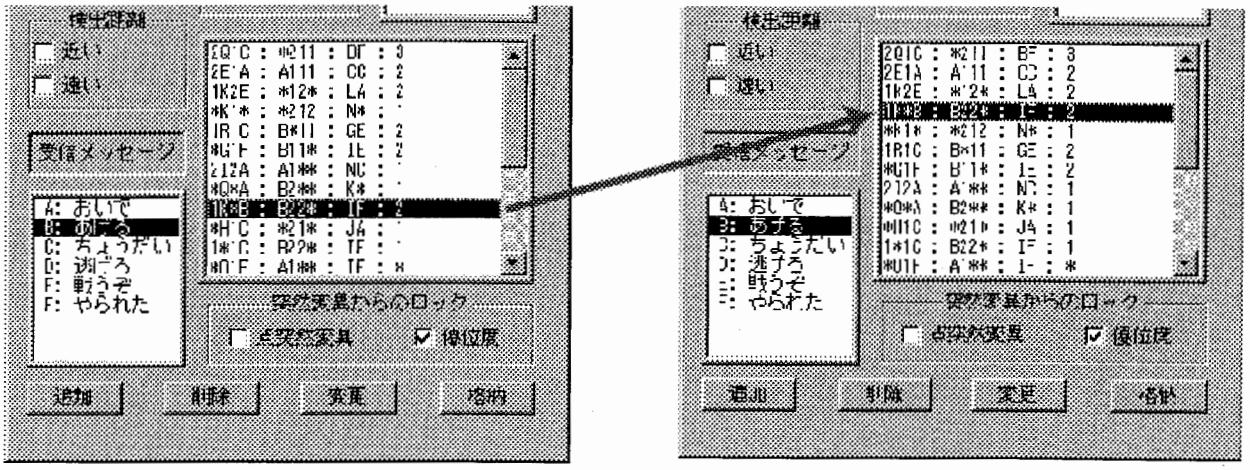
この操作には，かなりコツがいります．すぐにはできないかも知れません．円盤に簡単に乗れるようになったら，かなり上達したといえます．

●もっと賢くするために

【はじめに】の【賢い子供を造ってみよう】のところで説明したように，キャラクタが【Auto】モードのときは，キャラクタは自律的に自らの判断で行動しています．キャラクタは絶えず接触・視覚・聴覚センサを働かせて周囲の環境を認識して，これらの情報を内蔵された知能エンジンへ入力情報として送り込んでいます．またキャラクタは空腹の状態や，疲労度，感情などさまざまな内部の状態を持っています．キャラクタの各行動ルールは，これらの【入力情報】と【内部状態】の組み合わせから，次にとるべき【行動出力】を指示します．このような命令書が複数集まって一つのプログラムを構成しています．現在のキャラクタへの入力情報と内部状態を調べて，各命令書に書かれている入力情報と内部状態とのセットとを，照合していきます．もし，その命令セットの中に，自分のおかれた状態と入力にぴったりのものがあったら，その命令書の指示に従います．一つも一致するものがなければ，そのキャラクタがもともと持っていた行動癖に従って適当な行動が選ばれます．

ではプログラムを実際に覗いてみましょう．【プログラム】メニューから【設計】を選ぶと，既に以前に説明した遺伝子の設計用のダイアログ・ウィンドウが開きます．左の縦列が入力情報，真中の列が内部状態，右の列が行動出力です．右下の数字とアルファベットがたくさん並んでいるリストボックスは各命令書を数字化したものです．このリストボックスの適当な行をマウスでクリックしてみてください．違う行をクリックするたびに，入力・状態・出力の各項目のチェック状態が変わります．もうおわかりだと思いますが，右下のリストボックスの数字やアルファベットは，各項目のチェック状態を数字にして，順に並べたものです．この中にときどき【＊】の記号が含まれていることがありますが，これはその項目がどれも選ばれなかったことを示しています．実際のマッチングのときには，この【＊】記号はトランプのワイルドカードのような役目をしていて，一致したものと見なします．ただし，実際の数字やアルファベットが書かれていて一致したときより，低いポイントでマッチング得点を与えます．各ルールには順位があって上のルールほど優位度が高くなっています．決定論的なシステムの場合は，上から順に照合をして，そのマッチング得点を計算していきます．この点が予め決めた得点を超えたら，そのルールを採用して，そこに書かれている行動出力を行います．確率論的なシステムの場合は，全部照合して各ルールのマッチング得点を計算します．そのマッチング得点を総和で割ることで，そのルールが選ばれる確率を計算します．円グラフにそれぞれの確率に応じて扇型の開く角度を割り振ります．そして宝くじの抽選会のように，乱数で矢を飛ばして選ばれるルールを決めます．

ルールの優位度を変更したいときは，右下のリストボックスのルールをマウスでドラッグして好きな位置に持っていって下さい．既にあるルールの各項目のチェック状態を変更したいときは，項目をチェックしなおした後で，右下の【変更】ボタンを押してください．もし，そのルールが必要なくなったら【削除】ボタンを押してください．自分でチェックしなおしたルールを新たなルールとして追加したいときは【追加】ボタンを押します．このとき以前に選んでいたルールの下に追加したルールが加わります．

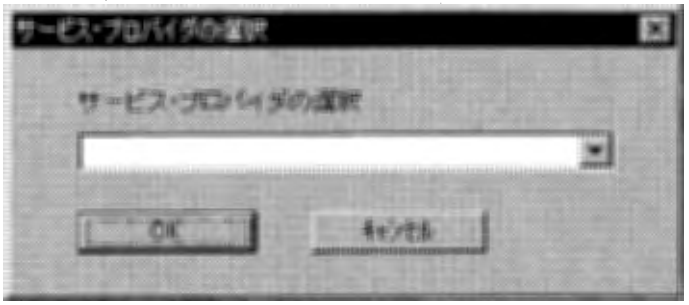


このようにしてユーザはアンケートに答えるような感じでプログラムを行っていきます．キャラクタのプログラムがうまく組まれていなければ，いずれは空腹になり過ぎたりして，死んでしまいます．このときに進化エンジンによって，プログラムの変更が行われます．【プログラム】メニューの中で【進化基準】を選ぶと，最初に図で示した【進化基準】ダイアログが開きます．このなかに表示されている【点突然変異率】に従って，各ルールの中身が変化します．例えば，あるルールの入力項目の‘りんご’が‘バナナ’に変更されたりします．また，【優位度突然変異率】に従って，ルールの優位度，つまり順番が入れ替わります．もし，ユーザがせっかくプログラムした内容を進化エンジンによって変更されたくない場合は，先ほどの【設計】ダイアログの右下の【突然変異からのロック】の【点突然変異】をチェックして下さい．逆に内容は変更されてもかまわないけど，順番だけは変えてほしくない場合は，その右にある【優位度】をチェックしてください．どちらも変更されたくないときは，両方ともチェックしてください．このロック機構は各ルールに個別に設定できて，いつでも変更することができます．この仕組みによって，ユーザのプログラムと進化エンジンとをうまく融合させることができます．全ての追加・削除・変更が完了したら，【格納】ボタンを押して，新しい知能モジュールをキャラクタにダウンロードしてください．
【プログラム】メニューの中にある他の項目については，既に説明済みなので【はじめに】の【賢い子供を造ってみよう】の図と解説を参照してください．

●ネットワーク接続

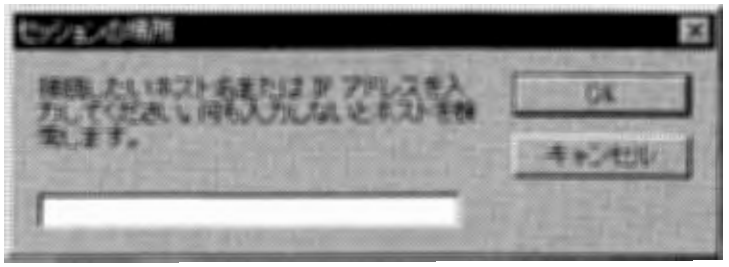
ネットワーク環境があれば，複数のコンピュータを繋いで，同じ仮想世界の中で遊ぶことができます．まず，メニューから，接続を選んでください．

ここで Connect を選んでください．既にネットワーク接続を開始している場合は，その下の Disconnect で，接続を切ることができます．
Connect を選ぶと，次のようなダイアログが出てきます．

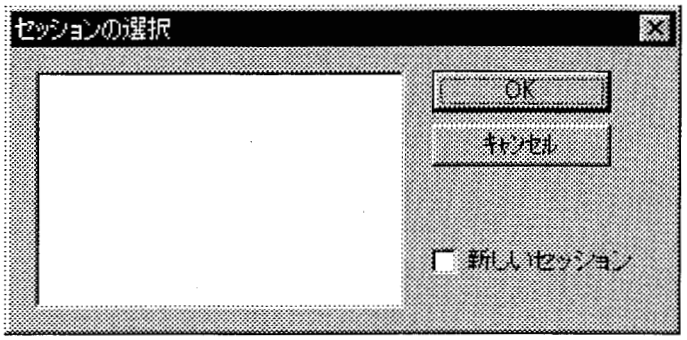


これは，通信をするために使う方式を選ぶダイアログです．

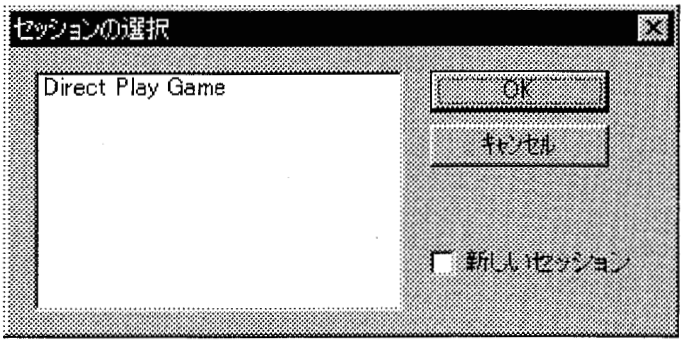
上から順に，TCP/IP，IPX，Modem，Serial という形式がありますが，通常はもっとも一般的な，TCP/IP を使います．TCP/IP を選択して，OK ボタンを押してください．



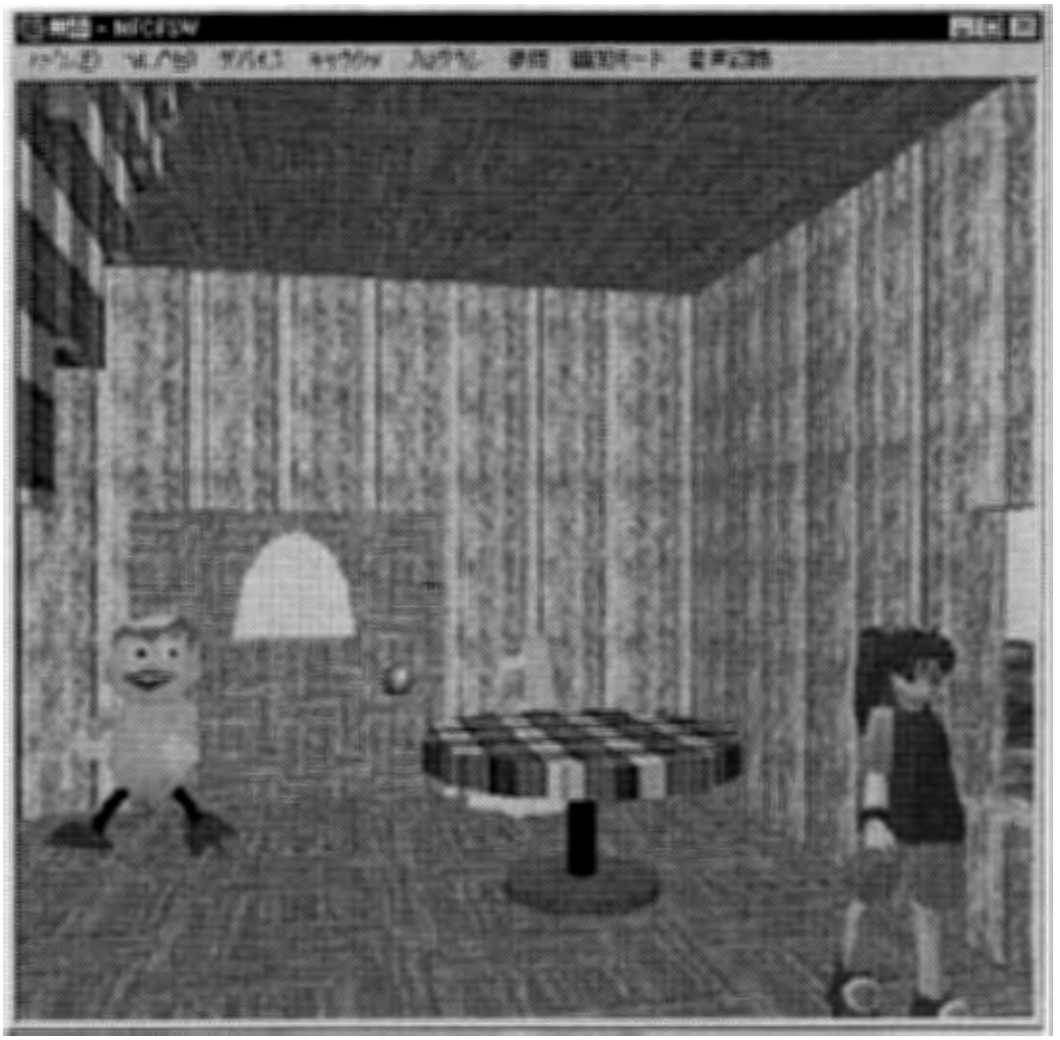
次に，既にゲームに参加しているコンピュータのアドレスを指定するダイアログが開きます．自分で新しくゲームのセッションを開始するときや，既にあるゲームセッションのアドレスがわからない場合は，何も入力せず OK を押してください．．



しばらくすると，既にあるゲームをリストアップしたダイアログが表示されます．誰もゲームをしていない場合は，何も表示されません．その場合，あるいは自分一人でゲームをはじめたい場合は，”新しいセッション”をチェックして，OK を押してください．既に誰かがゲームをしている場合には，そのゲーム名がリストアップされます．



参加したいゲームをクリックして選択すると，その行が黒く反転するので，これを確認してから OK を押してください．

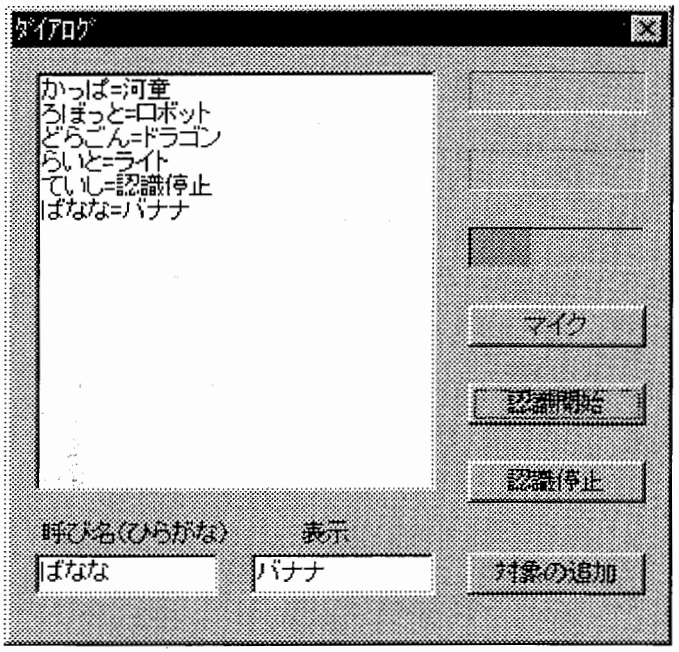


ネットワーク接続が開始され，既にゲームに参加している人のキャラクタが表示されます．「柔軟な仮想世界処理システム」のところで触れたように，ネットワークを通じて，同じ仮想世界を複数のユーザで共有することができます．あなたが取ってきたキノコを他の人にあげることもできるし，鍵のかかったドアを中から開けてもらうこともできます．

大勢でいっしょにゲームをすることができるので，いっしょにバナナを取りに行ったり，宇宙を探検してみたり，いろんな遊び方ができるでしょう．

●音声認識

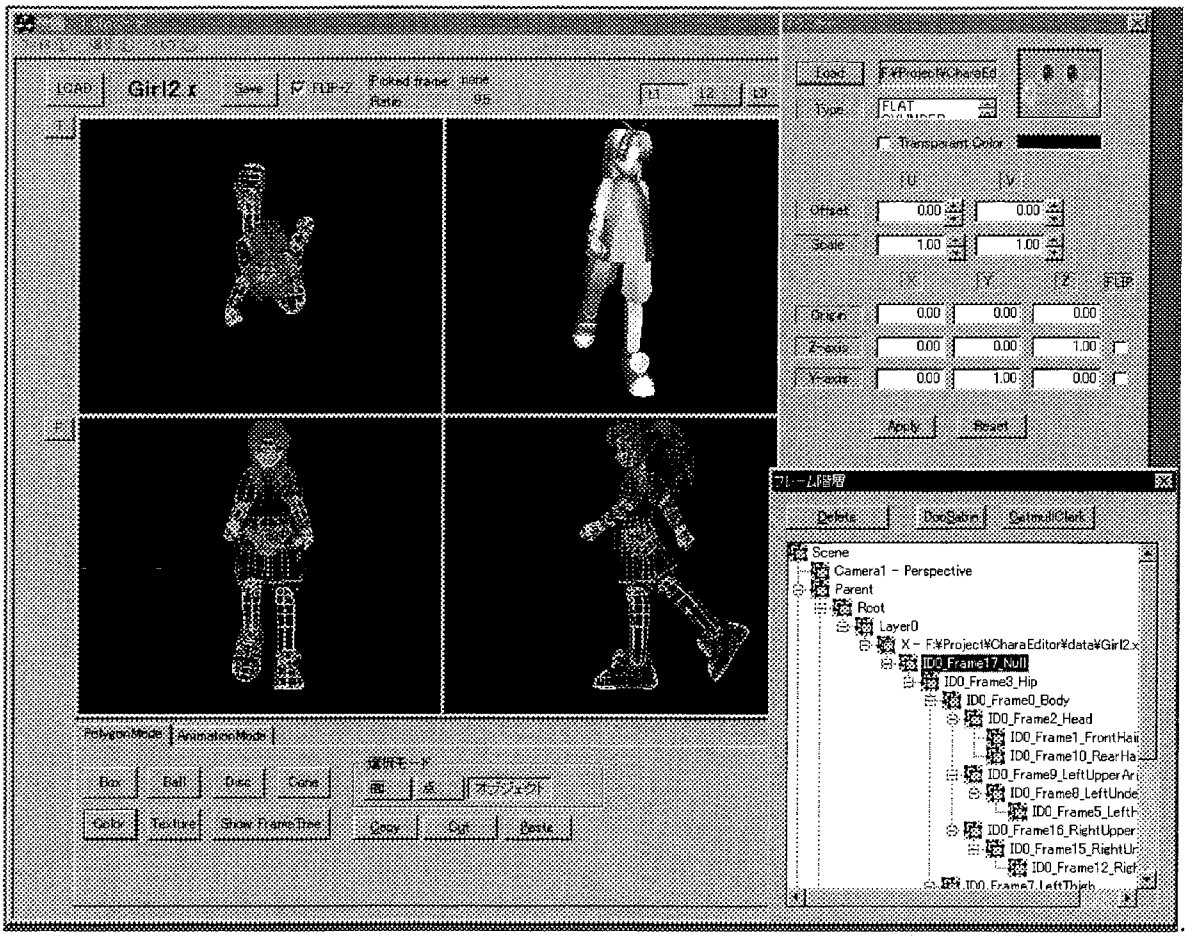
一番右のメニューの【音声認識】ダイアログでは音声認識の設定を行うことができます．【Manual】モードにするキャラクタの変更指示，ライトのON/OFFなどを音声の指示で行うことができます．マイクの音量が大きすぎたり小さすぎたりするときには，【マイク】ボタンを押して調節してください．【認識開始】ボタンを押すと音声認識が始まります．終了は【認識停止】ボタンで行います．認識対象の追加は【対象の追加】で行いますが，現在オブジェクトとのリンクに問題があり，まだ完全な実装を行っていません．今後に期待してください．

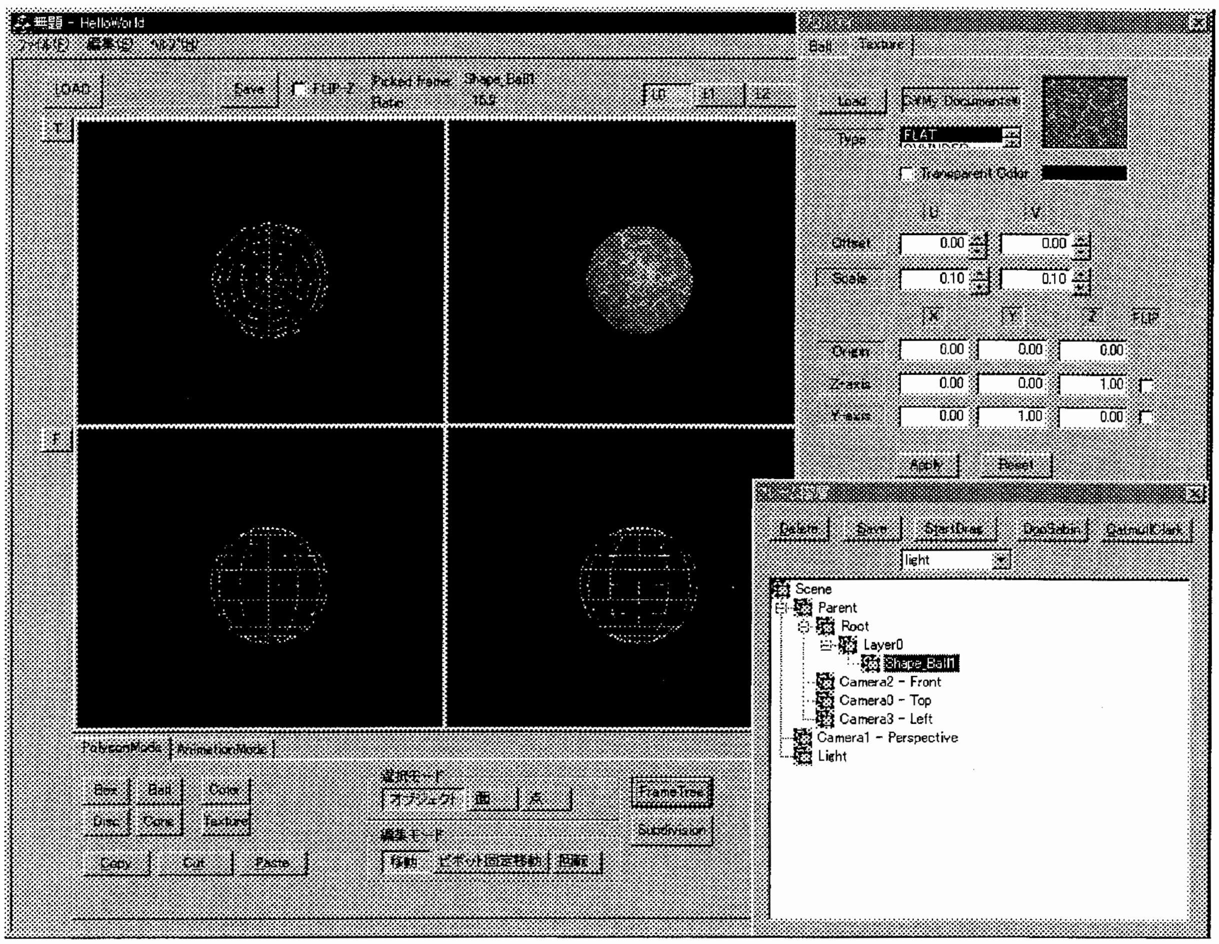


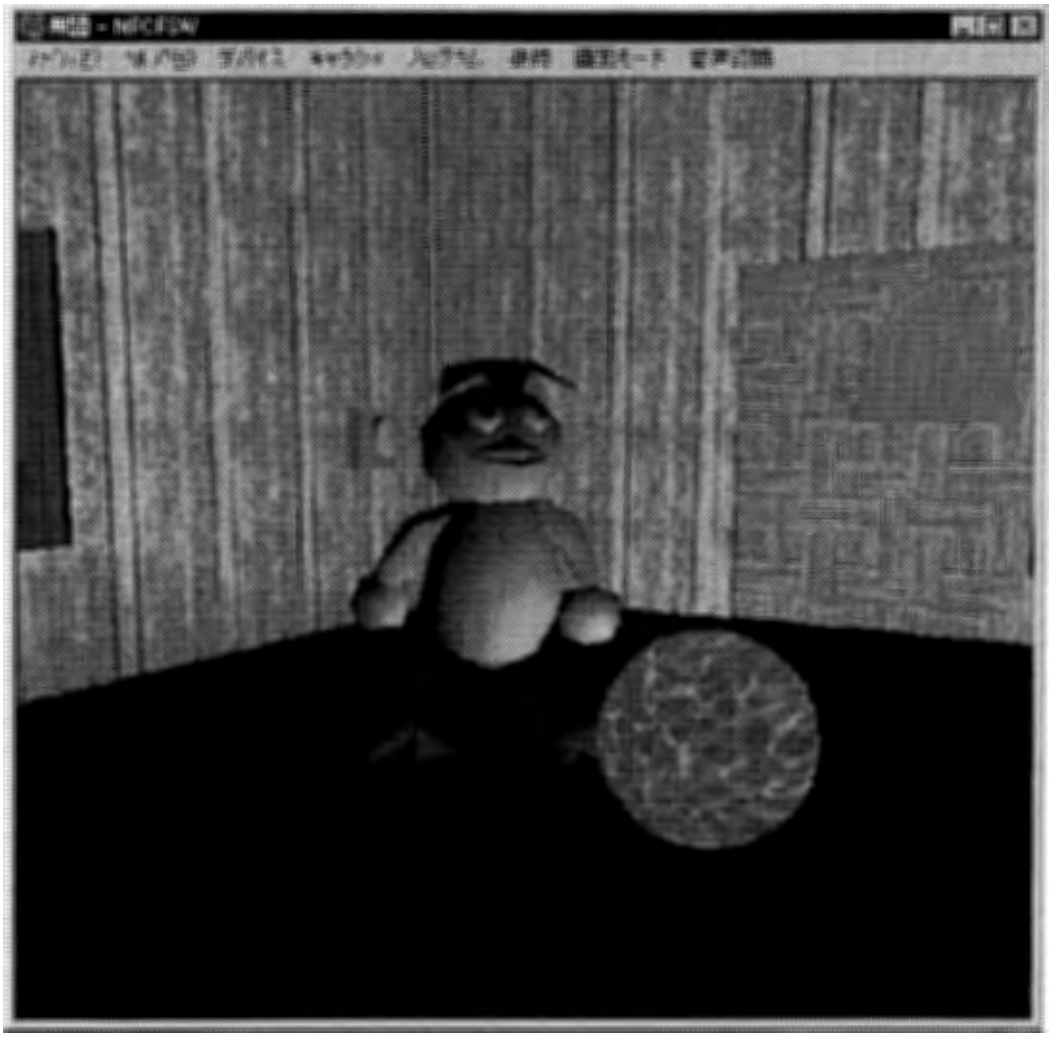
●自分達でお気に入りのワールドを作ろう

今まで説明してきた仮想世界通信システムに対応した専用のモデラを使って，ユーザ自らが少しずつ自分達の世界を構築していくことができます．例えば仲間どうしで相談して，山の上に集会場を作ろうということになったときに，モデリングの得意なAさんは集会場の骨組みを作り，絵を描くのがうまいBさんは壁の模様を描き，動く物が大好きなCさんはクリックしたらシュッという音と共に開閉する半透明の窓を作ったり，というように分担を決めながら楽しく作業を進めることができます

フィールドでの経験が増えてくると，それに応じてモデラの機能も順に増えていくようになっているので，いずれはキャラクタ自身を自分で作ることもできるようになります．また食べる・走る・蹴るなどの動作のアニメーションも自分でつけることができるので，面白いキャラクタやアニメーションを作ってネット上でお互いに見せ合うのも楽しいかもしれません．




無題 - HelloWorld
LOAD
Save
FLIP-Z
Picked Frame Shape_Ball1
Rate 15.9
Load
Type
FLAT
Transparent Color
Offset
Scale
Origin
Apply
Reset
Delete
StartDrag
CatmullClark
light
Scene
Parent
Root
Layer0
Shape_Ball1
Camera2 - Front
Camera0 - Top
Camera3 - Left
Camera1 - Perspective
Light
PolygonMode
AnimationMode
Box
Ball
Color
Disc
Cone
Texture
Copy
Cut
Paste
FrameTree
Subdivision



ワールドの中のフルーツや魚などの食べ物の形や属性もユーザによって作ることができます．後で順を追って説明していきますが，キャラクタはいろいろな遺伝子を親から受け継ぎます．食べ物の消化能力もその一つです．この栄養消化に関係した遺伝子は家系によって異なるので，あるキャラクタにとって紫色のキノコがとても栄養価に富んだ食べ物であっても，他のキャラクタには猛毒かもしれません．それはキノコも遺伝子を持っていて，その遺伝子によって決まる栄養とキャラクタの消化能力の組合せで栄養価や毒性が変わってしまうからです．またキノコもキャラクタに食べられることによって，より広まることができるのなら，結果的には世代と共にそのキャラクタにとって，より栄養価の高い特性を持つように進化していくことになるでしょう．この食べ物の遺伝子は突然変異などを制御する進化エンジン（ワールドの進化を促進するための機構）によって変化していきますが，ユーザが遺伝子を直接操作してフィールドで栽培することもできます．

●子供の結婚相手を選んでみよう

人工的に造られたキャラクタは親からの形質を受け継ぎつつ，ある程度の突然変異によって，少し親とは異なった特徴を持っています．ユーザがある程度キャラクタに感情移入をしやすくするために，性別のないバクテリアのような増え方ではなく，やはり性別があって，有性生殖によって増えていく方がより自然に感じます．また理論生物学的な観点からも有性生殖の方が，遺伝子の欠損やコピーミスなどのエラーに対して強く，かつ環境の変動に適応しやすいことが証明されています．性別があるときに問題なのが結婚の問題です．まずどのような基準で結婚させるのか，ということを考えてみましょう．今までの進化システム論を使ったシュミレータでは，与えられた課題をよりよく解けた個体を選んで増やしてきました．このような評価・選択の過程を何世代も繰り返すことによって，次第に課題をよりうまく解く個体が現れることを期待するのです．しかしこの発想は，あくまでも主体はシステムもしくは設計者の方であり，個々の個体はただ造られて，選ばれたり捨てられたりするだけの存在です．私達はできるだけワールドの住民の人権？を尊重してあげたいので，世代交代のタイミングや結婚するときの結婚相手の条件も全てユーザ側にまかせることにしました．しかしコンピュータの性能には限りがあるので，無限に子供を増やすことはできません．自然界でも食べ物などのリソースに限りがあるときには個体数も飽和してしまいます．私達のシステムは，基本的には男女のキャラクタのカップルから二人子供が授かるとしています．性別は2分の1の確率で決まります．どちらの子供を選ぶかはユーザどうしでよく話し合って（チャットして）決めた方がよいでしょう．しかし，必ずしも結婚する必要はないので，一生独身を貫くこともできます．そのときは予めユーザが設定した寿命がきたら命が尽きてしまうので，ユーザは新たにキャラクタを作成します．また独身者が増えてきているにもかかわらず，死亡後に新たにキャラクタを作成するユーザが減ると，総個体数が減ってきます．このように個体数に空きができたときには，ちょうど運良く結婚適齢期になったカップルが3つ子を授かったりします．結婚をせずに恐ろしいほどの長老になることもできてしまいますが，これはこれで生まれ変わるチャンスを失うので，必ずしも得策とは思えません．ましてや，環境がどんどん変わっていくのに，世代交代がなくなってしまうと，進化がストップしてやがて適応能力が極端に落ちてしまいます．ユーザは通常，キャラクタを自分で造るときは，まずモデンリングしてテクスチャを貼り，そして各動作のアニメーションを付けます．そしてそのキャラクタの行動基準となる遺伝子セットをプログラムして，さらに性格付けも行います．このときに，死んでしまうときの条件と，結婚相手の条件も合わせて指定しておきます．例えば死亡条件は，年齢や過度の空腹状態，仕事の失敗回数，などを指定します．結婚条件は，“魚つりがうまくて，ポリゴンの少ない若い男性”などを指定します．お互いの条件が合ったときに，結婚して両親の遺伝子を併せ持った子供が生まれます．

細かな取り決めごとは，それぞれのコミュニティにまかせるべきだとは思いますが，私達は，基本的には“捨て子禁止”の方針を持っています．道義的にも問題がありますし，捨て子を管理するサーバの必要が生じてしまうからでもあります．ただし，ユーザどうしの合議の上であれば里子も可能にしようと考えています．

●今後のバージョンアップの予定

ワールドオブジェクトをよりフレキシブルに生成するための機構や，音声認識システムを有効に活用したユーザインターフェースなどを現在開発しているので，次回のバージョンアップを期待していてください．

文責：
和田健之介・中口孝雄・和田佳子