



Theoretical Science Group

理論科学グループ

```
int fp7220::OpenSocket( void )
{
    struct hostent *hp;
    struct sockaddr_in sin;
    unsigned timeout;
    int result, s;

    // make socket
    if ( !(hp = gethostbyname(szServerLoc)) )
        return -1;
    if ( (s = socket(AF_INET, SOCK_STREAM, 0)) < 0 )
        return -1;

    // setup parameters
    memset(&sin, 0, sizeof(sin));
    sin.sin_family = AF_INET;
    sin.sin_port = htons(7220);
    memcpy(&sin.sin_addr, hp->h_addr, hp->h_length);

    // set timeout and handler before connect
    signal( SIGALRM, SIG_IGN );
    timeout = alarm( uConnectTimeOut );
    result = connect(s, (struct sockaddr *)&sin, sizeof(sin));
    signal( SIGALRM, SIG_DFL );
    alarm( timeout );
    if ( result < 0 )
        return -1;

    return s;
}
```

猫も杓子もネット沸く?(謎)

部報 210号  
— 駒祭パンフ号 —

目 次

駒祭企画紹介	1
ケルト十字展開法によるタロット占い . . . . . 【-1ParaGRAPH】	1
- - コンピュータと音楽 - - . . . . . 【うえ】	1
最弱オセロリーグ . . . . . 【げる】	2
3D フラクタル . . . . . 【HASM】	3
ネットワーク対戦型 RPG 制作記 . . . . . 【わたる (wataruk@tky.3web.ne.jp)】	4
ネットワーク対戦落ちゲー for Win95 . . . . . 【ばんだい】	6
Map Friends Project - mfp - . . . . . 【すーゆー】	7
一般記事	9
PPP を作ろう . . . . . 【TEA(定兼 邦彦)】	9

## 駒祭企画紹介

### ケルト十字展開法によるタロット占い

-1ParaGRAPH

タロットの起源についてはインド説、中国説、エジプト説など諸説ありますが、その中に表れている世界観、モチーフ等からエジプト説が有力です。さらに言えば、古代エジプト起源のものをカバラ思想に結び付けて理論体系が出来上がっているとされているようです。

タロットの構成の中で驚くべきことは「0」のナンバーを持つ「愚者」のカードが存在することです。これはタロットを作り出した人々が既に0の概念を持っていたことを示しています。このあたりがインド起源説の基となったのではないのでしょうか。

無と有の狭間にある万象を人間の深層心理にあるイメージと結び付け、占者を介して具現化させるタロットは心理学者ユングも研究の対象としたと言われていました。

タロットには大きく分けて、太陽や月、死、正義などのモチーフを単独で司る22枚の大アルカナと金貨、剣、聖杯、こん棒を、それぞれについて1～10の数字とキングクイーン、ナイトページで司る小アルカナがあります。

小アルカナはナイトとページがジャックに統合されることによって、現在のトランプへと変化しました。今回行うケルト十字展開法は22枚の大アルカナのみを用いて行い、一つの事柄を過去、現在、未来にわたって詳しく占うのが特徴となっております。

タロットは被占者の深層心理を写す鏡。

たとえ好ましくない結果が出たとしても、悲観すべきではありません。死は新たな再生の始まり、運命の輪は回り続け、太陽はまた昇ります。

### - - コンピュータと音楽 - -

うえ

DTMという言葉をご存知ですか？ Desk Top Music の略で、コンピュータで音楽を作ることです。

コンピュータを使うと、自分のイメージがあっという間に形になります。作曲用のソフトウェアは充実しており、ちょっと音楽の知識のある人ならすぐにオリジナルの音楽を作ることが出来るでしょう。

## 最弱オセロリーグ

---

また、現在の標準音源といえる MIDI は（低いレベルではですが）汎用性が高いので、自分の作品を多くの人に鑑賞してもらうことができます。

今回、私は主に「ネットワーク対戦 RPG」に使う曲を作曲してみました。音源はローランドの SC-88Pro です。

電子音楽に興味のある方は、是非聞いてみてください。

### 最弱オセロリーグ

げる

### 最弱オセロとは

皆さんご存知のオセロです。通常のオセロのルールでどこまで弱くできるのか、それぞれの思考ルーチンで対戦し、最弱のルーチンを決めようというものです。

それぞれのルーチンと人間との対戦も出来るようになっていきます。最弱のものは普通の人では絶対に負けられないくらい弱くなってます。

腕に自信のある方はどうぞお試し下さい。

やり方は見れば分かると思いますが、分からなかったらそこら辺の人に聞いて下さい:)

### ルール

- 黒が先手
- 相手の石を 1 個以上返せる場所にしか打てない
- どこかに置ける場所がある場合にはパスはできない
- どちらの石も置けなくなった時に終了
- 終了時に空白がある場合には折半する

### 終り

今、自分でも思考ルーチンを書いているけど、全然弱くない。どうしよう(^^; 果たして、当日までに間に合うのか...

## 3D フラクタル

HASM

フラクタルというのは自己相似性を有し、かつ無限に細かく複雑な構造を持った図形のことです。さまざまな種類のものがあり、どれも面白い形をしたものばかりです。有名なものにマンデルブロー集合やジュリア集合などがあります。自らプログラムを書いて描いてみたことのある方もいらっしゃると思います。

フラクタル図形は平面図形 (2D) として描かれることが多いのですが、今回は何種類かの 3D のフラクタル図形を PC-9821 上で描くプログラムを製作しました。3D にすることによって計算量、使用メモリともに莫大なものになるので、全てアセンブリ言語で書いた上で、特殊な方法によりリアルモードからプロテクトメモリを直接利用しています。

紙面の都合により今回扱っている図形を全て紹介することはできませんが、そのうちの 1 部の図形について説明をしたいと思います。

1 つ目は、1 つの正 4 面体の各面に、新たに辺の長さが半分の正 4 面体を付け加え、さらにその半分の辺の長さの正 4 面体を各面に付け加え...という操作を無限に繰り返して得られる図形です。これはいわゆる再帰図形と呼ばれるもので、2D ではコッホ曲線が有名です。

フラクタルとして十分に許される程度まで細かく正 4 面体を付け加えて行くと、最終的に描く正 4 面体の数は大体 1000 万個程度 (実際にはプログラムの都合によりもっと増える) という膨大な数になります。そのため、全てアセンブリ言語で書いて Pentium 上で動作させていながら、なお描画速度には難があります。

2 つ目の図形は、簡単にいうとジュリア集合の 3D 版です。2D のジュリア集合の描画については、漸化式

$$z_{n+1} = z_n^2 + A, z_0 \in C, A \in C$$

について  $A$  の値を固定し、複素平面上で  $z_0$  を動かし、 $z_n$  の収束発散をみて、それに対応した色の点をうつという方法が一般的ですが、ここで  $A$  を固定せずに  $z_0$  と合わせて動かすと複素数の変数が二つとなり、4 次元空間におけるフラクタルとなります。その 4 次元空間から適当に 3 次元の空間を切りとって表示すればめでたく 3D フラクタルとなるわけです。

当たり前のことですが、そのまま点を塗りつぶすと空間が埋めつくされて何が表示されているのか分からなくなってしまうので、透過処理などを施しています。

まあ、こうした理論はともかく、百聞は一見にしかずということととにかく見てみて下さい。フラクタル図形の美しさを満喫して頂けると幸いです。

## ネットワーク対戦型 RPG 制作記

わたる (wataruk@tky.3web.ne.jp)

### はじめに

Diablo, Ultima Online, Might and Magic VII, ...

パソコンゲームの世界では、ネットワーク対戦の RPG が大流行しています。これは、ゲームの進化の最終形態かも知れません。そこで、私たち TSG でも作ってみることにしました。

クライアントは、DirectXのおかげでゲーム OS としてもすっかり定着した感のある、Windows 95 用になりました。サーバは、やはり流行の PC-UNIX で作ることにしました。

なんてミーハーなんだ。(笑)

### サーバは単なるチャット

peer to peer で形成したネットワークゲームも作ってみたいし、負荷分散という点から、それが理想だとは思いますが、しかし、やはり現状のインターネットのインフラでは無理です。28800bps のモデムで PPP 接続している場合、同じデータを何回も送信する余裕はまったくありません。データを効率的に転送するために、どうしても中継地点が必要です。それがサーバになります。

プロトコルは、TELNET 形式で ASCII 文字列をやりとりするだけにしました。拡張やデバッグが容易であり、バイトオーダーの問題もなくなるからです。パフォーマンスは UDP に劣りますが、倍も違うわけではありません。もしネットワークの帯域不足が問題になるようならば、プロトコルを換えるより、通信量自体を削減すべきでしょう。

以上のことまで考えたとき、実はサーバは単なるチャットでよいことに気がつきました。たとえば座標 (1,1) に移動するときは、「/move 1 1」といった具合に発言するわけです。NIFTY-Serve などでは、古くからチャットでテーブルトーク RPG がプレイされていましたし、CRPG だって当然可能なはずですが。

クライアントは、他のプレイヤーとモンスターや NPC を、チャットを経由して同列に扱うことができます。モンスターや NPC の処理を、他のマシンに分散させるのも容易です。チャットのメンバーに『ゲームマスター』サーバを参加させれば、流行の 3 層構造になって、さらに愉快かも知れません。

よいことばかりですね。

## キャラクターの位置情報の問題

プレイヤーのステータスで一番頻繁に変化するのは、位置座標です。先ほどの例のように、「/move 1 1」などというメッセージを送ることにすると、一秒間に幾度も似たようなパケットを発信することになります。ネットワークの情報流量削減のために、ひと工夫したいところです。フライトものや、Quakeのようなアクションゲームは、自分の座標と移動速度を定期的送信する方式が多いようです。これをRPGにそのまま適用するのは、どうもうまくありません。

ここでDiabloを思い出してみましょう。操作はフルマウス・オペレーションで、キーボードでは歩けません。移動先の地点をマウスでクリックすると、そこに向かって自動的にコースを判断し歩き出します。もうわかりましたね。移動開始地点と目標地点を送信してやればよいのです。各クライアントがちゃんと同じアルゴリズムを使えば、移動経路も等しくなるはずですよ。

この方式を導入すれば、ネットワークの負荷は劇的に軽くなりそうです。しかし、私はこれを用いるのをやめました。マウス操作と移動コースの自動判定を実装するのが、めんどくさかったからです。(こら)

情報量の削減がどうしても必要になったら、手を着けるつもりです。

## 話の続きは

残念なことには紙面が少ないので、短いですがこれでひとまず終わりにします。NPCの行動アルゴリズムや、乱数やフラクタルからのマップ生成など、おもしろいテーマはいろいろ残っています。駒場祭の会場か、どこかの仮想世界で、話の続きをやれるといいですね。



## ネットワーク対戦落ちゲー for Win95

ばんだい

最初は落ちゲーだけでした。しかし、落ちゲーといってもただの落ちゲーではありません。史上初のブロックの絵や落ちるスピード、消えるまでの個数、...その他いろいろなゲームのプロパティを自分で自由に書き換えられる、いわば「落ちゲーツクール」的なものを目指していました。

しかし、制作途中でさたんだけだかぶれすてでかで「落ちゲーデザイナー、作ってポン」とかいうソフトが出て、先を越されたか、とか思いましたが、あれはかなりへぼくさいので無視。で、製作は続けました。

### 無理でした

開発当初は VC4 を使い始めたころで、かつ、Windows プログラミングも始めてだったので、MS に流されるまま、(今思えば)MFC という超弩級にあやしいもの<sup>1</sup>を使って作っていました。

しかも、ちゃんとドキュメントビュー構造<sup>2</sup>です。しかし、作っていくうちにクラス使うのが面倒になってきたり、訳が分からなくなってきたりしたので、やめかけました。ここでは、とりあえずブロックが落とせて、左右に動かせて、でも、消えない、ということまでできていました。

その時、私は急にネットワークにするか、と思い始めました (というか、周りではやってそうだったから)。ネットワーク対戦落ちゲーも割と行けるかも知れんな、と思い、急遽ソケットのプログラミングに走りしました。

最初は Win95 で TCP/IP チャットサーバーを作り始めました。で、一応動いたのですが、よく考えると、このゲームはインターネットにつないでできる (予定) なわけだから、Win95 じゃだめじゃん、と思い、急遽 UNIX 版も作ることにしました。で、一応できました。TCP/IP でどこでもチャットができるようです<sup>3</sup>。

さて、ソケットの方はできましたが、肝心のゲームのクライアントの方ができていませんでした。しかし、まだ、ソースはわけがわかりにくいクラス (さすがにもう MFC は使ってない) が大量にあって、直す気もおきません。そこで、私は、しょうがないので、エディット機能を削ることにしました。しかし、ここはネットワーク対戦にするとき、エディット機能で対戦のルールが違っていたら困るので、どうしようかとも思っていたところでした。そしてさらに、クラスも捨てることにしました。つまり、C で (C++ じゃない) 一からまた書き始めました。さらに、滑らかに落ちる機能も削りました。でも、まだ動いていません。本当にできるのでしょうか。

<sup>1</sup>ほとんど Win32API と同じで、クラスにしてる意味が無いと思うのですがね。

<sup>2</sup>ゲームには向かん

<sup>3</sup>私がいるときだけ駒場でちょこっと動かしたりします



しかし、チャットサーバは一応、動くので、それを展示したりして。  
あ、でもそれなら italk があるか。  
いや、Win 版 italk サーバ、とかいって....  
意味ねー。

## Map Friends Project - mfp - mfp is not My くるフロップ

すーゆー

### THE Friends Project とは...

<http://sodan.komaba.ecc.u-tokyo.ac.jp/~kazuhofriends/> 参照。と言うだけでは寂しすぎるので(^^;;、少し説明すると、情報棟を少し利用しかけると誰もが作りかける finger 連発シェルスクリプト<sup>1</sup> じゃああんまりだということで、有志(おくーほさんと Aleph-NUL さんとたなかあさん)が作り上げた fingerd キャッシュデーモンとそのクライアント群の総称です。

駒場に瞬く間に広がった friends ですが、問題が無かったわけではありません。ecc-as50, ecc-as51 (当時) に負荷をかけまくったり、急造プログラムであったがゆえに良く落ちたり、connect 制限がきつかったりとかかなり重かったものと記憶しています。

1997 年度になって sodan.komaba.ecc.u-tokyo.ac.jp の創設、friendsd の改訂(新プロトコルの実装、非駒場依存)が行われました。

そこで、私がクライアントの作成を買って出たのですが、、、それまで使われていた Aleph-NUL さんのクライアント rfpmap を新プロトコルに対応&キャッシュの実装させるだけで半年近くお茶を濁していたのでした。

### Map Friends Project とは...

なんか駒祭近いです。TSGer は企画を持たなくてはなりません。しかも何らかの展示をしなければ罰ゲームとして、良く知らないアニメ夢のクレヨン王国のシルバー王女のコスプレをしつつ、オープニングテーマン・パカ マーチをカラオケで歌わなくてはならないそうではないで

<sup>1</sup>全てのサーバに finger @ する邪悪シェルスクリプト。負荷が大きいですので真似禁止:)

すか<sup>2</sup>！

そこで、急遽ここでまたお茶を濁して済ます案を模索して出てきたのが、以前から作っていた mfp だったと言うわけです<sup>3</sup>

というわけでお分かりでしょう。mfp とは Friends Project 下で動作する非駒場依存な新クライアントプログラムです。

### 何が大変かと言うと...

クライアントソースコードを非駒場依存にする際にもっとも大変なのが検索です。依存して良いのであれば `int xst[54]` みたいに静的に確保した配列にチェックを入れるだけで良いところが、

1. 情報収集 (友人情報, ログイン情報, 地図情報)
2. 地図 (端末名) をキーにログイン情報からログイン者 ID を検索
3. ログイン者 ID をキーに友人情報を検索し友人が判定

てなことを動的に行わなくてはならないので、友人情報とログイン情報には AVL 木を使用しました。

これが大変だろうなぁと思ってしばらく (半年) やる気を失せさせていたのですが、ふとしたきっかけでやってみるとこれがあっさりできたので、やってみるかなと現在に至っています。

しかし、実は一番大変なのはユーザインタフェースであった事に最近気づいて戸惑っています (^\_^;; 駒祭が終了しても開発は続行しなくてはならないので、今からまた気が滅入ってきたなぁ。

駒祭ではデモが動くまでにできていたら誉めてやってください！)

---

<sup>2</sup>自分で決めた事だけど (お

<sup>3</sup>djgpp libc v2.01 for PC-AT/PC-9800 PATCH を展示しても良かったんだけど、知名度低いし:D

## 一般記事

### PPP を作ろう

TEA(定兼 邦彦)

#### ごあいさつ

こんにちは。すっかりごぶたさの TEA です。現在製作中の DOS 用 PPP について説明します。

#### 動機

PPP というのはみなさんご存じでしょうが、Point-to-Point Protocol の略で、パソコンからモデムでネットサーフィンするときに使います。ぼくは普段は NIFTY SERVE を使っていますが、料金が非常に高かったのです(注:12月からは月15時間まで2000円になるのでそんなに高くない)。そこで、安いインターネット接続プロバイダを探しているのですが、どこも PPP が使えないと繋げません。

Windows や UNIX ではちゃんと PPP が使えるので問題はないのですが、僕の場合はメール読みと italk だけで十分なので、それだけのためにメモリを食う Windows を入れたくありません。というわけで 9801 の DOS から PPP で telnet を動かしたいなと思ったのです。

ちなみに DOS でも DOS/V ならば EtherPPP や dosppp というソフトがあり、ちゃんと DOS から PPP できます。本当は DOS/V を買ってしまふのが一番いいのでしょう。

#### 息ぎれ(違)

PPP を作ろうと思ったのはずいぶん昔なのですが、よくわからないのでやっていませんでした。でも ZZZ というプロバイダがあって、そこがダイヤル Q2 を使って 1分3円ということをやっていたので、試しに普通の通信ソフトでアクセスしてみました。すると

## PPP を作ろう

---

```
login:zzz
Password:
PPP session from (203.138.254.226) to 203.138.254.231 beginning...
~ }#!}!}!} }4}"&} } } } }%}&
```

こんなかんじでした。これを見て、なんか作れそうだなと思いました。適当ですね。

### 何が必要か

「DOS 用 PPP を作ろう!」と思ったのですが、よく考えたらいろいろ作らないといけないんですね。少なくとも

- パスワード入力
- PPP
- IP
- TCP
- telnet
- 漢字コード変換

を作らないといけないですね。なんか不毛だ。本当なら汎用のドライバにするべきなんだろうけど、まずは italk ができるようにしようと思ってやっています。というわけで、italk をするのに最低限必要なことを説明します。

telnet で文字列を送る場合、まずその文字列の前に TCP のヘッダをつけます。次にその前に IP のヘッダをつけ、さらに PPP のヘッダを付けて、エスケープ文字を入れてモデムに送ります。反対に、モデムから何か送られてきたら、PPP のヘッダを取り、エスケープ文字を取り、TCP/IP のヘッダをとると文字列が得られます。

モデムとのデータのやりとりは、DOS 標準の関数を使います。C 言語では stdaux ですかね。なお、この部分は竹島くんの AUXTERM を参考にさせて頂きました。

### PPP

PPP のデータは 0x7e で始まり 0x7e で終わります。データ中に 0x7e が入っている場合は、エスケープ文字 0x7d と 0x7e の 0x20 = 0x5e の二文字になります。また、エラーチェックのために 16bit の CRC がついているので、送るときは CRC を計算しなければなりません。

PPP で相手とデータをやりとりするには、決めなければならないことがあります。これは LCP (Link Control Protocol) で決めます。ここでは、PPP ヘッダの圧縮や、制御コードの送りかたを決めます。相手がヘッダの圧縮を使いたいと言ってきた場合、こちらでそれが実装さ

れていれば ACK を返します。実装されていない場合はそれはできないと断ります。こちらと相手が共に相手の要求をのむまで妥協しながら繰り返します。

なお、LCP のあとに PAP や CHAP でパスワードを聞かれるかもしれません。

## IP

IP の場合も、IPCP (Internet Protocol Control Protocol) を使って決めることがあります。IP ヘッダの圧縮と、こちらの IP アドレスです。相手の IP アドレスは送られてきたものを使います。こちらの IP アドレスは 0.0.0.0 を使うと相手に伝えると、それは拒否されてこのアドレスを使いなさいと言ってくるので、それを使いますよと返事すればいいです。

これが済めば、IP のやりとりが出来るようになります。

## TCP

TCP のコネクションを張るには、TCP ヘッダの SYN フラグを立てて、こちらからのデータの順序番号を送ります。すると、相手からは SYN に対する ACK と、相手からのデータの順序番号が送られてきます。これに ACK を返せばいいです。

あとは TCP のヘッダをつけてデータを送るだけです。IP のヘッダに自分と繋げたいところの IP アドレスを書き、TCP のヘッダに相手のポート番号 (telnet なら 23、italk なら 12345) を書きます。なお、IP アドレスが分からない場合は、どこかの DNS に聞く必要があります。

TCP はヘッダとデータに対してチェックサムを計算する必要があります。サムと言ってもアム口のだんなではありません。このときに、疑似ヘッダと言う、TCP のヘッダには含まれない IP アドレスを入れたヘッダを作り、これを含めてチェックサムを計算します。ただし疑似ヘッダは実際には送られません。また、データが奇数長の場合は最後に 00 があるものとして計算します。

なぜだか理由は分かりませんが、IPCP がすんだあとにすぐに TCP のパケットを送ってもだめでした。そこで少し wait を入れています。なんか間違ってるのかな。

また、italk の場合はこれでいいのですが、telnet では最初に相手から FF FD 18 が送られてくるので、こちらは FF FC 18 を送らないといけないそうです (Applause くんありがとう)。

## これから

やることはたくさんあります。

## PPP を作ろう

---

- window の制御, パケットの再送
- DNS を使って名前でひけるように
- ICMP (ping や traceroute に反応するように)
- PPP および TCP/IP ヘッダの圧縮
- PPP での認証 (PAP, CHAP)

そのうちやるかもしれません。(お

### 連絡先

感想などは、[sada@is.s.u-tokyo.ac.jp](mailto:sada@is.s.u-tokyo.ac.jp) までお願いします。  
ホームページは <http://naomi.is.s.u-tokyo.ac.jp/~sada/> です。

### 参考文献

- [1] 西田 竹志: TCP/IP, (株) ソフト・リサーチ・センター
- [2] RFC Index Search Form, <http://www.nexor.com/public/rfc/index/rfc.html>

## 編集後記

### 最近の出来事

1. 対駒祭、ゲーム禁止令発令中
2. 対駒祭、非常事態宣言発令中

今年は集まるマシンはかなり豪華になる予定。が、内容がそれに比例するものになるかどうかはこれからの追い込みにかかっています（お。

このパンフは何と駒祭の3日前に制作されていますので、情報としてはかなり古いものとなっています。ここに載っていない企画が展示されてることもありえますし、逆も然りです。

プログラマの展示も御愛敬（^^;;。広い心で大いに笑ってやってください。

---

### 理論科学グループ 部報 210号

1997年11月19日 発行

発行者 植原 洋介

編集者 坂本 崇裕

発行所 理論科学グループ

〒153 東京都目黒区駒場 3-8-1

東京大学教養学部内学生会館 305

Telephone: 03-5454-4343

---

(C) Theoretical Science Group, University of Tokyo, 1997.

All rights are reserved.

Printed in Japan.

理論科学グループ部報 第 210 号  
— 駒祭パンフ号 —  
1997 年 11 月 19 日

*THEORETICAL    SCIENCE    GROUP*