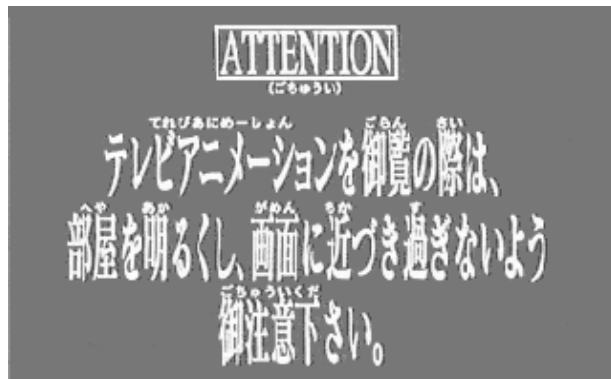


# TSG

Theoretical Science Group

理論科学グループ



部報 215 号  
— ぶほ・駒場祭準備特集号 —

目 次

駒場祭準備特集	1
駒場祭企画紹介 . . . . . 【ぶほ編集部】	1
こまさいれびゅー . . . . . 【Tellur】	12
過去を振り返る。 . . . . . 【Nishi】	15
続・駒場祭新戦力の顔ぶれ . . . . . 【ぶほ編集部】	17
一般記事	19
自己生成的プログラム作成の試み . . . . . 【TACT】	19
Perl 講座 第 1 回 . . . . . 【こんの】	25
MP3 といふモノ 第二回 . . . . . 【わいりー】	27
TSGer Mailing List (改訂再掲) . . . . . 【大岩 寛】	32
セキュリティー情報サイト構築プロジェクト(仮) . . . . . 【大岩 寛】	33
外部 HTTP サーバのファイルの情報の取得について . . . . . 【Lef】	34
(追加) 駒場祭企画最新情報 . . . . . 【ぶほ編集部】	36

## 駒場祭準備特集

### 駒場祭企画紹介

予定策定未定仮定暫定決定

ぶほ編集部

アプリ開発の負担は年々重くなっていくのに、駒場祭プロジェクトの始動時期が年々遅くなっていく気がする今日このごろ、皆様いかがお過ごしでしょうか。

駒場祭に向けて動き出したプロジェクトをほぼすべて列挙することができましたので、ここにお届けします。

各項目についている *project sheet* は、現時点での編集部スタッフの総意（ていうか私の独断）によります。駒場祭当日までにどんな大逆転劇が起こるか、ほんとに楽しみにしています。

なお「参加表明者」欄の（ ）囲み表示は、現段階で協力表明と受け取れる発言をしている人を示します。  
（福林一平 / 文殊壱平）

### ファミリーマニア（仮称）

*project sheet* KF1998

企画名	ファミリーマニア（仮称）
参加表明者	ギャラクタア，村上，岩崎， （はっぴい，むらき）
気合い指数	
修羅場指数	
客ウケ指数	
完成度期待	
財政貢献度	

なんかアレな名前ですが，別に妹属性の人とかおやぢ専な人とかが出てくるゲームではありません（ヤだな，それ）。

今，世間様で大流行しているピー マニア。

当企画は，そのートマニアを，ファミリートレーナーを利用することで，PC上でよりアクティブに再現しようという趣旨のゲームです。要するに，音楽に合わせてファミリートレーナーを手でぺしぺし叩くだけなんです。

……断じて，KONAMIの某ゲームのパクリではありません。

……いや，信じてくれという方が無茶かもしれませんが。

しかし、当企画は「アレ」とは異なるセールスポイントを保有しています！

まず、足で踏むのではなく手で叩くことで、見た目が地味に！

さらに、曲はまだ未決定！

とどめに、プログラムは完成の目処すら立っていません！

・  
・  
・

.....すいません。

ちょっとせつなさ炸裂しそうなんで、東京に無言電話かけていいですか？(謎)

.....ともかく、制作不調！(涙)

期待してお待ちください！.....いや、裏切られるかもしれません。

(企画無責任者 = 高木宏明 / ギャラ)

*gyara@mtc.biglobe.ne.jp*

(Thu, 29 Oct 1998 20:58:02 +0900)

## チャットサーバによるねとわくゲームの習作

*project sheet KF1998*

企画名	チャットサーバによるねとわくゲームの習作
参加表明者	こんの(個人企画)
気合い指数	
修羅場指数	
客ウケ指数	
完成度期待	
財政貢献度	

## 事の成り行き

もともと何かのパズルを作ろうかと思っていましたが、僕はまだ C や Java でグラフィックを扱うソースを書いたこともないので、どうしようかと困っておりました。

で、それとは関係なく、ちょうどここ 1 週間くらい、1 年半くらいは親しんでいる Perl でチャットサーバを作っていました。最初は fork って何、という状態から始めたのですが、次第に自分の書いているソースの意味を理解しながら、やっと複数人が同時進行で会話ができるサーバになりました。と言っても、今でもデッドロックしまくりだーぜー。(やや嘘

1998 年 10 月 27 日(火) 15 時 13 分 11 秒 JST 現在、このそこそこ形になりつつあるチャットサーバの他に、クライアントを何かの言語で書いて、ネットワークカードゲームを作ろうかともくろんでいます。これから Tcl/Tk に手を出すつもりはないので(笑)、クライアントは Java で書く可能性が一番高いですが、いかんせん Java はよく分からない™ ので、誰かに応援

を頼む可能性は高いかも。(おい

## 現在の鯖の開発状況

サーバは現在、以下のような実装になっています<sup>1</sup>。

サーバの環境 Perl5 なら大抵大丈夫だと思われます。今は Perl4 でも動くだろうし。ちなみに、SysV系IPCを使ってクライアント・サーバ間でログをやり取りしているため、IPCが使えないと話になりません。絶対に JPerl for Win32 とかでは動きません。あう。今気付いたけど、これって駒祭当日大丈夫なんだろうか。(汗  
クライアントの環境 サーバと telnet(1) もしくは同等のプログラムを使って接続できればあとは何でも大丈夫なのではないでしょうか。一番問題なのは、やっぱり見栄えのするクライアントを書けるかどうかですね。

上りのコード もともとは \*euc-japan\* 限定で書いていましたが、拡張して \*junet\* にも対応しました。混合でも OK だと思います。コントロールコードの取り扱いが激甘なので、なんとか処理ルーチンを書かないとですね。

内部のコード 内部では文字コードを \*euc-japan\* に変換して処理しています。変換は自前ではやらず、Perlらしく jcode.pl です(笑)。上りのコードの拡張と言っても、ようは jcode.pl 使っただけなんだが(爆)。

下りのコード 下りは内部のコードの \*euc-japan\* をそのまま垂れ流しています。改行コードは 現在 LF のみとなっていますが、近々 CR/LF に変更する予定です。

相手側強制切断の検出 全然出来ていません。いない人がいつまでもゾンビになって残っています。ゲームにする時は、自分のターンで一定時間飯能しないと「Computer 代打」君に登場してもらえばいいんでしょうけど。

プロセス数 現在  $2n + 2$  です。減らせる自信はないです。まあ、大人数でゲームやることはないと思うので、大丈夫だと思いますが.....。

子プロセスの取り扱い 全然なっていません。<defunct> なプロセスが残ってしまうことがよくあります。

チャット機能 もともとチャットサーバなので、そこら辺はまあまあ得意分野かと思われます。幸い秘話機能はまだついていません。:-)

ロバスト性 未定義値です。

## おまけ

クライアントは、サーバから受け取った文字ベースのデータを解釈して、カードの画像などを表示し、再び文字ベースのデータをサーバに送り返すといった、ごくごく簡単な処理しかしません。でも、個人的にはクライアントの方が大変そうだなあ。はたして完成するのでしょうか。

<sup>1</sup>なお、文中の「上り」とは「クライアント → サーバ」のデータの流れを指し、「下り」はその逆を指します。

## 駒場祭企画紹介

ちなみに、鯖は最初 Perl4 の入門書を見ながら書いたもので、素直に Perl4 で書いていたのですが、最近本屋に行った時「プログラミング Perl 改訂版」の本の前で気を失って以来、Perl5 も使っています。はっきりいってぐちゃぐちゃで訳の和歌欄ソースになっているので、そろそろ Perl5 に準拠して書き直そうかと思っています。

(今野俊一/この)

*toknn@ijk.com*

(Fri, 30 Oct 1998 01:22:53 +0900)

### R-Xs ( 仮 )

*project sheet KF1998*

企 画 名	R-Xs ( 仮 )
参加表明者	むらき ( 個人企画 )
気合い指数	
修羅場指数	
客ウケ指数	
完成度期待	
財政貢献度	

### 予定

決まっていること シューティング

未定のこと 2D か 2D+3D

### 進行状況

まだ本体のプログラミングにはほとんど取り掛かっていません。現在いろいろなクラスを大体作り終わったところです。これから本体に取り掛かります。

(村木亮太/むらき)

*mrkryt@fa2.so-net.ne.jp*

(Thu, 29 Oct 1998 12:28:46 +0900)

ぶほ編集部より

総決起コンパの席では Direct3D を使う予定だと宣言していましたから、なにやら期待できそうです。……え、平面ラスタライザですか?(汗)

なお、その後の話し合いでむらきくんは、ファミリーマニア企画(1ページ)の進捗状況によっては、そちらのプログラミングもサポートすることになったようです。

## ネットワーク対戦ゲーム

*project sheet KF1998*

企画名	ネットワーク対戦ゲーム
参加表明者	TACT (わいりー)
気合い指数	
修羅場指数	
客ウケ指数	
完成度期待	
財政貢献度	

2年前にプログラマ廃業宣言をしたはずなのに、なんでまた開発やってんだろ、なんていう疑問はさておき。

Microsoft VisualJ++ 6.0 の登場により、Windows 用の開発環境として生まれ変わった Java。VJ6TP1 を手にしたとき、こいつを使って何か面白いことがでけへんやろか、と思ったのがこの企画のそもそもの始まり。VJ6 の実力をいかに発揮したプログラムが書ければいいんですが、WFC を解説した書籍がいまだに無いことや時間的制約もあり、次のような方針を取ることにしました。

- JDK1.1.x に準じた Java アプリケーションとして開発する。awt を使い、WFC や AFC は使わない。
- ジョイスティックの入力のために DirectInput を使う。
- TCP/IP を使ったサーバ/クライアント型のネットワークゲームとする。サーバ/クライアント間は TCP ソケットで通信する。
- 斜め上から見た立体的な画面表示を目指す。

とまあなんとなく方向性は見えてるんですが、問題も山積みです。

- awt を使うため、特にグラフィック周りが遅く、貧弱。
- ゲームのルールも何も決まっていない(爆)
- 無印 Pentium 150MHz で画面サイズ 800 × 600 での開発は結構つらい。
- 開発時間がない。
- 眠い。
- 最近キータイプ速度が遅くなった気がする。
- 電話料金残高がまたマイナスになったようだ。<sup>2</sup>
- VJ6 試用版の期限が切れた。製品版をいれねば。
- だれかキャラデザやってくれー。

というわけでかなり萎え気味なんですけど、人前にプログラムを展示する機会なんてそうあるもんじゃないんでなんとか動くものを作ります。

今度こそ、これが僕の人生最後のプログラムになるかもね(笑) (下村哲人/TACT)

*tact@246.ne.jp*

<sup>2</sup>【編註】「日吉」ならでは。部報第 213 号「用語基礎知識」参照。

ぶほ編集部より

TACT くんは猛烈な量の製図課題が襲いかかったのは、この企画書が書かれた直後でした。

その後彼からは「書きかけコードを流用して動くものを作りたいと思っているが、当初構想のものを仕上げるのはとても無理」との連絡がありました。おそるべし建築学科。

## BM98 プレイヤーの実装

*project sheet KF1998*

企画名	MyApp01
参加表明者	文殊吉平（個人企画）
気合い指数	
修羅場指数	
客ウケ指数	
完成度期待	
財政貢献度	

## BM98 について

BM98 は、フリーの音楽ゲームソフトです。ただし、実質的にはコナミのヒット作「ビートマニア」シリーズのクローンです。開発開始当初は、ビートマニアトレイナーとして数多く制作された「類似ゲーム」のうちの一つだったのですが、ソフト作者の強い開発力と、ステージデータ作者たちの強力なサポートの中で、本物と見紛うほどの「クローン」に成長してしまいました。

このため現在では、BM98 本体の公開は完全に停止され、作者は、本体と BM98 曲データとの独立性を主張しています。こうすることで、データの蓄積を保護するとともに、現存ソフトウェアの存在を正当化している模様です。すなわち、一般的な音楽の制作・演奏のためのデータフォーマット

## 「Be-Music Data Format '98」

に準じたデータがいわゆる「BM98 曲データ」であり、これに準拠したプレイバックが、BM98 本体を含め、何種類かリリースされているのだという主張です。

この主張に従って、私も新しい BM98 プレイヤーを制作することにしました。1 年生のメイン企画と内容的に非常に重なり合いますが、11 月 3 日現在、互いに干渉せず、並立を予定しています。



## マイクロソフト ヴィジュアル スタジオ 6.0

このプロジェクトは、Microsoft<sup>®</sup> Visual Studio<sup>™</sup>、Microsoft<sup>®</sup> Visual C++<sup>®</sup> 6.0 上で開発し、MFC (Microsoft<sup>®</sup> Foundation Class) 6.0 および DirectX<sup>®</sup> 6.0 を利用します。DirectX<sup>®</sup> SDK は、もしかしたら利用しないかもしれませんが。相手は COM オブジェクトですから、仕様がわかってさえいれば、DLLがあればそれで十分なのですね。あ、でも、DirectInput とかの定数の嵐は SDK から持ってこないと辛いかな。

MFC AppWizard といえば、App くらす、Frame クラスに、びっちり View クラスが張りつき、こいつに Doc クラスがもれなくついてくるといって、非常に厚ぼったいスケルトンを強要されるか、

### さもなければダイアログベース

にするしかない仕組みのため、DirectX<sup>®</sup> を使うようなゲームには全く使えないと思われていました。かといって、MFC アプリケーションをゼロから自作するなんて、1 度たりともやりたくありません。Win32 SDK で書いたほうがよほどましです。それに、従来の DevStudio の内蔵エディタの出来がいまひとつで、統合環境を敬遠したくなる一因となっていました。

しかし、Visual C++<sup>®</sup> 6.0 の MFC AppWizard は、Doc-View アーキテクチャでないアプリケーションのスケルトンを吐き出す機能をもっています。チェックボックスのチェックを 1 つはずして「次へ」「次へ」……で、Frame、View、App (および AboutDlg) だけのプロジェクトができます。このうち View をポイして、かわりに DirectDraw をはめこんでやればいいわけですね。

たったこれだけのことですが、このおかげで、統合環境の機能をほぼ完全に利用できて、しかも MFC のお作法に縛られない開発ができるようになるわけです。

テキストエディタの方は相変わらず見栄えがしないのですが、シンボル名補完系のサポートが大変強力になったので、たっぷりお世話になるつもりです。

って簡単に言うけど、やってみると変なところにつかえるんだよね、きっと。特にリソースエディタとかが罨っぽいので、ダイアログ以外は作らないようにしよう、みたいな。

企画名称が未定なので、とりあえずデフォルトの名称 (笑) を書いておきました。マイクロソフト系開発環境に身をゆだねる覚悟が表れていますね。あ、べつにおおくぼくんと対決する気はありません (9 ページ)。

### 現在状況

Windows でアプリケーションを書くのは初めてなので、今のところデバッガで遊んでいます (笑)。11月2日現在、モードレスダイアログのエディットコントロールに、アプリ側から文字列を流し込む仕掛けが完成しました (お……デバッグ用ですね。ほんとは素直に CObject::Dump() を使えばいいんですけど。

(福林一平 / 文殊壱平)

[ippeii@t3.rim.or.jp](mailto:ippeii@t3.rim.or.jp)

## TRPG 用無脳

project sheet KF1998

企画名	TRPG 用無脳
参加表明者	ItaO (個人企画)
気合い指数	
修羅場指数	
客ウケ指数	
完成度期待	
財政貢献度	

### なんで ItaO はこんな企画に首を突っ込むことになったか？

某月某日 305 にて「italk 上で TRPG やりたいなー」のようなことを口走ってみる。

某月某日 駒祭の企画決めの部会にて本人不在の内に TRPG 無脳が駒祭企画になる。このころは裏企画だと思ってたんであんまし慌ててなかった。

某月某日 合宿に行き、一部有志で TRPG をやろうと試みるも忙しすぎてできず。<sup>3</sup>裏コンパにて某氏に TRPG 無脳のことを熱弁。

某月某日 ぶほにて TRPG 無脳がなんか他の企画と同等の扱いを受けていることに気付く。

### TRPG 用無脳ってなに？

.....という経緯で開発されることになったのが TRPG 無脳です。

具体的な構想は以下のような感じです。

- 対応 TRPG は「ソードワールド」(旧版)
- ダイスやレーティング表による判定を行う「判定君 sw」と、キャラクターごとの HP、MP などの管理を行う「CS 君 sw」の 2 本立ての予定。

以上のような感じです。

判定君はかの名作無脳「サイコロ君」<sup>5</sup> をチューンしたものを、CS 君は italk の /s や /w の機能を限定&拡張したようなものを想定してます。

.....ほんっとーに俺にできるんか.....。

### 目標(だけ)は壮大に

なんか弱気なことをいろいろ書いてきましたが、将来的にはソードワールド以外の TRPG にも対応した無脳も作りたいですねえ。あとは HTML でキャラクターシートを半自動生成してくれるようなツールとか。

<sup>3</sup>その節はすいません > 関係諸氏

<sup>4</sup>国産のファンタジー RPG。日本で一番普及している TRPG だと思われる

<sup>5</sup>2D6 とか 3D12 とか発言するとサイコロを振ってくれる無脳。ぐらふいかるモードは感涙もの。

ところで、TRPG 関係のホームページを散策していると、意外にチャット上で TRPG やりたい、という希望も多いみたいです。そこで将来的にはちゃんと一般にも使える TRPG 用無脳をつくって「チャットで TRPG といえば italk」とか言われるようになったら嬉しいかも（高望み杉。  
（坂尾要祐 / ItaO）

[g740581@komaba.ecc.u-tokyo.ac.jp](mailto:g740581@komaba.ecc.u-tokyo.ac.jp)

(Tue, 27 Oct 1998 13:10:58 +0900)

## 音声入力の応用

*project sheet KF1998*

企画名	noname00 (仮)
参加表明者	おおくぼ (個人企画)
気合い指数	
修羅場指数	
客ウケ指数	
完成度期待	
財政貢献度	

ぶほ編集部より

音声入力パッケージ ViaVoice を利用したデモないしゲーム、ないしエミュレータ (?) を作成するそうです。

「はどお~拳っ」っと叫ぶと技が出る、とかそういう熱いやつ希望。

企画タイトルを聞きそびれました。上記仮称は、Borland 使いのおおくぼくんを尊重したつもりです(笑)

## ネットワークカードゲーム

*project sheet KF1998*

企画名	ネットワークカードゲーム
参加表明者	ばんだい (個人企画)
気合い指数	
修羅場指数	
客ウケ指数	
完成度期待	
財政貢献度	

ぶほ編集部より

ネットワークカードゲーム制作中。サーバは無脳で作っているが、クライアントができていない。最悪の場合、クライアントは telnet(1) に。(お なお、ギャルゲーツクールは頓挫しました。  
(取材=こんの)

## 対戦型占い

project sheet KF1998

企画名	対戦型占い
参加表明者	はっぴい(間中)
気合い指数	
修羅場指数	
客ウケ指数	?????
完成度期待	
財政貢献度	

ぶほ編集部より

名前から先に決まったので、現段階でどういうものが構想されているのかさっぱりわかりません(笑)

はっぴいくん個人は、総決起コンパの席で、ファミリーマニア(1ページ)への協力を表明していました。

## 中ゲタ '98

project sheet KF1998

企画名	中ゲタ '98
参加表明者	-1ParaGRAPH, (わいりー)
気合い指数	
修羅場指数	
客ウケ指数	?????
完成度期待	
財政貢献度	

ぶほ編集部より

Visual Scheme をつくるかもしれないし、駒場シミュレーターを作るかもしれない、と。

何をか言わんやって感じですね。ちなみに初代「急造戦艦中ゲタリオン」というネーミングを最初に叫んだのは私だ(と主張してみるが自慢にならん) (壱)

## その他の声

コンパ会場などで聞かれた声の数々。こんのくんとItaO さんに取材を協力していただきました。

わいりー ばらか TACT に協力したい。でも、2人とも何をするのがよく分からない。

こんの ねとわくゲームって実は、ばんだいさんともろにかぶってしまう。

ひでき 各企画のお手伝いということで。

齊藤 個人企画必須って「2年生」だけの適用だよな？

壱平 この時点で自分の意思表示をしていない人は全員雑用係だよな。

.....言ってることがいろいろだ。合唱。

## チャンネルそのまま，このあとすぐ！

次のページから，過去の駒場祭のレビュー記事が続いています。ネタ切れ頭に注入しましょう。こんな場でなんですが，ご協力いただいた先輩方に感謝します。

## こまさいれびゅー

Tellur

### ご注意

以下の文書では、化石、死語の世界が屡々現れるかもしれませんが、ふるーい話なものでその辺は我慢してください。

### はじめり

ええと、福林先輩の2年後輩にあたる Tellur です。ども :> <sup>1</sup>

さて、むかーしの駒祭企画のレビューをしてくれーってたのまれたんですが、ほんとは先輩のほうが詳しいはずなんですけどねえ。編集長なんだし、過去の部報のありかもご存知のはず……。

まあ、こっちも過去の TSG 部報をひっくり返して、これは特筆ものとかいう企画を拾い出してみますか。

ごそごそ

どんがらがっしゃーん(お

ええっと、うげげ、軟派なヲで通しているおいらが、こんな硬派な記事を書いてたとはねえ、しかも OB 化してる時に。さっさと隠匿しましょ(謎

### 格言

「いま後悔しているのは、駒祭の準備を始める前に、去年の部報をみておかなかったこと。」

— '89 ぶてふ, Vol.143

……なにごと温故知新ですな。

### ライトサイクル

CG 黎明期の怪作映画「トロン」にでてくる、ライトサイクルというゲームは、バイクの軌跡が壁になってライバルの進路を妨害でき、最後まで走りつづけられたバイクの勝ちという、ゲーム性の高さのために、駒祭企画でも何度か出没したような気がします。

<sup>1</sup>【編註】(T-T)

3D 表示のものも有ったような気がしますが、今度作るのを企んだ方には、ぜひ、

## 「バイクをレンダして 60fps」

を実現して頂きたいっ！

### 電光掲示板

1990 年頃、笹原先生らのグループによって制作され、永らく駒祭やオリで使われていたものです。

16 × 16 の赤色 LED アレーを並べて TTL でドライブし、文字の表示は PC で制御するというものだったと思います。

後に Applause 君等がスタンドアロンでも多彩な表示が出来るように改造したような。こういう息の長い展示物も味わい深いものがありますな。

### 亀虎

むかーし有った「キャメルトライ」という、ボールをころがしてゴールに辿り着けるようにするというゲームを、98 で実現してしまったという、気合の一作です。

と、こう書くと、さらに古い人は「マーブルマッドネスと同じでないかい」とか思われるかもしれませんが、こいつの迷宮は 2D であるかわりに「ぐりんぐりん回転する」んです。

tea 君の恐ろしさがよくあらわれてますです。<sup>2</sup>

### Sky Duel

ええっと、これはすごかったです。

3D の Voxel 分割で光線の当たり判定をして、地形を 3D 表示する、ヒコーキのタイムトライアルを FM TOWNS で実現したという、画期的な企画でした。

見た目よし、ゲーム性もよし、の満点あげてもいいような企画でしたな。(でもくみこむ賞はもらえなーい)

のちに、改良版が雑誌に載ったりしたので、知ってる人も多いんじゃないんでしょうか。

Makken 先生入魂の一作です。<sup>3</sup>

### 対戦型迷路

ぼくらの仲間、油先生のこさえた、「一人が壁をくんで、もう一人が脱出する」迷路ゲームです。

<sup>2</sup>【編註】キャメルトライ、マーブルマッドネス、いずれも X68k 版が高校の部室にあったなあ……。キャメルトライのステージエディットが熱い！

<sup>3</sup>【編註】部報第 212 号 7~8 ページ参照。

壁を作る方は、ゲームをやっているうちにずんずんいぢわるになって、けっこうダークなもんでした。

## まとめて占い

占いは、昔は素晴らしい企画だったのですが、最近はどこもかしこも「こんびゅーたうらない」をやっているの、場所とりのほうが重要になって、うまみはすくないですな。

そんななかでも、カバラは占いの選択として画期的だったかも。「常駐カバラにして、普段はゲームやプログラミング、お客さんが来たらちょちょいと占い」というくだりは、古き良きDOSの時代の話なのでおいといて。

あとは、いつぞやの「真面目に計算する星占い」で、お客さんから「結果がちと違う」とクレームが来て、良く調べるとデータに誤差があったとか。けっこうマニアックな客がいたりするのが占いの醍醐味(お

## おしまいに

駒場祭企画でいつも思うのは、

## 見た目が肝心

という点です。

どうも TSGer はそういうところに気の回らない事が多くて、ぐちぐち.....なんてぐちってもしようがないのですが、兎に角お客様を引き付けるような外観にするというのは、とても大事です。

そういう観点で見ると、いままでの占いは

## 「お客様に綺麗な画面が見えない」

というあたりがいけないのであります。<sup>4</sup> 皆さんも見た目を大事にしましょう。

では、派手な企画をたてればいいのかというと、それだけでもだめなので、せっかく祭りの前に書くチャンスが来たことだし最後に強調しときます。

## 「部屋の飾り付けはー :D~」

おしまい

( 経澤重徳 / Tellur )

---

<sup>4</sup>【編註】97年のはすごくきれいだったんですけどね、斜め向かいにナンバ体育会系の同種企画がいたもので負けましたな。



## 過去を振り返る。

Nishi

私は 94 年入学で比較的若い部類に入るので、そんなに昔のことは書けません:) 聞いた話によると、私の入学以前には、もっと愉快なことが行われていたようです。<sup>1</sup>  
とりあえず私の覚えている範囲で。

### 94 年

Zephyr.....98 4 台による通信対戦 3D 戦車ゲーム。当然 10BASE-T など繋がってたりはしなくて、シリアルポートを数珠つなぎにしてリング作ってバケツリレー式に通信していたような。1 台でも遅いマシンがあると、そこで速度が決まってしまうため、速いマシン (Ap や Ap2 とか) で固めた。最初は As もあったが、それでは遅いので、はらだ 486GR に置き換えた記憶が。コントローラはスーフファミパッドで、この部分は Applause さんがやったんだっけか。

Makken.....Sky Duel (飛行機で飛んでいく(?) 3D ゲーム。TOWNS ㊦)

Nishi.....占い (98 版開発をやるが、事前調整をしっかりとやらなかったため失敗(^^; この年、占いは TOWNS (ちょらんま) のみ。NAO ルーチンは最終日にやっとそれらしい結果を返した。)

### 95 年

上映会の年。エヴァンゲリオンのその時点での放送分までを見た。(セーラームーンを見るために離脱した人がいたような記憶があるので、土曜の夜ではなかろうか?) みんながエヴァを見てる際に、寿司を食いまくったような記憶もある。

Makken さんの Marionette の年でもある。はらださんや私が画面の中を走りまわっていた:)

GANNA.....X 版平安京エイリアン。記録に残っているハイスコアは部報 195 号によれば、げる君の 7600 点。

Nishi.....ゲーム開発 (それも Win32) の実演 (お 当時は DirectX などというものはなかった。WinG はあったけどね(^^; 駒場祭前日が勤労感謝の日で、この日は Windows 95 の発売日だったような。結局 Win95 の展示となる。いきなり Hover を立ち上げ、マシンを落としていくガキがいた。む、1ha (無圧縮 lh5 を吐くアーカイバ) の年でもあるのか(^^; まあ、作っ

<sup>1</sup>【編註】12 ページ参照。

<sup>2</sup>【編註】部報第 212 号 7~8 ページ参照。

過去を振り返る。

---

ただけだが。

## 96年

八重樫あうあう(ネットワーク対戦3Dシューティング。液晶シャッター付きゴーグルが1台だけあって「あうあう」が体験できた)。ちなみにLAN敷設はこの年から。

たけしま.....tcommand。その他にもRPG、フライトシミュレータ等。

Nishi.....NXSvr 初公開。X版平安京エイリアンの「表示」(おやたら不安定ですぐに鯖が落ちた(^^; 現場開発をやろうとしたが、NTなGANAマシンを八重樫さんに奪われてしまったため、急遽(確か)うえマシンにVC++を入れる。気がつくとVC++が消されていて、代わりにMPEG-AUDIOなデータが入っていた。

## 97年

去年。オセロの他に何やったっけ?(^^; <sup>3</sup>

#以上、記憶違いな箇所もあると思う。

(西澤信行 / Nishi)

ふぼ編集部より(畏は世代を超えて.....)

追加情報。

95年駒祭企画「超超爆速lha」

このストイックな企画は94入学のNishiくんのもの。コマンドプロンプトのみが展示用ディスプレイに表示されるという潔さに感動し、私は展示のための看板まで作ってその日を待っていた.....がNishiくん

「やっぱり展示の仕様がないう.....(^^;」

良識的な彼の判断のもと、日の目を見る事がなかった。残念。(文責 sigma)

編集担当は、Nishiさんからの投稿よりも先にsigmaさんのメッセージを読んだので、完璧に「騙され」ました。うう:)

そういえば高校時代、ばらぐらふが、lhaでは絶対に圧縮できないファイルを生成するプログラムを書いて大ウケを取っていたような(笑)。

---

<sup>3</sup>【編註】ばらぐらふ+わいりーのPhotoShop系占いがありました。:)

続・駒場祭新戦力の顔ぶれ  
1998 年度新規 TSGer コメント集 ( 2 )

ぶほ編集部

続・奥ゆかしい人たち！

前回 ( 第 214 号 ) での積み残しを掲載します。各項目は前回と同様，

- 1 「出身地域」
- 2 「TSG を知ったきっかけ」
- 3 「駒場祭への意気込み」
- 4 「その他，TSG に入るようになったきっかけや，TSG で今後こんなことをしてみたいというようなことを自由に ( 長さは自由。0 行でも 100 行でも ) 」

として募集しました。

番組スタート

2 週間のご無沙汰でした，ぶほ編集部の壱平です。今回も，「TSGer になったよ，よろしくね」っていう人からたくさんメールが寄せられています。今日も，こいつを，到着順に紹介していきましょう。

Personal Data

木村香里 ( きむら・かおり )  
かおり EX

理科一類 ( 仏 )  
g840999@komaba...

- 1 ▷ 東京の最果て、東村山市。1 歳までは石川県にも住んでいました。
- 2 ▷ サークルオリで、高校の頃

駿台で数学を

教えていただいた油すまし氏と再会。そのままずるずると入部してしまいました。

- 3 ▷ やらなきゃいけないのに何もやってませんな。どうしてくれよう。うりうり。
- 4 ▷ たればんだ撲滅委員会会長としてがむばってゆきたいと思います。

説明不要。今ごろは駒場祭に向けて活躍中ですね。たぶん。うりうり。

石井浩一(いしい・こういち)

理科一類(中)  
kkii@mail.interq.or.jp

- 1 ▷ 千葉県柏市駅前の岡田病院。
- 2 ▷ 友だちの誘い。しかし、その友だちは現在幽霊部員と化してしまった...
- 3 ▷ 運動会の部活に入っているので、そっちのほうが厳しく、手伝うこともあまりできないかも...
- 4 ▷ TSGに入った当初はプログラミングを極めようと思っていたのですが、思うように時間がとれず、ほとんどあきらめてます。というわけで、これからはゲームの方を極めようかと思っております。どう極めるかは未定。現在は beat mania にはまっています。Normal mode での最高点は 40 万点台です。41 万点台を出された方、いらっしやいます？

ちなみに部長も運動会部員ですよ。:)

今 305 に来ると、beat mania について情報を欲している人がけっこういるはずなので、彼らにレクチャーしてあげてください。

### 続々・奥ゆかしい人たち!(予告)

でもって、まだ何人が未紹介な気がします。名前がわかんないんで名指しできないんですが、もしかすると清木くんとかかもしれません。次回の冊子に掲載しますので、ぜひ私宛メールでお寄せ下さい。フォーマットは今までに準じます。

(福林一平 / 文殊壱平)

ippeii@t3.rim.or.jp

## 一般記事

## 自己生成的プログラム作成の試み

Java 廃なクラスたち 第 2 回

TACT

## 原理

まず Java プログラムの動作原理から説明します。Java ソースファイルをコンパイルすると、バイトコードと呼ばれる仮想のマシン語に変換されます。バイトコードは Java インタプリタによって実行時に解釈され、プログラムが動作します。

このときバイトコードは Java のクラスごとに独立に .class ファイルに収められ、静的にリンクされることはないため、どのファイルのどのコードが実行されるのかは実際に実行するまで決まりません。DLL の振る舞いに似ているといえれば解りやすいでしょうか。(Java の動的コードリンクは Lisp 譲りだとも言われますが、僕は Lisp を良く知らないので本当かどうか分かりません。)

また、Sun が提供する JDK に含まれる Java コンパイラは Java で書かれているため、Java プログラムから呼び出して実行時に任意のファイルをコンパイルさせることができます。

これらの知見から、直ちに次の結論を導き出すことができます。すなわち「実行時に Java ソースファイルを吐き出し、コンパイルし、そして実行する、という自己生成的な Java プログラムの作成が可能である」。

## 当たり前にして当たり前にあらず

偉そうなことを書きましたが、実は当たり前のことなんです。でもここで僕は声を大にして言いたい。

「当たり前のことが当たり前に行えるのがすごいことなのだ」

と。

このような自己生成的プログラムが当たり前のように作れるのは、Java が非常にうまく設計されているからだと言えます。そこでこの記事では、実際に自己生成的なプログラムを作成しながら「当たり前のこと」が Java ではどのように実装されているのかを見ていきたいと思えます。

なおここでは JDK の非公開クラスを利用するため、Sun の JDK が必要です。(最近 JDK1.1.6 が雑誌の付録として収録されていました。最新版は <http://java.sun.com/> からダウンロードできます。) Visual J++ 6.0 から試みましたが、コンパイルはできるものの動作しませんでした。Microsoft の JavaVM と Sun のライブラリはやはり相性が悪いようです。

### Java High Classes

#### java.lang.StringBuffer クラス

String クラスと StringBuffer クラスの違いは重要です。String クラスは文字列定数をクラスとして扱うためのラッパー<sup>1</sup> ですから、中身を変更することはできません<sup>2</sup>。それに対し、StringBuffer クラスは変更可能な文字列を表します。長さに制限はありません。

ここでは、出力する Java ソースファイルの中身を作成するために StringBuffer クラスを使います。append() メソッドを呼び出して文字列を連結していただくだけです。StringBuffer から文字列を取り出すには toString() メソッドを使います。

ちなみに、+ 演算子によって文字列を連結するコードを書くと、StringBuffer クラスのオブジェクトを作って append() し toString() するようなコードにコンパイラによって置き換えられますので、知らず知らずのうちに頻繁にお世話になっているはずで

#### java.io.FileWriter クラス

Reader/Writer という概念は JDK1.1 になって導入されたものです。以前は InputStream /OutputStream のコンストラクタに直接ファイル名を渡していましたが、Reader/Writer の導入に伴い deprecated (非推奨) になりました。文字コードの変換がどうのこうのとかいう説明をどっかで読んだ気がしますが詳しいことは忘れま

した。ともかく素直に Reader/Writer を使いましょ

- public FileWriter( String file ) (コンストラクタ)  
ファイルを開

---

<sup>1</sup>String クラスの内部表現は 16 ビット char の配列になっています。

<sup>2</sup>裏技的に String クラスの中身を変更する方法はありますが、おすすめしません。

- `public void write( String str )`  
文字列を出力します。
- `public void close()`  
クローズします。

## sun.util.javac.Main クラス (非公開クラス)

このクラスこそが JDK の Java コンパイラ (javac) の正体です。  
主なメソッドは次の通りです。

- `public Main( PrintStream out, String name )` (コンストラクタ)  
out にはエラーメッセージを出力する PrintStream を指定します。System.out か System.err を指定すれば良いでしょう。name はコンパイラの名前です。何を与えても構いません。
- `public boolean compile( String[] args )`  
args は javac のコマンドライン引数そのものです。オプションも指定できます。成功すると true を、失敗すると false を返します。失敗しても例外を投げないのはちょっと Java らしくないですね。
- `public static void main( String[] args )`  
ご存知 main() メソッド。このメソッドを直接呼び出してコンパイルすることもできるのですが、困ったことにコンパイル後に内部で System.exit() を呼んでいるため、JavaVM ごと終了してしまいます。System.exit() でも終了しない方法はあるはずですが現在調査中です。

このクラスのメンバを知りたければ、

```
javap sun.util.javac.Main
```

を実行してみてください。 `javap -c classname` で逆アセンブルもできます。Java のバイトコードを読んでみるのも一興でしょう。

ソースコードが欲しければ逆コンパイルしましょう。僕は逆コンパイラとして JDK1.1 対応の「Jasmine」を使っています。 <http://www.srctec.com/jasmine.htm> で入手可能です。一般的な Java クラスはほぼ完璧に逆コンパイルできるので、他人が書いた Java プログラムを読んで勉強することができます。なお JDK のクラスは lib/classes.zip に含まれているので、逆コンパイルするためには unzip する必要があります。

## java.lang.Class クラス

Class クラスは、Java のクラスの実行時表現 ( RTTI = Run Time Type Identification ) を表すクラスです。非常に抽象的な概念なので分かりづらいのですが、今回の目的には次の2つのメソッドを知っていれば十分でしょう。

- `public static native Class.forName( String name )`  
    throws `ClassNotFoundException`  
    name で表された名前を持つクラスの RTTI を取得します。
- `public native Object newInstance()`  
    throws `InstantiationException, IllegalAccessException`  
    RTTI が表すクラスのインスタンスを取得します。Object 型で返されるので、  
    適当な型に明示的にキャストする必要があります。

なお、`Class#newInstance()` メソッドでは引数なしのコンストラクタが呼び出されるので注意してください。コンストラクタを作成していない場合はスーパークラスのコンストラクタが呼ばれます。JDK1.1 から導入されたクラスリフレクション ( `java.lang.reflect` パッケージ ) を使うと任意のコンストラクタを呼び出すこともできますが、結構めんどうなので素直に引数なしのコンストラクタを作った方がよいでしょう。

もう一つ注意して欲しいのが、`Class#newInstance()` の戻り値を適当な型にキャストする必要がある点です。Java では、継承ツリーをさかのぼる方向のキャスト ( 子から親へ向かうキャスト ) は自動的に行われますので明示的なキャストは必要ありませんが、ダウンキャスト ( 親から子へ向かうキャスト ) は明示的に行う必要があります。Java のオブジェクトは自分自身がどのクラスのインスタンスであるかを知っていますので、本来親子関係にないクラスにキャストしようとすると `ClassCastException` が発生しますから注意してください。ただし `ClassCastException` は `RuntimeException`<sup>3</sup> に属しますので明示的に例外処理を行う必要はありません。

## インタフェースの利用

さて、ここまでの知識があれば「ソースファイルを吐き出し、コンパイルし、インスタンスを生成する」ところまで実装できますね。最後に、生成したインスタンスにアクセスする方法を示します。いよいよ完成ですね。

未知のクラスのインスタンスにアクセスする方法は、僕の知る限りでは3つあります。

最も簡単なのは継承を使う方法です。これは新しく作るクラスを既知のクラスの派生クラスとするだけですから、これ以上の説明は要らないでしょう。ただし Java は多重継承をサポートしないのがポリシーですので、他に継承したいクラスがある場合は使えません。

---

<sup>3</sup>たとえば `NullPointerException` のように、非常に多くの場所で発生しうるため明示的な例外処理を強制されない例外。



最も Java らしい方法が、インタフェースを使う方法です。インタフェースは、定数の定義と abstract メソッド (C++ の純粋仮想関数に相当) の宣言だけが含まれる abstract クラスとほぼ同じ働きをしますが、extend ではなく implement する点が違います。インタフェースを継承することはできませんし、当然インスタンス化もできません。インタフェースはいくつでも implement できますので、継承を使うよりスマートな方法だと言えます<sup>4</sup>。

最後に、先にも述べたクラスリフレクションを使う方法があります。実際に試してはいませんが、リファレンスを見る限りではなんだかめんどくさそうなのでパスです。

## サンプルプログラム

以下に、これまでの方針に基づいたサンプルプログラムを示します。このプログラムでは Child0.java ~ Child4.java の 5 つのソースを吐き出してコンパイルし、IChildClass インタフェースを通してそれぞれの reply() メソッドを呼び出します。

JDK1.1 以上でコンパイルし、Parent を実行してください。動作確認は Windows95OSR2 + JDK1.1.6 で行っています。

### IChildClass.java

```
public interface IChildClass
{
    public void reply();
}
```

### Parent.java

```
import java.io.*;

public class Parent
{
    public static void main( String[] args ) {
        Parent sg = new Parent();
        sg.makeSrc();
        sg.compile();
        sg.test();
    }

    private int files = 5;
    public void makeSrc() {
        for ( int i = 0; i < files; i++ ) {
            System.out.println( "Making \"Child"+i+".java\" " );
        }
    }
}
```

<sup>4</sup>【編註】多重継承の弊害を避け、インターフェイスのみを受け継いでいくという考え方は、C++ 設計時にはほとんど考慮されていなかったものです。が、実は C++ 言語の範囲内で実現可能です。具体的には「クラスのネスティング」を利用するわけですが、ソースはかなり見づらくなります。こいつを活用して開発しようと思ったら、マクロ定義を大量にするしかありません。その実例がまさに COM です。inside COM する勇気があれば、この事実を確認できます。

```
StringBuffer src = new StringBuffer();
src.append( "public class Child"+i+" implements IChildClass\n" );
src.append( "{\n" );
src.append( "    public void reply() {\n" );
src.append( "        System.out.println( \"This is Child\"+i+\".\" );\n" );
src.append( "    }\n" );
src.append( "}\n" );

try {
    FileWriter writer = new FileWriter( "Child"+i+".java" );
    writer.write( src.toString() );
    writer.close();
} catch ( IOException e ) { System.out.println( e ); }
}

public void compile() {
    sun.tools.javac.Main javac =
        new sun.tools.javac.Main( System.out, "javac" );
    for ( int i = 0; i < files; i++ ) {
        String[] arg = { "Child"+i+".java", };
        System.out.println( "Compiling \"" + arg[0] + "\"" );
        javac.compile( arg );
    }
}

public void test() {
    IChildClass[] children = new IChildClass[5];
    for ( int i = 0; i < files; i++ ) {
        try {
            Class childClass = Class.forName( "Child"+i );
            children[i] = (IChildClass)childClass.newInstance();
            System.out.print( "Calling Child"+i+"..." );
            children[i].reply();
        } catch ( Exception e ) { System.out.println( e ); }
    }
}
}
```

### 終わりに

今回は、Java における動的コードリンクや動的インスタンス化を示すために自己生成的なプログラムを取り上げました。実際 Java でこのようなプログラムを書く必要に迫られることはまずないでしょうが、この記事で述べたような知識は Java を理解する上で助けになることと思います。もしかしたら、あっと驚くような利用法があるかもしれません。Java らしい面白いプログラムの登場を期待しています。

( 下村哲人 / TACT )

*tact@246.ne.jp*

## Perl 講座 第 1 回

こんの

## Perl とは

Perl とは、Larry Wall 氏によって作られた、The Practical Extraction Report Language ( 実用的情報抽出レポート言語 ) のことです<sup>1</sup>。最近ではホームページにカウンタや掲示板を作成する際に使用する言語として非常に向いているという特長から、CGI ( *Common Gate Interface* ) で多用されています。僕も CGI で Perl を知った 1 人です。本屋でも CGI に内容が特化された Perl の本をよく見かけます。しかし、中には CGI に内容が片寄っていない本もあるはずですが。Perl は CGI にしか能がないわけではありません。テキスト処理、プロセス管理、ネットワークプログラミング、データベースの操作、オブジェクト指向プログラミングなど、Perl にできることは意外と多岐に渡ります<sup>2</sup>。これから数回、特に Perl の初心者を対象とした Perl 講座を開きます。簡単なログの処理といったことはスクラッチ<sup>3</sup> から 1 人で作れるようになってほしいな。Makken さんのお弁当のように。( 謎

## 簡単なスクリプト

ECC には最初から Perl がインストールされています。せっかくだから、まずは簡単な Perl のスクリプト ( プログラム ) を書いてみましょう。次のように mule に入力してください。

```
print "Howdy, world!\n";
```

これを howdy.pl という名前でも保存して下さい。Perl スクリプトは、このように .pl と名前をつけるようになっています。別に付けなくてもあまり問題はありませんが。 :)

さて、次にこれを実行してみましょう。

```
% perl howdy.pl
```

このようにコマンドラインで入力すると、画面に

```
Howdy, world!
```

<sup>1</sup>マニュアルには Pathologically Eclectic Rubbish Lister ( 異常に折衷主義的ながらくた作成プログラム ) とも書いてあったりします。 :)

<sup>2</sup>ちなみに、筆者はオブジェクト指向を全然分かっていないので、そこんところはよろしく。( 激お

<sup>3</sup>mule を立ち上げた時に表示されているパツファのこと。パツファ名が \*scratch\* と出ているはず。

と出たはずですよ。簡単でしょ？

このように、Perl は「簡単なことは簡単に」できるようになっています。もちろん「難しいことも可能」であることも重要です。さらに、Perl は C や Java といった言語と違い、プログラムを書いてコンパイルし、それから実行するという手順を踏まず、いきなりそのまま実行できます。これも Perl がお手軽でとっつきやすいという特長を作っています<sup>4</sup>。

なお、ここでは perl を直接人間が呼び出していますが、スクリプトから自動的に呼び出すこともできます。さっきの howdy.pl を以下の通りに変更してください。

```
#!/usr/local/bin/perl
print "Howdy, world!\n";
```

これをまず、chmod で実行可能にしてください。そうすればスクリプト名だけで実行できるようになります。

```
% chmod +x howdy.pl
% howdy.pl
Howdy, world!
```

これからはこの #!/usr/local/bin/perl の部分を抜かしてスクリプトを出すこともありますが、実際に試す時には各自で補完して下さい。

### howdy.pl の解説

それでは、このスクリプトの内容を順に見てみましょう。……って言っても実質 1 行だけだけど。(笑)

まず、さっき付け足した #!/usr/local/bin/perl は、これは perl によって実行されるスクリプトであることを指定しています。

2 行目からが実際の内容です。Perl のスクリプトは C や Java のように、行の終わりが ; となります。付け忘れないように注意してください。

print はそのまんまです。ここではどこに内容を表示するのかが指定されていないので、標準出力<sup>5</sup>に表示されます。その後のダブルクォーテーションでくくられた部分が表示される内容ですが、実際に表示されていない文字列があるはずですよ。\\n がその文字列ですね。これは「改行」を表わす文字<sup>6</sup>で、その位置で改行することを表わします。これは目には見えませんが、普段メールを書いている時でもリターンキーを押せばこの文字が入力されています。

そのことを実感するために、ちょっと \\n を抜かした howdy.pl を実行してみましょう。分かりやすくするため、ここではプロンプトを ECC 風にします。

<sup>4</sup>同時に実行速度が遅いと言う欠点をも作っているのですが……。

<sup>5</sup>その名の通り、プログラムの標準の出力先。普通はディスプレイ。あまり気にしなくていいです。

<sup>6</sup>\\n で 1 個の文字を表わします。

```
g740208@ecc-as50> howdy.pl  
Howdy, world!g740208@ecc-as50>
```

次の行に来るはずの、次のプロンプトが実行結果の行と同じ行に表示されていますね。これは、実行結果に改行がなかったため、ビックリマークの直後にそのままプロンプトが出てしまっているのです。\\n を入れた時の結果と比べると、「改行」を表わす文字の存在が少し実感できると思います。

## 次回予告

今回はこんなところにおきます。ていうか™，もう寝させてくれ～。(お  
次回はさっそく ECC の WWW サーバのログを使ってアクセスランキングを作るまでの一連の流れを追いたいと思います。1 回じゃ終わらないと思うけど、あれだけでも結構多くのことがつまっているのです、楽しみにしてください。  
(今野俊一 / こんの)  
toknn@ijk.com

## MP3 といふモノ 第二回 ～混沌の淵に～

わいりー

さて第二回です。

はじめに、前回「ふざけんな> JASRAC」と書きましたが、この辺(個人的な MP3 データ利用)に関しては微妙な部分が多くあるようです。問題はいくつかあります。

### 1. 録音補償金の問題

まず、現在 MD のメディア価格には、録音補償金が含まれています。この録音補償金については、政令で定められています。したがって、レンタルしてきた CD を MO にダビングしても、音楽業界にお金が入るようになっているのです。著作権保護という観点からは有効な手段だとは思いますが、この制度に批判があるのもまた事実です。例えば、自分で音楽を作って MD に録音しようとしても、録音補償金を払わされることになってしまいます。

MD の場合、その用途がほとんど音楽録音に限られるため、録音補償金の問題はあまり起きていないのですが、MP3 となると話は複雑です。前回書いたように、MP3 と、MD で用いられ

る圧縮方式 (ATRAC) の基本原理はほぼ同じです。それなりに音質に配慮したパソコンを用意すれば、十分に MD デッキの代わりになりますし、パソコンを使わなくても、MP3 の録音、再生をするデッキを作ることは可能でしょう。実際、フラッシュメモリを利用したポータブル MP3 プレーヤが作られています。MP3 やその他、パソコン上で扱われるオーディオファイルについては、現在何の法律、政令も制定されておらず、録音補償金を払う必要はありません。

しかしながら、JASRAC や RIAA (アメリカの音楽業界団体) などは、前述のように MD 同様に利用できる MP3 やその他のパソコン用音楽フォーマットからも、金を取りたいところでしょう。MP3 の場合には、MD のように、メディアに録音補償金を上乗せすることは不可能です。汎用の記録媒体すべてに録音補償金をかけることはできません。そこでこの種の団体はこう考えます。

「ではエンコーダやプレーヤなどのソフトに上乗せすればどうか」

これもフリーのエンコーダやプレーヤが多く存在していて不可能と思われたのですが、最近になって激震が起きました。これについてはソフトウェア紹介で後述します。

## 2. データの再配布の問題

更に、MP3 ファイルの再配布の問題があります。MD のような「閉じた」フォーマットではありませんから、ごく通常のファイルと同様に流通できてしまいます。とりあえず、ネット上で不特定多数にばらまくのは「放送」に相当し、現在の法律でも明らかに違法とされます。

微妙なのは友人や家族の間での流通です。MD のデジタルコピーが一世代に限定されている趣旨から考えると、「CD を借りてきて MP3 化 友達にプレゼント」は違法ではないかということになりますが、現在の MD などについて、このあたりの法律がどうなっているのか全然知らないので (お 識者のフォローをいただきたいところです。例えば「友達から頼まれて CD を MD にコピーして渡す」のが違法なのかどうかと.....恐らくは厳密に言えば違法にあたるけれども黙認されている、というのが現状ではないでしょうか。ただでさえグレーな部分が大きそうなのですが、これが汎用のデジタルデータとなると「MD で渡すのが黙認されるなら、ネット経由で特定少数に渡すのも黙認されるのか」とか、あるいは

「再生のために一時的に HDD やメインメモリ上にコピーするのも含めて  
デジタルコピーは一切禁止っ！」

(MPEG2 ビデオ規格制定時のハリウッドの意見)

のような新機軸なものまで、諸説吹き出してきました。ていうか

混乱してきました。

法律関係について、すげえ大嘘書いてる気がしてきましたので、JASRAC の Q & A でも読んでみましょう。

Q テレビやラジオの放送，レンタル CDなどを自分で楽しむために，録音したいのですが，申請が必要ですか。

A 必要ありません。

著作権法 30 条では私的使用の著作物の複製（コピー，録音など）について以下の条件を満たした場合のみ，著作権の行使を制限しています。

1. 個人的に又は家庭内，その他限定された少数のグループ内で楽しむ場合
2. 公衆（特定，不特定かかわらず多数の者）が使用することを目的として設置されている複製機器を使って複製するのではないとき（ただし，MD など政令で定められたデジタル方式の録音，録画については，私的使用であっても補償金の支払いが義務づけられています）。

ですから，録音したものを多数の人の間で回覧するような場合などは，著作権者の許諾が必要となります。

読んでみたところで灰色ですね……「楽しむ」がいかようにも解釈できてしまいます。おそらくユーザーと JASRAC の見解には長大な隔たりがあると思われませんが。この項目に関しては，MP3 の登場以前から議論が絶えないのですが，深入りは避けます。

- （例）
- 既存の楽曲をベースにした MIDI ファイルの配布について
  - TV 放送の録画代行について

これらのケースについては，もしかしたら既に明確な結論が出ているのかもしれませんが。ついでに 1. の録音補償金についての Q & A です。

Q 自分が聞くために CD などを MD に録音したいのですが，申請は必要ですか。

A 必要ありません。

著作権法 30 条では個人で楽しむためであっても，MD など政令で定められたデジタル方式の録音，録画については補償金の支払いを義務づけています。

しかし，私的録音・録画行為を個々に捉えることは事実上困難であることなどから，著作権法 104 条の 4 において，補償金は文化庁長官の指定する管理団体が，記録機器・媒体の購入時に支払いを請求できることになっています。現在，私的録音については，私的録音補償金管理協会 SARAH が各メーカーの協力のもと，メーカーから支払いを受け，関係著作権管理団体へ分配しています。

ですから，個人がデジタル方式の私的録音，私的録画について許可を求める必要はありません。もちろん，MD に録音して多数の人に配る場合は，私的録音に該当しませんので，別途，著作権者の許諾をとる必要があります。

だそうです。

最近になって，PC 用でない単体の民生用 CD-R ドライブが政令指定機器になり，これにもなって家庭用 CD レコーダと，録音補償金を含み，更に MD 同様のコピープロテクト（デジタル デジタルは一代しかコピーが作れない）を施した専用のメディアが発売されました。PC 用の CD-R ドライブ & メディアに比べて高価かつ制約が多いため，普及するかどうかは甚だ疑問だと思います。

個人利用に関する結論ですが、現在の著作権法は、このようなデジタルデータに対応しておらず、違法、適法を論じることさえ困難ではないかと思えます。とりあえず、現時点において、CDからのMP3ファイルの作成や、少数のグループ内での流通が、法律違反となることはない、というのが私の見解なのですが、

## 大嘘書いてるかもしれません。

詳しい知識がありませんし、

この連載はデジタル著作権一般を論じるものではないので、深入りは避けます。ただMP3を語る上で、著作権論議はどうしても避けられないため、今回は少しばかり書いてみました。デジタル著作権については、詳しい方にフォローをお願いしたいところです。

### 追加 ポータブルMP3プレーヤ“Rio”の衝撃

この連載の最中に、MP3界にとって大きなトピックがありました。DiamondMultiMedia社(以下DMM)による携帯型MP3プレーヤ、Rioの発表と、その製造中止を求めて起こされた裁判です。

MP3データの録音、再生は別にパソコンを使わなくとも可能です。据え置き型のMP3デッキというのは聞いたことがありませんが、Mpmanというポータブルプレーヤが、韓国のセハン社によって発売され、アキバなどで販売されて、ここ数ヶ月話題になっていました。しかしながら、マイナーなメーカーであることや、高価(MDデッキとMDウォークマンが買える)なことから、広く一般に普及する様子はありませんでした。ネットではMpman以外のポータブルプレーヤ開発の噂も出回っていましたが、いずれもマイナーなメーカーによるもので、噂の域を出るものは少なかったように思います。

そんななか、DMMから、Rioがアナウンスされました。DMMはビデオカードやサウンドカードのメーカーとして、一般家電業界ではともかく、パソコン用パーツの世界ではかなりの大手です。Rioのハードウェアは、セハンがMpmanの次期モデルとして開発を進めていたものそのものです。これに、パソコンとの接続キットや、MP3エンコーダなどをセットにして¥30000以下と、実に魅力的です。

さて、RIAAはこの話を聞いて焦りました。更に、レコード会社はMP3の存在自体を快く思っていない。理由としては

- MP3からは録音補償金のような金がとれない。
- インターネット上には違法にアップされたMP3がわんさか。
- アーティストがレコード会社を通さずに直接消費者と結びつくことが可能になる。

などがあります。そこでRIAAはRioの恒久的な製造中止を求めた裁判を起こすと共に、判決までの間の生産停止の仮処分申請を出しました。とりあえず審問開始まで10日間の生産停止



命令が出されましたが、10/26の第一回審問の結果、仮差し止め命令の申請は棄却され、これで、今後の裁判、及び控訴審で生産中止命令が出ない限り、Mp3プレーヤを販売することが出来ることになり、Rioは11月中にも出荷されることになっています。裁判は長引くことが予想されるため、もし仮処分が認められていれば、MP3プレーヤの出荷は事実上不可能になるところでした。DMMはPCパーツ業界でこそ大手に入りますが、けして大企業ではなく、アメリカの音楽業界の90長引く訴訟を維持する財力はありません。

この訴訟はアメリカの、家庭での音楽記録に関する法律(AHRA)に基づいており、日本の場合とはまた異なりますが、おそらくこの訴訟の結果が世界的なリファレンスになるのではないかと思います。(著作権の扱いの違いに関する問題では、日本はほぼアメリカに全敗ですし。)DMMとRIAAの主張は両者のホームページに原文が掲載されていますが、ZDnetニュースのバックナンバーに主張の要点がまとめられています。

それにしても著作権がらみの話をし始めるとキリがないですね.....。

では今宵はこの辺で。

(和井田寛則 / わいりー)

*waita@tky2.3web.ne.jp*

### 今回参照したページ

DimandMultiMedia	<a href="http://www.diamondmm.com/">http://www.diamondmm.com/</a>
RIAA (全米レコード協会)	<a href="http://www.riaa.com/">http://www.riaa.com/</a>
ZDnet ニュース	<a href="http://www.zdnet.co.jp/news/">http://www.zdnet.co.jp/news/</a>
JASRAC ホームページ	<a href="http://www.jasrac.or.jp/">http://www.jasrac.or.jp/</a>
MP3tidalWave	<a href="http://mr.simplenet.com/layer3/title_1.html">http://mr.simplenet.com/layer3/title_1.html</a>
MP3.ne.jp	<a href="http://www.mp3.ne.jp/">http://www.mp3.ne.jp/</a>

(TidalWaveは昔からあるマニア向けの情報サイトです。MP3.ne.jpは、MP3の一般への浸透を目指して作られたサイトで、MPmanの輸入販売を行っている会社と縁があるそうです。ダークな面のないMP3情報サイト、を目指してます。)

## TSGer Mailing List (改訂再掲)

大岩 寛

駒場生・本郷生・アクティブな OB / OG などを含む TSGer の連絡用に、TSGer Mailing List を開設しました。

### 参加資格

TSGer 及びコンパなどの行事に参加している OB / OG など関係者とします。

### 内容

TSGer 全体への連絡・活動の公表などなど。TSG という団体としての活動には限定しませんので、TSGer の個人プロジェクトの参加者募集なども流して構いません。

### 投稿アドレス

tsg@italk.ne.jp

### 管理人アドレス

tsg-request@italk.ne.jp

### 参加方法

tsg@italk.ne.jp に、本文に

```
# subscribe your-name
```

の一行を書いたメール (*your-name* は自分の名前に置き換えて下さい) を送って下さい。

よくわからない方は、手動での登録も受け付けますので管理人までお気軽に御連絡下さい。

お問い合わせは [tsg-request@italk.ne.jp](mailto:tsg-request@italk.ne.jp) までどうぞ。

(大岩 寛)

## セキュリティ情報サイト構築プロジェクト（仮）

### 参加者募集のお知らせ

大岩 寛

インターネットの急激な発展に伴い、ソフトウェアのセキュリティホールなど、セキュリティに関する情報の重要性が増してきています。最近では2日に1件は何らかのセキュリティに関する情報が発表される状況です。一方で、セキュリティホールを攻撃して計算機に侵入するクラッキングも増加の一途を辿っていて、コンピュータ緊急対応センター（JPCERT/CC）によれば、国内での7月から9月までの3ヶ月間の不正アクセスの件数は、JPCERTに報告されただけで299件になっています。これは実際には氷山の一角で、実際の不正アクセスはこれよりかなり多いものと思われます。

しかし、これらの情報はあちこちのサイトで発表され、またほとんどが英語であること、また対策をとるのが面倒である上に非生産的な「後向き」の作業であることから、一般にこれらの情報を入手する努力はほとんどされておらず、不正アクセスは事実上野放しになっています。

最近でも Netscape Navigator や Internet Explorer に対する攻撃や、FreeBSD や Linux の走る計算機の管理者権限を奪う攻撃が発生しており、これらの情報をいち早く入手して対策をとることは急務になっています。

これまで「Security Watch」という DeerPark さんによる個人サイトがあり、これらの情報をいろいろな方面から集めて日本語の説明をつけて公開してくれていましたが、最近の情報量の増大もあり、負担が大きかったようで、残念ながらこの10月27日に店仕舞いしてしまいました。

そこで、自分たちでセキュリティに関する情報の収集し、それに関して自らの知識を少しでも高め、ついでにその情報を公開することで少しでも一般に役に立つことができたらということで、italker の Lef さんと私を中心に、セキュリティの情報を収集して Web で公開するプロジェクトを開始することになりました。しかしながら、これには大量のマンパワーを必要としますので、この2人だけではとても運営していくことはできません。このプロジェクトと一緒に進めて下さるメンバーを募集しています。

作業の内容としては、

- CNET, WIRED などのニュースサイトや、Microsoft や Netscape などのソフトメーカのホームページを巡回して、その中からセキュリティに関係のある記事を見つけてメーリングリストにその存在を知らせること
- 上の情報やセキュリティ情報センターの出すリリースなどを元に、それらの内容を要約（場合によっては翻訳）して、具体的な対策の必要な OS や対策方法などの情報をまとめること
- それを Web で公開するための準備
- これらの内容に関しての技術的なコメント付け

などがありますので、これらのうちで自分の提供できる時間と能力に応じて、少しでも力を貸して下されば幸いです。

## 外部 HTTP サーバのファイルの情報の取得について

また、すぐには作業に参加できなくても、セキュリティーに関する知識を得るために参加するという方も歓迎します。とりあえずはこれらの議論のためのメーリングリストを覗いて、これらの情報に触れてみるだけでも構いませんので、積極的に参加して下さいをお願いします。  
# 特に元気のいい駒場生の参加を期待します :)

少しでも興味を持って頂けたら、[oiwa@is.s.u-tokyo.ac.jp](mailto:oiwa@is.s.u-tokyo.ac.jp) までメールで連絡して下さい。  
または italk などでも連絡をとってくださっても構いません。

よろしくおねがいします。

(大岩 寛)

[oiwa@is.s.u-tokyo.ac.jp](mailto:oiwa@is.s.u-tokyo.ac.jp)

## 外部 HTTP サーバのファイルの情報の取得について

Lef

ぶほ編集部より

筆者 Lef さんは、TSG に在籍した経験はありませんが、TSG と歴史的に縁のあるチャットシステム italk などを通じて、多くの TSGer と交流を持っている方です。今回特別に、TSG 部報にご寄稿をいただきました。

なお Lef さんは、本誌 33 ページの記事で紹介されている、セキュリティー情報サイトに関するプロジェクトにも深くかかわっておられます。そちらもぜひご覧ください。

WWW に存在する無数のページはとても有益なものが多いですが、それらのページが更新されたかどうかを手動で確認するのはとても大変です。そのため、クライアント側で更新を自動的に確認する WWWC などのツールは、とても便利なものとして使われています。

しかし、更新確認をしたいというのは多くの方が思っていることであり、その更新情報そのものを Web ページに載せたいと思うこともあります。例えば、特定ジャンルのリンク集のページで、更新状態も表示したいなどという場合です。

これは、国内の Web 上で流行っている「日記」のリンク集でよく行われています。各日記のページの更新状態を確認して、「\*\*さんの\*\*日記(更新時刻\*\*/\*\*/\*\* \*\*\*)」などとリンク集のページに掲載されていると、毎日複数の日記を読むような人にとってはとても便利になるというわけです。

italk<sup>1</sup> の main サーバに集まる人達の日記を集めたページを作る上で、筆者も同じことを行いたいと思い、既に更新確認の自動化を実現しているページではどうやっているのかを調べました。そして、これらの更新確認に使われているツールとして、朝日奈アンテナ<sup>2</sup> と、kinofinger<sup>3</sup> というツールを知ることができました。

<sup>1</sup>【編註】<http://www.italk.ne.jp/~yuki/italk/> 参照。

<sup>2</sup><http://www.kabe.co.jp/~masshy/hina/release/>

<sup>3</sup><http://www.kinotrope.co.jp/~nakahiro/src/>

朝日奈アンテナと kinofinger の最大の違いは、前者が更新時間によってソートを行ってくれるのに対し、後者が HTML の特定部分を SSI の様に置き換えてくれるということです。

italker 日記 Links<sup>4</sup>では、後者の kinofinger に手を加えた上で、出力されたファイルを更に自作の Perl スクリプトに通し、ベースとなる HTML ドキュメントとマージすることで実現しています。

kinofinger を選択した理由ですが、構造がシンプルだったことや、朝日奈アンテナの機能以外の部分が筆者の好みに合わなかったというのが理由です。:)

kinofinger では、更新時刻順に項目をソートするため、HTML ドキュメントも行単位で順番を入れ換える必要があります。そこで、ベースとなる HTML ドキュメントの

```
<!-- check block start -->
<!-- check block end -->
```

などというコメントで囲まれた範囲をフィルターで動的に書き換えているわけです。この作業は、kinofinger が出力ファイルをそのまま HTML ドキュメントとして使うように作られているのに対し、別の HTML ドキュメントとマージして使おうとしているために必要になっています。これは、kinofinger の設計を無視して使っている筆者の使い方が間違っているといえるでしょう。

kinofinger はシンプルでいいのですが、出力されるファイルがあまりにもひどいことと、設定の仕方がかなり使いにくいものになっているため、好みの形になるように手を入れざるを得ませんでした。変更点はほんの数行ですが、出力形式と設定ファイルの形式を変更し、小さなバグを取ってあります。ただし、kinofinger とベースとなる HTML ドキュメントの二種類の部分で同じデータを管理しなくてはいけなくなっているため、構造的にとても美しいものとなっているのが現状です。

手を加えた筆者の kinofinger とその filter ですが、非公開のつもりはないものの、前述の問題を含めて様々な部分に不満があることや、場当たりの改良をしているために運用時にかなりノウハウが必要だったりするため、お薦めできないものとなっています。そのため、近々完全にスクラッチから書き直そうかと思っているので、できればそれを待っていただいたほうがいいでしょう。

なお、個人のページで応用したい場合などは、kinofinger を使わずに朝日奈アンテナを使うほうが便利だと思われます。また、手元で更新確認をするだけなら WWWC などを使ったほうが効果的でしょう。朝日奈アンテナは、X-TT と同じように名称の付け方などのセンスを疑いたくはなりませんが :)、機能的にはよくできているものの一つだと思います。

更新確認というのは Web 上では思ったより以上に便利に使えるので、興味のある方は是非もっと便利なものを作って欲しいと思います。:)

---

<sup>4</sup>URL は直接 italk に来て入手してください。

(追加) 駒場祭企画最新情報

11月4日現在

ぶほ編集部

## ヒヨシ部隊のゆくえ

こんなメールが流れてます。

---

Date: Wed, 04 Nov 1998 03:23:09 +0900  
From: Tohru TSUJIKAWA <tohru.tsujikawa@nifty.ne.jp>  
To: ml-tsg@ml.246.ne.jp  
Subject: [ml-tsg 805] Project 2468  
Message-Id: <363F4A0D1F4.78CBTOHRU.TSUJIKAWA@mail.angel.ne.jp>  
Reply-To: ml-tsg@ml.246.ne.jp

日吉 2468 部隊のプロジェクトが発進しました。

詳細はわいりー氏が近日中に公表する予定です。

-----  
辻河亨@東京大学 理学部情報科学科 内定 (g740340@komaba.ecc.u-tokyo.ac.jp)  
mailto: para@246.ne.jp

---

Date: Wed, 04 Nov 1998 03:58:52 +0900  
From: Tetsuto SHIMOMURA <tact@246.ne.jp>  
To: ml-tsg@ml.246.ne.jp  
Subject: [ml-tsg 806] 日吉勢力結集  
In-Reply-To: <363F4A0D1F4.78CBTOHRU.TSUJIKAWA@mail.angel.ne.jp>  
References: <363F4A0D1F4.78CBTOHRU.TSUJIKAWA@mail.angel.ne.jp>  
Message-Id: <363F526CC8.1C69TACT@mail.246.ne.jp>  
Reply-To: ml-tsg@ml.246.ne.jp

TACTです。

- > 日吉 2468 部隊のプロジェクトが発進しました。
- > 詳細はわいりー氏が近日中に公表する予定です。

WE826 部隊と検討を重ねた結果、

「僕の企画を継続するのは無理である」

という結論に達したため、僕も WE826 部隊に  
合流することにしました。  
「スピ魅」よりはまともなものを作るので、  
期待しててください(お

=== == = TACT/下村哲人 = == ==  
=== == = tact@246.ne.jp = == ==

---

(ぶほ編集部)

(Wed, 4 Nov 1998 09:48:20 +0900)

## 編集後記

TSGer ML

先月末のコンパの際、大岩さんより、駒場生も各自登録してほしいとの発言がありました。32ページ参照。

pub/prv 分離中止

企画紹介がわりとイミあるものに仕上がったので、駒場祭配布用の学外版は特に用意しないことにしました。

著者取材のため九歳（意味不明）

いえ、PDK サボってるわけじゃないんです。ただ、駒場祭関係の記事をきっちり構成していったら、わりと厚くなってしまって。40ページ超にはしたくなかったんです。

駒場祭接近

緊迫感の場  $T(x, y, z, person)$  にムラがあるような気が。でも  $t \rightarrow (\text{駒場祭当日}) - 0$  で  $\nabla T \rightarrow 0$ ,  $T \rightarrow +\infty$  が境界条件。

今回の表紙

テレビ東京系で放映中のアニメ番組「彼氏彼女の事情」オープニング前のカットです。監督は「エヴァ」の庵野秀明、以下 GAINAX 関係の名前がわらわら並びます。

「ポケモン」光過敏症騒動以来、テレビ東京系の TV アニメには、本編開始後すぐに「部屋を明るくして、近づきすぎないように……」のような字幕が出るようになり、これがアニメファンには不評でした。

それが、彼の手にかかれば演出の一部（？）の制約にめげず、作品の構成に取り込んでしまうあたり、学ぶべき姿勢のような気がするのですが……。

なお庵野氏に関しては、映画作品でもたびたび「映倫」マークや制作・配給会社のタイトルを強調する演出を行なった実績があります。

次回締切は 19 日正午

印刷の都合が厳しいので、絶対に延びません。本物のデッドラインです。駒場祭当日のスペックを的確に予測して、自分の企画の紹介記事を早いうちから用意しておいて下さい。

---

### 理論科学グループ 部報 第 215 号

1998 年 11 月 4 日 発行

発行者 今野 俊一

編集者 福林 一平

発行所 理論科学グループ

〒 153-0041 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

---

©Theoretical Science Group, University of Tokyo, 1998.

All rights are reserved.

Printed in Japan.

理論科学グループ部報 第 215 号  
— ふほ・駒場祭準備特集号 —  
1998 年 11 月 4 日

*THEORETICAL    SCIENCE    GROUP*