

D言

「D言語」
群馬高専電算部

2010

電
N



付録 CD について

CD には以下のファイルが含まれております。

- 新作プログラム（文発 2010）

リバーシ・改 作：jinkai

MineSweeperMODOKI 作：tk

- 旧作プログラム（工華祭 2009）

FeliCa Battler 作：akina ※この作品の動作には PaSoRi RC-S330（ソニー製）が必要です。

- 旧作プログラム（コミックマーケット 76 出展作品）

クイズの国のアリスたん 作：しまぱん、からぼん（グラフィック：ますたー）

- 過去のプログラム

シューティング 作：JJ

リバーシ（旧アルゴリズムバージョン） 作：麿芥

東方工華鏡（シューティング） 作：akina（グラフィック：ますたー）

<卒業生製作プログラム>

から風（落ち物系） 作：輝凜

TypeHell（タイピング） 作：Skyblue

流れ星（シューティング） 作：いちご、よもぎ、霧ヶ峰ジュン、さっちゃん

- 画像など 作：ますたー、EXILIAS

- その他 過去部誌 PDF 版など

収録されているプログラムは Microsoft Windows XP Professional SP3、Windows Vista Business SP2（いずれも 32 ビット版）で動作確認を行っております。なお、すべてのコンピュータにおいて動作を保証するものではありません。また、各プログラムの著作権はそれぞれの制作者にあります。無断使用・転載は大歓迎します。バグの報告、ご感想などありましたら、下記メールアドレスまでお待ちしております。

メール gnect.densan@gmail.com

※本誌中に現れる製品名・商品名・ロゴ等は各社の商標または登録商標です。

もくじ

付録 CD について

表紙裏

Xperia レビュー 2

高専 1 年が Android 説明したり。 6

MHP2G ～効率狩りのススメ～ 10

Small Basic 12

群馬高専隔離スレ 16

弱いオセロを作る過程を
赤裸々に記す。 19

Befunge 22

きょーじゅとじょしゅの
中二病のおはなし（漫画） 24

NTT ドコモ／ソニー・エリクソン XPERIA レビュー

2010年4月1日、NTT ドコモから満を持して発売されたソニー・エリクソンのスマートフォン・Xperia（エクスペリア）SO-01B。2度のファームウェアアップデートを経て、いかに進化したのだろうか。実機を手にもレビューする。

著：akina

外観と基本仕様

●一般的な携帯電話に比べて少し大きめの本体

まずは大きさから見てみよう。一般的な折りたたみタイプの携帯電話（写真右・NTT ドコモ FOMA N906i）に比べ、厚みはないが、縦・横ともに大きい。表面にはその大部分を占める約 4.0 インチの液晶と Android 機に共通する「メニュー」「ホーム」「戻る」の 3 つのハードキー、受話スピーカー、フルカラー LED がある。左側面には着信音などが再生されるモノラルスピーカー、右側面には音量調節キーとカメラボタンがある。上側面には電源ボタンとイヤホンマイク／ステレオヘッドホン用のオーディオミニプラグ端子、充電・通信用の microUSB 端子を備え、下側面には送話マイクとスマートフォンには珍しいストラップホールがある。裏面には約 810 万画素のカメラレンズと LED フラッシュライトがある。ライトはかなり明るく、懐中電灯代わりに使用するためのアプリがあるほどだ。また、ほぼ全体が電池カバーとなっており、レンズとライトの部分が切り抜きになっている。

カバーを開けると、大部分を電池パックが占めている。本機の内部記憶は共有部分で約 440 MB と少ないが、microSDHC スロットを備え、16 GB のものが添付されているのでストレージ容量に困ることはほとんどないだろう。



大きさを比較



厚みを比較（底面）
底面にはストラップホールがある



上面とカメラレンズ（ホワイトモデル）
（NTT ドコモ Web サイトより引用）

●持ちやすいデザイン・ボディーカラーによって異なる質感



筆者の基本的な持ち方（片手持ち）

側面がラウンドになっており、持ちやすい。筆者は基本的に 4 本の指で本体を支えるように持ち、親指で操作する。または、手のひら全体で支え、逆の手の人差し指で操作する。筆者の手はあまり大きい方ではないが（手を広げて親指から小指までが約 21 cm）、前者のように片手で操作することができる。ただ、画面端、とくに（左手持ちの場合）右下付近は多少きつく感じ、アプリによっては後者のような両手操作の方が楽だ。多くの男性は片手操作ができるだろうが、女性は両手操作が基本になるだろう。

ボディー（手で触れる電池カバー部分）の質感はカラーによって異なり、ブラックモデルはレザー風の非光沢、ホワイトモデルはツルツルで光沢感がある。



両手持ちの場合
(NTT ドコモ Web サイトより引用)

●余裕のスペックだが、電池の持ちは残念

リッチなユーザーインターフェース (UI) を提供するスマートフォンでは、一般的な携帯電話ではあまり気にすることのない CPU やメインメモリのスペックが重要だ。本機は CPU に 1 GHz 駆動の QSD8250 と 384 MB の RAM を搭載している。数多くの Android 機のなかでもハイスペックな製品だ。

当然、高機能機の消費する電力は大きい。電池容量は 1500 mAh とかなり大きい*が、電池持ちの悪さは否めない。公称の連続待受時間 (3G 移動時) は約 190 時間だが、実際に使用してみると、移動中のブラウジングとメール十数通・通話数分のようなあまり使わない使い方でも 2 日が限度で、各種アプリを使用するなど「スマートフォンらしい」使い方をすれば、結果的には毎日充電して使うようになる。もちろん、一般的な携帯電話でも毎日充電するくらいの人ならば気にならないだろう。ちなみに、フル稼働 (液晶常時点灯・頻繁な通信) 状態では 6 時間程度で切れる。

●スマートフォン用 OS の大本命・Google Android OS

スマートフォン用 OS としては、アップル社製品専用の iOS や同様の BlackBerry などのをのぞけば、Google Android か Microsoft Windows Mobile のどちらかに二分される。Windows Mobile には長い歴史とサポートにより、企業ユーザーに喜ばれている。国産初スマートフォンとして有名なウィルコム (WILCOM) の W-ZERO3 シリーズに搭載されていることでもおなじみだ。

後釜である Android の強みといえば、なんといってもオープンソース開発による柔軟性と Google サービスとのシームレスな連携があげられる。Gmail や Google カレンダーなど、PC 版と同期して使用できたり、Gmail がキャリアのメールのようにプッシュ配信で受け取れたり、内蔵 GPS 以外に電話帳の住所データともリンクした Google マップの経路探索などに Google らしさを感じられる。

Android が iPhone と根本的に違うのはマルチタスク機能だ。iOS のマルチタスクは対応アプリに限られるが、Android はそれが前提、PC と同じである。だから、音楽を聴きつつブラウザを参照しながらメールを書くなど、PC では当たり前作業ができる。また、非純正の日本語入力アプリ**やファイラー、テザリングなどのシステムに直接関わるアプリがあったり、Android マーケット (iPhone でいう App Store) に登録されていないアプリを端末から直接または PC と USB 接続してインストールして使用できたりするのも、オープンソースならではの。

* 一般的な携帯電話の約 2 倍

** PC でいうところの IME



4 大スマートフォンのロゴ



Sony Ericsson

●「さすがドコモ」の通信環境

HSPA (FOMA ハイスピード) に対応する本機は、最大で
上り 2.0 Mbps、下り 7.2 Mbps で通信できる。実測 (BNR
スピードテスト画像読込版) でも、本機単体でブラウザ
からの計測で約 1.1 Mbps、PC を接続したテザリング状態では約 2.8
Mbps で通信できた。計測場所は「SoftBank が圏外な」筆者宅であ
る。ほかにも、iPhone ではできない 3G 環境での YouTube 高画質モー
ド再生など、回線のクオリティには「ドコモらしい」良さがある。
もちろん Wi-Fi (IEEE802.11b/g) も搭載し、海外にも対応している。



日本語入力は POBox Touch を搭載し、
PC 入力のほかフリック入力にも対応
(画像は ITmedia より引用)

内蔵アプリケーション

●日本人向けにカスタマイズ、だが「ケータイ」に勝てない

電話としての重要な機能・電話帳は、今まで使ってきた携
帯電話からも microSD や Bluetooth を通して転送できる。
Gmail の連絡帳と同期しておけば、PC などから編集する
こともできる。しかし、任意のグループを本機からは作成できず、
今までのグループ設定も引き継げないなど、「ケータイ」に一歩お
よばない感じだ。

力レンダーも同様に同期して使用することができる。しか
し、これは「日本の祝日」を表示してくれない。どちら
にもいえることだが、Android デフォルトのものより UI
も含めてかなり日本人向けにカスタマイズされており、日本で売る
意気込みを感じられる。だが、今一歩感は否めず、今後の改良を期
待したい。もっとも、標準アプリが気に入らなければ、別のアプリ
を入れればいいと考えることもできる。

●普通に使える Mediascape・かなり使えない Timescape

Mediascape は CM でも使われている Xperia の目玉機能
のひとつで、撮影した写真や動画、本機内や Web 上の
音楽をまとめて管理できるアプリだ。なかでも Infinite
ボタンが秀逸で、再生中の楽曲や表示中の写真の人物を自動認識し
てその関連情報を得ることができる。

Timescape も同様に目玉機能のひとつなのだが、これが実
はいまいち使えない。時系列順に撮影した写真や通話・メー
ルや mixi・Twitter などの発言を見ることができなのがウリ
なのだが、電話帳で写真登録してなければそれは味気なく、mixi で
は直接日記投稿やコメント書き込みなどはできず、タッチしてもブ
ラウザで該当ページに飛ばされるだけのリンクにすぎない。



Mediascape (左) と Timescape (右)
(画像は ITmedia より引用)

“Xperia”、“Timescape” は Sony Ericsson Mobile Communications AB の商標または登録商標です。

“Google”、“Android”、“Gmail”、“YouTube” および関連のロゴは、Google Inc. の商標または登録商標です。

その他、商品名・ロゴマーク等は、各社の商標または登録商標です。

外部アプリケーション

Android のアプリは主に Android マーケットから入手できる。2010 年 7 月には 10 万本を超えたが、ここでは無料アプリにしぼって、特筆すべきものについて紹介する。

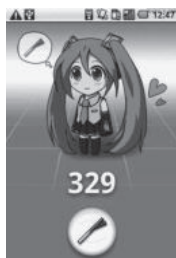
●必須はメールとタスクと Google 純正品。そして QR コード

メールは、Gmail と E メールが内蔵されている。しかし、9 月からはじまった i モードメールアドレスが引き継げる sp モードメールでは、専用アプリが必要だ。このアプリでは、一般的な携帯電話と同じように絵文字やデコメも使える。9/13 現在、アップデートは 1 度行われたが、なんといってもこのアプリは動作が重い。Xperia のスペックは申し分ないはずなので、アプリ側の問題といえるだろう。早めの改善を期待したい。

Android はマルチタスクと前述したが、アプリには終了の概念がないため、ホームに戻ったり別のアプリを起動したりしても、基本的にはバックグラウンドで動き続ける。そのため、多数のアプリの起動を繰り返していると、本機の動きが鈍くなってしまうことがある。タスクを終了させるには Taskiller free がおすすめだ。ウィジェットとしてホームでメモリ残量確認もできる。

Google 純正アプリと聞くと、内蔵されていないのかと疑問に思うだろう。しかし、一部の純正アプリはマーケットからダウンロードしなければならないものもある。そのなかでも、本機に話して「ググレ」る「音声検索」と、マップと連携して GPS で軌跡が残せる「My Tracks」はおすすめしたい。

赤外線通信機能は本機にはない。ケータイと電話帳のやりとりをする場合、標準では手打ちする必要がある。そんなときに便利なのが、自分の電話帳データを QR コード化するアプリだ。いくつも種類があるが、多機能な「シンプル QR コード」が秀逸だ。これで作成した QR コードを読み取ってもらえばいい。また、「QR コードスキャナー」も導入しておくとう便利だ。



「ねぎボタン」の画面
ただのカウンター。押すとミクが動く
(画像は AndroLib.com より引用)

Xperia と Android 全般について軽く触れてきたが、ここでは紹介できなかったことも多い。また、スマートフォンは高機能だが、日本人にとってはいわゆる「ガラケー」のほうが一日の長がある部分も多い。たとえば、おサイフケータイはガラケーならではの機能だ。お互いの機能が近づいていくことを期待したい。



Taskiller free の画面

アプリごとに終了・無視が選択できる



マップと連動した My Tracks の画面
軌跡 (GPS データ) の出力も可能
(画像は AndroLib.com より引用)

●必要になったら探せば出てくる

そのほか、必要だと思ったアプリは探せば出てくる。たとえば、mixi なら「TxMixiView」、2ちゃんねるなら「Tuboroid」がおすすめ。純正の「ヤマダ電機」携帯会員アプリがあれば、非純正だがマックのクーポンを表示する「McCoupon」、使い道のない「ねぎボタン」などのアプリもある。アプリはスマートフォンの醍醐味なので、探してみるとおもしろい。迷ったときはアプリ紹介本や Web サイトも参照するとよいだろう。



高専1年が Android 説明したり。 (ハロワもあるよ)

著：key

ここでは Android プログラミングにハマったとある高専生が Android について、簡単だけど実用性皆無のアプリの作り方を紹介したいと思います。

第1章：Android の説明

1. そもそも Android ってなに？

Android (アンドロイド) とは、スマートフォンやインターネット端末やタブレットなどを主なターゲットとして開発された、Linux ベースのモバイル用オープンソースオペレーティングシステム、ミドルウェア、主要なアプリケーションからなるソフトウェアスタック (集合) パッケージを基にしたプラットフォームです (Wikipedia より引用)。

要は Google の出している携帯用オープンソース OS です。企業によっては一部改造している企業もあります (SonyEricsson の Xperia など)。一般にこの OS を搭載した携帯電話はスマートフォンといわれ、日本では普通の携帯より高性能というイメージが強いようです。また、スマートフォンの OS はあの有名な iPhone に搭載されている iOS のほかに、BlackBerry、Windows Mobile などがあります。

2. なんで Android について書いたの？

筆者が Android のプログラミングをやっているため、というのとともに携帯電話が好きだったという二点からです。プログラミングについては次項に詳しく書くのでここでは割愛させていただきます。携帯が好きというのは、主にデザインですね。自分は一時期デザイナーに憧れておりまして、日本の携帯電話は多くのデザインがあって、なおかつこよかったため、携帯電話にひかれていったんです。それが原因で Android プログラミングを始めたようなものですから、結果としてよかったなあと思います。

3. iOS と Android の違いは？

スマートフォンに触れた事のない方の多くはこのように思っているのが現実です。まあ iOS (iphone の OS) と Android じゃあメディアの露出度がなあ…… (遠い目) というわけで両者の利点と欠点を挙げて簡単に説明したいと思います。

● iOS

利点……レスポンスが速い。アプリが豊富。

欠点……搭載機種が iphone のみ。AppStore の審査基準が分かりにくい。

● Android

利点……搭載機種が多い。成長の余地がある。

欠点……レスポンスが遅い。アプリが iPhone の半分程度（8/31 現在）。

➤ れだけ見ると iOS の方が上に見えますが、「レスポンスが遅い」は最新の製品では iOS に近い
↳ ともいわれ、アプリ数も急増しているので、iOS を抜くのもそう遠くはありません。むしろそ
うであってほしい。

また、開発者側から見ても大きな差があります。iOS は objective-C という C 言語をもとにした
ま 言語を使ってプログラムを組みます。それに対して Android は主に Java 言語を使ってプログラ
ムを組みます。また、取り扱える言語にも違いがあり、iOS では Flash が使えず HTML5 が使える
といった具合です。

第 2 章：Android プログラム

1. 使うものとその設定

Android プログラムを作るのに必要なものは、

- Android SDK
- Eclipse 3.5 ←ここ重要（後述）
- Java の知識（無いと大変苦労します）
- 少しの好奇心

ぐらいですね。

2 番目の Eclipse3.5 というのがなかなか重要な点でして、最新版の 3.6 だと XML ファイル（後述）
の編集時にエラーを吐きまくって実行できなくなります。実際 Android の中の人たちも 3.6 はお
すすめしていません。あと、筆者は Pleiades というプラグインで日本語化されたものを使っています。

➤ こまで来たのでセットアップまで説明します。
↳ まず、

<http://developer.android.com/intl/ja/>

から、お使いの OS に合ったものをダウンロードしてください（以降 Windows と仮定して進めます）。
説明文をよく読んで、OK なら I agree ～をクリックしてくださいね。落としたものは android-sdk-
windows と名前をつけて C ドライブ直下に解凍します（別に名前と解凍場所はどこでもいいです）。
そのあとは Path の設定をします。システムのプロパティ→システムの詳細設定→環境変数と進みま
す。その中でシステム環境変数が Path のものを探してください。あったらそれを選択して「編集」
をクリック、変数値の一番後ろに、

c:\android-sdk-windows\tools;

と追加します（解凍した場所によって適宜読み替えてください）。

もし Path がないというのなら、「新規」をクリックして、

変数名に「Path」

変数値に「c:\android-sdk-windows\tools;」

と入力します。これで Path の設定ができました。

次 は新しい変数を作ります。環境変数の設定画面にある「新規」をクリック、

変数名に「ANDROID_HOME」

変数値に「c:\android-sdk-windows\tools;」

と入力してください。その後「OK」を押して一通りウィンドウを閉じてください。

ここでやっと Eclipse のターンです。Pleoades のサイトから Java の 3.5 の zip をダウンロードし、解凍。中に実行ファイルがあるのでデスクトップかタスクバーに表示しておきましょう。実行後ワークスペースについて聞かれるので、適当な場所に作ってください。間違ってもデスクトップをワークスペースにしないでください。酷いことになります（ソースは筆者）。

無事に起動したら、ヘルプ→新規ソフトウェアのインストールをクリックし、表示されたウィンドウにある「追加」のボタンを押し、

名前に「Android-Plugin」

ロケーションに「<https://dl-ssl.google.com/android/eclipse/>」

と入力し、OK をクリック。出てきた「Developer Tools」をクリックして選択、以降 OK しまくってインストール、Eclipse の再起動で開発環境ができます。ですが、まだ大事なエミュレーターの設定が残っています。

Eclipse のメニューにある

ウィンドウ→Android SDK and ADV Manager

をクリック。「新規」をクリックして、名前とターゲット（OS のバージョン、とりあえず 2.2 を選択）を入力、create ADV をクリックするとエミュレーターができるので、作ったエミュレーターを選択して「開始」を押してみましょう。エミュレーターが起動します。とりあえずこの重さを実感しましょう。違うベクトルで衝撃を受けます。これで一通りの設定が終わりました。

なお、エミュレーターの日本語化は割愛させていただきます。

2. Hello World 表示プログラムを作る

今回はおそらく最も簡単なプログラムである「Hello,World!」と表示するプログラムを作りたいと思います。Eclipse を起動し、

ファイル→新規→プロジェクト→Android →Android プロジェクト

と行き、「次へ」をクリック、プロジェクト名に「Hello World」と入力、ターゲットに Android2.2 を選択、アプリケーション名に「Hello World」

パッケージ名に「net.hello」

Create Activity にチェックを入れ、「hello」

と入力して「完了」をクリック、何もせずに、

メニューの実行→「実行」

その後出てくるウィンドウで、

「Android アプリケーション」を選択

して実行。エミュレーターが起動するので放置。そうすると黒い背景に「Hello World,hello!」と出力されます。とりあえずエミュレーターは消してください。

次にこのプログラムの解説をしたいと思います（ソースがあっても無くても混乱する解説なので一発で理解したい方はネットで検索してくださいね！）。

1 行目：net.hello のソースコードであるということです。

3,4 行目：Activity メソッドと Bundle メソッドをインポートして使えるようにします。

6 行目：Activity クラスを継承した hello クラスを作成します。

7 行目：コメントです。これが無くても正しく動きます。内容は自力で訳してください。

8 行目：脚注です。以下のメソッドがオーバーライドされていない場合にエラーを吐きます。

9 行目：onCreate メソッドを作ります。また、Bundle 型の savedInstanceState という変数を作ります。

10 行目：スーパークラスの onCreate メソッドの savedInstanceState を参照しています（この辺

は筆者も実を言うとよくわからない)。

11 行目：R.layout.main を View にセットします。

12 行目：onCreate を閉じます。

13 行目：hello クラスを閉じます。

となっています。Java を知らない人にとっては何のことだかさっぱりです。Java やってる人もさっぱりです。筆者もさっぱ r (ry……

結構特殊な仕組みがあるのでそこについて説明します。ここで出てきている Android 独特の仕組みは Activity と View と onCreate です。Activity は PC でのウィンドウで View はウィンドウに表示されている内容、といきましょう。6 行目で Activity が作られています。また、11 行目で View の内容が決められています。この R なんとかについては後で説明します。onCreate は Activity ができたときに最初に実行されるメソッドです。ここでは onCreate しかないので大して意味がなさそうですが結構重要です。さて、R.layout.main について説明します。これは Hello World プロジェクトの中にある res フォルダの中の layout フォルダの中にある main.xml のことを指します。これには要約すると、「string にある hello を表示しろ」と書いてあります。めんどくさいです。ほんと。でこの hello は res フォルダの values フォルダの string.xml 書いてある、

```
<string name="hello">Hello World, hello!</string>
```

を指します。「hello」というのは変数のようなものと考えてください（なんで曖昧なのかって？それは筆者もよく分からないか r (ry)。つまり hello = Hello World, hello!。ここをいじれば Hello,World だけを表示させることも可能です。このようにしてこのアプリは成り立っています。

3. Hello World のあとに

➤ のように Android アプリは複雑です。他にも色々な変数名やらがでできます。正直言って大変です。ですが、アプリができたときの感動は格別です。特に実機で動いたときなんか言葉にできないくらい身が震えます。これを機会に Android について知ってみたい方はネットで検索してみるのをお勧めします。また、お勧めの書籍とサイトを紹介させていただきます。

● 「GAPSIS」 <http://www.gapsis.jp/>

Android 関係の情報ならまずここ。最新のものが多くそろっています。

● 「Android 入門」 <http://www.javadrive.jp/android/>

さまざまなクラスについて丁寧に書かれています。また、Java が分からない人はここの Home → Java 入門がお勧め。スーパークラスあたりまで完璧にしておくとなんか済みます。

■ 『Android 2.1 プログラミングバイブル』(ソシム)

色々なクラスについて、サンプルプログラムも交えて説明しています。

■ 『Google Android プログラミング入門』(アスキー・メディアワークス)

Android 関係の本で最も高いが初心者にお勧めしたいのがこれ。ただし Java の知識は当然要ります。

この記事で少しでも Android・Android プログラミングについて興味を持っていただけたら幸いです。



MHP2G

～効率狩りのススメ～

著：crin

去る 2008 年 3 月 27 日、PSP ゲームソフトで最高の、売り上げ本数 400 万本を達成した「モンスターハンターポータブル 2ndG」が発売された。大好評の「モンスターハンターポータブル 2nd」のアップグレード版として発売され、全国のファンを虜にした。

さて、モンハンで大事なことはなんでしょう？

一番大事なのはテクニック。これは誰でも答えるでしょう。

では次に大事なことは？チームワーク？ロマン？それとも運？

——答えは「効率化」です。異論はありません。認めません。

ということで、ここではモンハンにおける「効率狩り」の魅力について解説していきます。また、効率を求めるプレイヤーそれぞれに自分の考える効率化があるので、私の考える効率狩りであることをご了承ください。

なぜ、いまさらこんなことを書くかって？ さあ、なぜだろう。

まず、効率狩りとはどういったものなのか、それについて説明します。

効率狩りとは簡単に言うと、金やアイテムを惜しまず、一方的かつ短時間でモンスターを討伐することです。モンスターを完全パターン化して討伐できた時などは笑いが止まりません。そんな効率狩りで大切なことは、計算することと仲間を捨てること、そして作業を楽しむ心を持つことです。

とりあえずまずは武器選びから。武器を選ぶときはモンスターの肉質（ダメージの通りやすさ）を見て判断します。肉質はモンスターの部位によって異なり、さらには攻撃の種類によっても異なります。また、属性攻撃に対する肉質も部位によって異なります。

攻撃の種類には「斬撃」「打撃」「射撃」の3種類があり、前述の通り同じ部位でもその種類によって肉質は変わります。（例を挙げると、3種類を総合して肉質の軟らかいディアブロスの尻尾は斬撃や打撃でも結構なダメージを与えられるが、射撃での肉質が一番軟らかく最も大きなダメージを与えられる。）しかし、最も肉質の軟らかくなる武器を選べばいいというわけでもなく、その武器で連続して攻撃を加えられるかというのもポイント。攻撃を当てやすく、反撃の回避もしやすい武器であることも確認しなければいけません。

次に武器のステータスについて。これは色々面倒な説明になってしまうので結論だけ。

○近接武器は切れ味を第一に考える。常に白以上を保つように。

○肉質が軟らかいモンスターには攻撃力重視、攻撃力の高い無属性でも十分。硬いモンスターには属性も少し考慮する（属性ダメージは物理的な肉質に影響されないため）。

○属性値が等しければ大剣の一撃も片手剣の一撃も与える属性ダメージは等しい。

したがって、手数が多い武器は属性重視、少ない武器は攻撃力重視と考える。

○ボウガンは基本的にヘビィで、弾の特徴・攻撃力を熟知し、弓はモンスターの大きさによって連射と貫通を使い分ける。常にクリティカル距離を意識し、攻撃の手を休めない。

こんなところでしょうか。さらに細かいポイントもあるのですが、これくらいを把握していれば効率的に狩りができます。特に属性には注意してください。属性値に特化して攻撃は低いというのはお勧めしません。また、モンスターによっては状態異常も有効だったりします。火に弱いモンスターにはいつもコレではなく、肉質や攻撃力と属性値のバランスなども考えながら、モンスター毎に使用する武器を一つ一つ決めておくとういでしょう。

武器選びの後は実戦での立ち回り。スキルは当然ながら攻撃力UPや斬れ味レベル+1などの攻撃系で、モンスターの攻撃は極力回避です。先生のクチバンなど当たったら恥と思ってください。攻撃は特に部位破壊を狙わない限り、常に弱点やハメができる箇所を意識します。部位耐久値なども予測できるとモンスターの突進を攻撃で止めるなんてこともできます。アイテムを使うときは、量があるものは常時惜しまず使い、罾など数が少ないものは最も効果があるときに使います。大抵のモンスターは怒り時になるとアイテムの効果が薄れるので注意。

計算をするというのはこのようなことです。

そして、効率狩りでさらに大事な、仲間を捨てること。と、作業を楽しむ心。仲間と狩りをする
とモンスターの攻撃対象がバラバラになってしまい、結果的に事故多発で攻撃も当てにくく、グダグダと時間だけが過ぎてしまいます。そんなつまらないことで時間を無駄にするのは勿体無いです。モンスターの注意を自分だけに集中させ、その猛攻を決められた方法で避け、決められた方法で決められた位置に攻撃するというパターンをテンポよくこなすというのが効率狩りです。そして、そんな作業を苦と思わず逆に楽しむというやりこみの心が必要なのです。モンハンは仲間とワイワイ狩るのではなく、作業でモンスターを倒して俺 TUEEE することに快感を覚えるゲームなのです。

読み返してみると、武器中心の誰でも知っているようなことを堅苦しく長々と書いてしまいました。しかし、効率狩りについては少なからず理解していただけたかと思います。効率狩りというものはロマンであり時短テクニックでもあるのです。他にも書きたいことはあるのですがとりあえずここまで。痛々しい内容でしたがありがとうございました。これであなたも効率厨です。



Small Basic

著：tk

みなさんはプログラミング言語というものを知っていますか？

プログラミング言語とはパソコンのソフトウェア（Microsoft Office などのパソコン上で動くもの）を作成するための言葉のことです。

このプログラミング言語は、C 言語、C++、Java などたくさんありますが、今回はその中でも非常に簡単な言語である「Small Basic」について紹介したいと思います。

第 1 章：Small Basic とは

Small Basic とは Microsoft が開発した初心者向けのプログラミング言語です。プログラミングを知らない人でも理解しやすいように設計されているので、一番最初に触れる言語としてお勧めできます。

●少し詳しい Small Basic 参考：「About Microsoft Small Basic」 <http://smallbasic.com/>

少し詳しい話をしますので、このセクションは読み飛ばしてもらってもかまいません。

Small Basic の本質は以下の通りで、プログラミングに楽しさを取り戻すことです。

"Microsoft Small Basic puts the "fun" back into computer programming."

主な特徴としては "Simple" "Social" "Gradual" の 3 つがあります。

Simple とは、非常に短いコードでオリジナルのプログラミングやゲームを作ることができることです。IntelliSense（コードの入力候補を表示してくれる機能）が優れており、簡単にコードを書くことができます。

Social とは、自分の作ったプログラムを友達と簡単に共有できることです。作ったゲームを Silverlight（Web ブラウザ用のプラグイン）を用いて Web ブラウザで遊ぶことができます。

Gradual とは、この言語を基礎として他のプログラミング言語を学ぶことができます。特に Visual Basic とは相性がいいので、この言語を学んだ後は是非 Visual Basic にも触れてみて下さい。

第 2 章：コンピュータの準備

1. 開発環境のインストール

インストールは以下のサイトからダウンロードして実行するだけです。マイクロソフトの Web サイトから、次のページを見つけてみましょう。

「Download details: Small Basic - www.microsoft.com」

2. 開発画面

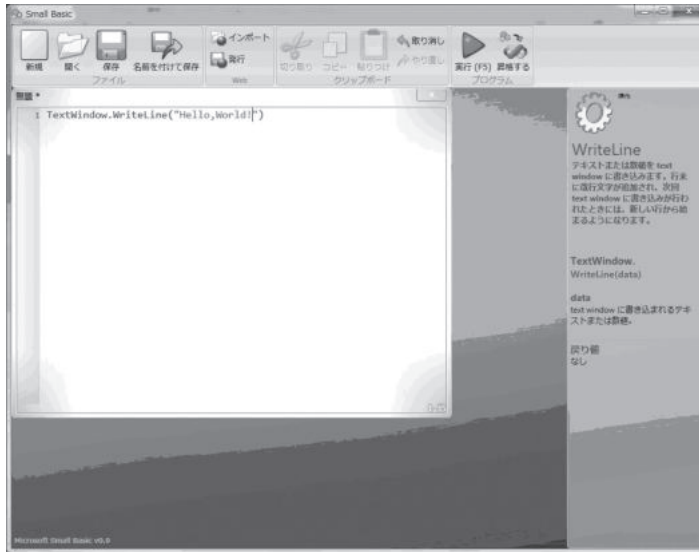
インストールは終わりましたか？

それでは開発画面を見てみましょう。

画面上部にあるツールバーでプログラムの作成、保存、実行などを行います。

右側にあるのがヘルプで、オペレーション（Operation：他の言語では関数やメソッドと呼ばれている機能です。3章で詳しく説明します）の説明などが表示されます。

左側の一番大きな場所がエディタで、ここにコードを書いていきます。



開発画面

第3章：実際にプログラミング

1. 実行してみよう

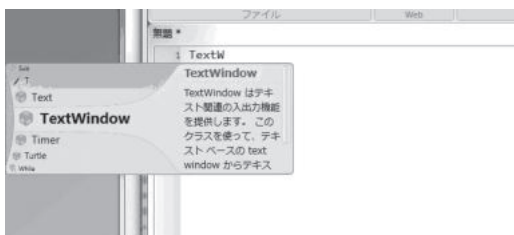
まずは世界で一番有名なプログラムである「Hello, World!」を出力させてみましょう。
Small Basic を起動し、エディタに、

```
TextWindow.WriteLine("Hello, World!")
```

と入力して下さい。途中まで入力すれば IntelliSense（コードの入力候補を出してくれる機能）がはたらいて (A)、入力候補が現れます。

入力後に右上の実行を押す (B) と、コマンドプロンプトが表示され、Hello,World! と出力されます (C)。終了するにはコマンドプロンプト上で何かキーを押して下さい。

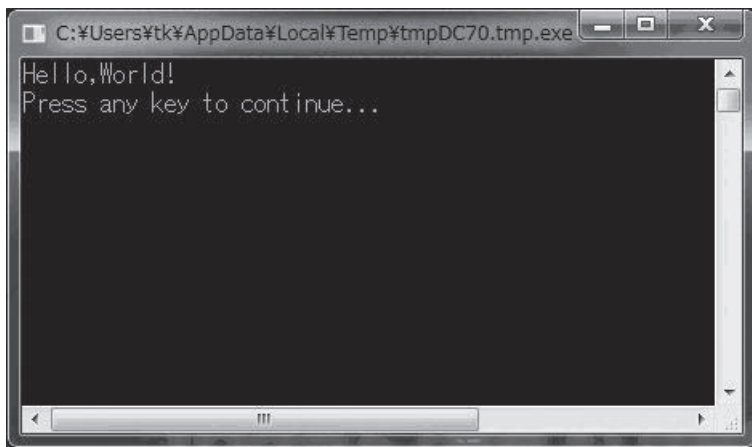
たったこれだけで、Hello,World を出力するプログラムが完成し、実行もできました。



(A) IntelliSense がはたらく



(B) このボタンを押して実行



(C) このようにコマンドプロンプト表示される

2. クラスとオペレーションとは

先ほど簡単なプログラム（Hello, World!）を書きましたが、このコードの意味を探っていきましょう。

「TextWindow」は、テキストの入出力をすることを表しています。

「.」（ピリオド）は、右側は左側に所属していますということを表しています。今回の例の場合「WriteLine」は「TextWindow」に所属しています。「WriteLine()」は（）の中を改行して出力することを表しています。この時文字列は「"」（ダブルクォーテーション・[Shift]+[2]）で囲みます。

アここで覚えてほしいのがオペレーションです。オペレーションとは操作の事で、他の言語では関数やメソッドと呼ばれています。（）の中の文字（引数といいます）を利用し、なにか操作をします。引数がない場合もあります。また、戻り値（オペレーションを行った結果の値）を持つものもあり、それを利用することもできます。クラスとはそのオペレーションをまとめているものです。今回の例では「WriteLine()」がオペレーション、「TextWindow」がクラスです。

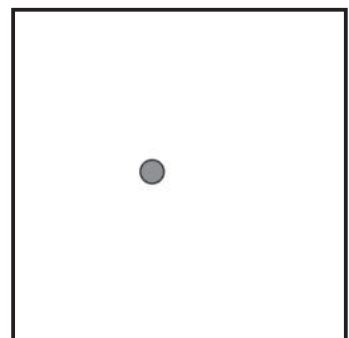
3. グラフィカルに作ってみよう

それではグラフィカルなプログラムを作ってみましょう。マウスを動かすと丸いボールがマウスについてくるプログラムです。コードと実行画面を以下に示します。

```
GraphicsWindow.Title = "Graphics"
GraphicsWindow.Height = 400
GraphicsWindow.Width = 400
GraphicsWindow.BackgroundColor = "white"
GraphicsWindow.PenColor = "#000000"
```

```
GraphicsWindow.MouseMove = MoveMouse
ball = Shapes.AddEllipse(30,30)

Sub MoveMouse
    Shapes.Move(ball, Mouse.MouseX, Mouse.MouseY)
EndSub
```



実行画面

➡ のようにわずか数行のプログラムで簡単にグラフィカル
なプログラムを作ることができます。

4. プログラムの説明

1 行目から 5 行目までがタイトルやサイズ、色の設定です。色は名前でもコードでも大丈夫です。なお、「=」は右から左への代入を表しています。例の 1 行目では、「GraphicsWindow.Title」に「Graphics」を代入しています。

7 行目はマウスが動いた時の設定です。マウスが動いている時に「MoveMouse」サブルーチンを実行します。

8 行目はボールを作っています。オペレーションである `Shapes.AddEllipse(30, 30)` の戻り値を `ball` に代入しています。

10 行目から 12 行目で `MoveMouse` サブルーチンを作成しています。サブルーチンはオペレーションと異なり、引数を持つことも戻り値を設定することもできません。

11 行目では `Shapes.Move()` オペレーションを用いてマウスの位置に `ball` を移動させています。

5. プログラムを Web で公開してみよう

発行ボタン (D) を押すと、インターネット上で公開することができます。発行が完了すると、ID が表示されます (E)。



(D) 発行ボタン



(E) 発行完了メッセージと ID

Small Basic プログラミングはいかがでしたでしょうか。このように Small Basic ではプログラムを書くのも、実行するのも、そして公開するのも非常に簡単に行うことができます。

この言語を基礎にしてプログラミング全般に興味を持ってもらえたらいいと思います。

筆者は以下の環境でこの文章を執筆しました。

Version: Small Basic v0.9 OS: Microsoft Windows 7 Professional (32bit)

CPU: Intel Core i5 750 Memory: 4.00GB



群馬高専隔離スレ

著：nega, bakin & taku

むしゃくしゃしてやった
反省はしている
だが後悔はしていない

bakin : で。

taku : お題はどうします？

bakin : なにすんだっけ？

taku : 語るんだろ？

nega : 適当なテーマ決めて適当に語るんだろ？

bakin : 正々堂々ネタで行こう

taku : アニメネタ？高専ネタ？その他？

bakin : 高専ネタとかありますか？

taku : 高専病とか

bakin : そうしよう

taku : まじでか

nega : 構わないんでね？

bakin : いくつかやってまともげなのにするべさ

taku : 高専病といったらなんだろうか

: 非オタがオタになるとか

bakin : オタが帰ってこれなくなったりとか

taku : 生物学的にメスならどうとかオスでもどうとか

: とりあえずほこりっぽくて鼻水でるとか

: 嫁ができるとか

: それが三次だったり二次だったり

bakin : 主に二次だけどな

taku : 三次は氏ねとかか

nega : 高専病とは、高等専門学校に在籍する学生の大多数が罹るといわれる病気である。

主な症状として以下が挙げられる。

大学入試試験の試練を乗り越える必要がないために、大学受験にしか使わない付け焼刃をこしらえるようなくならない受験勉強をする必要がない。

したがって勉学に対する意欲が減退してしまう。高等学校は中学校よりも授業が厳しいと世間一般では言われているが、高等専門学校は専門学校であり大学寄りなので低学年のうちはおろそかに中学校に比べ遥かにユルい(特に文系科目)ことも原因のひとつである。その分5年生のときは大学編入試験、卒業研究、就職試験と大忙しである。

・ストライクゾーンが広がる

前述の通り、高専は女子の比率が少ない為、大して可愛くもない女子であっても女子であるというだけで憧れの対象となってしまう。症状が進むと女子じゃなくてもいいと思うようになったりする。

とある高専では男子どうしが手を繋いだり抱きしめあったりする。

・自転車が大好きになる

高等専門学校は寮制を採用している学校が多いため、学生はある程度束縛される。そのため、自転車で家まで逃亡を図ったり、コンビニまで自転車で爆走したり、別に電車を使えばいいものをわざわざ自転車や原付で遠くまで行こうとする。

・やる気がなくなる

・オタクになる

元々オタク気質を持つものが多く集まる学校である為、いつの間にか本格的なオタクになってしまうことが多い。高専ではむしろオタクでない方がカッコ悪い、という認識が一般的である。

もともとは成績優秀スポーツ万能だった人も、在学しているうちにオタクの学生によってどんどん染められて……まあ、ロリコンはほとんどいないからその分ましではあるか？

仮面ライダー電王が放送終了した際、とある女子学生が嘆き悲しみ暴れまわった。

とある高専の近くのコンビニで、ローゼンメイデンが連載再開された号のヤングジャンプが発売後すぐに売り切れた。

文化祭でハレ晴レユカイを踊る。

雑談の話題は基本アニメかエロゲか下ネタである。

昼休みには遊戯王や MTG で闇デュエルが開催される。最近ではヴァイスシュヴァルツなどが流行っている。

低学年でも普通に教室に PC を持ち込むやからがいる。

女子なのにラブプラスをやっている奴もいる。

・BL になる

あまりにも男子が多く、寮生活など移動範囲が狭いため、普通の高校生に比べて異性との接触が極端に少ない。

そのため、同性に目覚める学生もいる。見た目では分からないがどこの高専にも必ず居る。

とある高専の寮では、実際に 5 年生が 1 年生を無差別に襲ったという事件が起きている。その寮ではその恐怖により、夜間に鍵をかける習慣がついた。

bakin : これどっかに載ってたの？ w

taku : アンサイクロですか

nega : ですね

(リンク省略)

taku : スカイクをニコニコしながらやってる

とか

bakin : ん？おれか

taku : お前もか

nega : 何だお前ら

taku : 高専病患者だよ

bakin : んだんだ

: >女子なのにラブプラスをやっている奴もいる。

nega : 凜子可愛いよ凜子

taku : nega じゃねえか w w w

nega : お前小早川さんはなあ！

: はにゃーんで可愛くて天使なんだよ！

taku : ローゼンメイデンについては共感するわ

bakin : この三人全員患者ってどうなんだよ w w

nega : 患者同士仲良くしようぜ

taku : 誰か、ここに病院を建てて

bakin : 病院は逃げて！！

: ついでに警察も逃げて！！

: ボカロ好きというかニコ厨も多いよな

nega : だって高専だからニコ厨は仕方ない

taku : ニコ厨だけどボカロはほとんど無知だよ

nega : ばっきばきにしてやんよ！

bakin : されてやんよ！！

taku : w w w w w w w w w

: だめだこりゃ w w w

bakin : ところで教室でゲームとかいつもやってるとさ

: ゲームやってるやつらはほんとくぜみた
いな空気にするのってどうなのよ

taku : デュエル大会は毎日だな

nega : 関わりたくないと思うんでね？

bakin : 相当な偏見だと思うんだよ

: たかがゲーマーじゃん w

: やってるのがエロゲじゃあるまいし w

taku : リア充>>>>>超えられない壁>
>>>>>患者ってこと？

bakin : リア充くたばれ！！

taku : エロゲの人だっているじゃないか

nega : 患者のリア充も居ますよ

taku : ああああああああああああああああ
あああ

: なんだよ！！何様だよ！！！！

bakin : とりあえずリア充は爆発しやがれええ
えええええええ！！

nega : リア充なんてええええええええええ
ええええええええ

taku : あ、俺たち……可哀想だな

nega : 今更何を

bakin : わかりきっていることをいうんじゃない
よw

taku : じゃあ部誌で嫁でも募集するか？き
つと痛いぜ

nega : 僕の嫁は凜子ちゃんただひとりでいい

taku : w w w w w w w w w w

bakin : 俺も長門だけでいい

nega : いやしかし二次元なら多妻でもいいん
じゃないかと思う

: ってことで朝倉ー！俺だー！結婚してく
れー！

bakin : しかし、長門はみんなの嫁だ

nega : みんなの嫁か

taku : こ、これが高専病なのか……

nega : 高専の歪みだよ

: 多分

bakin : ここでオチ！？

taku : お題高専病とは何か？ 答え長門はみ
んなの嫁

nega : 素晴らしいな

bakin : ふつくしい……

taku : せっかくだから最後にそれぞれの好き
なものをオヌヌメしないか？

nega : プリキュアあああああああああああ
ああ

taku : アリプロおおおおおおおおおおお
おお

nega : バトスピとトランスフォーマーも面白
いよおおおおおおおお

: もうだめかもしれないいいい

taku : アリカ様あああああああああああ
あ

: 殿おおおおおおおおお

: アリプロ友達募集中！！

bakin : お前らの嫁がどうかしたのか？

nega : 嫁かよおおおおおおおおお

taku : 嫁……画伯！俺だ！結婚してくれ！！

nega : bakin も何か好きなもの叫んだら？

bakin : そいそ————す！

nega : ……妹って、叫べよ

bakin : な、なんのこたあああああああ
あああああああああああああ

nega :

taku :

bakin : おまえら、どうしてもいわせたいの
か . . . ?

nega : う n

taku : やっちまえよ

: 我慢すんなよ

: 解き放てよ

bakin : なんか、酒進められてるみたい……

nega : ほら、最後まで言えよ

: 楽になれるぜ？

bakin : ミク————！

nega : 妹はあああああ？

bakin : だってキョンの妹だれかが嫁にしてた
し

nega : taku のことかああああああ！！

bakin : ちきしょうが！！妹萌えでなにがわる
い！！！！w w w

: こんなんでいいのか……？俺ら……orz

このスレッドにはもうかけません。
新しいスレッドを立ててください。



弱いオセロを作る過程を 赤裸々に記す。

著：jinkai

この文章にはアルゴリズム的な解説はほとんど含まれていない。なぜならそれらに関しては優れたページがネット上に沢山あり、私などが解説したところで読者を混乱させるのが関の山であると思うからである。また、「オセロ」は登録商標であり、以下より一般的に「リバーシ」と記すことにする。

ちなみに、さっとみればわかるが、図なんか用意していない。読みやすさなんてどこかに行ってしまった。

1. 弱いリバーシとは

弱いリバーシとは、通常のルールでは勝ちを目指すべきなのにあえて負けようと試みるリバーシ AI のこと。一般にゲームの楽しみ方は人それぞれ（制限プレイを考えて欲しい）であり、リバーシで負けようとしてみてもよい。しかし、対人戦では力を抜くことは失礼にあたる。そこでリバーシ AI に負けようとしてみる。しかししかし、通常のリバーシ AI は勝つことを目的にしているので、負けることは容易なのだ。そこで負けることにゲーム性を見出すため、弱いリバーシが考案される。

昔からリバーシを弱くしようとする活動はあった。某大学の某サークル主催の最弱オセロリーグなるものが開催されたこともあるし、あるゲーム（名は失念）ではあるキャラにリバーシで負けないと次のイベントに進まないなんてこともある。しかもすごぶる弱いらしい。

2. リバーシ AI において「弱い」とは

リバーシ AI を作る上で基本的な考え方はミニマックス法である。これは自分の手番において「お互いに最善の手を打つと仮定したとき、数手先の盤面が自分にとって最もよくなるように次の手を決定する」方法である。あきらかに、「お互いに最悪の手を打つと仮定したとき、数手先の盤面が自分にとって最も悪くなるように次の手を決定する」ようにすれば弱くなるだろう。

論理的に手、盤面の評価を反転すれば、アルゴリズムは通常の AI と同じで良いことがわかる。この考え方において「最善の手」「最悪の手」「最もよく」「最も悪く」という部分をどう定義するかによってどれくらい弱くなるかが決定される。このような基本的な考え方に関しては、私は以下のページを参考にした。

「オセロプログラムの作り方」 <http://hp.vector.co.jp/authors/VA015468/platina/algo/>

3. 私が作った弱いリバーシ（前）

さて、私が作成した弱い (?) リバーシについて説明する。

私がこのリバーシを作る上で目標としたことは、「一般の人間に十分実現可能な弱さ」である。なぜなら、私の目標は「私がこのリバーシに負けること」だからだ。しかし、わざと強くしたのは

意味がない。以上より、私はいくらかの制約をつけた。それは、

- ①突っ込んだ理論は用いない（リバーシをするときそんなことを考えるのは一般の人間ではない）
- ②定石は考慮しない（定石はイーブンで進むことが前提で、負けることに有益であるとは思わない）
- ③先読みは4手を最高とする（私は手を読むのが苦手だ。幸いにもこれは計算量の削減が図れる）

の3項目である。

「突っ込んだ理論」についてだが、一般的なコンピュータリバーシは様々な理論を考慮した計算が行われる。これについても多くの解説ページがある。たとえば、

「- 鶯教 - コンピュータ・リバーシ講座」 <http://uguisu.skr.jp/othello/index.html>

などが分かりやすいだろうと思う。

また、我が電算部は主な活動としてプログラミングを掲げる部活だから、プログラムのにも新たなことにも挑戦した。それは、リバーシの盤面をビットボードで表すことである。これに関しては、参考文献を載せるに留める。大変わかり易く、私のリバーシの大半はこれらのページからのコピペ・改悪で出来ている。

「RunRunDietOnline」 <http://vivi.dyndns.org/rrdo/rrdo.php>

「オセロの試合結果は何通りか？」 <http://www9.atwiki.jp/othello/>

ビットボードでリバーシの盤面を表現することの利点を簡単に上げておくと、

- ビットに対する演算で各種操作を定義することで、各種操作の速度向上が望める
- 盤面を表現するメモリ領域が二次元配列に比べ小さくてすむ
- 各処理が楽になることが多い

の3つだ。

最後の項目についてもっとも効果的なのは、石を置く処理だと私は思う。石を置く処理は排他的論理和で表現され、これにより、1度置いたところに同じように石を置く処理をすると置く前の盤面に戻すことができる。これは先読みを行うとき非常に便利である。

また、ビットボードは有効に使わなければ逆に遅くなることがある。通常の配列のように使いたくインデックスを求めようとしたりしては遅くなることは容易に想像できるだろう。

4. 私が作った弱いリバーシ（後）

私が作成したリバーシ AI は先述したミニマックス法を弱い方向に適用したアルゴリズムを用いている。より正確には α - β 法を弱い方向に適用している。また、盤面の値をキーとして、一度評価した盤面を再度評価することを省くためのテーブルを用意している。計算量削減のためであるが、これを実行するために必要な計算量も相当ある気がするのはいのちのせいだろうか。

ちなみに α - β 法の適用において、数手先までの盤面を生成するがそれぞれを別に作るわけではなく、石をおいて、元に戻して、という繰り返しなので、その点ではメモリの領域はとらない。しかし、 α - β 法は再帰で記述されるため、スタックオーバーフローにわりとすぐに陥る。その点では、同じ盤面について再評価しないのは効果があると言える。実行時間は犠牲になったのだ。

さて、リバーシ AI においてメインともいえる、盤面評価の関数について説明しよう。私が今回採用した評価要素は、各色のある盤面において、

- その盤上にある各石の数
- 相手の一手でひっくり返されない石の数

●盤上にふったポイントの総計

この三つの評価要素であるが、簡単にそれぞれにある係数をかけて足したものを評価とする。その係数は、局の前半、後半で別に定義される。この係数は一般に適当な値がわからない。私が、この係数を決定した方法は、

- ①ランダムに決定した係数の評価関数2つをそれぞれ用いたAIを用意する。
- ②それを対局させ、負けた方のAIを残し、新たにそのAIを元に微妙に変化させたAIを用意する。
- ③②を繰り返す。

AIの弱さが関数で記述できるとするなら、それは変数として係数を持ち、連続に変化するとすれば、

以 上の手順でローカルな極弱値を得る。また、ときどき完全ランダムな係数を発生させれば大域的に極弱値を得ると考えた。

ちなみに以上の理論は完全な後付であり、厳密さ、確証など微塵もないことを留意されたい。AIどうしの相性など考えてもいない。

5. これが私のリバーシだ！

以 上の考えにより作成したリバーシ。果たして、その弱さとは……！

(編注：新版・旧版ともに付録CDに収録してあります)

え、弱く……ない……？

6. おわりに

さて、弱いリバーシを目指して作ったものはあまり弱くなかったというエンディングですが、実は私はこのリバーシに負けることができません。どんなにやっても勝ってしまいます。私の読みゲーの弱さは折り紙つきであり、これはもう仕方ないのかもしれませんが。

結果だけ見ても、なんてことのないリバーシプログラムです。この駄文を読んだ後、感慨深くプレイするとより一層楽しめると思います。バグがあったら報告してくれると嬉しいです。でも多分バグフィックスはしません。

ソースは公開しますが、読まない方がいいでしょう。個人的な学習のため、あえて冗長なデータの表現を使っています。おぶじえくとしこう？なにそれ。コメントなんて忘れてましたし、どこが間違ってるかわかりませんよ。

それでは、またお目にかかることを楽しみにしています。読んでいただきありがとうございました。



Befunge

著：tk

みなさんはプログラミング言語というものを知っていますか？
プログラミング言語とはパソコンを低級（言語的表現）なレベルで好き勝手に遊ぶために覚えなければいけない処理体系のことです。普通の人は覚える必要はありません。覚えるだけなら大丈夫ですが、この道に進むと大きな犠牲を払うことになります。

1. Befunge とは

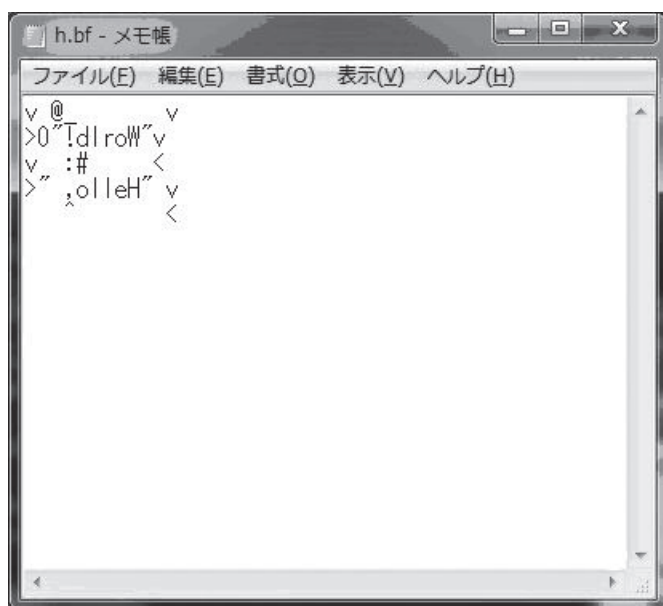
非常にやさしい初心者向けの言語のことです。わかりやすいスタック型です。プログラムサイズが固定されており、スパゲッティや力技が自動的に廃止されるという素晴らしい仕様です。実行の向きが上下左右に変化する面白い体系をしています。作者とか詳しいこととか知りません。

2. インストール

するまでもありません。
言語仕様が単純なので自分で作りましょう。

3. 開発画面

この言語の需要は非常に大きいため、開発環境の一部は Windows に標準搭載されています。



開発画面

4. 実行してみよう

ここでは世界で一番有名なプログラムである「Hello, World!」を出力させてみましょう。

(A) もしくは (B) に示す通りに、スペース、改行を入れて以下の文字を入力してください。
このように同じ処理でも違うコードで書くことができ、個性を表現することができます。

```
^@  
>0"l d l r o W"  
^: #  
>" x o l l e H"  
^
```

(A) コード例 1

```
>^"Hello world!"0<  
^:_25*,@
```

(B) コード例 2



実行画面

5. プログラムの説明

^v<> で上下左右に変化させています。見ていて非常に分かりやすいですね。

!や_はスタックの状態で上下左右への変化が変わります。他にもいろいろ説明しなければいけませんが、割愛させていただきます。疲れたので。

6. あとがき

➤のように Befunge ではプログラムを書くのも、実行するのも簡単に行うことができます。

⌋また、従来の言語と異なり、上下左右から読むことができ、可読性が向上していることがわかります。

この言語を基礎にしてプログラミング全般に興味をもってもらえたらいいなと思います。

筆者は以下の環境でこの文章を執筆しました。

開発環境：メモ帳 + コマンドプロンプト

開発機械：DELL の 10 万くらいするやつ

開発場所：研究室片隅



※注意

この漫画は偏見によって描かれた
ジョーク漫画です。あまり真に受けず
ありすの可愛さを楽しんでください。



う~~~~~ん

きょーじょって

こういうのが好きなんですか？

短剣印使、さる……

「こういうの」とは？

もっと具体的に
言うとどんな感じ？

え!?

なんていうか……

痛い？

中二病というか……

ニヤリ

中二病って何だ？

どんな症状だ？

何か問題はあるのか？

それじゃあ

うう

正直、よくわかりません

なんとなく見ているモヤモヤするので
中二病って言えばいいのかなって…
皆そう言って馬鹿にしてるし……

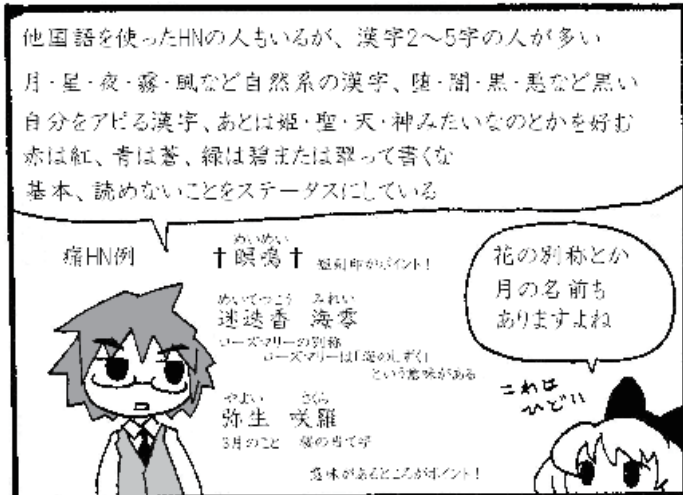
9F

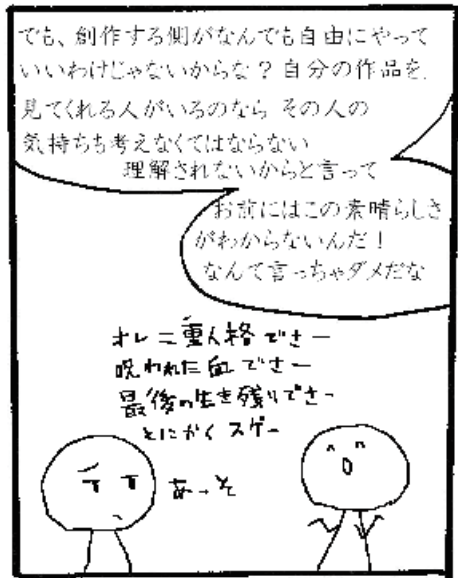
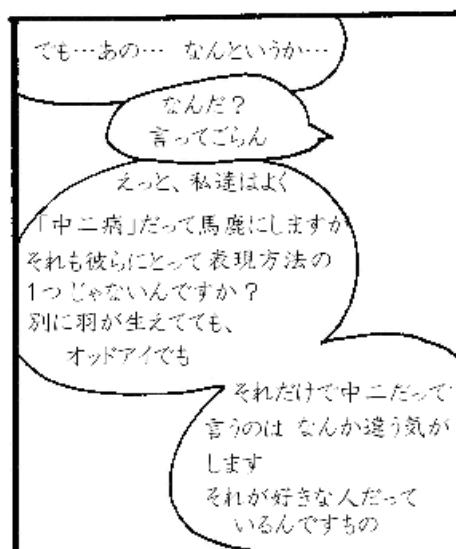
なんとなくモヤモヤする、
皆も馬鹿にしてるからって…
「痛い」って言われた人は
たまったもんじゃないな！

ごめんなさいー

僕もまだ研究中だが
一緒に中二病について
考えてみようか

お願いします！





あとがき



最初は、きょーじゅとじょしゅのはのぼのハートフル漫画にしようと思ったけど、なんかネタに走りました
支離滅裂な内容で、ごめんなさい。あと原稿出すの遅れてごめんなさい><

この漫画で言いたかったことは、中二病患者も「中二病バロス」って言う人も「調子にのっちゃダメー」
ってことです。なんでもほどほどがいいんだよ。

でも、「貴方は●●番目の片翼の天使です」なんて言われちゃたまったもんじゃないよね(暗黒微笑)

見てくれてありがとうございました！

またいつかお会いしましょう！



表紙イラスト：ますたー 裏表紙イラスト：EXILIAS
装丁・DTP： akina

部誌「D 言語」・2010 関東信越地区高専文化発表会号

発行日 平成 22 年 9 月 16 日 初版

発行者 群馬高専電算部

発行所 群馬工業高等専門学校

〒 371-8530 群馬県前橋市鳥羽町 580



これが最もくだらないアプリです。

サシミプラス

S A S H I M I P L U S



単純だからこそ奥深さがあります。

刺身の上にタンポポを乗せる。ただそれだけの単純な作業。

ですが、この非常にシンプルな作業の中に奥深さがあるのです。

刺身とタンポポの相性、全体の色合い、乗せる位置。

また、選択するタンポポの種類も重要です。食用の菊、プラスチック菊、はたまた本物のタンポポ……。挙げたらキリがないほど多種多様な「タンポポ」があるのです。

タンポポ乗せ職人はそれらを一瞬で判断し、刺身を彩ります。それはもう、美しく美しく。

さあ、あなたも刺身にタンポポをプラスしてみませんか。

いつでもどこでも、サシミカメラ。

サシミカメラを使うことによって、カメラまたはフォトライブラリーからあなたのお気に入りの写真を選択し、その写真の好きな位置にタンポポを乗せ、そしてその合成画像をフォトライブラリーに保存することができます。

スーパーの鮮魚コーナーでサシミにタンポポが乗っていなかったら、無性に「ああ、タンポポを乗せたい!」と思うはずです。

そんな時こそサシミカメラ。好きなときに、好きなものに、好きな場所へタンポポを乗せることができます。

そう、サシミカメラならね。



@exilias (twitter)

<http://exilias.net/>