ゼロからゲームプログラミング

平山　尚

私は何者か

- セガ社員８年。
- 元遺伝子工学。研究から脱落してゲーム屋。

関わったゲーム

電脳戦機バーチャロン
　マーズ

・ボスの攻撃いくつか
・前作の移植部分
・PS2アセンブラコード
・デモシーン再生装置

関わったゲーム

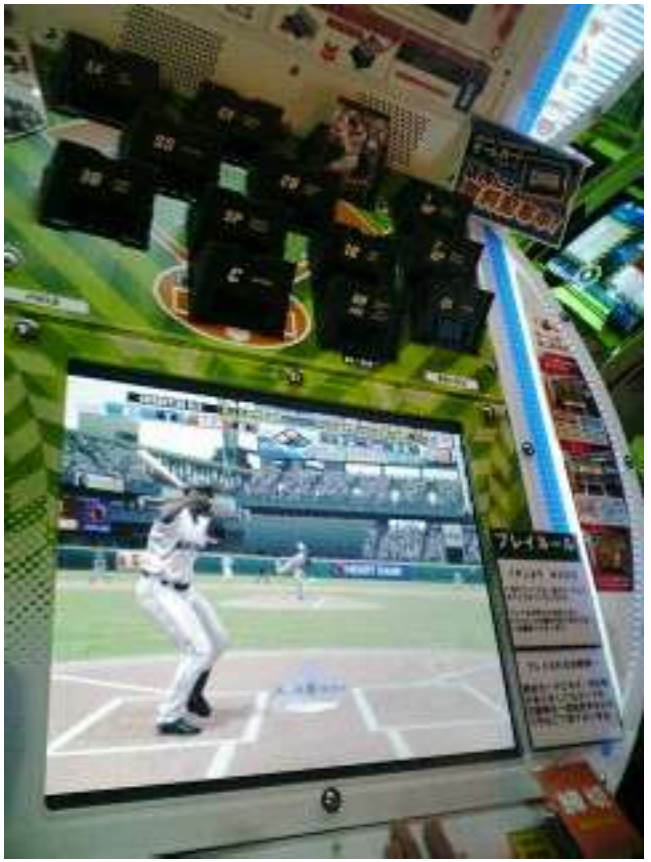
- PowerSmash3

- グラフィックス
- ハード叩き
- カメラ制御一部

関わったゲーム

- MLB CardGen

- グラフィックス
- ハードウェア抽象化



本書いてみたり

- この本に書いてあることができれば、だいたいゲーム会社でプログラマをやっても大丈夫だと思います。

この授業の主旨

- プログラミングを**がっつり**学んでもらいます。

何故か？

- 皆さんがプログラマになりたいわけでもないらしい、と言うことは聞いてます。

•だが、敢えてやる。

何故か？(2)

- ゲームに限らず（ゲームならなおさら）、工学部を出てつくような仕事で、コンピュータを使わない仕事はもうほとんどありません。
- コンピュータを使いこなすには、コンピュータが何者かを知るべきです。
- コンピュータが何者なのかを知るのに一番費用対効果が安い手段は、プログラミングです。

何故か？（3）

- ゲームに関して。

- 最近は絵描き（モデラー、アニメーター）もプログラミングが出来ると良いです。
- 最近は企画屋（プランナー、ゲームデザイナー）もプログラミングが出来ると効率が良いです。

何故か？（４）

- とりわけゲームデザイナ！

- 考えてるゲームはコンピュータで表現可能？
- 考えてることがプログラマに正しく伝わる？
- そもそも自分でプログラムした方が早いよ？

- 「スクリプト」なんて呼ばれます。

プログラミングをどう学ぶか？

- どんな言語であれ、自分で設計して自分で作っているなら、私が言うことは何もない。

- が、「授業でやった。以上」みたいな人はおそらくプログラミングのイメージがまだ頭にないと思われるので、今回の授業をうまく利用してほしい。

今回の授業

- 専用言語でゲームを作る。

早速デモ

- こんな感じのものを作れたらオーケー。
  - テトリス以外に作りたいものがあればそれで。

専用言語？！

- 既存の言語は覚えることが多すぎる。
- 既存の言語はできることが多すぎる。
- 既存の言語だとすでにやってる奴が強い。

- そこで、全員同じスタートラインに立てるよう、覚えることが恐ろしく少ない言語を作った。
  - ググっても無駄。

専用言語の特徴

- メモ帳と実行プログラムだけあればいい。
- 簡単に画面に何かを描ける。
- 文法は1時間で覚えられる(人によっては10分)

- 基本的に、何も用意しない。
  - ex.乱数？printf？作れば？

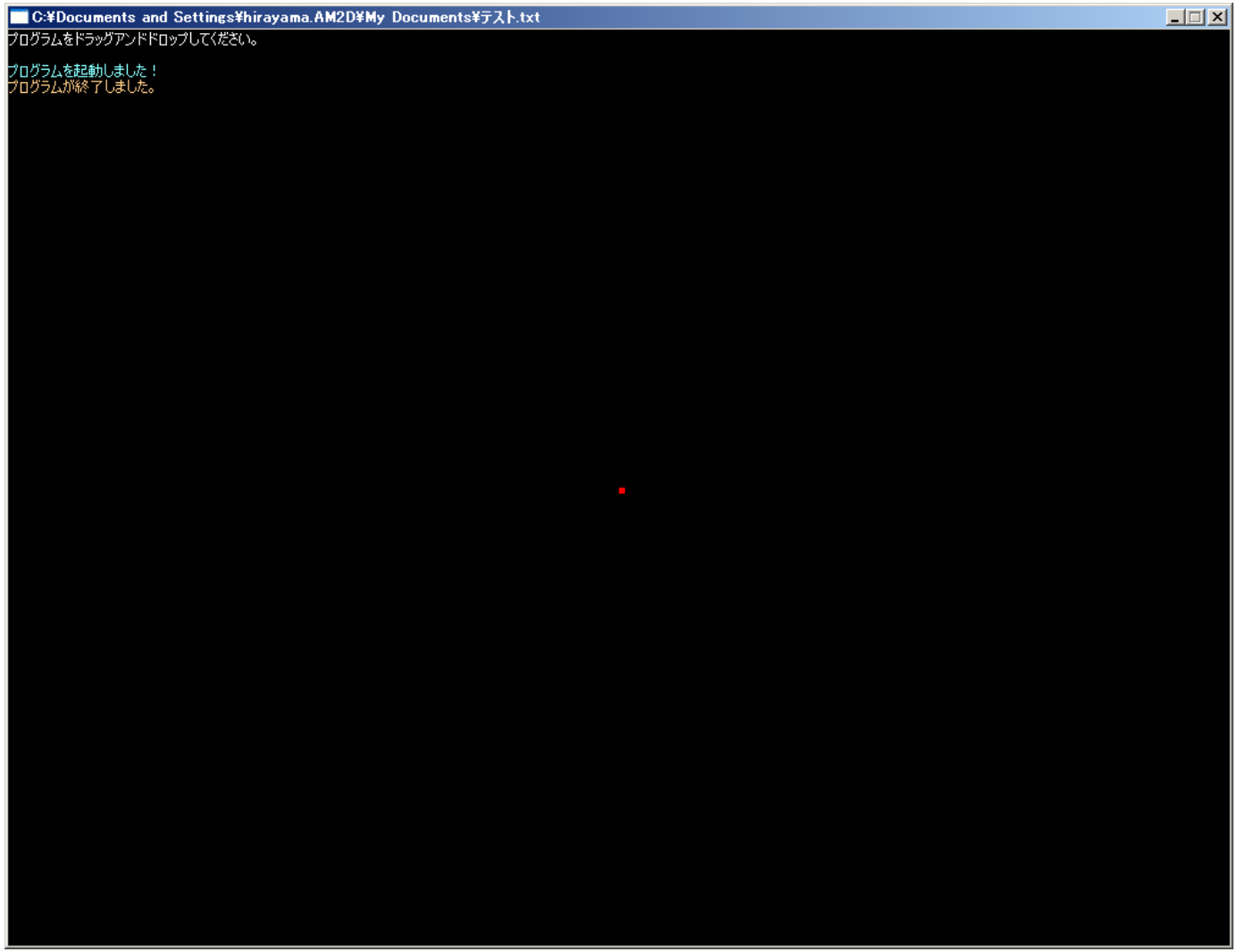
専用言語の特徴(2)

- 若干C言語に似た見た目
- 日本語を結構使える
- コンピュータに歩み寄ろう
  - クラス？オブジェクト？それは貴方の心の中にあればいい。コンピュータに見えるのは01です。
- 完成度が低いように見えるが、わざとだ。
  - バグは私のせいなので、直しますが。

とりあえず動かそう

- zipをD:¥で展開
  - bin¥SunabaPlayer.exeがあればオーケー。
- メモ帳を開く
  - 決まったエディタがある人はそれで。
- 以下のように書いて保存
  memory[75100]=990000;
- SunabaPlayer.exeを起動。
- 書いたテキストをドラッグアンドドロップ。

初プログラムおめでとう！



意味は？

- memory[75100]=990000;
- memory[x]で、「メモリのx番」
- =は代入。左辺の値が右辺になる。
- 右辺は数値。

- つまり、これは「メモリの75100番を990000にする」という意味。

何故代入で絵が出る？

- メモリの75100番は画面のある点の色を格納している。
- そこの値が変わると画面の色が変わるようにコンピュータとディスプレイが配線されている。

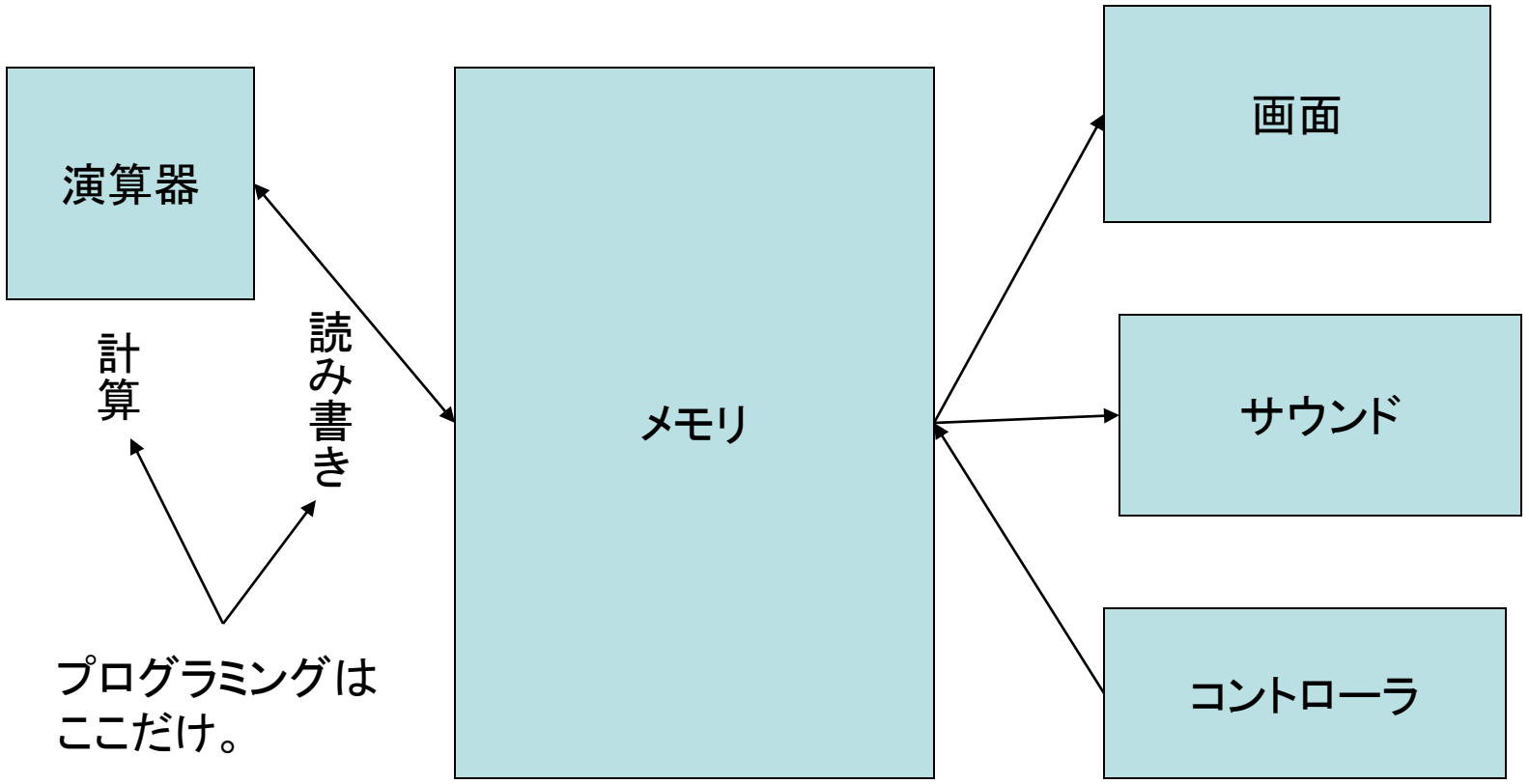
コンピュータは全部これ。

- コンピュータにできることは、「メモリの値を変えること」だけ。
- プリンタもネットワークも画面も、メモリのどこかに配線がつながっていて、メモリの値によって動きが変わる。

ゲーム機も一緒。

- 例えばプレステ2みたいな機械も一緒。
- メモリのx番にポリゴンの色を書き込んで、メモリのy番にポリゴンの座標を書き込んで、メモリのz番に1を入れるとポリゴンを描き始める…のような状態。
- メモリの何番にどういう順番で何を入れるかを制御すること＝プログラミング。

これがコンピュータだ！



これが一番言いたい。

- 正直、これがちゃんと伝わればこの授業の役割は十分果たされたと思ってます。
- 「もうわかったからいいじゃん」と言うかもしれないが、体でわからないと意味がない。

- 何故かみんなこういう説明をしない。だからコンピュータが難しそうに見えてしまう。

というわけで、体でわかろう。

- とりあえず、プログラミングに最低限必要な言語要素を紹介します。しばらくこれでいじってみてください。

- zipに入っているSunabaQuickReference.pdfを参照。
- book.pdfの付録にもう少し正確な仕様があります。

最小言語要素

- 分岐: if(式){ ... }
  - 式が0でないなら{}を実行
- 反復: while(式){ ... }
  - 式が0でない間{...}を反復実行
- 代入(左辺=式;)
- 演算子(+,-,*,/,<,>,!=,==)
- これだけあれば原理上なんでも出来ます。
- あと、コメントは「//」の後ろ行末まで。

メモリの番号

- 0-9999 : 使えません。
- 10000-39999 : 自由に使えます
  - 正確じゃないけど。まあ今はこれで。
- 40000-49999 : 使えません。
- 50000-89999 : 機械につながってます

メモリに何が入る？

- 1つのメモリにはプラスマイナス10億くらいの数値が入ります。

- ----------以下玄人向けに----------
- 符号付き30bitです。
- この言語では1バイト=32bitと考えてください。

機械につながったメモリ

- 50000 : マウスのX座標が入ってます
- 50001 : マウスのY座標が入ってます
- 50002 : マウスの左ボタンが押されてれば1、押されてなければ0
- 50003 : マウスの右ボタンが押されてれば1、押されてなければ0
- 50005 : コンピュータと機械の通信。ここに1を入れた瞬間に画面が更新されます

機械につながったメモリ

- 60000-89999 : 画面の色。
- 幅200、高さ150。左上から右へ行って、端まで行ったら1段下へ。

y=	x = 0	1		198	199
0	60000	60001	.....	60198	60199
1	60200	60201	.....	60398	60399
	.	.		.	.
149	89800	89801	.....	89998	89999

画面に渡す色

- 0-999999の6桁。
- 「赤赤緑緑青青」
- それぞれ0-99。
- だいたい%

000000	黒
990000	赤
009900	緑
000099	青
999900	黄（赤+緑)
009999	空色(緑+青)
990099	紫(赤+青)
999999	白(赤+緑+青)

演算子

- 二つの数字から一つの数字を作るもの。
- + : 左と右の和を作る
- - : 左と右の差を作る
- * : 左と右の積を作る
- / : 左と右の商を作る
- < : 左が右より小さければ1,でなければ0
- > : 左が右より大きければ1,でなければ0
- ==: 左と右が同じなら1,でなければ0
- != : 左が右と違えば1,でなければ0

演算子の優先順位

- 演算子を区別しない。左から計算。
- 4+5*6→(4+5)*6→54
- 自分で括弧をつけよう！
  - 4+(5*6)→4+30→34

演算子の優先順位

- <や>も区別しない。
- 1<2>3!=4==5
- (1<2)>3!=4==5
- (1>3)!=4==5
- (0!=4)==5
- (1==5)
- 0

以上を使った例

```
while ( memory[ 50003 ] == 0 ){ //右ボタン
  if ( memory[ 50002 ] == 1 ){ //左ボタン
    memory[
      60000 +  //画面の始まり
      memory[ 50000 ] + //マウスX
      ( memory[ 50001 ] * 200 ) ] = 990000; //マウスY
  }
  memory[ 50005 ] = 1; //機械と通信
}
```

簡単なお絵かきソフト。

- 前のページのプログラムを作ってみましょう。
- マウス左ボタンを押している間、カーソルがある場所に赤い点が打てます。
- 右ボタンを押すと終わります。

改造！

- 速く動かすと途切れる問題。
- 画面の範囲外に出ると異常終了問題。
- その他機能。
  - 色変えられたら楽しいよね。
  - 右ボタンで終了しなくてもいいです。
- 「お絵かき.txt」という名前でよろしく。提出してもらいます。

知っていると便利な機能

- 変数
- 番号アクセス
- 部品（いわゆる関数）
- デバグ用文字出力

変数

- メモリのある番号に別名をつけられ、これを変数と呼びます。
- 回数=35;
- のように代入文の左に書いたものが変数として作られます。

- 使わなくてもプログラミングできますが、便利です。

番号アクセス

- 変数名[式]と書くと、
- memory[変数の値 + 式]と解釈します。

- a = 10000; a[1] = 5;　は、
- memory[10000 + 1] = 5;に解釈されます。

- メモリの特定の部分に別名をつけるのに。
  - 画面=60000; 画面[20] = 999999; //便利。

部品（関数）

```
余りを出す( a, b ){
   return ( a - ( ( a / b ) * b );
}

答え = 余りを出す( 16, 5 );
```

と書くと、「答え」は1になります。

デバグ用文字出力

- memory[50012]に文字の番号を入れると、画面に文字が出ます(F1で表示切り替え)。
- 番号は、Unicodeです。
  - 0-9の数値なら48を足すとそれが出ます。
  - memory[50012] = a + 48;
    - aが0-9なら'0'から'9'が出ます。
- これなしでも、画面に点を出せればデバグ可
  - 特定条件で青い点を出すとか、まあいろいろ。

実はもうゲーム作れます。

- 基本的な道具はそろってます。
- ゲームは無限ループです。
  - 入力、状態更新、出力、でループし続けます。
  - お絵かきソフトにはこの三つが揃っています。
- 好き放題やってください。

お絵かきに飽きたら

- 倉庫番なんかいいかも。知ってます？
- テトリスが作れそうならがんばって。
- 「ルプさらだ」が作れたら即欲しい。

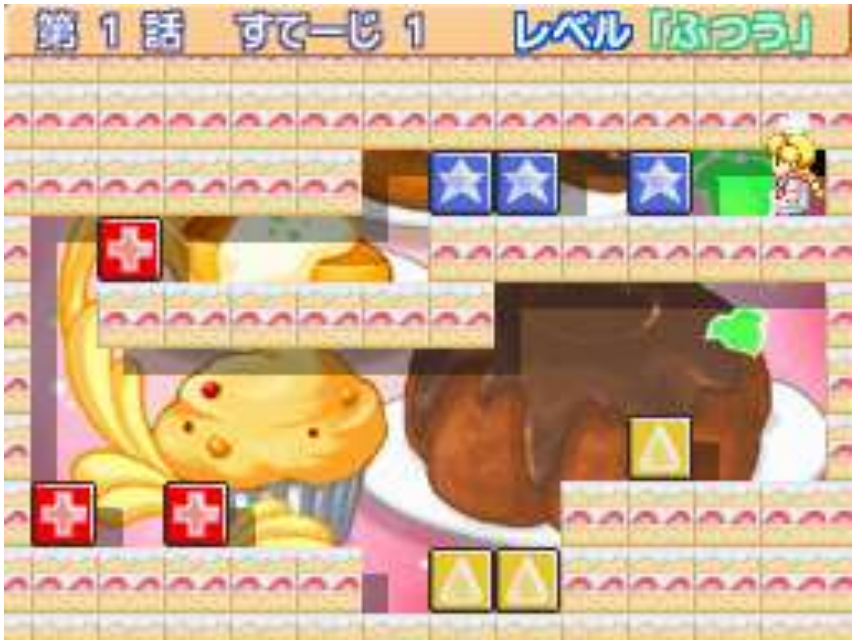
- 3回の終わりにテトリスが出来てたらいいな。
  - 出来たらポテンシャルとしてはプログラマ。

提出

- 今日書いたものは、とりあえずください。
  - 名前でフォルダを作って中に入れてください。
  - お絵かき.txtと、他に何かあれば。
- 面白いものは次回取り上げます。
  - 面白い間違い方なども含めて…
- 動かなくても気にしないように。動く駄目なコードより、動かない良いコード。
- 最終回は全員発表する予定。

ルプさらだ

- 初級ゲームプログラマにうってつけの課題
  - 新人研修で作らせたこともあります。
- 作るの、結構難しいです。



終わり

- バグ報告、質問はhirasho0@gmail.com

- 最新のSunabaPlayerは
- http://www.page.sannet.ne.jp/hirasho/
- インストーラが入ってます。家でも使ってみてね!
- doc/book/book.pdfは独習用のテキスト。でも途中。

おまけ

- 以下おまけの話。

ゼロから数える

- なんでゼロから数えるの？
- 15歳は15年生きてます。
- 100年1月-紀元前100年1月=199年間。
  - 紀元0年があればこんなことにならなかった！

- 0から数えると、「距離」が簡単に出ます。
- その代わり、最後がn-1になります。
- トレードオフです。多くの言語は0からですね。

字下げ

- ifとwhileの中身は字下げすることをお勧めします。
- 字下げされてないコードがどれだけ読みにくいかをお見せしましょう。

変数の使い方

- 変数といっても、計算に使うものばかりを入れるとは限りません。
- 色 = 0; //黒
- if (色番号 == 0){
- 　色 = 000099; //0番なら青
- }
- if (色番号 == 1){
- 　色 = 990000; //1番なら赤
- }
- if (色番号 == 2){
- 　色 = 009900; //2番なら緑
- }

プロへの壁

- 倉庫番は200行くらいで書けます。
- テトリスは500行くらいで書けます。
- ----------超えられない壁----------
- PowerSmash3は60万行
- バーチャロンマーズは80万行
- 200万行を超えてるゲームもあるなあ。
- Windows2000は数千万行だとか。

どうすれば大きくできる？

- ここで初めてオブジェクトやらクラスが生きてくる。
  - 私一人で毎回1-2万行は書いてます。少ない方。

- あと、人と話す能力ね…
- ゲームデザイナがプログラミングを理解してないと、大きいプロジェクトでひどいことになります。