

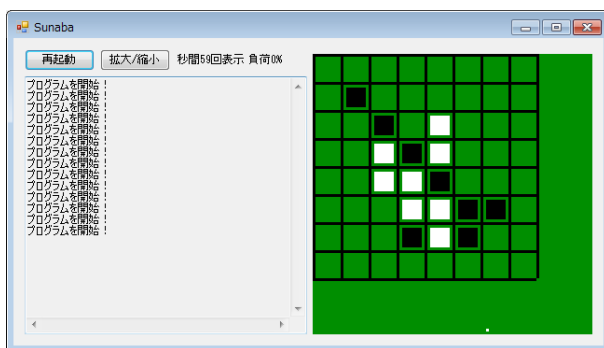
Sunaba 入門

リバーシができるまで

2014 年 4 月 9 日 0 時 9 分 (GMT) バージョン

この本の目的

まずはこの写真を見てほしい。



白と黒の石を取り合うあのゲームの画面だ。^{*1}。

この本は、このゲームを作り上げる経験を通して、コンピュータプログラム (以下「プログラム」) を作り上げる方法を学び、理解し、身につけるためのものである。

ただし、この本はプログラミングの入門書ではない。プログラミングを学校の授業等で多少なりとも学んだ人を対象としており、また if や while といった単語を見て意味がおぼろげに想像できる程度に英語に親しんでいることも前提としている。if やら while やらが散りばめられた何かしらのプログラミング言語で、何かしらのものを書かされた経験はあるが、細かいところ

^{*1} 諸般の事情により敢えて名前は言わない。

は理解していないし、自分でゼロから作ることもできない。そういう人が対象だ。つまり、「やったことはあるが、到底これで金を稼げるとは思えない」というレベルの人である。

さて、私はすでに似たような本として「プログラムはこうして作られる～プログラマの頭の中をのぞいてみよう～」を出版している。そちらを読んでいただけるのであれば、こちらの本は必要ない。というのも、そちらは完全な未経験者を対象としており、英単語も出てこず日本語で書けるし、説明もずっと親切で、かつ最終的により難しいプログラムを書けるようになるからである。

しかし、「プログラムはこうして作られる」は 500 ページ近くあり、説明もくどいという欠点がある。君がある程度勉強ができる高校生、あるいは大学生であるならば、あれほどの親切さはかえって邪魔になるだろう。また、授業で使う場合、細かい所は教師に聞けるのだからそこまで詳しい説明は必要ないし、授業の時間は限られており、あんなものを読んでいられるほどの時間はない。

そして本を薄くするために、目的とするゲームはより簡単に作れるものとした。「プログラムはこうして作られる」では動きのあるゲーム、つまり放っておいても画面が動くゲームを題材としたが、動きをプログラムで作るのにはかなりの慣れが必要になる。そこで、二人が交互に石を置くボードゲームを題材とすることにした。この結果、「プログラムはこうして作られる」での題材であるテトリスに似たゲームでは 180 行のプログラムを書かねばならなかったのが、今回は 140 行程度になっている。中もずっと簡単である。

0.1 独白言語

この本では、主に二つの理由から、独自の言語を使って説明をする。

- 特定の言語の知識を前提としないため
- 調べられなくして、全て自力で作る他ないようにするため

特定の言語を前提とすれば、その言語を知らない人は読めなくなる。これではスタートラインを揃えられない。また、どうしても言語特有のことにも触れねばならなくなり、私が学んでほしいと思うことでないことにページを割かねばならなくなる。だから、忘れても惜しくないくらいに簡単に覚えられる言語を別に用意することにした。文法は見開き 2 ページで足りるくらいしかない。満足する程度に動くものが作れたら、さっさと実用的な言語に移ってこんな言語のことは忘れてしまえば良いだろう。

そしてもう一つのより重要な理由は、調べるられなくするためである。今はわからないことがあれば気軽に検索できる時代だ。「こういうプログラムを書きたい」と思った時には、まず検索してやり方が出てこないか調べるのが当たり前の時代である。しかし、これではゼロから組み上げる力はない。独自言語にしてしまえば、調べても何も出てこないため、そういう危険はなくなる。もちろん、この独自言語が流行れば web 上にも情報が出てくるかもしれないが、決して流行らないように使い物にならない言語にしてある。「調べたらもっといい方法が見つかるんじゃないか」などと気をもむ必要はないので、安心して自分で考えてほしい。

なお、普通の言語には「標準ライブラリ」なるものがあって、よくやりそうな処理は簡単に書けるようになっているが、この言語にはそのようなものはない。ライブラリのマニュアルを読んで欲しい機能を探すのに時間を使う必要はない。

0.2 動作するコンピュータ

独自言語を動かすには windows か mac のコンピュータが必要だ。windows は XP 以降、mac は 10.9 以降が必要である^{*2}。

^{*2} より古いコンピュータに関しては、動くかどうかがわからない。動かないとは言わないが、試していないのでわからないのである。特に mac は 10.8 でも動きそうな作り方はしている。ただ、確認はしていない。

0.3 前提知識

この本は、何かしらプログラミングをやったことがある人を対象としている。プログラムを書くということは、何かしらのテキストファイルを書いて、それを何かしらのプログラムに処理させて実行させるものである、ということを知っていてほしい。

また、if という英単語が「もし」という意味であることや、「while」という英単語が「何々な間」という意味であることは知っていてほしい。文章が読めたり書けたりする必要はないが、英語の単語を見て怖がるようだと、そこで引っかかって理解を妨げてしまうからだ。ただし、プログラムで良く使う英単語は偏っていて、普通の英語ではあまり出てこないのに、出てくる単語については全て脚注などで補足することにする。

独自言語は日本語でも動くのでそれでもいいのだが、授業の場に留学生がいるケースもあるし、他の言語は英語ベースなので、何かしらの言語をやったことがあるならばそこが障害になることもそうないだろう。だからこの本では英語風文法で説明している。

あとは、Windows あるいは mac の基本的な使い方は前提とする。プログラムを起動する手順や、ドラッグアンドドロップやダブルクリックなどの操作、それにファイルやフォルダについての理解も前提とする。

目次

この本の目的	iii
0.1 独自言語	iv
0.2 動作するコンピュータ	v
0.3 前提知識	vi
第 1 章 準備	1
1.1 独自言語が動く状態にする	1
1.1.1 お手本を動かしてみる	2
1.1.2 Sunaba の見掛け	3
索引	7

第 1 章

準備

この章では、プログラムを書く前の準備をする。面白くはないが、難しくもないので、さっさと片づけてしまうことにしよう。

まずは、最初に写真を示した「あのゲーム」を君のコンピュータで動かしてみることにする。

1.1 独自言語が動く状態にする

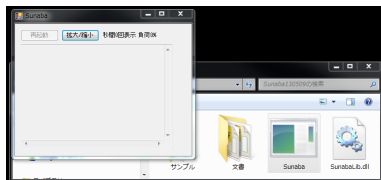
まずやることは、独自言語のプログラムを動かすプログラムをコンピュータに入れることだ。

この独自言語は、名前を Sunaba(砂場) と言う。名前の意味は、そのまま「砂場」だ。実用になるものは何一つ作れない。転んでも怪我をしない。砂しかないので想像力なしには何もできない。子供の一時期だけ夢中になり、大きくなれば忘れてしまう。そんな砂場のイメージがそのまま当てはまるプログラミング言語である。

web ブラウザで「プログラミング言語 Sunaba」と検索すれば、私の配布サイトに辿りつくので、そこで「Sunaba140324.zip」といった具合のファイ

ルをクリックしてダウンロードしてほしい*1。さらに、「Sunaba 追加パッケージ 140324.zip」といった具合のファイルもダウンロードしておく。

まずは「追加パッケージ」でない方を展開しよう。その後、中に入っている Sunaba.exe を起動すれば Sunaba のウィンドウが開く。



windows8 の場合「windows によって PC が保護されました」と言われて実行できないことがあるが、「詳細情報」という所を押して、さらに「実行」を押せば良い。

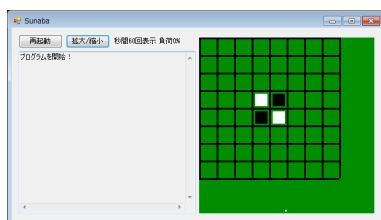
また、「お使いの PC にあるアプリには、Windows の次の機能が必要です」と言われることもあるが、この時には「この機能をダウンロードしてインストールする」を選べばいい。少し待てば、Sunaba を実行できる。

1.1.1 お手本を動かしてみる

次に、「追加パッケージ」の方を展開する。Sunaba フォルダの中に「サンプル」というフォルダがあるので、開く。中に「リバーシサンプル」というフォルダがあり、その中に「リバーシお手本.txt」というファイルがある。これを Sunaba のウィンドウにドラッグアンドドロップする。

すると、ゲームが動き出す。マウスで石を置く所をクリックすれば、石が置かれる。残念ながら思考プログラムは入っていないので、二人で遊ぶか、一人で交互に白黒の石を置いてほしい。

*1 140324 という数字は、2014 年 3 月 24 日の意味で、私が更新すれば変わる。プログラムには必ず間違いがあり、更新を繰り返しながら直していくものだ。だから、日付けがわかった方が都合がいいのである。



この状態でもう一度動かしたければ、もう一度ドラッグアンドドロップすればいい*2。

このサンプルはプログラムを短くするために、わざと不具合を残したままになっており、例えばどこにも石を置けなくなるとゲームを進められなくなるし、全部のマスを埋めた後でどちらが勝ったかを表示してくれることもない。自分で数えるしかない。さらに、今白と黒どちらの番なのかすらわからない。

そういった問題を直したり、思考プログラムを書いたりすることは読者の課題としておこう。

なお、Sunaba のウィンドウにある「拡大/縮小」ボタンを押す度に、表示画面の大きさが変わる。大きくしたい時や、小さくしたい時には押してみしてほしい。

1.1.2 Sunaba の見掛け

では、実際に Sunaba のプログラムがどんな形をしているか見てみよう。「リバーシお手本.txt」をダブルクリックしてみてほしい。標準であればメモ帳が起動して中身が見える。ここでは少し長くなるが、雰囲気を感じるために全文を載せてみよう。

```
memory[55001] -> 1
drawBackground()
initializeBoard()
side -> 0
while 1 #infinite game loop
    x -> memory[50000]
```

*2 「再起動」というボタンを押してもいい。

```

y -> memory[50001]
cx -> x / 10
cy -> y / 10
if memory[50002] != 0
    if placeDisk(cx, cy, side) != 0 #successfully placed.
        side -> 1 - side
        memory[55000] -> 1
#---program end here. followings are subprogram definition---
def initializeBoard()
    i -> 0
    while i < (8 * 8)
        memory[i] -> 2
        i -> i + 1
    set(3, 3, 0)
    set(4, 4, 0)
    set(3, 4, 1)
    set(4, 3, 1)
def placeDisk(x, y, side)
    out -> 0
    if (x >= 0) * (x < 8) * (y >= 0) * (y < 8) #inside board
        if square(x, y) = 2 #vacant
            if (placeDiskInsideBoard(x, y, side) != 0)
                out -> 1
def placeDiskInsideBoard(x, y, side)
    out -> 0
    out -> out + tryTurn(x, y, 0, -1, side) #12時半
    out -> out + tryTurn(x, y, 1, -1, side) #1時半
    out -> out + tryTurn(x, y, 1, 0, side) #3時半
    out -> out + tryTurn(x, y, 1, 1, side) #4時半
    out -> out + tryTurn(x, y, 0, 1, side) #6時半
    out -> out + tryTurn(x, y, -1, 1, side) #7時半
    out -> out + tryTurn(x, y, -1, 0, side) #9時半
    out -> out + tryTurn(x, y, -1, -1, side) #10時半
def tryTurn(bx, by, dx, dy, side)
    out -> 0
    x -> bx + dx
    y -> by + dy
    enemySide -> 1 - side
    turnCount -> 0
    while square(x, y) = enemySide
        x -> x + dx
        y -> y + dy
    if square(x, y) = side
        endX -> x
        endY -> y
        x -> bx + dx
        y -> by + dy
        while (x != endX) + (y != endY)
            turn(x, y)
            turnCount -> turnCount + 1
            x -> x + dx
            y -> y + dy
        if turnCount > 0
            set(bx, by, side)
    out -> turnCount > 0
def turn(x, y)
    memory[(y * 8) + x] -> 1 - memory[(y * 8) + x]
    drawDisk(x, y)
def set(x, y, side)
    memory[(y * 8) + x] -> side
    drawDisk(x, y)
def square(x, y)
    out -> -1
    if (x >= 0) * (x < 8) * (y >= 0) * (y < 8)
        out -> memory[(y * 8) + x]
def drawVerticalLine(x, up, down)
    y -> up
    while y < down
        memory[60000 + (y * 100) + x] -> 0
        y -> y + 1
def drawHorizontalLine(y, left, right)
    x -> left
    while x < right
        memory[60000 + (y * 100) + x] -> 0
        x -> x + 1
def drawBackground()
    fillGreen()
    drawGrid()
def fillGreen()
    x -> 60000
    while x < 70000
        memory[x] -> 005500
        x -> x + 1
def drawGrid()
    c -> 0 #horizontal
    while c < 9
        drawHorizontalLine(c * 10, 0, 80)
        c -> c + 1
    c -> 0 #vertical
    while c < 9

```

```
drawVerticalLine(c * 10, 0, 80)
c -> c + 1

def getSideColor(side)
  out -> 0
  if side = 0
    out -> 999999
  if side = 1
    out -> 000000

def drawDisk(cx, cy)
  color -> getSideColor(square(cx, cy))
  beginX -> (cx * 10) + 2
  beginY -> (cy * 10) + 2
  y -> 0
  while y < 7
    x -> 0
    while x < 7
      memory[60000 + ((beginY + y) * 100) + (beginX + x)] -> color
      x -> x + 1
    y -> y + 1
```

いきなり気が遠くなったかもしれないが、どれくらいの量なのかを感じておいてほしいので、敢えて全部載せた。雰囲気がわかれば良い。140 行足らずの量だ。400 字詰め原稿用紙は 20 行あるので、一行の長さを無視すれば 7 枚くらいだ。文字数で言えば、2000 字より少し多いくらいとなる。あのゲームを作るには、これくらいの量が必要だということだ。

索引

Sunaba のインストール, 1