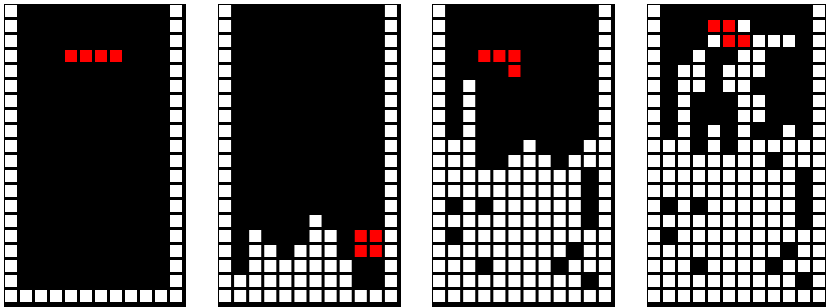


プログラムはこうして作られる
ゲーム作りで学ぶプログラミングの思考法

2013 年 8 月 26 日 11 時 41 分 (GMT) バージョン

この本は誰に何を提供するか

まずはこの写真を見てほしい。



落ちてくる四角をうまく詰めるゲームだ*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、身につけるための入門書である。

入門書であるから、君がプログラムを一度も作ったことがなく、コンピュータに関係した勉強をしたこともないと仮定する。そんな君に、私はプログラムを考え出す能力を身につけてもらいたいと思う。他人が作ったものに手を加えたり、他人の作ったものを寄せ集めたりする能力ではない。全てを自分の頭の中からひねり出し、作り上げる能力である。

*1 有名な何かに似ているが、諸般の事情により敢えて名前は言わない。

なぜこの本か？

さて、この本の主旨はわかっていただけたかと思う。しかし、これだけでは読む気にはなるまい。

プログラム作りを学ぶなら、他にも手段はたくさんある。学校で学ぶこともできるし、職場で学ぶこともできる。プログラム作りの入門書は掃いて捨てるほどあるし、無料の web サイトも多い。

なのに、なぜこの本なのか。何が他と違うのか。それを知ってもらう必要がある。

少々長くなるが、この前書きにて話しておきたい。この本を選ぶかどうかはそれを聞いてから決めてもらえればと思う。

0.1 これまでの教え方が持つ三つの問題

私が思うに、これまでの教え方には、三つほど大きな問題点がある。

- 個々の要素がバラバラに教えられ、組み立て方が教えられない
- いきなりプロ向けの道具を使うために、初期に挫折しやすい
- できない人がいても仕方ないと思われる

このそれぞれについて、少し詳しく扱おう。

0.1.1 要素がバラバラ

プログラム作りには、多くの道具がある。「プログラミング言語」という道具が特に大切だが、それだけではない。開発環境と言われるソフトウェアも道具だし、数学や情報工学のような学術的な知識も道具だ。それぞれの分野について本や web サイトがあり、授業や研修も行われる。

しかし、これらの道具を使って実際にどうプログラムを作るかは、ほとんど教えられない。

まだ君はプログラム作りを良く知らないだろうから、料理で例えてみよう。

鍋や包丁の使い方や食材についての本はたくさんあるのに、なぜかレシピ集がない状況を想像してほしい。乱切りや千切り、鍋の種類や扱い方、調味料の種類、などが網羅された本があるのに、カレーに入れる人参をどう切ればいいのかはわからない。そもそもカレーに人参を入れるかどうかすらわからない。

もちろん、才能と根性があれば、レシピがなくてもどうにかなるだろう。食べたカレーを思い出しながら使う食材を決め、切り方や煮方を試行錯誤する。そうやって経験を積むうちに、「煮込みの時にはこの切り方」「この野菜にはこの調味料が合う」というようなこともわかるだろう。いずれは、料理をアレンジしたり、料理を生み出したりできるようになる。

しかし、そこまで行ける人間が何人いるだろうか？ レシピがあれば、人参を入れるかどうかを迷う必要はないし、切り方を迷う必要もない。いろいろな料理をレシピに従って作っていれば、そう時間をかけずに料理の基本が身に付く。才能や根性はそれほど必要ない。

「煮物の理論」とか「調味料事典」のような本を初心者が買うことはない。なのに、プログラム作りに関する本はそういう本ばかりなのだ。

プログラムにレシピ本はあるか

だが、プログラムにレシピがないのには理由がある。

まず、プログラムの種類によって、使う技術が違いすぎる。ワープロとゲームでは使う技術も作り方も違う。料理でも寿司とビーフシチューはかなり違うわけで、寿司の本でビーフシチューが作れるようにはなるまい。

加えて、プログラムは一冊の本で説明するには大きすぎる。以前私が参加したとあるゲームは、7人のプログラマが1年間にわたって毎日10時間以上取り組んで完成させた*2。こんなに大きなものを、一冊の本で詳しく説明することはできない。

*2 ゲームとしては大きいとは言えない。プログラマが50人を超えるゲームは珍しくない。

プログラム作りの本が、鍋や包丁の使い方のような個々の要素に偏っているのには、そのせいだ。

もちろん、全体を見せようとする努力や工夫はされている。しかし、全体を見ようとすれば細部は省略するほかない。「鍋具材セットを買ってきて、言われた通りに野菜を切って煮るだけ」に近い状態になる。それでは本当に学んだことにはならないのだ。

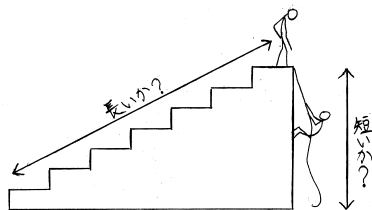
0.1.2 プロ向けの道具がもたらす挫折

多くの皿を一度に洗う食器洗い機は、お店を効率良く回すには必要なものだ。しかし、プロになった後で必要だからといって、プロになる前から使い方を学ぶべきだろうか。学ぶべきことは他にあるのではないだろうか。

プログラムに関しては、こういう当たり前の疑問が大事にされていない。最初から複雑で大きな道具を渡され、何十人も人間が協力して作る時に初めて必要になるような考え方や進め方を押しつけられる。

挫折して当然である。

確かに、プロと同じ道具を使うことでやる気が湧く、という事はある。教育用と言われると萎^なえるものだ。初心者は野心に燃えているものであって、プロの道具に憧れもあろうし、それが一番の近道だと思しやすい。だが、それは「チマチマ階段なんて登ってられるか！俺はロープで登るぜ！」と言うに等しい。



登れる人もいる。それは素晴らしいことだ。だが、普通の人には階段を用意すべきだろう。

0.1.3 できなくても仕方ないと思われている

もう一つ、さらに深刻な問題がある。

できるようにならない人がいても仕方ない、と思われていることだ。

プログラムの道で教師になったり本を書いたりするような人は、当然ながらプログラムが得意である。得意な人は、得意でない人のことがわからない。できないのは努力をしない本人が悪い、と思いがちだ。教え方が悪いとはなかなか思えない。

もちろん、学ぶ側にも責任はある。努力は大抵足りない。才能の違いも大きい。しかし、それはそれだ。教える側はベストを尽くすべきである。私が知る範囲では、教える側がベストと尽くしているとは思えない。

また、一見向いていないように見えても、本当に向いていないとは限らない。本人が苦手だと言っている、本当に苦手とは限らない。私はそれを大学生や後輩に接して知った。

人はいろいろな人生を歩んできており、^{つまづ} 躓き方はさまざまだ。「こんなことで！」と思うような事で躓く。しかし、そこを越せば、あとはどんどん伸びていくかもしれない。むしろ初期に他人が躓かないような些細なことでしっかりと躓いて、それを乗り越えてこそ、後の上達が早いということもある。

「マイナスにマイナスを掛けてプラスになるのが納得行かない」と言う生徒は、むしろ数学に向いているかもしれない。それを「才能がない」「不器用」と言って否定するのは、犯罪と言っている。私はそういう人を助けようと思う。

0.2 この本は何が違うか

この本は、私に見えている範囲で、これらの欠点を除く努力をした。大まかに言えば以下になる。

- 一つのプログラムを最初から最後まで作る
- 専用に道具を用意する
- 本当に初めてプログラムに触れる人を対象とする

0.2.1 一つのプログラムを最初から最後まで作る

この本は、料理でたとえれば「カレーを通して料理を学ぶ本」である。何に使うかもわからない要素をバラバラに紹介することはしない。全てはカレーを作るための手段として登場する。

そして、登場する手段については、それを何に使い、なぜ必要なのかを必ず説明する。そこが納得できないままで本気で勉強できるような人は、こんな本を必要としない。

このために犠牲になること

もちろん、いい事ばかりではない。

君がカレーに興味がない場合、私の作戦が裏目に出る。君が最初に写真を見せた「あのゲーム」に魅力を感じない場合、君にとってこの本は苦痛だろう。私は「カレーならみんな好きだろう」と思っているが、君がどうかはわからない。

また、カレーに必要なことしか出てこないことを、君は物足りなく思うかもしれない。千切りはしない。フライパンも使わない。醤油も出てこない。一つのゲームを作る中で全ての技術を網羅することはできない。

順番の問題もある。カレーを作る工程の順に説明する以上、分類や整理は一切なされない。第一章切り方、第二章調味料、第三章加熱、のような分類は無理だ。玉葱をみじん切りにして、炒めてから、人参を乱切りにする、となれば、「みじん切り」と「乱切り」は離れた場所で説明されてしまう。だから個々の技術を身につける効率は劣る。^{*3}。

^{*3} 普通の本は「第一章:変数」「第二章:分岐」というように技術の種類で章分けされている

新しい道具を出す前に必要性を納得してもらう、ということにも問題がある。必要性を納得するには、それがない時にどう困るのかを知る必要がある。口で言っても理解は難しいだろうから、君には体験してもらう他ない。そのため、君が十分に困るまで、私は道具は出さない。のび太が困る前に道具を出すドラえもんにはなりたくない。だから、どうしても時間がかかる。

しかし、私はそうした。それが必要だと思うからである。

0.2.2 専用に道具を用意する

プロになれる技術を身につけた後であれば、君にとって最高の道具はプロの道具である。しかしそれは、素人である今の君にとっては最高の道具ではない。

これは当たり前のことだが、不思議と当たり前と思われない。一流になるためには一流の人間に教わるのが早道、と言う台詞をたまに聞くが、ノーベル賞物理学者が小学校にいても役には立つまい。

そういうわけで、私は君のために専用の道具を用意することにした。プログラミング言語、開発環境、そしていくつかの用語や概念まで、初心者である君のために用意した。

機能は最小限だし、覚えねばならない規則も最小限だ。この本以外から情報を集める必要もない。

このために犠牲になること

もちろん、これも見方によっては欠点となる。

まず、この本だけでプロにはなれない。プロの道具は後で学ぶ必要がある。私はそれを遠回りとは思っていないが、君が遠回りと感じるならば仕方がない。

ので、こういう欠点はない。なお、この本を読み終えた人向けに、機能別の普通の説明書も用意しておいた(追加パッケージ内の「文書」フォルダ内にある「Sunaba 上級ガイド」)。

また、私のプログラミング言語では、大したことはできない。売り物のプログラムのような複雑で大きなものは作れない。開発環境も貧弱で、慣れてくれば機能不足を不満に思うだろう。

もう一つ、習慣の問題もある。「最初から正しいやり方をしないと悪い癖がつく」と脅す人は多い。最初からプロの道具で、プロのやり方をする事で、良い習慣が身に付くのだという。私はまるでそうは思わないが、そう思う人が多いことは知っている。

そして、専用の言語ということは、情報がこの本の中にしかないということでもある。君はこの本から得られる情報だけを材料にして、考え、組み上げねばならない。調べる必要がない、と言えば聞こえがいいが、調べることができない、と言えば長所には聞こえまい。調べる技術は身につかないし、調べることで他人の優れたやり方を知ることもしない。そういう種類の勉強を、私は完全に排除した。

今の君にはそれが必要だと思うからである。

0.2.3 本当に初めての人を対象をする

「初心者向け」という言葉は曖昧だ。そんな言葉を私は使わない。だから、「初めてプログラム作りに触れる人」とはっきり言おう。

説明をできるだけ丁寧にして飛躍がないように努め、日常の感覚と結びつけて理解するためにたとえを多く使い、また、用語の一つ一つに至るまで理解を妨げないように検討しなおした。抽象的な概念は極力後回しにし、具体例を先に出すようにした。

結果、以下のような特徴を持つ。

他の本には依存しない

本を選ぶのは手間だ。本を買うのも手間だしお金がかかる。本を読むのもまた手間だ。

複数の本を必要とするということは、その面倒でお金がかかることを何度

もやらねばならないということだ。初心者のやる気を削ぐに十分である。

だから私は、あのゲームが完成するまでに他の本が必要になることはないようにした。「プログラミング言語の文法は別の本で」「開発環境の使い方は他の本で」などとは言わない。

前提条件を極力下げた

私は君が初めてプログラム作りを経験すると仮定する。変数、関数、ループ、分岐、ソースコード、のような用語は一つも知らないと仮定している*4。プログラムを作る手順も一切知らなくていい。

英語も必要ない。プログラムと言えば英語圏発祥のものなので、英語が必要だと思われがちだが、それはプロになってからの話である。今は無用だ*5。

数学もいらない。小学校の算数で足りる。

しかし、残念ながら Windows の基本的な使い方は前提とせざるを得ない。プログラムを動かすとか、ドラッグアンドドロップ、ダブルクリック、といった操作は知っているとは仮定するし、「ファイル」というものが「フォルダ」なるものに収められていることも理解してほしい。Windows が動くコンピュータも必要である*6。

このために犠牲になること

もちろん、こういった特徴は欠点にもつながる。

おそらく君は、一度ならず「くどい」と感じるだろう。「何度同じ説明をするのか」「それは前提として先に行けよ」「もっといいやり方があるのに、なぜこんな馬鹿なやり方をする？」などと思わずに済むことは、たぶんない。

*4 そもそもこれらの用語は出てこない。

*5 必要になってからの方が本気で学べる。その時が来たら学べば良い。後で必要だから学んでおくべき、と言う人間を私は信用しない。必要になる前に本気で学べるはずがない。

*6 windows は 7 と 8 を主な対象とし、Vista と XP も動くように努力する。2000 でも動くかもしれない。mac については要望次第で対応する。また、将来的には iPhone などを含めて何でも動くようにするつもりでいる。

そして、君は大したことはできるようにならない。たかだか「あのゲーム」を作れる程度だ。すでに言ったように、大きなプログラムを丸ごと作ろうとすれば、どうしても細部は抜け落ちる。一冊の本に入らない。細部を保とうとするならば、小さなプログラムにするしかない。誰でも登れるようにと坂を緩やかにすれば、登れる高さは低くなるのである。

だが私はこれらの欠点を受け入れ、「二階へ上がるために 100 メートルの坂道を歩く」ような本とした。

0.3 では、始めよう！

まとめれば、

- 具体的なプログラム作りを通して学ぶ
- 入門専用のプログラミング言語を使う
- 本当に初めてプログラムを作る人を対象とし、一冊で基本を身につけてもらう

となる。

もし君がこの本に合わないと感じたならば、無駄な金と時間を払うことはない。しかし、ここまで読んだ上で納得してもらえたならば、私は君のために全力を尽す。

それでは、始めよう。

目次

この本は誰に何を提供するか	iii
0.1 これまでの教え方が持つ三つの問題	iv
0.2 この本は何が違うか	vii
0.3 では、始めよう！	xii
第1章 プログラム作りを始める前に	1
1.1 あのゲームを作るためにはどうしたらいいか	1
1.2 どこから手をつけるか	15
1.3 この章で伝えたかったこと	17
第2章 メモリ操作～四角を描く～	19
2.1 書いたプログラムを動かす	19
2.2 四角を描く	23
2.3 好きな場所に四角を描く	40
2.4 この章で伝えたかったこと	54
第3章 くり返し～プログラムを短くする～	57
3.1 行数を減らす	58
3.2 くり返しの使い方	61
3.3 くり返しで四角を描く	77
3.4 この章で伝えたかったこと	98

第 4 章	プログラムの変形～四角をたくさん描く～	101
4.1	くり返しで壁を描く	101
4.2	もっと短くしたい	114
4.3	この章で伝えたかったこと	127
第 5 章	部分プログラム～四角をわかりやすくたくさん描く～	129
5.1	くり返し以外の行を使い回す手段	130
5.2	Sunaba における「参照」	135
5.3	部分プログラムを使って壁と底を描く	140
5.4	もっとわかりやすく	149
5.5	この章で伝えたかったこと	158
第 6 章	動きのあるプログラム～四角を落とす～	161
6.1	次は何をすべきか	162
6.2	四角を落とす	163
6.3	もっと速く塗りたい	180
6.4	ここまでを組み合わせる	188
6.5	この章で伝えたかったこと	193
第 7 章	名前付きメモリ～プログラムを楽に～	195
7.1	今のプログラムは何が悪いか?	196
7.2	Sunaba でメモリに名前をつける	204
7.3	名前付きメモリのルール	208
7.4	この章で伝えたかったこと	214
第 8 章	操作できるプログラム～四角を動かす～	215
8.1	操作するとはどういうことか	216
8.2	操作できるようにする	220
8.3	動きっぱなしを避ける	231
8.4	落ちてくる四角を動かせるようにする	246

8.5	この章で伝えたかったこと	248
第 9 章	条件実行～ゆっくり落とす～	251
9.1	ゆっくり落とすということ	251
9.2	条件実行	260
9.3	Sunaba で条件実行	267
9.4	計算の本当の意味	275
9.5	この章で伝えたかったこと	286
第 10 章	連続したメモリ～四角を積み上げる～	289
10.1	当たるとのこと	290
10.2	メモリをかたまりで使う	300
10.3	積み上げる	313
10.4	この章で伝えたかったこと	336
第 11 章	部分プログラムへの入力～列を消す～	339
11.1	とりあえず消すだけ	339
11.2	プログラムを良くする	353
11.3	落とす	369
11.4	この章で伝えたかったこと	384
第 12 章	部分プログラムの出力～回す～	387
12.1	2 つに増やす	389
12.2	回すとはどういうことか	394
12.3	回転を組み込む前に	406
12.4	回転を組み込む	426
12.5	ちゃんと動くのか?	436
12.6	この章で伝えたかったこと	446
第 13 章	完成	449

13.1	3 マスづつ落とす	449
13.2	4 マスづつ落とす	470
13.3	この章で伝えたかったこと	496
第 14 章	始まりの終わり	499
14.1	ここまで使ってきた考え方	499
14.2	さらに進むために	506
14.3	実用言語の大変なところ	509
索引		519
終わりに、そして感謝		521