ことだまonSqueakで学ぶ論理思考とプログラミング

大岩 元…監修
松澤芳昭・杉浦 学…編著

イーテキスト研究所

ことだま on Squeak で学ぶ
論理思考とプログラミング

大岩 元 監修

松澤 芳昭　杉浦 学 編著

はじめに

Squeak は，それを使って遊べば遊ぶほどいろいろなことを発見できる不思議なおもちゃです．しかし，Squeak を起動しても，最初は何もない画面が出るだけです．テレビゲームのように，遊ばせてくれるおもちゃではありません．レゴブロックのように，自分で組み立てて遊ぶおもちゃです．ゲームを作ることもできるし，世界をシミュレート（模擬実験）をすることもできます．いろいろな遊び方ができる代わりに，おもしろい遊び方を自分で見つけないと使えないおもちゃです．

実は，パソコン自体もそのような機械で，「パソコンはソフトウエアが入っていなければただの箱」などとよく言われます．たとえソフトウエアが入っていたとしても役に立つかどうかは使う人しだいです．例えばワープロソフトが入っていたとしても，パソコンは文章を自動的に作ってくれるわけではありません．自分で文章を打ち込む必要があります．パソコンは，使い方によっては自分でソフトウエアを作ることさえできる機械ですが，やはり使い方を自分で見つけないと使えない機械です．

Squeak を作ったのは，パソコンを作った人と同一人物です．アラン・ケイ博士です．ケイ博士がパソコン（個人用コンピュータ）を考えたのは 30 年以上前で，当時コンピュータといえばプロフェッショナルのための機械で，それだけで部屋が埋まってしまうほど大きく，画面には文字だけが表示されるものでした．操作するのも難しく，個人がコンピュータを使うようになるとは誰もまだ想像できませんでした．そんな時代に，ケイ博士は子供たちでも使えるくらい簡単な道具になるようにパソコンを作りました．

今日，パソコンは日常のものとなり様々な場所で使われていますが，ケイ博士は，「パソコン革命はまだ始まっていない」とおっしゃいます．今日，ほとんどの人はパソコンに用意されたソフトウエアの使い方を学んで利用しているだけです．不満があってもコンピュータに従うしかありません．しかし，ケイ博士がパソコンを作ったときに想像していたのは，使う人たちのそれぞれが，自分たちをよりよく助けてくれるようにパソコンを変えたり，新しいものにしていくような使い方なのです．人間はパソコンに使われるのではなく，使うことで人生を豊かにしていかなければなりません．

Squeak はケイ博士が想像する使い方が簡単にできるように作られたソフトウエアです．人間が想像したことであればなんでもシミュレートするソフトウエアを作ることができます．ソフトウエアを作ることをプログラミングといいます．プログラミングというと，プロの人がなにやら文字を打ち込んでいるというイメージがありますが，それはプログラミングのほんの一部分で，実は何かを想像するところがプログラミングの一番面白く，難しいところです．何をするかを想像することは，コンピュータがいくら進化したとしても人間がやらなければならないことです．

では，なぜ人間はパソコンを使ってプログラミングをしなければならないのでしょうか．それは，人が何かを学習したり，理解したりすることと関係があります．シーモア・パパート博士は，子供にプログラミングをさせながらそのために必要な数学や科学を教えることで，子供たちが大人でも難しいことを理解してしまうことを発見しました．現在ほとんどの大人が数学を難しい学問だと考えていますが，それは紙の上で意味のわからない数式を書きつづる，何に役に立つのかわからない数学のことであって，何かをシミュレートするために使う数学は，役に立つ魔法のようなものになります．人間は役に立つとわかれば，年齢に関係なく理解することができるのです．

さらに，パパート博士は，子供がプログラミングをする過程で，いままで学校ではあまり教えてくれなかった「大事なこと」を学習していることに気づきました．例えば，何かものを作ったことがある人は，何

かを作り上げるときに，一回で想像したものが想像通りにできてしまうことはないことを知っています．プログラミングをやってみるとよくわかることですが，作ったものが一回で意図したとおりに動くことはまれです．プログラミングでは，一回で正解を作るのではなく，間違っているところを修正して，試行錯誤をしながら正しく動いていくようにしていけばよいのです．これはプロでもやっていることです．

このプログラミングの過程は，人が何かを学習する過程によく似ています．最初からテストで 100 点を取れる人はいません．間違った考えを試行錯誤しながら修正していって，理解にたどりつきます．最初の間違った考えに「なぜ？」という疑問を持ったとき，初めて学習が始まります．学校ではいつも間違いは悪いことだと教わりますが，社会ではそうでないこともたくさんあります．プログラミングをすることで，そういった学習の方法も学習していることになります．そして何より，そうしてプログラムが完成したとき，作ったものの仕組みを完全に理解していることになります．これらは，「大事なこと」の一つの例にすぎません．

ケイ博士は，このようなパパート博士の研究に大きな影響を受けました．そしてそれをより簡単に子供が使えるようにした現代版のソフトウエアが Squeak です．あとは Squeak を使って遊んでみて，いろいろなことを発見してください．作品の出来を楽しみにしています．

Happy Squeaking!

2008 年 4 月
監修者・編著者 一同

読者のみなさんへ

高校生のみなさんへ

高校生になると，数学は急にわけがわからなくなります．私も，高校の数学で落ちこぼれた部類の人間です．どうしてかというと，急に現実からかけ離れた，何の役に立つかわからない数式になっていったと感じたからだと思います．特に微分・積分は意味がわからないものの象徴で，やる気も成績も芳しいものではありませんでした．私自身も大学院生になって，Squeak で遊んでみて，やっと微分・積分の意味がわかってきたような気がします．このテキストにも微分・積分を利用したプログラミングがたくさん含まれています．実は，最初の車に円を描かせる問題がすでに微分方程式を解く問題なのです．皆さんにとっても数学がそうした意味のあるものになることを期待しています．

また近年では Squeak を使って新しい問題を解き，論文を書く高校生もいるようです．Squeak で培ったものづくりの経験を，文化祭などの企画実行に活かしたりする生徒もいます．自分で考えた世界を構築し，シミュレートして，高度なことを発見し，現実の生活に役に立つ学習ができることを期待します．

大学生のみなさんへ

Squeak をやろうというと，大学生なのだから，Java とか C などの実用言語を勉強すべきだという学生がよくいます．最終的にはそうなるべきです．しかし，プログラミングの初歩的な考え方を学ぶ入門コースは，Squeak が適していると私は考えています．

プログラミング教育は，自分の考えをモデル化して，表現できるようになる教育です．コンパイルエラーに悩まされ，コンピュータに従うことを強制される訓練ではありません．プログラミング言語を覚えることはそのうちのほんの一部分でしかありません．

そのための言語は何でもよいのですが，簡単にできるものがよいでしょう．Squeak でできることを Java で書いてみて，比較してみましょう．Squeak のほうが簡単にできることがわかるはずです．大学生なのですから自分の考えをモデル化することに集中し，複雑な問題を解くことを目指してください．

それでも Squeak にはいろいろ制限があるよ，と主張する人は，Squeak の裏側で動いている Smalltalk という言語と Java を比較してみるとよいでしょう．Squeak は裏側でうごいている Smalltalk の美しさでも評価されています．Squeak の裏側で動いている Smalltalk のソースコードは，全て見ることができ，改造もできるように設計されています．プログラミングが得意な人にとっても面白い教材となるはずです．

このテキストの使い方

Squeak は遊んでいれば学習できるように設計されています．しかし，「はじめに」で指摘したように，Squeak は「遊ばせてくれる」ものではありません．最初は遊び方のガイドが必要です．このテキストはそのガイドを務められるように設計しました．また，皆さんが Windows や Mac の基本的なファイル操作ができることを前提に書いてあります．

人の学び方はそれぞれ異なります．つまずくところも，理解にかかる時間も，理解する内容ですら人それぞれです．ですから，自分なりにテキストを読み，遊んでください．面白いことを発見したら寄り道をしながら進んでみてください．基礎的なカリキュラムを追っていくのが性に合わない人は作品製作から始めてみて，つまったら必要なところをつまみ食いする，という方法もあるかもしれません．

このテキストの作成にあたっては，なるべくたくさん練習問題を作ることを考えました．例題から発展する「やってみよう」や「考えてみよう」という小さな問題も作ってみました．「やってみよう」よりも「考えてみよう」のほうが難しく，少し深い考えが必要な問いになっています．余裕がある人は挑戦してみてください．友達と議論してみても面白いと思います．

興味を引く面白い問題が人を学習へと導きますが，すべての人が面白いと思う問題を作ることはできません．そこで，なるべくいろいろな種類の練習問題を用意するよう努力しました．しかし，それでもみなさんの興味の範囲をとてもカバーし切れません．是非自分で面白い問題を考えて，解いてみてください．テキストの練習問題をすべて解くよりも，自分で考えた問題を一つ解くほうが価値があります．いい問題ができたら，是非教えてください．（連絡先：squeakers@crew.sfc.keio.ac.jp）

最後に，どのように進めるかの参考として，めやすの時間を示します．初心者を対象とした実験では，練習問題を全て解き，例題も何故そう動くのかを考えながら進むと，Project 7 まで 25 時間位かかることがわかりました．なるべく早く進みたい人は，例題だけをテキストどおりに作ってみてください．そうすると Project 7 までが 2，3 時間で終わります．でも例題を追っていくだけでは，ただ作業をしているだけで，考えないで作業をすることになります．勘のいい人はそれでわかってしまいますが，そうではない人もいると思います．お勧めの方法は，例題に加え，全ての「やってみよう」と各 Project の練習問題を 1，2 問解いてみる方法です．8 時間位で Project 7 までが学習でき，Squeak を使って何ができるかがわかると思います．

「ことだま on Squeak」とは

「ことだま on Squeak」は，大島さん，阿部さんを始めとする日本人 Squeaker が作成した日本語版 Squeak（Squeak2005J）を基に，2005 年に経済産業省の支援を受けて，慶應義塾大学 大岩研究室の岡田 健を中心とするチームによって開発されたものです．

この「ことだま on Squeak」は，タイルの語順等を変更し，日本人にとって自然な日本語になるようにしてあります．これによって，日本人の学習者はタイルに書かれている言葉を読んでその意味を理解することができるようになりました．もはや，先生がタイルの意味を解説する必要はありません．学習者は母語を使い，論理的思考に集中することができます．

また，「ことだま on Squeak」には，私達の 5 年余りにわたる教育実践を踏まえて，従来の Squeak では操作しづらい箇所のインターフェイスの変更を行っています．教育上不必要なタイルは隠してあります．音楽ファイルの取り込みなどの便利な機能も搭載されています．これによって，学習者が指摘する操作方法の不満はほぼ解消されており，たくさんの創造的な作品が毎年誕生しています．

「ことだま on Squeak」は，よりよい学習環境を目指して現在でも改良が続けられており，インターネット（http://www.crew.sfc.keio.ac.jp/squeak/）からフリーでダウンロードすることができます．2008 年 4 月現在の最新バージョンは 1.1.8 です．バグの報告や，改良の提案がありましたら，ご連絡をいただければ幸いです．（連絡先：squeakers@crew.sfc.keio.ac.jp）

目次

第 I 部　Squeak 環境の準備 1

Project 0 Squeak を使えるようにしよう 3
0.1 インストール . . . 3
0.2 起動と終了 . . . 5

第 II 部　Squeak の基本操作とスクリプティング 7

Project 1 車を描いてみよう 9
1.1 フラップ . . . 9
1.2 オブジェクトとハロ . . . 10
1.3 プロジェクト . . . 13
1.4 お絵かきツール . . . 14
1.5 作業結果の保存 . . . 18

Project 2 車を動かしてみよう 21
2.1 ビューアーと命令の実行 . . . 21
2.2 スクリプトの作成と繰り返し . . . 24
2.3 命令の組み合わせ . . . 28
2.4 命令の実行順序 . . . 30
2.5 スクリプトの便利な操作 . . . 31

Project 3 車を道に沿って走らせてみよう 39
3.1 場合分け（1）– 各場合に実行する命令が 1 つのとき . . . 39
3.2 場合分け（2）– 各場合に実行する命令が複数のとき . . . 43
3.3 フローチャート（1）– 場合分けが 1 つのとき . . . 46
3.4 フローチャート（2）– 場合分けが 2 つのとき . . . 48

Project 4 障害物を作ってみよう 55
4.1 変数 . . . 55
4.2 変数と値の変更（1）– 値指定 . . . 57
4.3 変数と値の変更（2）– 差分指定 . . . 59
4.4 変数を使った場合分け . . . 62

Project 5 車をハンドルで運転できるようにしてみよう 69
5.1 他の変数を使った現在値の変更（1）– 値指定 . . . 69
5.2 他の変数を使った現在値の変更（2）– 差分指定 . . . 72
5.3 2 つの変数を使った場合分け . . . 76

Project 6 レーシングゲームを作ってみよう 81
6.1 計算 . . . 81
6.2 乱数 . . . 83

Project 7 車をアクセルで加速できるようにしてみよう 91
7.1 速度の概念と変数の作成 . . . 91
7.2 加速と減速のシミュレーション . . . 93

第 III 部 作品づくりプロジェクト 97

Project 8 作品づくり 99
8.1 作品作りのプロセス . . . 99
8.2 企画の立案 . . . 100
8.3 仕様の策定 . . . 102
8.4 スケジューリング . . . 107
8.5 実装 . . . 109

Project 9 まとめと評価 111
9.1 プレゼンテーションの作成 . . . 111
9.2 作品の評価 . . . 114
9.3 報告書の作成 . . . 114

第 IV 部 アルゴリズムの組み立て 115

Project 10 並び替えをしてみよう 117
10.1 アルゴリズム . . . 117
10.2 手作業による並び替え . . . 118
10.3 コンピュータによる並び替え（1）– カードの準備 . . . 123
10.4 コンピュータによる並び替え（2）– 並び替えのプログラム . . . 128

Project 11 辞書を作ってみよう 133
11.1 検索のアルゴリズム . . . 133
11.2 辞書のアルゴリズム . . . 139

第 V 部 付録 147

付録 A Squeak テクニック集 149
A.1 ハロ . . . 149
A.2 お絵かきツール . . . 152
A.3 プロジェクト単位の保存 . . . 153
A.4 軌跡の描画 . . . 154
A.5 刻み値 . . . 154
A.6 方向と重心 . . . 155
A.7 全スクリプト . . . 156
A.8 リモートスクリプティング . . . 156
A.9 見た目を似せる . . . 157
A.10 ボタン . . . 158

A.11 ジョイスティック 159
A.12 スライダー 159
A.13 テキスト 160
A.14 出現させる・隠す 160
A.15 キーボード入力の受け取り 161
A.16 音楽ファイルの取り込み 161
A.17 入れ物 163
A.18 カテゴリレベルの変更 166
A.19 フラップの初期化 166

付録 B Mac で使うには 167
B.1 インストール 167
B.2 起動と終了 169

付録 C 最小値選択法ワークシート 171
C.1 Step1. カードの準備 171
C.2 Step2. 並び替えの手順の理解 172
C.3 Step3. 並び替えと計測 173
C.4 Step4. グラフ作成 174
C.5 Step5. 考察 174

参考文献一覧 177

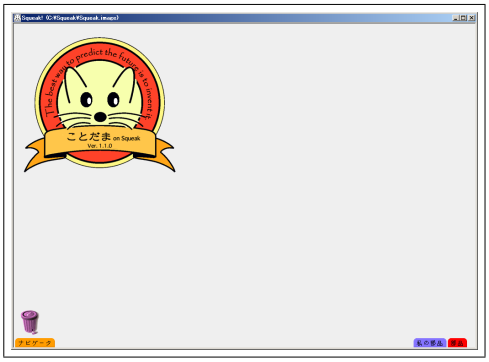
索引 179

第 I 部

Squeak 環境の準備

Project 0

Squeak を使えるようにしよう



ようこそ，Squeak の世界へ．

このプロジェクトでは，これから「ことだま on Squeak」を使ってプロジェクトを進めるための準備を行います．「ことだま on Squeak」はフリーソフトですので，誰でも自由に自分のコンピュータにインストールして使うことができます．この Project 0 では，Windows 版のインストール方法を説明します．

付録 B（P.167）に Mac 版のインストール方法も掲載してあります．

キーワード

インストール，起動 / 終了，image ファイル，changes ファイル

0.1　インストール

M先生　それでは，これから「ことだま on Squeak」を使った勉強を始めるよ．

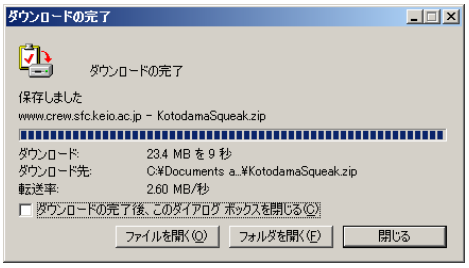
Ｓくん・Ｔさん　よろしくお願いします．

M先生　さっそく“インストール”を始めよう．まずは Squeak[*1]をダウンロードしてみよう．URL は http://www.crew.sfc.keio.ac.jp/squeak/ だよ．

Ｔさん　ダウンロードするファイルはどこに保存すればいいですか？

M先生　とりあえず好きなところに保存しておけばいいよ．ファイル名もそのままにしておいてね．

Ｓくん　ダウンロードが完了しました．

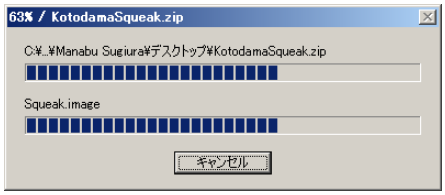


[*1] 2008 年 11 月時点で最新の「ことだま on Squeak」は Ver.1.1.8 です．本テキストの「ことだま on Squeak」に関する記述は Ver.1.1.8 時点のものです．

M先生　次はダウンロードしたファイルを解凍しよう．ダウンロードしたファイルの拡張子は zip になっているよね．

Ｓくん　はい．圧縮されているということですね．

M先生　そうだね．では，そのファイルを解凍しよう．いつも自分が使っている解凍ツール[*2]を使ってね．



Ｓくん　今解凍しています……終わりました．

M先生　最後に，解凍してできたフォルダを適切な場所に配置しよう．まずはそのフォルダの名前を「Squeak」に変更しよう．半角で入力するようにね．

Ｓくん　Squeak ですね．わかりました．

Ｓくん　先生，名前の変更が終わりました．

M先生　さて，そのフォルダはどこにあるかな？

Ｓくん　僕はデスクトップにあります．

Ｔさん　私はマイドキュメントの中ですね．

M先生　なるほど．2 人ともそれではまずいな．

Ｔさん　どうしてですか？

M先生　Squeak のあるフォルダのパスに全角文字やスペースが入っていると，データの保存や読み込みをするときにトラブルが発生する可能性があるんだ．

Ｔさん　ではどこに移動すればいいですか？

M先生　次の 2 つの条件を満たす場所であれば好きな場所でいいよ．

1. フォルダまでのパスに全角文字が入っていないこと．（Windows だとデスクトップやマイドキュメントなどは駄目）
2. フォルダまでのパスに空白が入っていないこと．（Windows だと Program Files 以下などは駄目）

Ｓくん　では，いつも使っている C ドライブに Squeak フォルダを移動します．

M先生　OK．これで Squeak を起動するための準備が整ったことになるね．これから「Squeak フォルダ」と言ったらそのフォルダのことだから．

Ｔさん　ところで，ダウンロードした zip ファイルはどうしましょう？

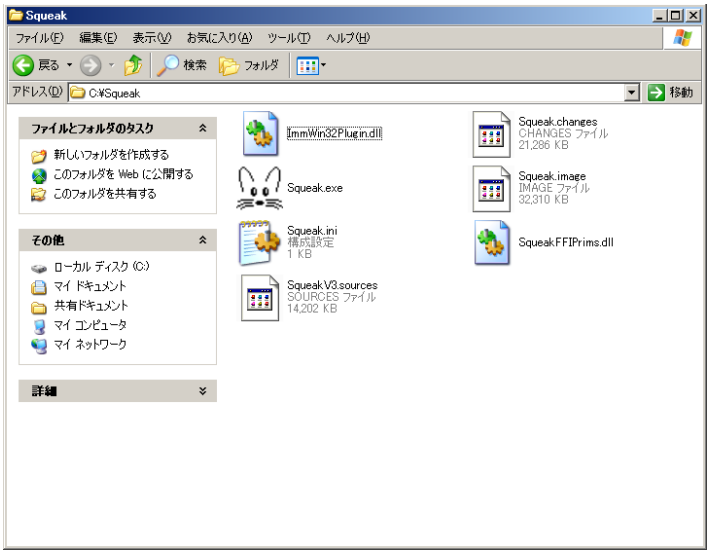
M先生　もう使わないから削除して構わないよ．

[*2] Windows XP の場合，解凍したいファイルを右クリックして，「全て展開」を選択すると解凍できます．

0.2　起動と終了

M先生　さて，Squeak フォルダの中身を確認してみよう．

Ｓくん　はい．全部で７つのファイルがあります．



Ｔさん　どれをクリックすれば起動できるんですか？

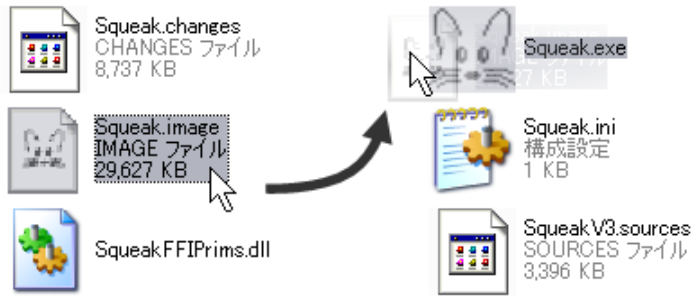
Ｓくん　Squeak.exe[*3]というファイルがありますね．これをダブルクリックすれば起動できるんですね？

M先生　いやいや，それだけでは無理なんだ．Squeak.image というファイルがあるよね？

Ｓくん　はい．

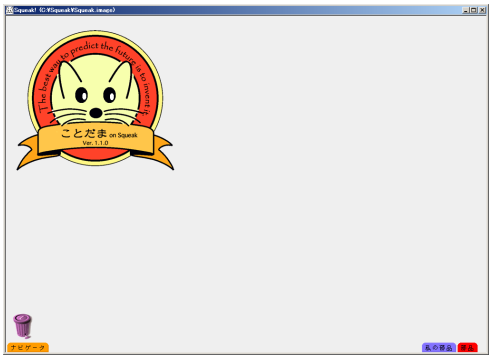
M先生　Squeak では，そのファイルにプログラムが保存されているんだ．Squeak.exe は起動役のプログラムだから，Squeak.exe に Squeak.image を読み込ませて起動する必要がある．Squeak.exe に Squeak.image をドラッグ＆ドロップすると“起動”できるよ．やってごらん．

Ｓくん　こうでしょうか？



[*3] 環境によっては，拡張子（「.exe」）非表示モードになっている可能性があります．その場合は Squeak アイコン（ねずみの顔）で判断してください．

Ｓくん　あっ，ウィンドウが開きました．Squeak が起動できたようです．



M先生　その画面が出れば起動は成功だよ．

Ｔさん　先生，Squeak フォルダには Squeak.changes というファイルもありますよね．このファイルは何ですか？

M先生　それはプログラムの修正履歴を保存しておくためのファイルなんだ．“changes ファイル”は同じ名前の“image ファイル”とセットで作成されている．だから誰かに自分の Squeak のデータを渡したいときは image ファイルと changes ファイルの両方を渡すようにね．

Ｔさん　わかりました．

Ｓくん　起動の方法はわかりましたが，Squeak を終了させるためにはどうすればよいのでしょうか？

M先生　Windows であれば，ウィンドウ右上にある×ボタンを押せば“終了”できるよ．

Ｓくん　×ボタンを押すとメッセージの書かれたウィンドウが出ますね．

M先生　Squeak の修正を Squeak.image に保存しないまま終了してよいかどうかを聞かれているね．だから「はい」を選択すると終了できるよ．

Ｔさん　今回は起動しただけですから，とくに何も保存する必要は無いわけですね．

M先生　そのとおり．だからとりあえず今回は「はい」を選択して終了して構わない．詳しい保存の方法は後で説明するからね．

Ｔさん　わかりました．

No. 0 – 1　**考えてみよう！**

Squeak という単語はどのような意味でしょうか．
また，なぜこのソフトウェアの名前が Squeak なのかを考えてみましょう．

M先生　それでは，いよいよプログラミングを始めようか．

Ｓくん　がんばるぞー！

第 II 部

Squeak の基本操作とスクリプティング

Project 1

車を描いてみよう

このプロジェクトでは，Squeak の基本操作の練習をします．Squeak 以外の環境に慣れている人にとって，最初は Squeak 特有の操作方法に戸惑うかもしれません．しかし，Squeak では全てのものが「オブジェクト」として扱われており，一貫性のある操作環境が提供されているので，操作を習得するのは非常に容易です．

一通りの基本操作に慣れたら，「お絵かきツール」を使って車の絵を描いてみましょう．この絵はこれから取り組むプロジェクトでもずっと使うので，愛着のもてるかわいい絵を描いておいてくださいね．

✎ キーワード

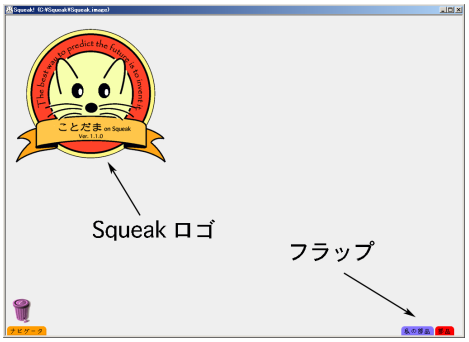
オブジェクト，フラップ，ハロ，プロジェクト，お絵かきツール，ワールドメニュー，保存

1.1　フラップ

M先生　まずは Squeak で遊んでみよう．Squeak は起動してるかな？

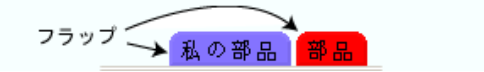
Ｓくん　起動しました．

Ｔさん　目が動く Squeak のロゴと，周りにタブのようなものがあります．



M先生　それが，Squeak の初期画面だよ．そのタブのようなものを“フラップ”というんだ．フラップは，Squeak で使用する様々な道具や部品が収められている引き出しのようなものと考えるといいね．フラップにはいろいろな種類があって，様々なものが収められているんだ．

M先生　フラップは，クリックすると引き出すことができるよ．例えば，「部品」というフラップをクリックしてみて．



Sくん　色々な部品が出てきました．

M先生　OK．部品の引き出し方については次で説明するね．フラップは，ドラッグで引き出すこともできるよ．

Tさん　フラップをたたむにはどうすればいいですか？

M先生　フラップをもう一度クリックするか，ドラッグするとできるよ．

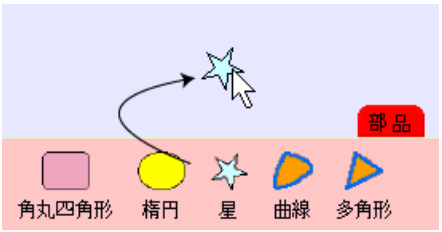
1.2　オブジェクトとハロ

M先生　では，次に“オブジェクト”を出してみよう．

Sくん　先生，「オブジェクト」って何ですか？

M先生　オブジェクトとは，Squeak 上で世界を構成する部品のことだよ．まずは部品フラップから「星オブジェクト」を取り出してみよう．オブジェクトの取り出しは，部品フラップからドラッグ＆ドロップでできるよ．

Sくん　こうですか？



M先生　OK．オブジェクトは，ドラッグして動かせるよ．

Tさん　面白いですね．

M先生　Squeak のオブジェクトは，動かす以外にも，大きさを変えたり，色を変えたりと，様々な操作ができるよ．オブジェクトに対する基本操作はすべて“ハロ”を出すところから始まるんだ．Alt（Mac の場合は Apple キー）を押しながらオブジェクトをクリック[*1]してごらん．

[*1] Windows でホイールのついているマウスを使っている場合，マウスのホイールをクリックしてもハロを表示できます．

Ｓくん　オブジェクトの周りに変な記号がいっぱい出てきました．

M先生　その変な記号のことをハロというんだ．ハロには色とアイコンがついていて，色やアイコンの名前でハロの種類を区別するので覚えておいてね．例えば，黄色いハロは「黄ハロ（きはろ）」と呼ぶよ．

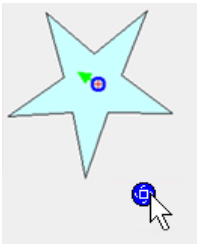
Ｔさん　わかりました．

M先生　では，最初に黄ハロを使ってみよう．黄ハロをドラッグするとどうなるかな？

Ｓくん　大きさが変わりました．

M先生　そうだね．じゃあ青ハロをドラッグするとどうなる？

Ｓくん　回転します．



M先生　次は緑ハロを 1 回クリックしてから，マウスを動かしてごらん．

Ｔさん　星がコピーできました．

Ｓくん　すごいっー！

M先生　桃ハロでそのオブジェクトを削除できる．この場合は，ドラッグではなく，ハロをクリックするだけでいいんだ．

Sくん　ゴミ箱にドラッグ＆ドロップしても消せますか？

M先生　もちろん．

Tさん　消してしまったものを元に戻すことはできますか？

M先生　ゴミ箱をダブルクリックしてごらん．

Sくん　あっ！ ゴミ箱の中身が見えた！

Tさん　捨ててしまったオブジェクトを戻したい場合はゴミ箱の中から引っ張りだせばいいんですね．

M先生　ほかにも様々なハロがあるので，あとは自分でいじってハロを研究してみてね．あと覚えておいてほしいことは，Squeak 上に見えるものは，ほとんど全てのものがオブジェクトだということ．フラップも，ワールド[*2]も，ゴミ箱もオブジェクトなんだよ．例えば，ゴミ箱や，Squeak ロゴも，クリックすればハロが出るよ．

Sくん　おもしろいですね．ゴミ箱を消してしまったらどうなるんだろ……．

No. 1 – 1　やってみよう！

黄，青，緑，桃ハロ以外のハロを使うとどんなことができるでしょうか？ ハロをクリックして研究してみましょう．

☞ ヒント

P.149（A.1）に様々なハロの解説があります．

[*2] 次節（1.3）で説明するように，Squeak では作業する画面をいくつも増やすことができます．ワールドは，その一つ一つの画面のことを指します．

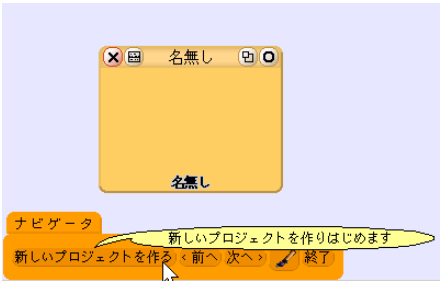
1.3　プロジェクト

M先生　これから色々な作品を作っていくんだけど，1つの画面に色々な物を作っていくとごちゃごちゃになってしまうよね．だから，新しくもう1つ画面を作ってみよう．

Sくん　どうやればいいのでしょうか？

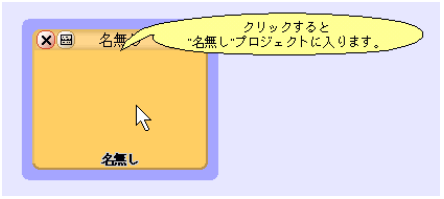
M先生　Squeakの1つの画面は“**プロジェクト**”という名前で呼ぶんだ．新しいプロジェクトを作成するには，「ナビゲータ」というフラップを開いて，「新しいプロジェクトを作る」ボタンをクリックするんだ．やってごらん．

Sくん　何か出てきました．



M先生　それがプロジェクトだよ．できたプロジェクトをクリックするとプロジェクトの中に入れるよ．

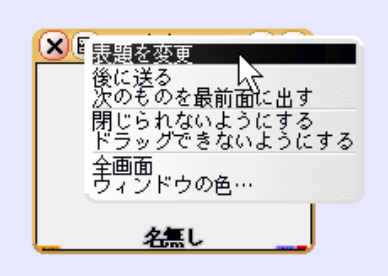
Sくん　こうですね．



M先生　ナビゲータフラップの「前へ」をクリックすると前の画面に戻れるよ．ためしに，何か適当なオブジェクトを置いてから，前のプロジェクトに戻ってみよう．プロジェクトの中身が見えるでしょ？

Sくん　星が置いてあるのが見えますね！

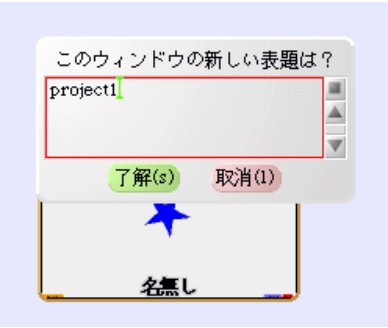
M先生　あと，プロジェクトの名前を変更しておこう．プロジェクトの×ボタンの右にあるメニューマークのついたボタンをクリックしてごらん．メニューが出てくるから，「表題を変更」を選択すると，プロジェクトの名前を入力できるよ．



Tさん　先生，どんな名前にすればいいですか？

M先生　これから始めるのは project1 なので，project1 という名前をつけておこう．

Sくん　了解です．



1.4　お絵かきツール

M先生　さっきは星オブジェクトのような，はじめから用意されているオブジェクトで遊んだけど，今度は“お絵かきツール”でオブジェクトの絵を自分で描いてみよう．

Sくん　どこのプロジェクトで作業をすればいいですか？

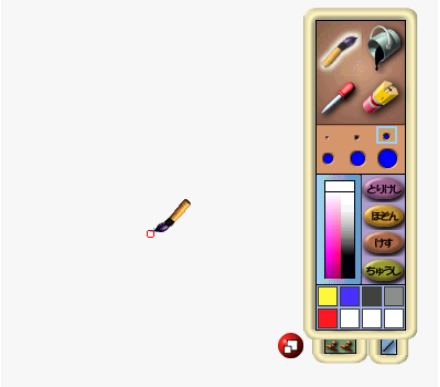
M先生　さっき作った project1 の中で作業をしてみよう．

Sくん　了解です．

M先生　ナビゲータフラップにある筆の描いてあるボタンをクリックしてみてね．

M先生　パレットが表示されていれば絵が描ける状態だから．

Ｓくん　できました．



M先生　OK．それでは早速絵を描いてもらおう．上から見た車の絵にしてくれるかな．ちなみにペンの太さや色はパレットで変えられるよ．

Ｓくん　はい．描けました．

M先生　では，描いた絵をオブジェクトにしよう．パレットの「ほぞん」ボタンを押してみて．パレットが出ている間はまだオブジェクトになっていないから，注意してね．

Ｓくん　「ほぞん」ボタンを押したらパレットが消えてしまいました．

M先生　それでいいんだよ．お絵かきモードが終わるとパレットは消えるんだ．お絵かきモードが終わると描いた絵はオブジェクトになるから，ハロを使った操作ができるようになるよ．

Tさん　あ，ほんとだ．

M先生　描いた絵のオブジェクトは最初「スケッチ」という名前になっているね．たくさん増えてくるとややこしいので，描いたものはすぐに名前を変えておこう．名前を変えるには，ハロの下の名前をクリックするんだ．今回は「車」という名前にしておくといいね．

Sくん　わかりました．

M先生　描いた絵は灰色の描き直しハロを使うと描き直すこともできるよ．描き直しハロは自分で描いたオブジェクトにしかない特別なハロなんだ．星オブジェクトにはないよね．

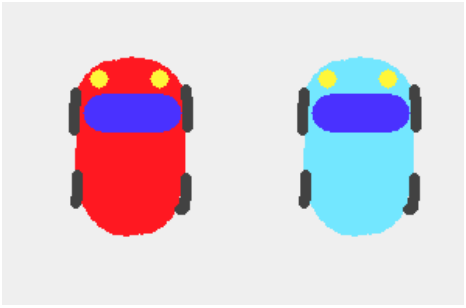
Sくん　先生，描き直しで間違えてしまったのですが…….

M先生　描き直したものを，取り消してオブジェクトに戻したい場合はパレットの「ちゅうし」ボタンを押すとできるよ．

Sくん　元に戻りました．ふー，よかった．

No. 1 – 2　やってみよう！

緑ハロと描き直しハロをうまく使って，形はまったく同じで，色違いの車を作ってみましょう．



☞ ヒント

お絵かきツールのバケツボタンを使うと，色の塗りなおしが効率よくできます．

No. 1 – 3　やってみよう！

S くんはパレットのボタンの役割がよくわからないようです．パレットのボタンの役割を調べて，S くんに教えてあげましょう．
英語版の Squeak では，図の右のようにボタンの名前がつけられています．日本語訳がどうしてわかりにくくなってしまったのかを考えてみるとよいでしょう．

No. 1 – 4　やってみよう！

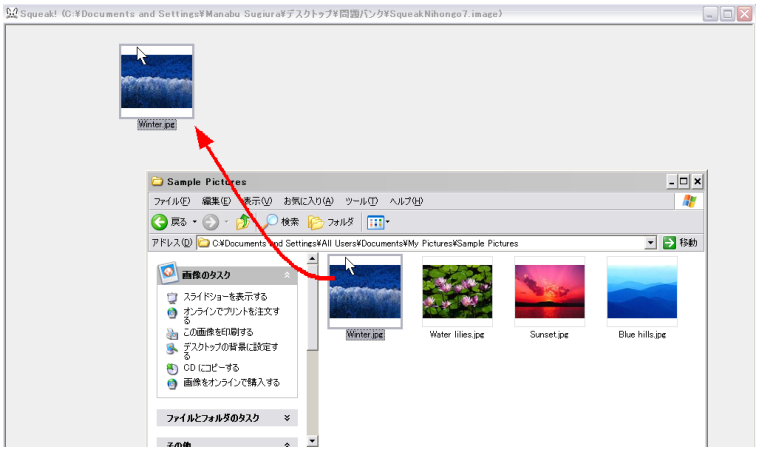
お絵かきツールには色々な機能があります．お絵かきツールを駆使して色々なオブジェクトを作ってみましょう．

☞ ヒント

P.152（A.2）にお絵かきツールの使い方の解説があります．

No. 1 – 5　やってみよう！

自分のコンピュータに保存されている画像ファイル[a]（どんな画像でもよい）を Squeak の画面にドラッグ＆ドロップしてみましょう．



[a] 名前の最後に bmp，png，jpg，gif などがついているのが画像ファイルです．画像ファイルがない場合は Windows のペイントで絵を描いて保存し，そのファイルを Squeak の画面にドラッグ＆ドロップしてみてください．

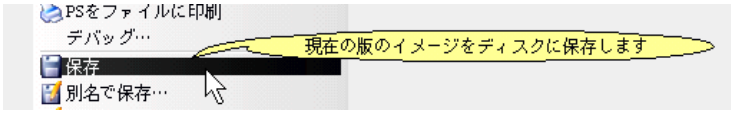
1.5　作業結果の保存

Sくん　先生，作ったものを保存したいのですが．

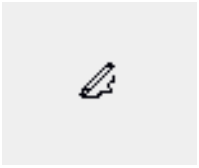
M先生　そういえば，まだ“保存”を教えてなかったね．キーボードの ESC キーを押してみてくれるかな．

Sくん　何か出てきました．

M先生　これを“ワールドメニュー”というんだ．「保存」というメニューを選択すると保存できるよ．



Sくん　マウスカーソルがペンマークになりました．



M先生　ペンマークに変わっている間は保存中だから．しばらくすると，元のカーソルに戻るよ．

Sくん　ほんとだ．

Tさん　これまで作業したデータはどこに保存されたんですか？

M先生　Squeak.image というファイルだよ．作業の続きをしたいときは，最初に教えたように Squeak.image を Squeak.exe にドラッグ＆ドロップすればいいんだ．Squeak に夢中になってしまうと保存するのを忘れがちになるので，こまめに保存することを忘れないようにね．

Sくん　気をつけます．

Tさん　先生，ワールドメニューは他にも色々なことができそうですね．

M先生　そうだね．研究してみるといいね．

No. 1 – 6　やってみよう！

Squeak には，image ファイルに全てのデータを保存する以外に，1 つのプロジェクトのデータだけをファイルに保存する方法があります．この方法を利用すると，データを保存するためのファイルのサイズを小さくすることができます．project1 だけを project1.pr というファイルに保存し，Squeak.image とサイズを比較してみましょう．

☞ ヒント

P.153（A.3）に 1 つのプロジェクトだけを保存する方法の解説があります．

Project 2

車を動かしてみよう

このプロジェクトから，いよいよオブジェクトを操作する命令の組み立てが始まります．この作業のことを「プログラミング」と呼びます．プログラミング環境として，子供たちのために設計された「Etoys」を使います．Etoys を使うと，マウスを使って「ビューアー」から「命令タイル」を取り出し，それらを組み合わせることによってプログラムを作ることできます．まずは，車をまっすぐ走らせる，曲がりながら走らせるなどの練習をしてみましょう．このプロジェクトを通して，「命令」，「繰り返し」，「順次実行」というプログラムの基本概念の習得を目指しましょう．

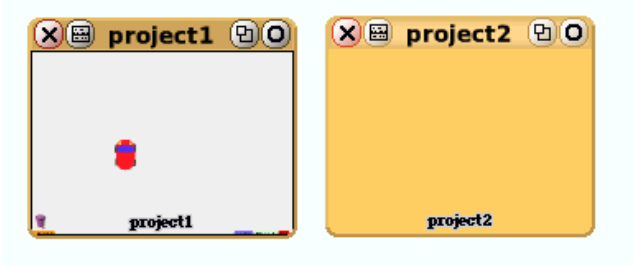
キーワード

命令，繰り返し，順次実行，ビューアー，命令タイル，スクリプト，カテゴリ，方向，重心，刻み値

2.1　ビューアーと命令の実行

M先生　では，次のプロジェクトに進もう．早速復習だけど，project2 という名前のプロジェクトを作ることができるかな？

Sくん　おっと，project1 の中に作ってしまってはだめですよね．いったん project1 から出て，新しいプロジェクトを作って……．できました．



M先生　そうしたら，project1 で作った車を project2 へコピーしてほしいんだけど．

Sくん　ん，できない……．

M先生　できないよね．プロジェクトをまたいでのオブジェクトのコピーや移動は，「私の部品」というフラップをうまく使うとできるんだ．project1 に入り，「私の部品」フラップに車をコピーして入れてごらん．

Tさん　フラップは，どこのプロジェクトでも共通なんですね．

M先生　そのとおり．車がフラップに入ったら，project2 へ移り，フラップから車を取り出せることができるね．

Sくん　はい．できました．

M先生　車をコピーしたのでオブジェクトの名前が「車 1」になってしまったね．「1」を消して「車」にしておこう．

Sくん　わかりました．

M先生　それでは車を走らせてみよう．オブジェクトに“命令”を出すときには，まず“ビューアー”を出す必要があるんだ．「水色（目玉）ハロ」をクリックしてみてくれるかな．

Sくん　ハロを出して，目玉をクリックですね．

Tさん　右側に何か出てきましたね．

M先生　それをビューアーというんだ．この中に様々な命令をするための部品が入っているのが見えるでしょ？ これを“命令タイル”というので覚えておいてね．

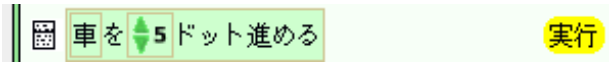
Ｓくん　命令をするものなのに，なぜ「ビューアー」というんですか？

Ｔさん　ビューアーなのだから，何かを見るためのものだと思いました．

M先生　そうだね．ビューアーは命令をするためだけでなく，オブジェクトの状態を見るためにも利用するんだ．そうした使い方については，もっと先で紹介するよ．

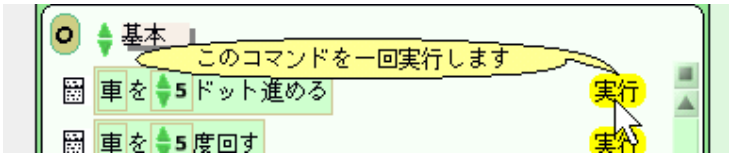
M先生　では早速命令タイルを使ってみよう．一番上に，「車を 5 ドット進める」という命令タイルがあるのがわかるかな？

Ｓくん　はい．



M先生　タイルの右側にある「実行ボタン」を押すと，この命令を 1 回実行できるよ．やってごらん．

Ｓくん　わぁ，車が前に進みました．



Ｔさん　この 5 という数値は変更できますか？

M先生　マウスで矢印をクリックしてもいいし，キーボードから数値を入力しても変更できるよ．

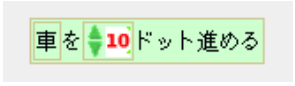
Ｓくん　先生，数値を入力したのですが，車の動きに変化がありませんね．

M先生　キーボードで入力した数値が赤いうちは編集中の状態だから，タイルの変更が確定されていないんだ．エンターキーを押せば確定するはずだよ．

Ｓくん　できました．

No. 2 – 1　やってみよう！

進める命令タイルのドット数を変更して，車がどんな動きをするか観察してみましょう．



数値の横についている矢印をマウスでクリックすると 1 ドットずつ数値が変更できます．矢印を上下方向にドラッグしてみるとどうなるかも試してみましょう．

2.2　スクリプトの作成と繰り返し

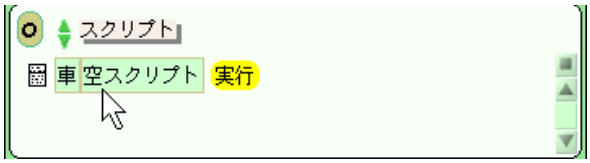
M先生　次は“スクリプト”の作成と命令の“繰り返し”に挑戦してみよう．

Tさん　スクリプトとは何ですか？

M先生　スクリプトは命令を複数組み合わせて実行できるようにするものだよ．スクリプトを使うと用意されている命令を自分で組み合わせて，新しい命令を作って実行できるんだ．

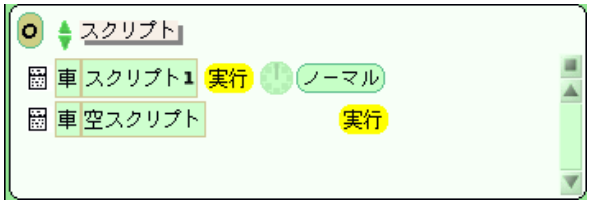
Sくん　すごそうだけど，よくわからないなー．

M先生　まずは「空スクリプト」というタイルをビューアーの外にドラッグ＆ドロップしてみて．



Sくん　スクリプト 1 という名前のオブジェクトができました．

M先生　ビューアーの「スクリプト」カテゴリの中に「スクリプト 1」というのが新しくできているはずだよ．



Sくん　本当だ．

M先生　まずは「車を 5 ドット進める」というタイルをマウスでドラッグして，スクリプト 1 の中に入れてみよう．

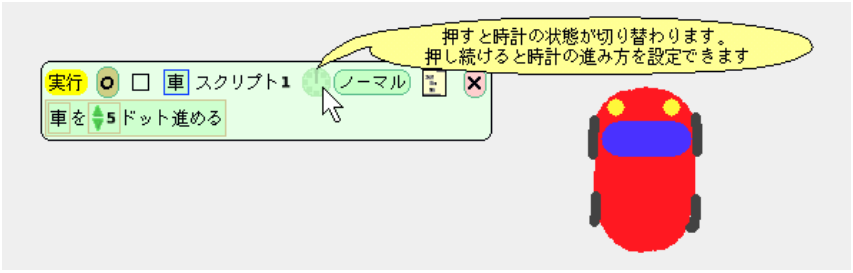
Sくん　できました．

M先生　できたら，スクリプト 1 についている実行ボタンを押してみよう．

Sくん　前に進みました．ビューアーの実行ボタンを押したときと同じ動作をしますね．

M先生　それでは，次に繰り返し実行してみよう．時計のマークをクリックしてみて．

Sくん　はい．



Sくん　おっ！ 前進するようになりました．

Tさん　「車を 5 ドット進める」という命令を何回も繰り返して実行するから，前に進み続けるんですね．

M先生　そのとおり．繰り返して実行するのを止めるためには，時計をもう一度クリックするんだ．

No. 2 – 2　やってみよう！

Sくんは間違えて，横向きの車を描いてしまったようです．このままだと，車は上に向かって進みます．この車を右に進ませるための方法を調べて，Sくんに教えてあげましょう．

> ☞ ヒント
>
> オブジェクトの進む「方向」を変更する方法の解説が P.155（A.6）にあります．

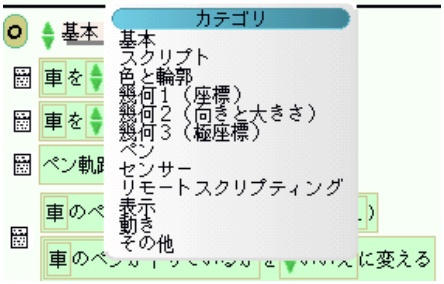
No. 2 – 3　やってみよう！

車をバックさせる方法を考えてみましょう．全部で何通りあるでしょうか？

> ☞ ヒント
>
> 車の進む「方向」を調節する方法の解説が P.155（A.6）にあります．また，進むドット数にマイナスの数値を設定することもできます．

Tさん　先生，はじめから用意されている命令は，今ビューアーで見えているものだけなんですか？

M先生　「基本」と書いてあるラベルをクリックしてごらん．他の“カテゴリ”が選択できるはずだよ．

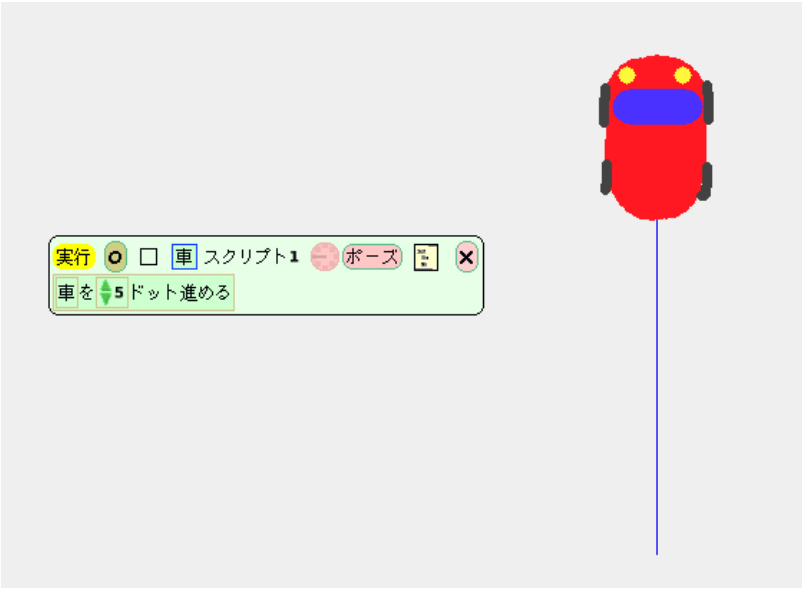


Sくん　いろんな種類のタイルがありますね．これは覚えられそうにないなー．

M先生　一生懸命覚えようとしないで，使いながらいろいろ研究してみてね．使っていくうちに徐々に覚えていくものだよ．

No. 2 – 4　やってみよう！

車が移動した軌跡を描画するようにしてみましょう．描画した軌跡を消すことにも挑戦してみましょう．



> ☞ ヒント
>
> 「ペン」カテゴリの命令を使って軌跡を描画する方法の解説がP.154（A.4）にあります．

No. 2 – 5　考えてみよう！

以下の 2 つのスクリプトを作って，車を動かしてみてください．各スクリプトによって車の動きが違う理由を考えてみましょう．

☞ ヒント

2 つのスクリプトでは刻み値が違うことがポイントです．刻み値に関する解説は P.154（A.5）にあります．

2.3　命令の組み合わせ

M先生　次は，複数の命令を組み合わせたスクリプトを作ってみよう．「車を 5 度回す」という命令を「車を 5 ドット進める」というタイルの下に追加してみて．

Sくん　おっ．車が円を描くようになりました．

M先生　スクリプトを使うと複数の命令を組み合わて，複雑な命令を作ることができるんだ．

Tさん　すごーい．

M先生　なぜ円を描くようになったかわかるかな？

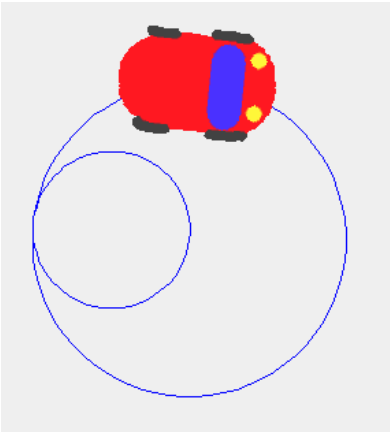
Sくん　えーっと……．

M先生　では，これは宿題にしよう．

Sくん　教えてくださいよー．

No. 2 – 6　やってみよう！

直径が倍の大きさの円を描いてみましょう．

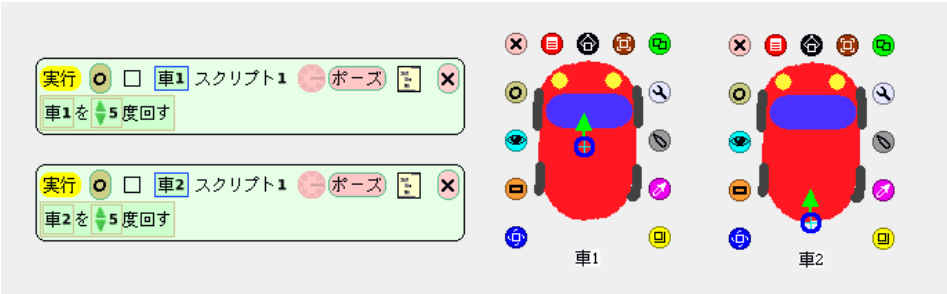


No. 2 – 7　考えてみよう！

S くんは，以下の 2 つのスクリプトを実行し，描画された円を重ねてよく観察してみると，微妙に形が違うことを発見しました．何故そうなるのか，理由を考えてみましょう．

No. 2 – 8　考えてみよう！

車 1 と車 2 があります．両方のスクリプト 1 を繰り返し実行すると，動きにどのような違いがあるか考えてみましょう．



☞ ヒント

2 つの車の重心が違うことがポイントです．重心に関する解説は P.155（A.6）にあります．

No. 2 – 9　考えてみよう！

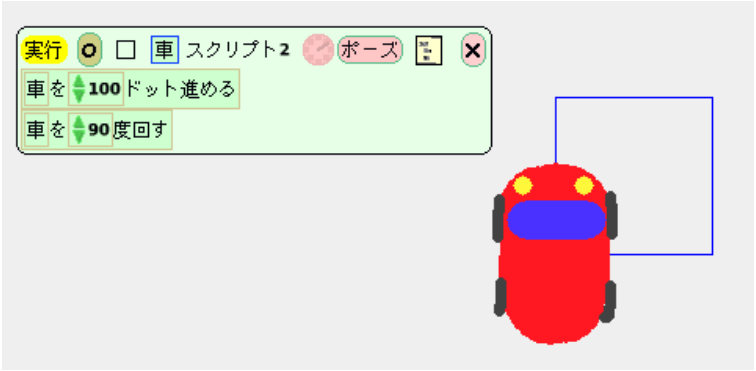
「車を 5 ドット進める」と「車を 5 度回す」という命令を組み合わせると，なぜ円が描けるのでしょうか．理由を説明してみましょう．

2.4　命令の実行順序

M先生　次は正方形を描いてみよう．できるかな？

Sくん　円を描くスクリプトとは別に新しいスクリプトを用意した方がよさそうですね．えっと，空スクリプトをビューアーの外にドラッグ＆ドロップして．よしっ，スクリプト 2 ができたぞ．

Tさん　一辺を描いて，90 度曲がるのを繰り返せばいいから……．こうですね．

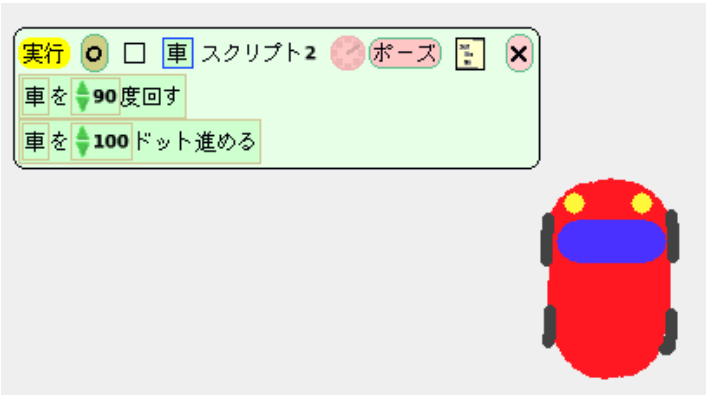


Sくん　そうか，別に 4 回だけ繰り返さなくてもいいんだ．

M先生　そうだね．この場合，100 ドット進んだ後に，90 度曲がるという動作を繰り返すことになるんだ．命令は上から順番に 1 つずつ実行されるということがポイントだよ．ちなみにこれを“順次実行”というんだ．基本だから覚えておいてね．

No. 2 – 10　**考えてみよう！**

例題で作った正方形を描くプログラムを以下のように変更してみました．プログラムを実行せずに，描画される正方形にどんな違いがあるか予想し，その理由も説明してみましょう．



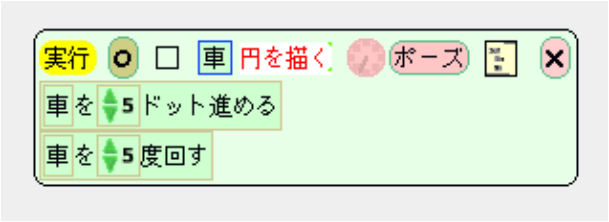
2.5　スクリプトの便利な操作

M先生　ところで，円を描くスクリプトの名前はどうなっているかな？

Ｓくん　スクリプト１のままですね．

M先生　自分で作ったスクリプトには名前を変更することができるよ．名前のところをクリックしてみてね．

Ｓくん　じゃあ，「円を描く」に変更します．



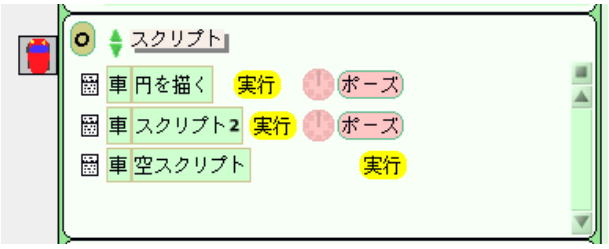
Ｔさん　スクリプトについている○ボタンはなんですか？

M先生　それをクリックすると，スクリプトをたたむことができるよ．

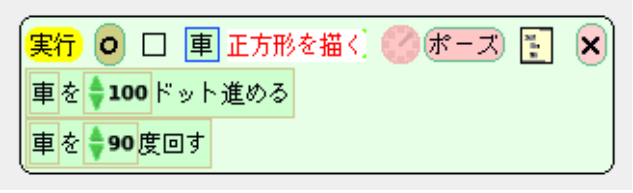
Ｔさん　たたんだスクリプトを再度表示させたいときはどうするんですか？

M先生　「スクリプト」カテゴリを選択して，表示したいスクリプトをビューアーの外にドラッグ＆ドロップすれば表示できるよ．

Ｓくん　スクリプトにわかりやすい名前をつけておくと，ビューアーで探しやすいですね．スクリプト１のままにしておいたら，たたんでしまうと中身がわからなくなってしまうや．



Ｔさん　スクリプト２も「正方形を描く」という名前に変更すれば完璧ですね．



M先生　そうだね．

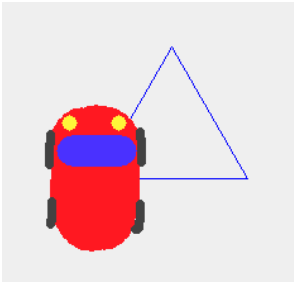
M先生　スクリプトの操作について，まとめておくよ．新しいスクリプトを作るときは，空スクリプトをビューアーの外にドラッグ＆ドロップだったね．新しく作ったスクリプトの操作に関しては以下のとおりだよ．

練習問題

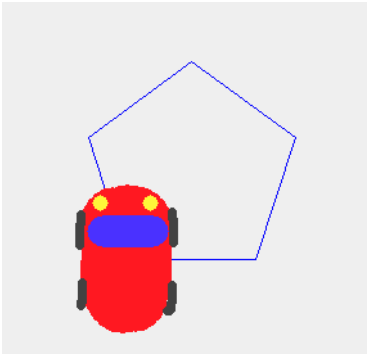
練習問題 2.1

色々な図形を描いてみましょう.

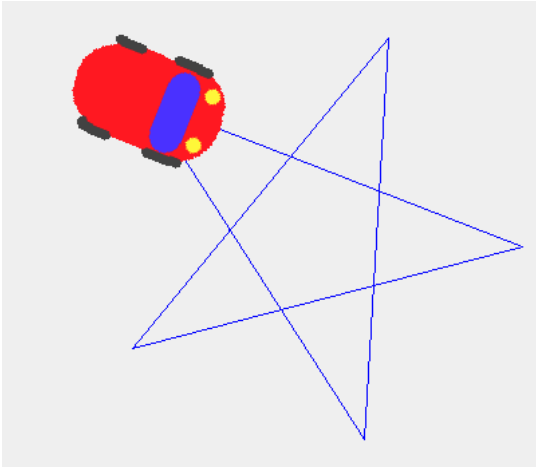
1. 正三角形



2. 正五角形



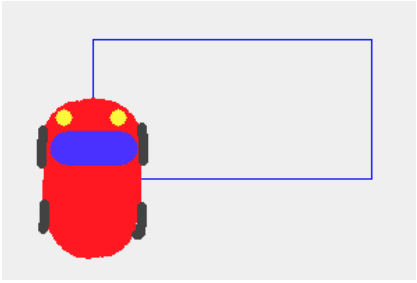
3. 星形



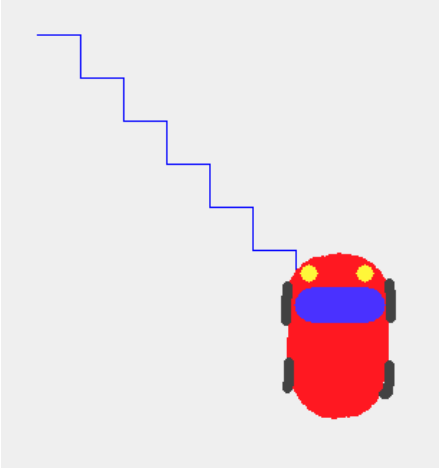
練習問題 2.2

指示された個数の命令タイル（「進める」と「回す」だけ）を使って，色々な図形を描いてみましょう．

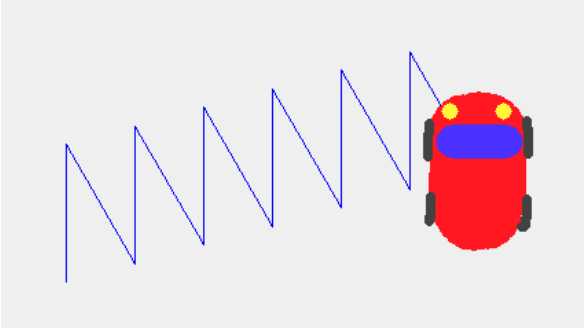
1. 長方形（命令タイル 4 個以内）



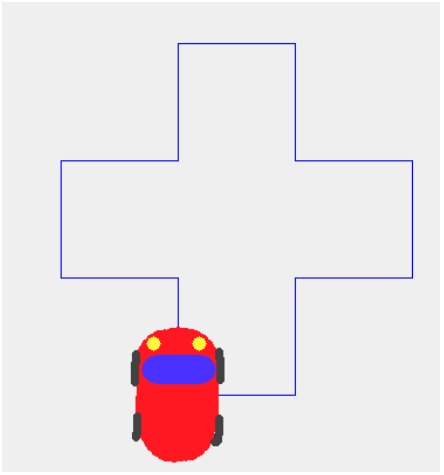
2. 階段（命令タイル 4 個以内）



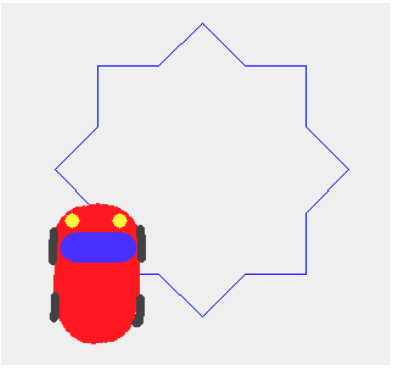
3. ギザギザ（命令タイル 4 個以内）



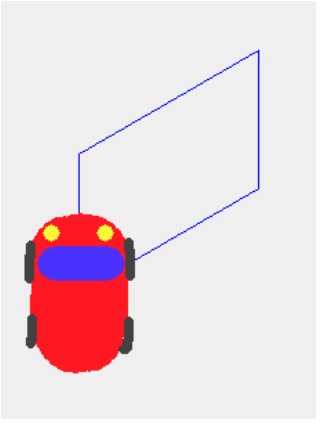
4. 十字（命令タイル 6 個以内）



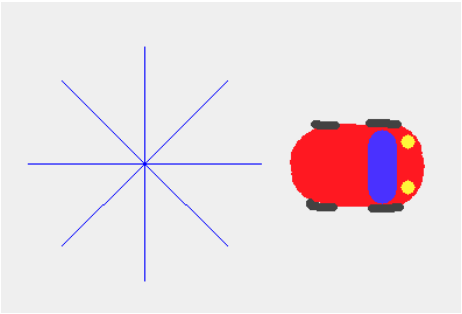
5. こんぺいとう（命令タイル 4 個以内）



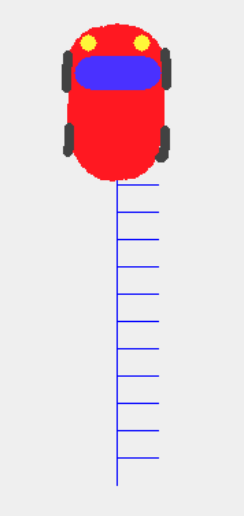
6. 平行四辺形（命令タイル 4 個以内）



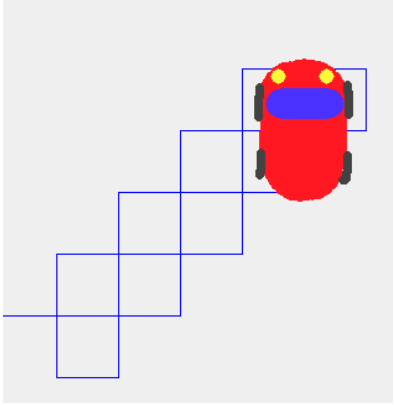
7. アスタリスク（命令タイル 3 個以内）



8. 定規（命令タイル 5 個以内）



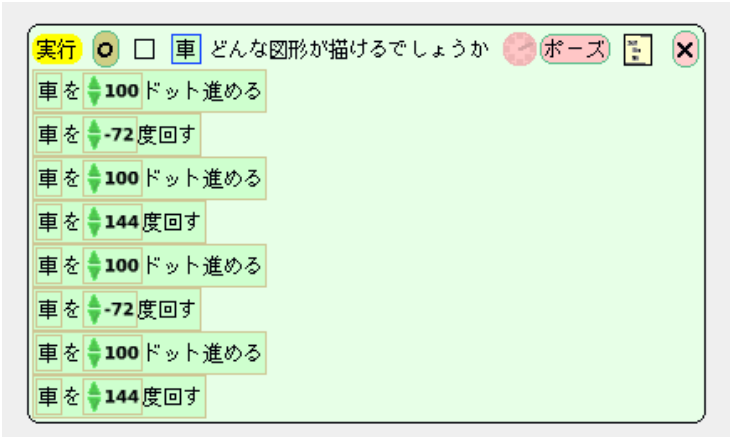
9. 四角形の帯（命令タイル 8 個以内）



10. 紹介したような問題を自分で作ってみましょう（みんなが挑戦したくなるような面白い図形だといいですね）.

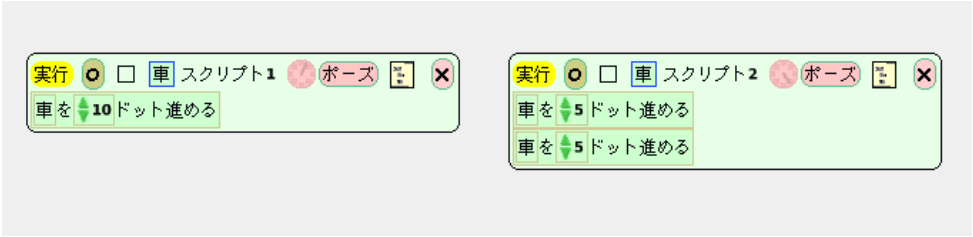
練習問題 2.3

次に示すスクリプトを繰り返し実行するとどんな図形が描けるか考えてみましょう．
どんな図形が描けるかわかったら，無駄な命令を省いて，タイル数を減らしてみましょう．



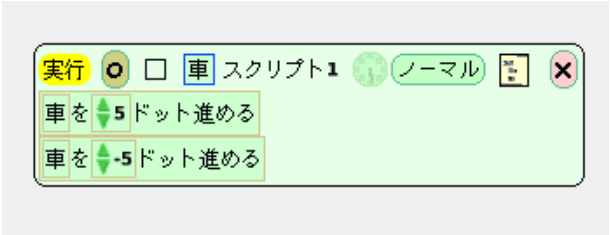
練習問題 2.4

以下の 2 つのスクリプトを繰り返して実行すると，車の動きに違いがあるでしょうか．プログラムを実行する前に予想をして，その理由を説明してみましょう．



練習問題 2.5

以下のスクリプトを繰り返して実行すると，車はどのような動きをするでしょうか．プログラムを実行する前に予想をして，その理由を説明してみましょう．



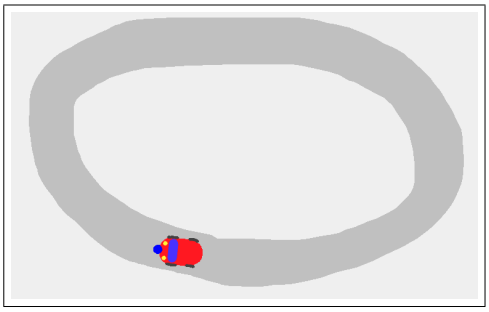
練習問題 2.6

本物と同じように時を刻むアナログ時計を作りましょう．

> ☞ ヒント
>
> 針の回転軸を変更するためには，重心を調整する必要があります．重心の変更方法は P.155（A.6）に解説されています．
>
> 刻み値を変更すると針が回転する速度を調整することができます．刻み値については， P.154（A.5）を参考にしてください．本物の時計と同じように針を動かすためには秒針，長針，短針が 1 秒間に何度動くかを計算する必要があります．

Project 3

車を道に沿って走らせてみよう



このプロジェクトでは，直線の道を折り返したり，円形の道に沿って走ったりする少し知的な車をプログラミングすることに挑戦します．これを実現するためには，道を判別する「センサー」と「場合分け」という概念を学習する必要があります．

プログラムもだんだんと複雑になってくるので，それを整理するために「フローチャート」という図法を活用します．まずは既にあるプログラムをフローチャートで表現することで，プログラムの構造を理解してみましょう．次に実行前のプログラムのフローチャートを見て，どんな動作をするのか予想してみましょう．

キーワード

センサー，場合分け，フローチャート，入れ子，場合分けタイル

3.1　場合分け（1）– 各場合に実行する命令が 1 つのとき

M先生　新しく project3 という名前のプロジェクトを作ってもらえるかな．今まで使ってきた車も用意しておいてね．

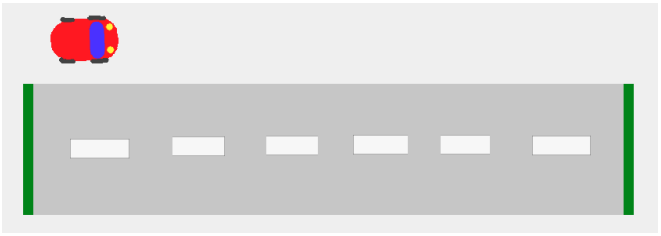
Sくん　はい．

M先生　まずは道路を往復する車を作ってみよう．道の端に車が来ると，折り返して走り続けるようにしてみようか．

Tさん　すごい．そんなことができるんですね．

M先生　先に必要な部品を作ってしまおう．まず横向きの直線道路をお絵かきツールで描いてくれるかい？ 後で必要になるから，道路の両端に印をつけておこう．あとは道路の真ん中にセンターラインも入れておこう．車は青ハロを使って横向きにしておいてね．

Sくん　はい．道路の両端の印は緑色で描いておきました．こんな感じですかね．



M先生　いいね．次は車を少し改造するよ．道の端に来たかどうかを判別するために車に“センサー”をつけよう．

Ｓくん　えっ，また車を描くのかー．面倒くさいなー．

Ｔさん　描き直しハロを使えば，センサーだけを描き足すことができるよ．

Ｓくん　おっと．そうか．

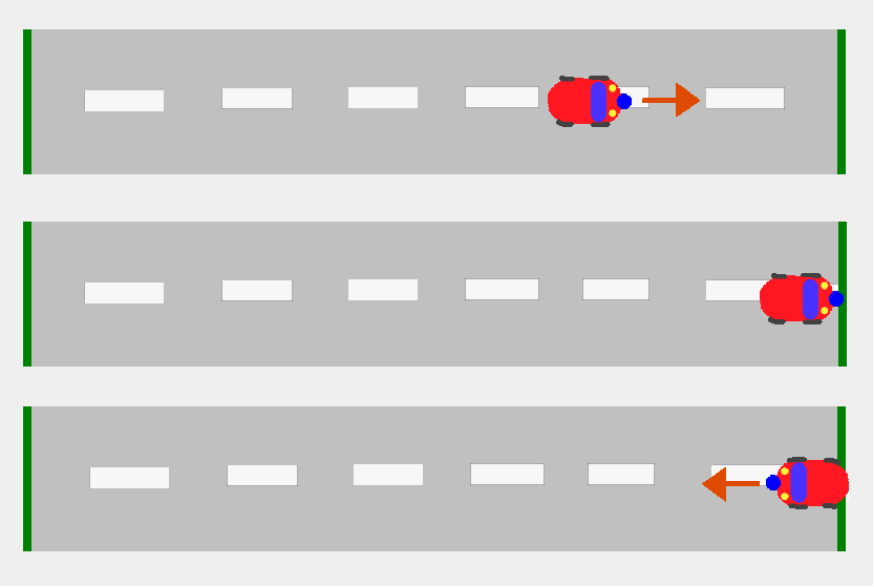
M先生　センサーは車のボディの色と違う色にするのがポイントだよ．詳しくは後で説明するけれど，こうしておけば車が道の端に来たかどうか検知できるんだ．

Ｓくん　センサーをつけ終わりました．

M先生　よし，これで部品は全部そろったね．次は完成させたい車の動きを図で確認してみるよ．

Ｔさん　はい．

M先生　次の図をよく見てね．

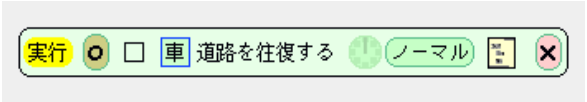


M先生　車は道に沿って直進するんだけど，道の両端に来たらそれまで走っていた方向と逆向きに反転して，また直進．これを繰り返したいんだ．

Ｓくん　先生，動きはよくわかりましたが，どうやってプログラムするかはわからないです．

M先生　まあまあ，慌てないで．これから一緒にやっていくからね．まず，車に「道路を往復する」という名前のスクリプトを作ってね．

Ｓくん　はい．できました．



M先生　さて，今回のプログラムは，「道路を走っているとき」と「道路の端に車が来たとき」で異なる命令を実行する必要があるんだ．

Ｔさん　道路を走っているときは進めて，端に来たら曲がるんですね．

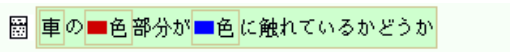
M先生　そのとおり．場合に応じて異なる命令を実行したいときは“場合分け”のタイルを使う必要があるんだ．

Ｓくん　先生，その場合分けのタイルはどこにあるんですか？

M先生　場合分けをするためには，今どんな状況かを調べることが必要なんだ．だから，「〜を調べる」というタイルが必要になる．

Ｓくん　でも，「センサーが道の端を検知したかどうか調べる」というタイルはないなー．

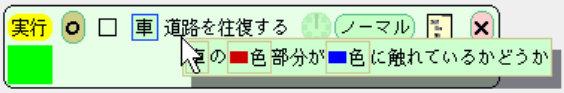
M先生　そうなんだ．この場合は，「センサーが道の端を検知したかどうか調べる」ことをもっと具体的にしなければいけないんだ．今回は「色を調べる」タイルを使ってみよう．「部分が色に触れているか」というタイルがあるよね．



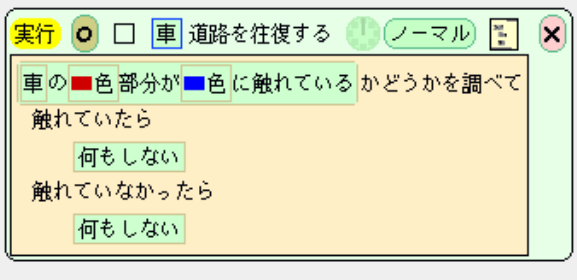
Ｔさん　なるほど．このタイルを使えば「車のセンサー（青色）が，道の端の印（緑色）に触れているか」を調べることができますね．

M先生　そのとおり．では実際にやってみよう．「部分が色に触れているか」タイルを「道路を往復する」スクリプトに入れてみて．

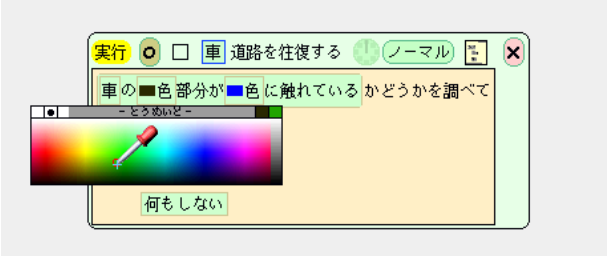
Ｓくん　はい．



Ｔさん　わぁ，タイルが変化しましたね．



M先生　次に色の設定をしよう．まず左側の色から．左側の色の部分をクリックすると，色を選ぶスポイトが表示されるよね．



Sくん　このパレットから色を選べばいいんですね．センサーの色にするのが微妙で難しいなー．

Tさん　そんなことせずに，センサーを直接スポイトで選択すればできるよ．

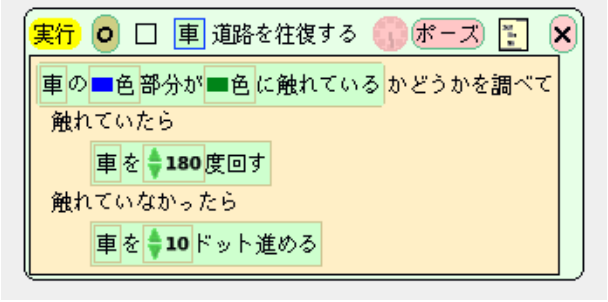
Sくん　おっ．簡単だー．

M先生　あとは右側も同じ要領で，道路の端の印の色（緑色）を選択すればいいね．

Tさん　残るはそれぞれの場合に実行する命令ですね．「何もしない」と書いてある部分にタイルが入るんですね．

Sくん　えっと，反転させるためには回すタイルを使えばいいから……あれ？

Tさん　先生，私の方はできました．



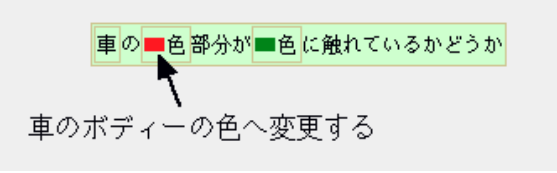
No. 3 – 1　やってみよう！

道路の両端に印をつけないで，同じ動き方をする車を作ってみましょう．

No. 3 – 2　考えてみよう！

例題で作ったプログラムに以下のような改造を加えると，車はどのような動きをするか予想し，その理由を説明してみましょう．

・色に触れているかの判定タイルの左側の色を車のボディーの色に変更する



・車を進めるドット数を大きくする．（ドット数がある値以上になると，車はどのように動くでしょうか）

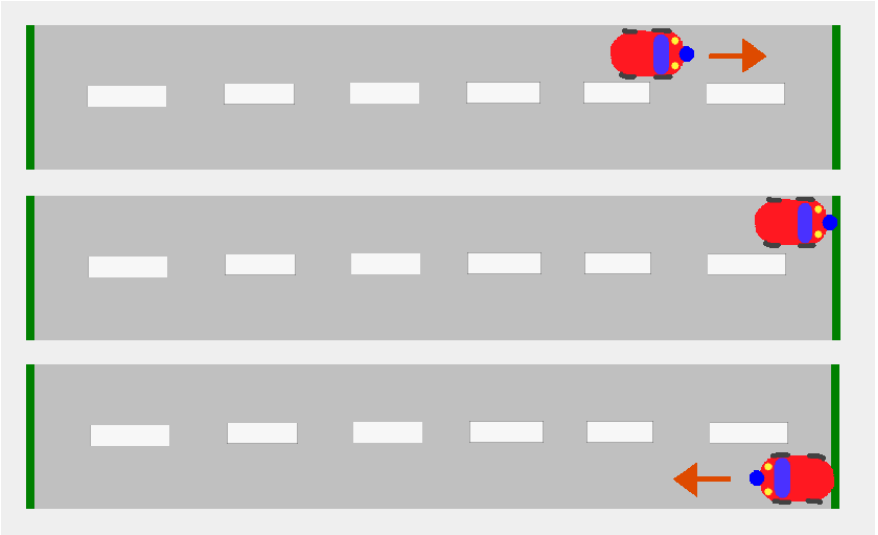
☞ ヒント

どうしても理由がわからないときは，「実行ボタン」を押してスクリプトを一回ずつ実行し，車の動きを観察してみましょう．

3.2　場合分け（2）– 各場合に実行する命令が複数のとき

M先生　これまで車は一直線に走っていたけれど，次は道の端で折り返すときに車線を変更するようにしてみよう．

Sくん　車の動きを図にするとこんな感じですね．こうすれば車が 2 台になっても正面衝突しなくなりますね．

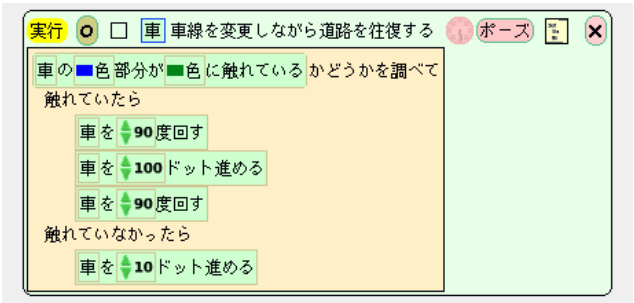


M先生　よし，ではプログラムを組み立てていこうか．

Tさん　スクリプトを新しく作る必要はありますか？

M先生　そうだね．「車線を変更しながら道路を往復する」というスクリプトを作ろう．

Tさん　車線を変更してから反転させればいいのだから……．



Tさん　先生，できました．それぞれの場合に実行する命令を入れる部分には，複数の命令を入れることができるんですね．

M先生　そうだね．複数の命令があれば上から順番に実行されるんだ．

Sくん　反転してから車線を変更してもいいですよね？

M先生　そうだね．ただ，反転してから車線を変更するとプログラムがややこしくなるはずだよ．

Sくん　えっ？　なんでだー？？？

M先生　じゃあSくんは反転してから車線を変更するという方法でやってみたら？

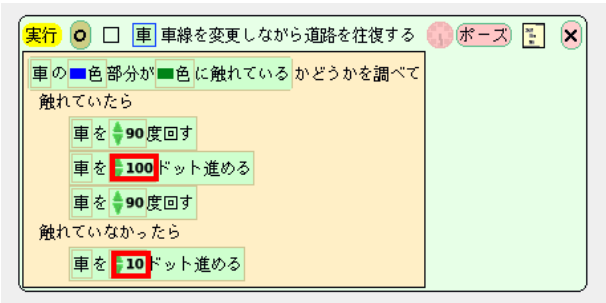
Sくん　うーん．できないよー．

No. 3 – 3　考えてみよう！

Sくんが選んだ，「反転させてから車線を変更する」方法ではなぜプログラムがややこしくなるのでしょうか．理由を考えてみましょう．

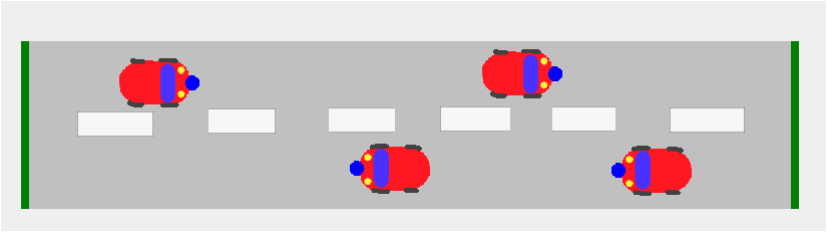
No. 3 – 4　考えてみよう！

例題として作った以下のスクリプトをよく見てください．進める命令タイルが2つ入っていますね．四角で囲んだ2つのドット数（100ドットと10ドット）にはどんな意味があるでしょうか．道は縦幅より横幅の方が長いはずです．縦方向に走るときのドット数（100ドット）の方が，横を走るときのドット数（10ドット）より大きいのはなぜでしょうか．



No. 3 – 5　やってみよう！

車をコピーし，複数の車を道路に走らせてみましょう．オブジェクトをコピーすると，スクリプトも一緒にコピーされます．コピー元のオブジェクトのスクリプトを実行したままコピーするとどうなるかも試してみましょう．コピーしたオブジェクトのスクリプトを削除してもコピー元のオブジェクトのスクリプトには変化がないことを確認しましょう．



☞ ヒント

複数のオブジェクトのスクリプトを一度に実行する場合は，部品フラップにある「全スクリプト」ツールが役に立ちます．P.156（A.7）に使い方の解説があります．

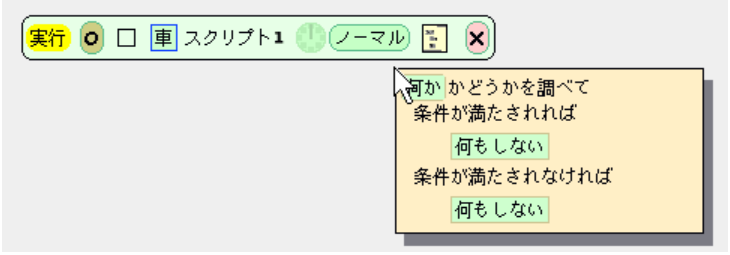
M先生　ちなみに，センサーカテゴリのタイルがなくとも，場合分けタイルだけを取り出すこともできるんだ．

Sくん　どうやるんですか？

M先生　スクリプトについている時計の横の四角いアイコンをクリックしてみて．

Sくん　はい．

M先生　出てきたタイルをスクリプトの中に入れて，「何か」の部分に調べたい事柄を表現したタイルを入れればいいんだ．

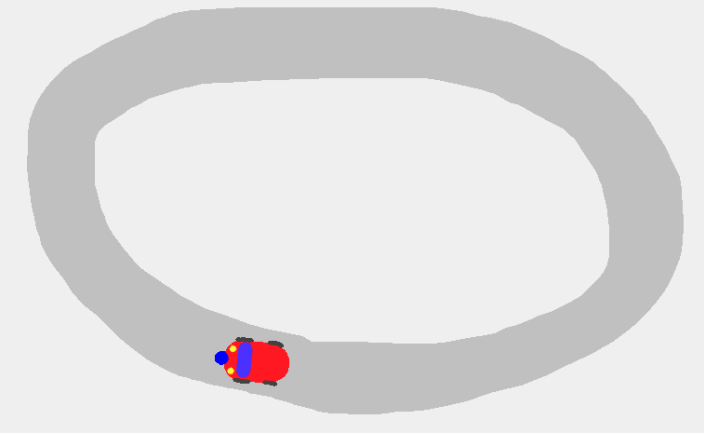


Tさん　ちょっと面倒くさいですねぇー．

M先生　あまり使わないと思うけど，念のため説明しただけだよ．

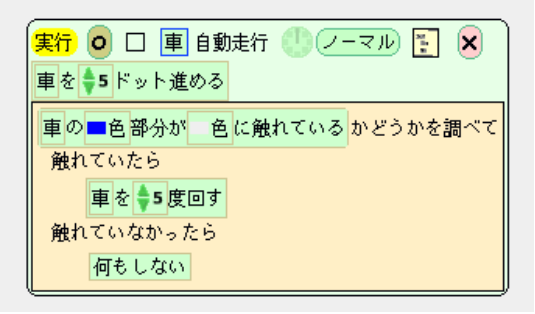
3.3　フローチャート（1）– 場合分けが 1 つのとき

Sくん　先生，今度はコースを丸くしてみて，どっちが先に道に沿って走る車を完成できるか勝負してます．

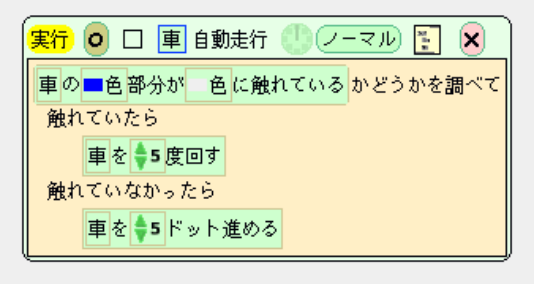


M先生　なるほどね．ちょっとスクリプトを見せてくれるかな？

Sくん　僕のはこうです．



Tさん　私はちょっと違います．



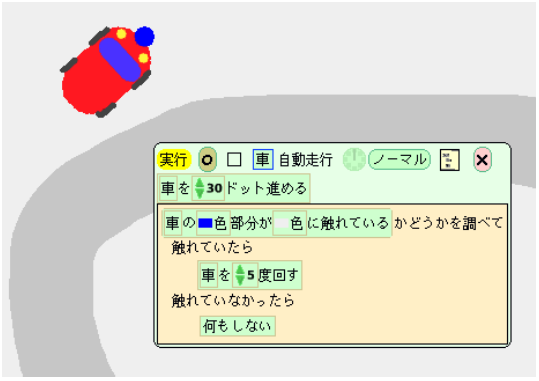
M先生　なるほど．直線道路のプログラムと違うのは，センサーの部分が道路の外にあるかを調べているところだね．どちらのプログラムでも車はちゃんと道に沿って走れそうだね．

M先生　では，どちらが早くコースを一周できるか競争してみよう．車が速く走るようにプログラムを改造してもいいよ．

Sくん　車を速くするためには，進むドット数を大きくすればいいんだ．30 ドット位にしてみようかな．

Tさん　私も．

Sくん　あっ，進むドット数を 30 にすると，僕の車はコースアウトしてしまいました．



Tさん　私の車はコースアウトしませんね．

M先生　どうして違う結果になったのかな？

Tさん　車が道から外れたときの動きが違うのかも……．

Sくん　僕のプログラムでは，車を進める命令が場合分けタイルの外にあるよね．

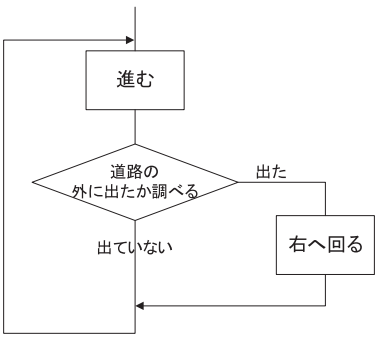
Tさん　そうね．車が道から外れたときの動作の違いが問題のようね．

Sくん　場合分けタイルの外に進める命令があると，道から外れた状態でも車は進んでしまうのか．

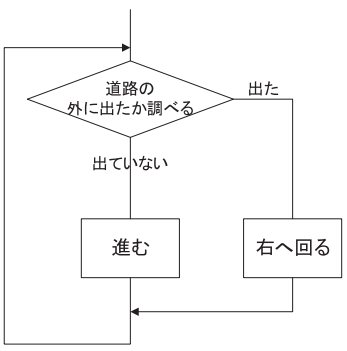
Tさん　私の場合は，車が道の方向に戻るまで進まないみたいね．

M先生　2 人ともいい線いってるね．では，より整理して考えるために“フローチャート”という図法を紹介しよう．2 人のプログラムをフローチャートで表現するとこうなるね．

Sくんのプログラム



Tさんのプログラム



Sくん　微妙に違いますね．

M先生　フローチャートでは，ひし形が場合分けを表していて，四角が命令を表現しているんだ．どこが違うかわかるかな？

Sくん　僕のプログラムは，まず進めるの命令が実行されてから，場合分けが実行されるんだ．だから，進むドット数が大きすぎるとコースアウトしてしまうんですね．

Tさん　私のプログラムは，車が道から外れている場合は，回るだけです．だから進めるドット数を大きくしても大丈夫なんですね．

M先生　そうだね．Sくんのプログラムは，場合分けタイルの外に進めるという命令があるよね．これだと車が道から外れているかに関係なく，常に進める命令が実行されてしまうんだ．

Sくん　なるほど．だから進めるドット数を大きくすると，コースアウトしてしまうんですね．

M先生　今回は既に作ってあるプログラムを整理するためにフローチャートを使ったけれど，プログラムを作る前にフローチャートを書いておくと，プログラムがどのように動作するかを考えやすくなるよ．

No. 3 – 6　やってみよう！

3.1 と 3.2 で作った 2 つのプログラム（道路を往復する，車線を変更しながら道路を往復する）について，それぞれのフローチャートを書いてみましょう．

3.4　フローチャート（2）– 場合分けが 2 つのとき

Sくん　うーん．車を道の真ん中を走らせるためには……．

Tさん　どうしたの？

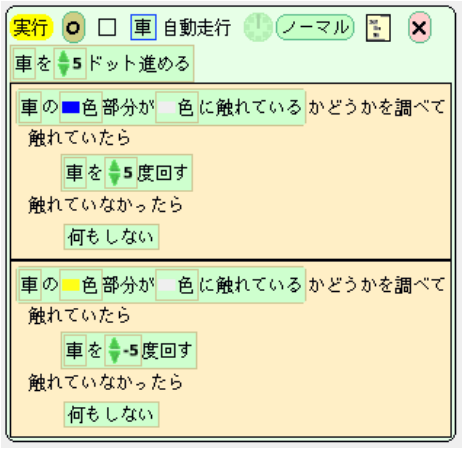
Sくん　今までの車では道の端を走ってしまうから，道の真ん中を走らせようとしているんだ．

M先生　それなら色違いのセンサーを 2 つ用意するのがオススメだね．こんな感じかな．

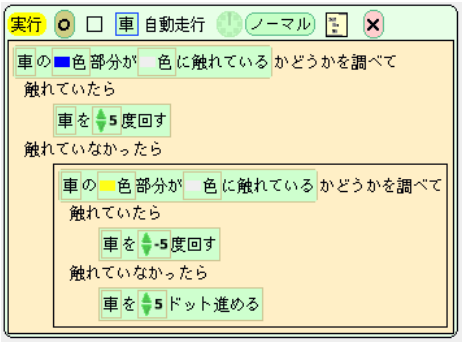
Sくん　そうか．そうすると場合分けタイルが 2 つ必要になるのか．

M先生　プログラムを組み立てたら，車を走らせる前に先生に見せてくれるかな？

Sくん　僕のプログラムはこうなりました．



Tさん　私のはこうなりました．



Sくん　わー，場合分けタイルの中に場合分けタイルが入るんだ．はじめて知ったよ．

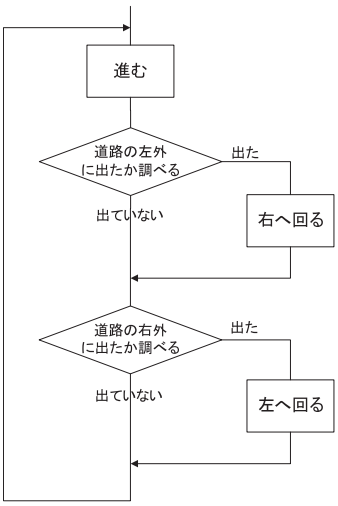
M先生　そうだね．こういう構造を“入れ子”と呼ぶよ．

Tさん　入れ子って，箱が箱の中に入っているイメージね．

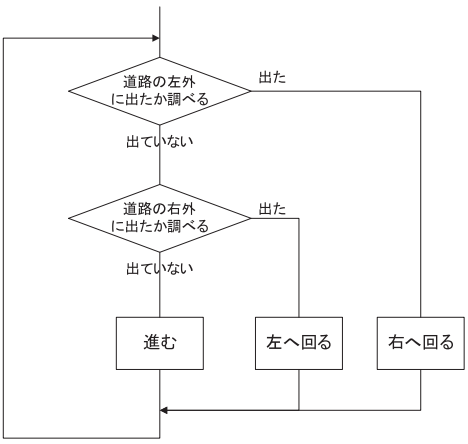
M先生　それでは，それぞれのプログラムをフローチャートで書くとどうなるか，書いてみてくれるかな．

Tさん　こうでしょうか．

Sくんのプログラム



Tさんのプログラム



M先生　OK．では，この 2 つのフローチャートを見て，それぞれの車の動きが違うかどうか検討してみよう．

Sくん　えっと，僕のは前に作ったのと同じで，必ず進む命令が実行されますね．だから，進めるドット数を大きくするとコースアウトしてしまうと思います．

Tさん　私のプログラムは道からはみ出ていないときだけ車が進むから，S くんの車と違ってコースアウトはしませんね．

Sくん　実行する前からコースアウトすることがわかるなんて……．

M先生　では実験してみよう．

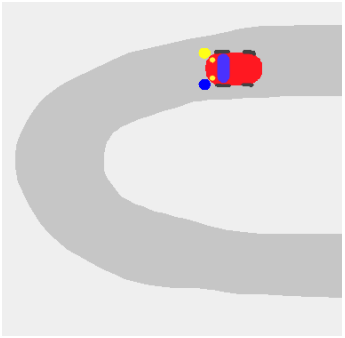
Tさん　やっぱり進めるドット数を大きくすると S くんの車はコースアウトね．

M先生　まぁまぁ．フローチャートを書くことで，実行する前からプログラムの動作を予測することができたね．S くんだってどのように直せばコースアウトしないかわかったはずだね．

S くん　はい…….

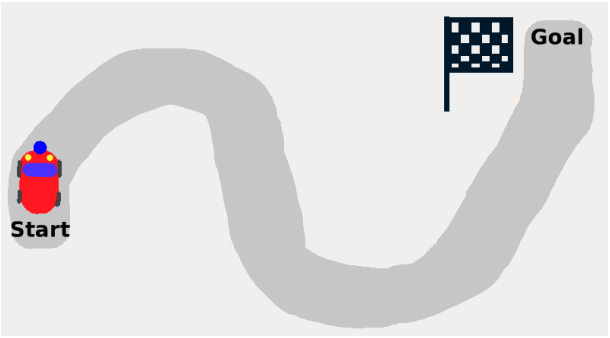
No. 3 – 7　考えてみよう！

以下のようなヘアピンカーブのあるコースをうまく走るためにはどうしたらよいでしょうか．



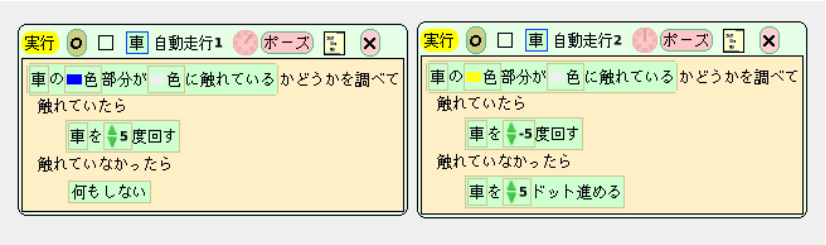
No. 3 – 8　考えてみよう！

以下のようなコースはセンサーが 1 つの車でうまく走れるでしょうか．うまく走れるかどうかを予想し，その理由を説明してみましょう．



No. 3 – 9　考えてみよう！

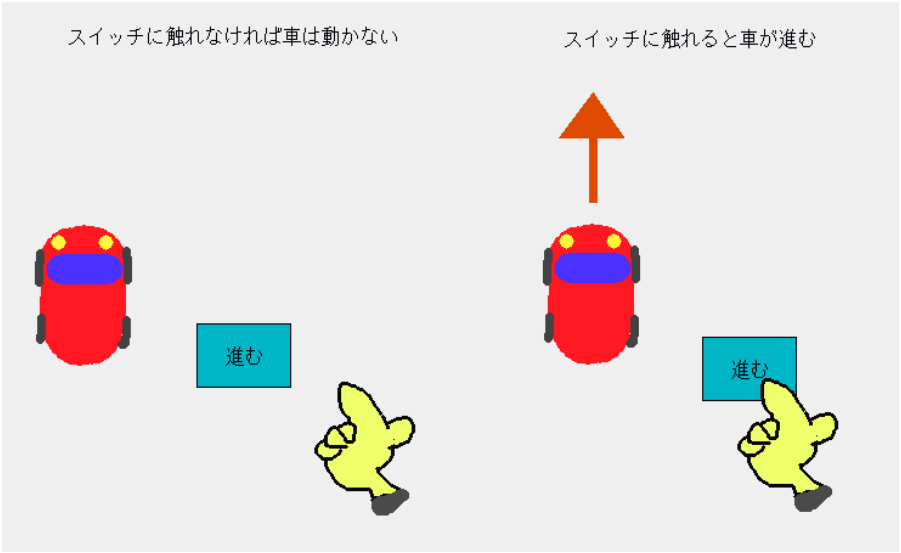
例題で作ったプログラムを，以下のように別々のスクリプトに分割し，同時に実行すると車はどのような動きをするでしょうか．



練習問題

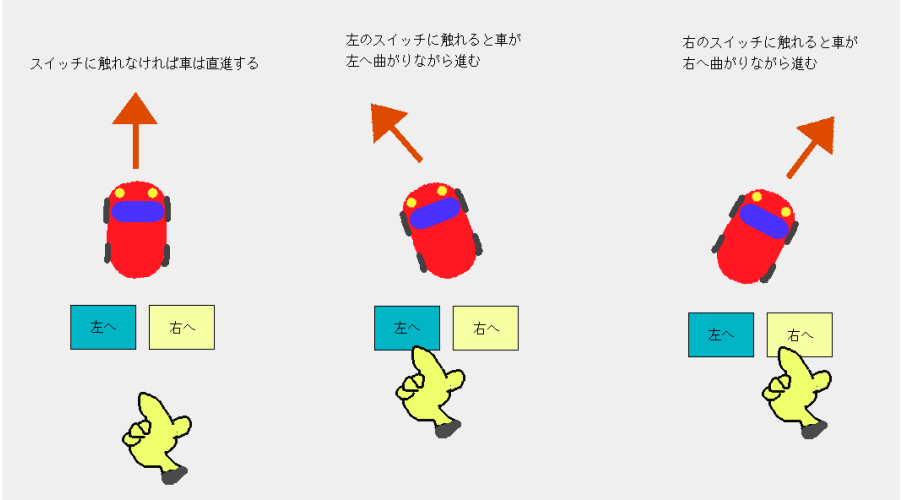
練習問題 3.1

車を走らせたり，止めたりできるスイッチを作りましょう．



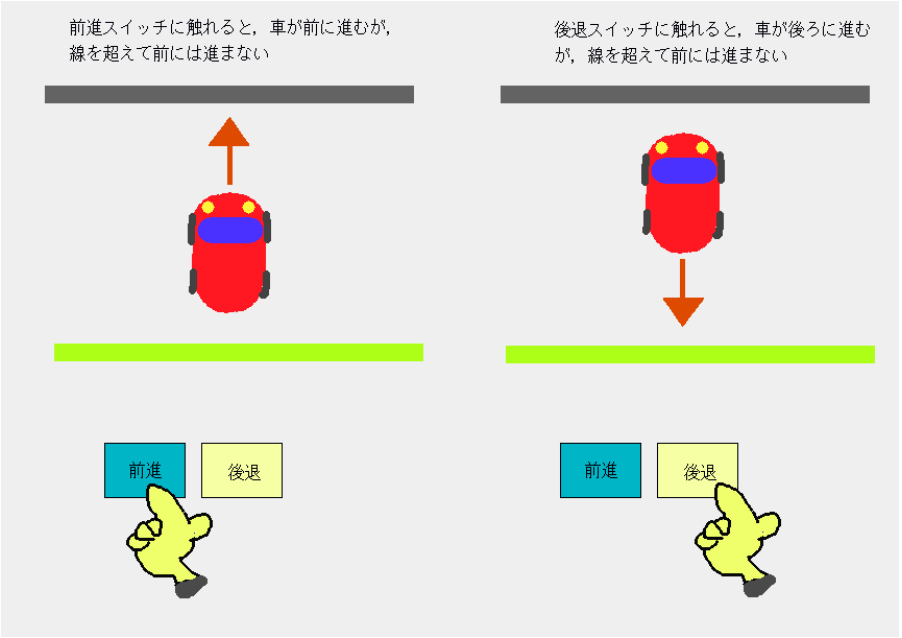
練習問題 3.2

車を左右に操縦できるスイッチを作りましょう．



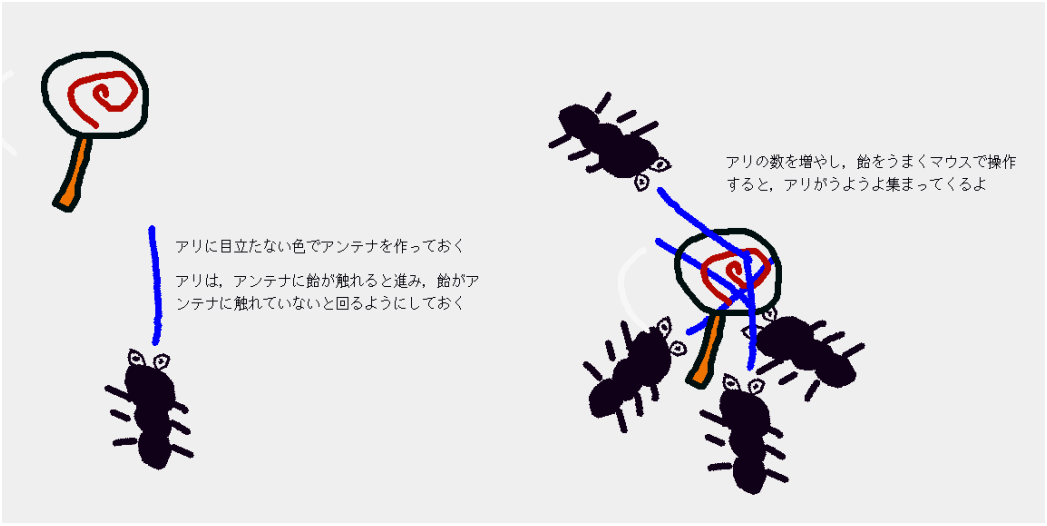
練習問題 3.3

決められた範囲内で車を前後に操縦できるスイッチを作りましょう.



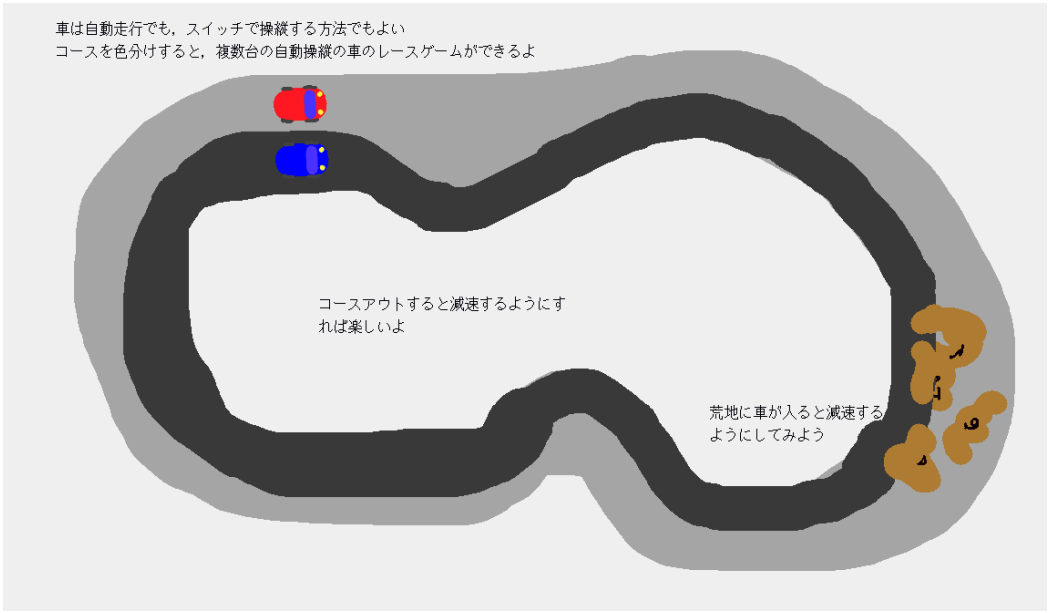
練習問題 3.4

アリ集めゲームを作りましょう.



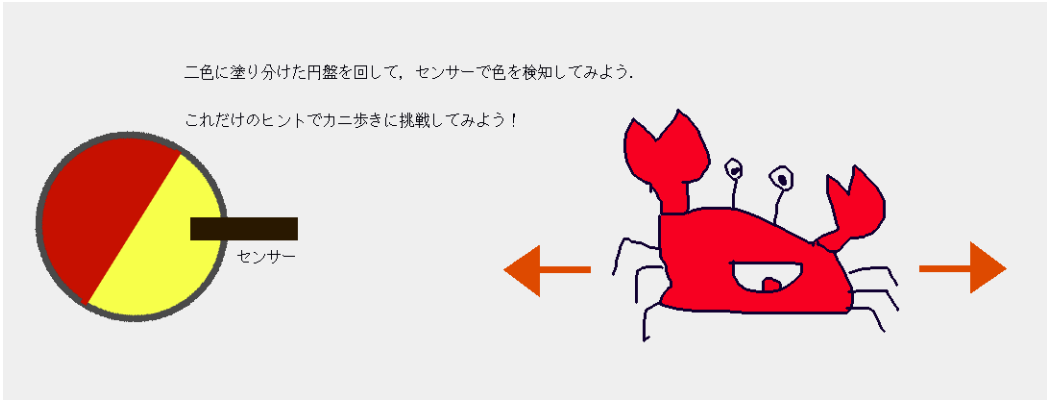
練習問題 3.5

複数の車を競争させることができるレースゲームを作りましょう.



練習問題 3.6

一定の幅で左右に移動する反転するカニを作りましょう.



Project 4

障害物を作ってみよう

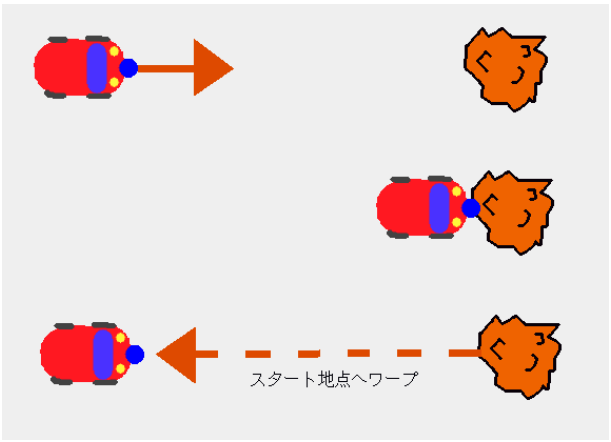
このプロジェクトでは，車のワープや障害物の製作を通じて，オブジェクトの持っている「変数」とその「値」という概念を学びます．変数の値を変更するための「変更タイル」を使えば，「座標」変数を使ってオブジェクトを動かすことができます．また，「変数タイル」を場合分けと組み合わせることにも挑戦します．

キーワード

変数，値，座標，変数タイル，変更タイル，ワープ，ボタン，見た目を似せる，テキスト

4.1　変数

M先生　さて，今度は project4 だね．このプロジェクトでは，障害物の岩石を作ってみよう．まずは岩石が車に衝突すると，車がスタート地点に“ワープ”するようなものを作ってみよう．



M先生　車をワープをさせるためには，ビューアーの新しい使い方を勉強する必要がある．車のビューアーを開いてごらん．

Sくん　はい．

M先生　これまでは基本やセンサーといったカテゴリを使ってきたけれど，今回は「X 座標」や「Y 座標」と書かれたタイルを探してくれるかな？

Ｓくん　ありました．

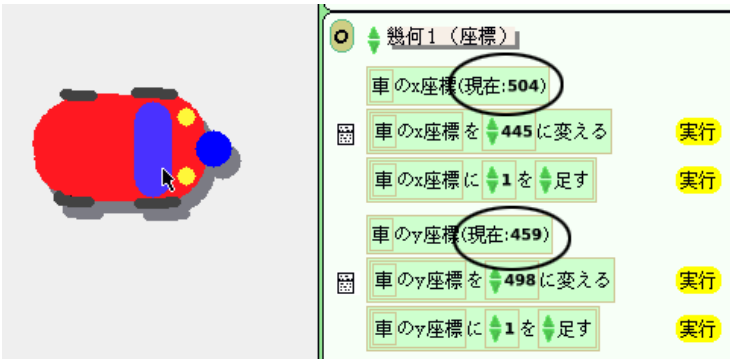
M先生　これを“変数タイル”と呼ぶんだ．車をマウスで動かすとタイルの表示はどうなるかな？

Ｓくん　数値が変わりますね．今までは気にしてなかったけど，「方向」とか，他にも色々な変数タイルがあるんですね．

Ｔさん　あっ，オブジェクトの変数を見るためのものだから，ビューアーという名前なんですね？

M先生　そのとおり．では，車をマウスでつかんで動かすと“座標”の“値”はどうなるかな？

Ｓくん　車の位置によって変化しますね．



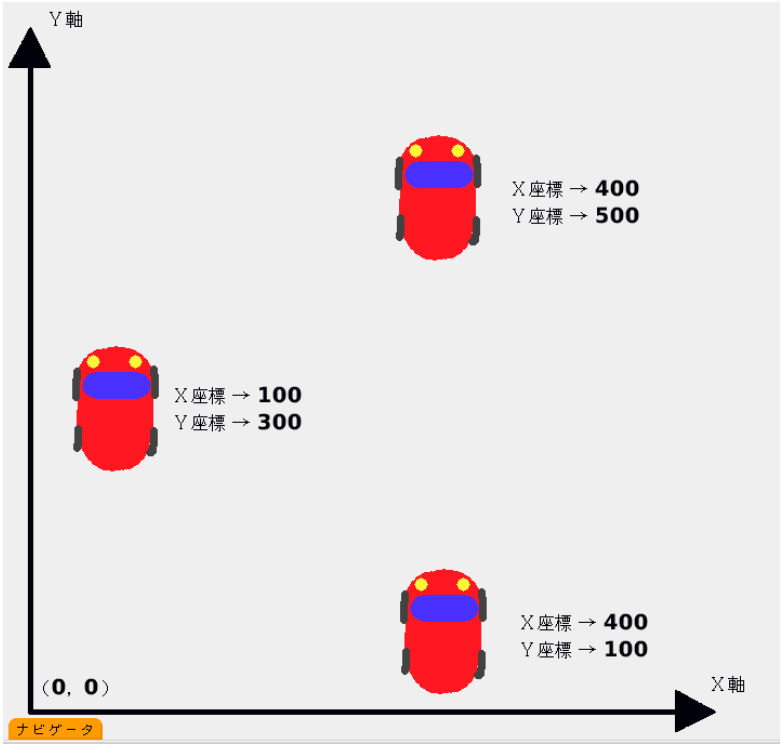
M先生　色々動かして，Squeak の画面の座標がどうなっているかを調べてみて．

Ｓくん　はい．

Tさん　X座標は横方向の位置を表しているようですね．Y座標は縦方向です．

Sくん　画面の右上に行くにしたがって数値が増えていくから，左下が（0,0）となりますね．

Tさん　図で描くとこんな感じかな．



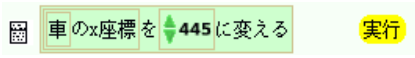
Sくん　数学でやったXとYの座標軸とおんなじだ．

4.2　変数と値の変更（1）– 値指定

Sくん　でもこれが，ワープと何の関係があるんですか？

M先生　マウスで車を動かすと，座標の値が変わるのがわかったよね．逆に座標の値を変更できれば，オブジェクトを移動させることができるんだ．そのためのタイルが，変数の“**変更タイル**”だよ．

Tさん　これですね．



M先生　この命令を実行すると，X座標の値が設定した数値に変更されるよ．

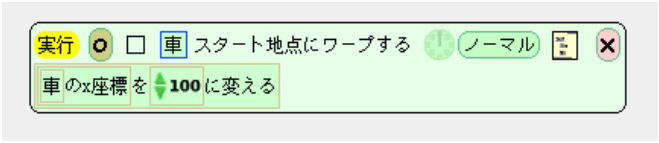
Sくん　つまり，車がワープするんですね．

M先生　それでは，はじめの練習として，実行すると車がスタート地点にワープするようなスクリプトを作ってみよう．

Sくん　えっと，スタート地点のX座標がわからないや．

Tさん　車をスタート地点に移動させて，ビューアーで確認すればいいじゃない．

Sくん　そうか．えっと，スタート地点の X 座標は 100 だから……こうですね．

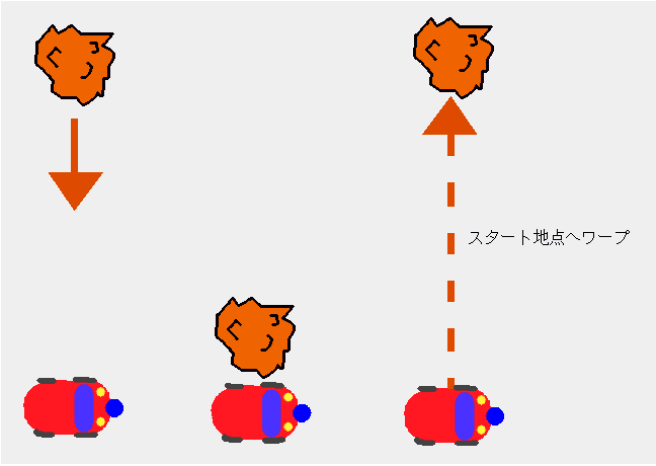


M先生　じゃあ次は，車が岩石と衝突したか調べて，衝突していたらワープするようにすればいいだけだよ．これは前の project の復習だね．

Sくん　はい．できました．

No. 4 – 1　やってみよう！

車を固定し，上から落下してくる岩石が車に衝突したら，岩石が落下開始地点に戻るようにしてみましょう．



No. 4 – 2　やってみよう！

車の重心を動かして，座標との関係を調べてみましょう．

☞ ヒント

X，Y 座標の値はオブジェクトのどの点の値でしょうか．重心を変更すると X，Y 座標の値はどのように変わるか試してみるとよいでしょう．重心についての解説は P.155（A.6）にあります．

4.3　変数と値の変更（2）– 差分指定

M先生　何をしているの？

Sくん　岩石が置いてあるだけではつまらないので，落石みたいにしようかと思って……．

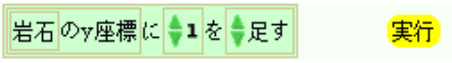
M先生　いいねえ．うまくできるかな？

Sくん　回転しながら落下するようにしたいんですが，「回す」と「進める」を組み合わせても，やっぱり円を描くだけなんですよ．

M先生　ではヒントを出そう．「進む」タイルを使わないでオブジェクトを移動させることはできないかな？

Tさん　さっき習った座標を使うのではないですか？ Y 座標をだんだんと減らしていくと，岩石は画面の上から下に進むのではないかしら．

Sくん　そうか，こっちの変更タイルを使えばいいのか．



Ｓくん　あれ，岩石が真横に動くー．

Ｔさん　Ｙ座標を減らせば下に落ちていくはずでしょ．

Ｓくん　Ｙ座標から 5 を引くことを繰り返せばいいんだ．「足す」とか「引く」とかは変更できるんだね．できた．落石っぽい動きになったよ．

Ｔさん　あれ，お気に入りの落石が画面の外に行っちゃったよ．どうしよう．先生，助けてください．

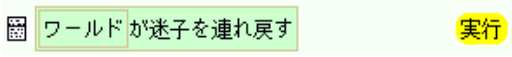
M先生　落ち着いて．岩石のビューアーは開けるかな？ オブジェクトが画面になくとも，画面の右端のタブをクリックすればビューアーが開けるはずだよ．

Ｓくん　はい．あっ，Ｙ座標が-100000 になっている．まずはスクリプトを止めて，Ｙ座標を画面内の値に変更すればいいんですね．

M先生　そうだね．

Ｔさん　もし，ビューアーが開けないときはどうしたらいいんですか？

M先生　その場合，ワールドのビューアーを開いて，「遊び場」カテゴリの「迷子を連れ戻す」を実行すればいいんだ．



No. 4 – 3　考えてみよう！

進める命令タイルだけでは，回転しながら落下する岩石を作れないのはなぜでしょうか？ 説明してみましょう．

> ☞ ヒント
>
> 進めるタイルを使ったときにオブジェクトが進む方向はどうやって決まるのか考えてみるといいですね．オブジェクトの方向についての解説は P.155（A.6）にあります．

No. 4 – 4　考えてみよう！

スクリプト 1 とスクリプト 2 で車の動きに違いはあるでしょうか．プログラムを実行する前に車の動きを予想してみましょう．

No. 4 – 5　やってみよう！

スクリプト 1 とスクリプト 2 で車の動きに違いはあるでしょうか．プログラムを実行して試してみましょう．

No. 4 – 6　やってみよう！

落下するにしたがって，落石が大きくなっていくようにしてみましょう．

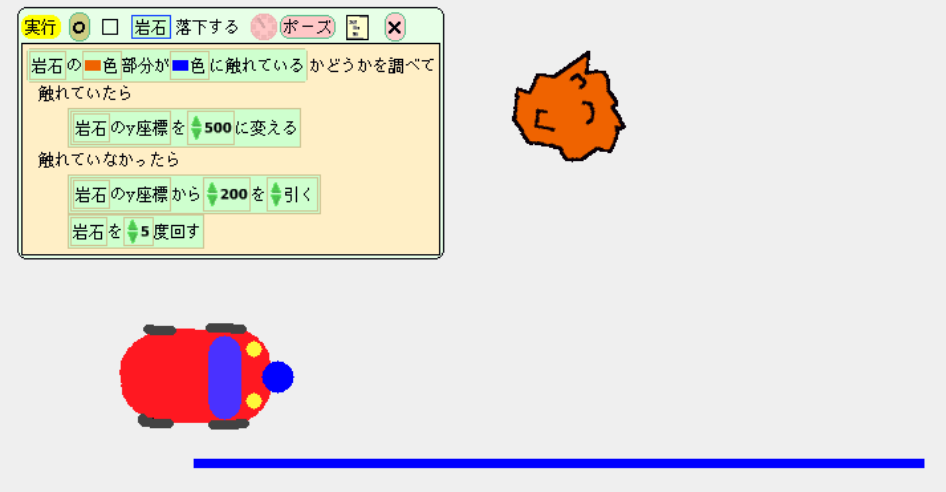
> ☞ ヒント
>
> オブジェクトのサイズを変更するには「幾何 2（向きと大きさ）」カテゴリにある，拡大率を使うとよいでしょう．急に大きくなりすぎる場合は，拡大率に追加する数値を小数点に設定してみましょう．

4.4　変数を使った場合分け

Sくん　うーん．これだとうまく動かないなー．

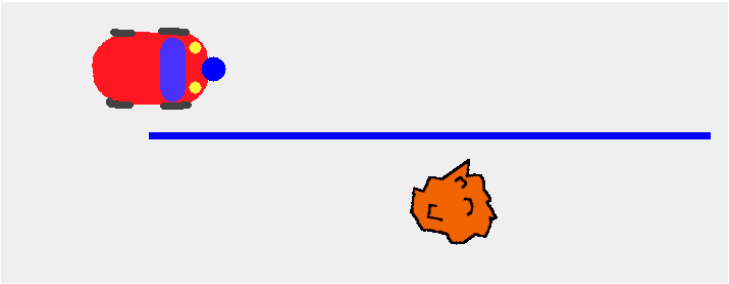
Tさん　どうしたの？

Sくん　車がワープするのと同じ要領で，岩石が繰り返し落ちてくるようにしたんだ．みてみて．



Tさん　すごいじゃない．それで？

Sくん　岩石の落下スピードを速くすると，うまくワープしなくなるんだ．どうしてかな？



Tさん　スクリプトが 1 回実行されると岩石は何ドット進む？

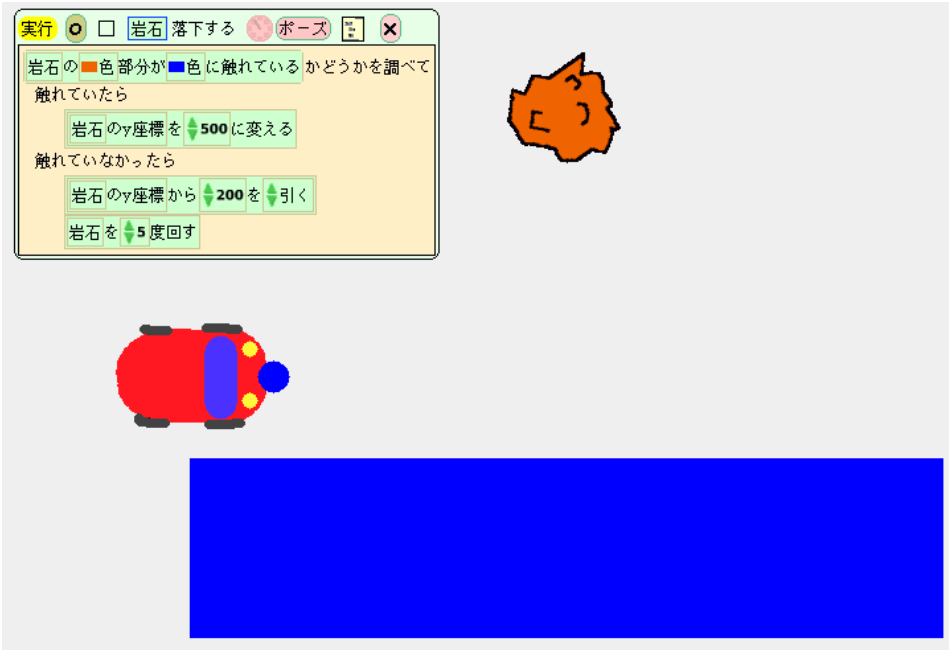
Sくん　200 ドットかな．

Tさん　落下中は，ワープの開始地点に来たかどうか調べて，開始地点でなければ 200 ドット進む，を繰り返すよね．

Sくん　そうか．ワープの開始地点かどうかを調べるのは 200 ドットおきなんだね．

Tさん　ワープ開始地点の目印をもっと太くすればいいんじゃない？

Ｓくん　そうか……．こうすれば，ああ，うまくいったよ．



Ｍ先生　そういう方法もあるね．

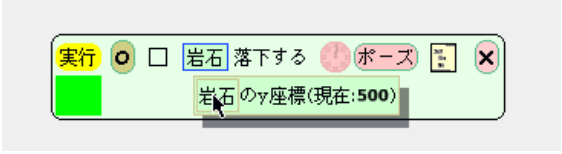
Ｓくん　えっ先生，他に方法があるんですか？

Ｍ先生　座標を調べることができれば，同じことができるはずだね．

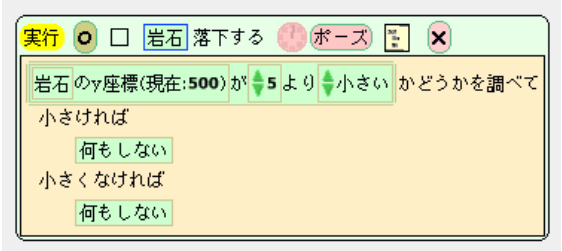
Ｔさん　落石のＹ座標がワープ開始地点の座標かどうかを調べればいいんですね．

Ｍ先生　そうだね．Ｙ座標の変数タイルを直接スクリプトの中に入れてごらん．

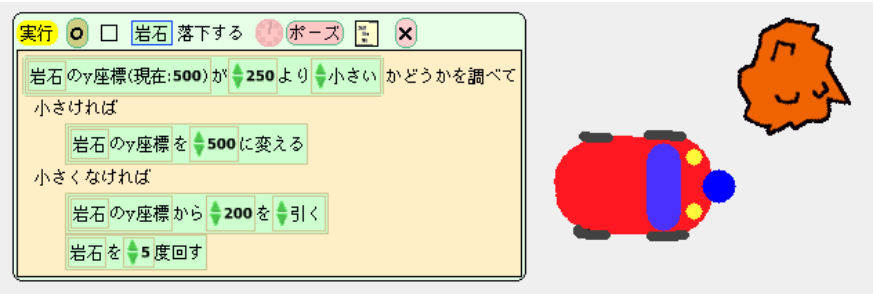
Ｓくん　こうですね．



Ｓくん　あっ，タイルが変わった．



Tさん　これで，ワープ開始地点のＹ座標を入力すればできあがりね．



No. 4 – 7　やってみよう！

「No. 4 – 6 のやってみよう！」を改造し，岩石の大きさを元に戻すリセットボタンを作りましょう．

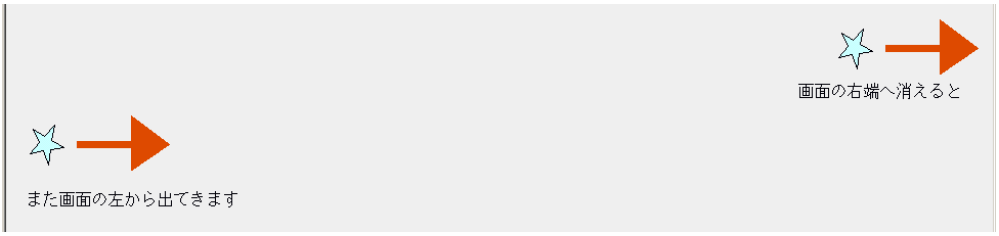
☞ ヒント

ボタンの使い方の解説が P.158（A.10）にあります．

練習問題

練習問題 4.1

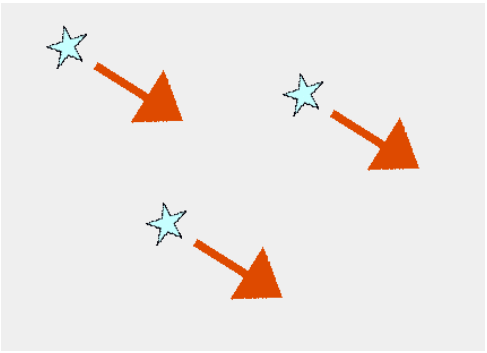
画面の左から右へ流れ続ける流れ星のアニメを作ってみましょう．



> ☞ ヒント
>
> 1 つ目ができたらコピーをして，たくさんの流れ星を作りましょう．きれいですよ．複数のスクリプトを一斉に実行するには，全スクリプトが便利です．全スクリプトについての解説が P.156（A.7）にあります．

練習問題 4.2

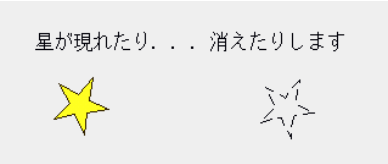
練習問題 4.1 を改造し，画面の左上から，右下へ斜めに星が降るようにしてみましょう．



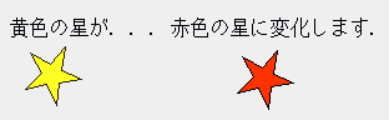
練習問題 4.3

星を使ったアニメーションを作ってみましょう．

1. 星が消えたり現れたりする



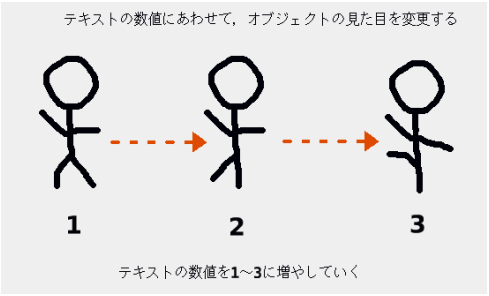
2. 星が黄色と赤色に点滅する



> ☞ ヒント
>
> 色を変化させる場合は，色違いの星オブジェクトを 2 つ使います．場合分けタイルを使って，画面外と画面内の座標に交互に移動させるようにします．点滅の間隔をゆっくりするために刻み値を下げるとよいでしょう．刻み値についての解説は P.154（A.5）にあります．

練習問題 4.4

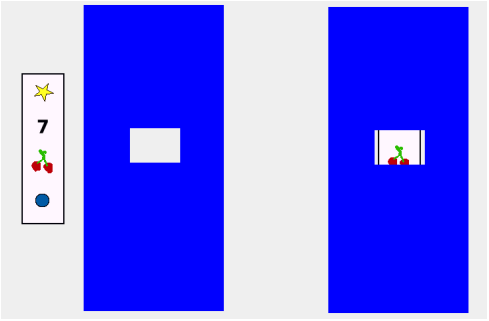
人間が踊るアニメーション（パラパラ漫画）を作ってみましょう．



> ☞ ヒント
>
> オブジェクトの見た目を次々に変更するためには，見た目の変更と，テキストを使います．テキストに表示するコマ数（1〜2）を設定し，その数値にあわせて，オブジェクトの見た目を変更します．オブジェクトの見た目を変更する方法の解説が P.157（A.9）にあります．まずは 2 コマのアニメに挑戦し，それができたら，3 コマ以上のアニメを作ってみましょう．

練習問題 4.5

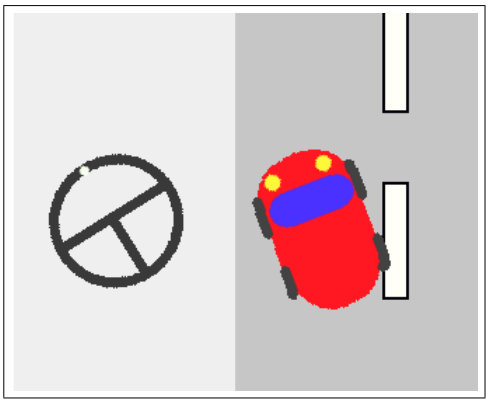
スロットマシーンを作ってみましょう．



> ☞ ヒント
>
> 図のように，絵柄を描いたパーツとカバーを組み合わせると簡単です．絵柄が描いてあるオブジェクトを動かせば，絵柄が回転しているように見えますね．

Project 5

車をハンドルで運転できるようにしてみよう



このプロジェクトでは，ハンドルで操作できる車の作成に挑戦します．ハンドルで車の動きを制御するためには，2 つのオブジェクトの「変数」と「値」をうまく連携させてプログラムを組み立てる必要があります．そろそろプログラミングにも慣れてきたころだと思いますから，プログラムを作る前にフローチャートを書き，それをもとにプログラムを組み立てることにも挑戦してみましょう．

✎ キーワード

変数値の利用，ジョイスティック，スライダー

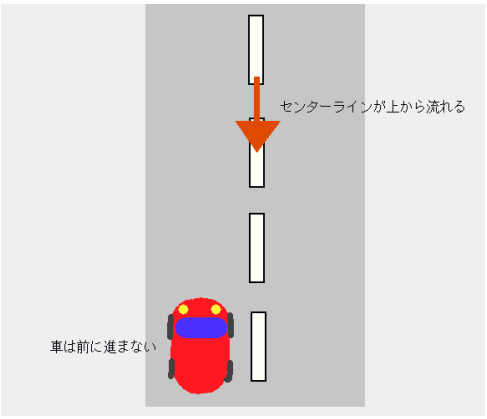
5.1　他の変数を使った現在値の変更（1）– 値指定

M先生　さて，今回から車を使ったゲームを作り始めようか．

Ｓくん　ゲームで遊ぶのは得意です．

Ｔさん　遊ぶんじゃなくて，作るのよ！

M先生　今までは車を前進させていたけど，今度はちょっと趣向を変えて，道路の方が動くようにしてみよう．センターラインが画面の上から下へ流れていくようにすると，車が動いているように見えるから．



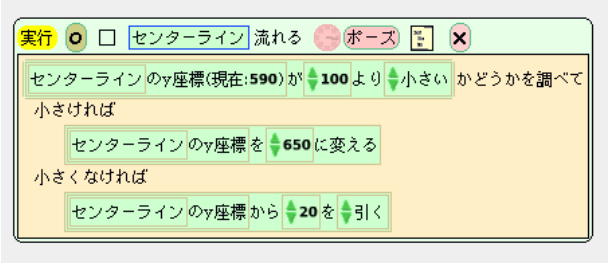
Sくん　なるほど．車が走っているように見えますね．

Tさん　センターラインを道路と別のオブジェクトにしておいて，センターラインにはSくんが大好きな落石と同じようなスクリプトを作ればOKですね．

Sくん　しまったー．道路に直接センターラインを描いてたよ．

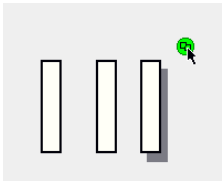
M先生　それだと難しいね．

Sくん　よし．まずは1つセンターラインを作って．同じものを作るのは面倒くさいな……．

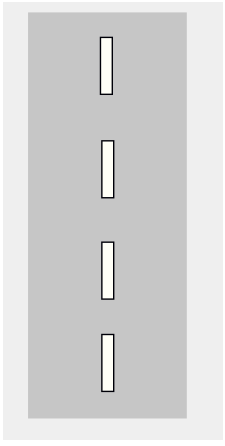


Tさん　緑ハロでコピーすればいいんじゃない？

Sくん　そっか．スクリプトも一緒にコピーされるんだったね．



Sくん　できたぞ．4個くらいあればいいかな．



Sくん　先生，車はどうやって操縦するんですか？

M先生　今回はハンドルを作って，それで操縦するようにしよう．

Tさん　本格的ですね．

M先生　まずはハンドルを描いてみてくれるかな？ 形はお任せするけど，どこが上なのかがわかるようなものがいいかな．

Ｓくん　できました．上がわかるように印をつけておきました．

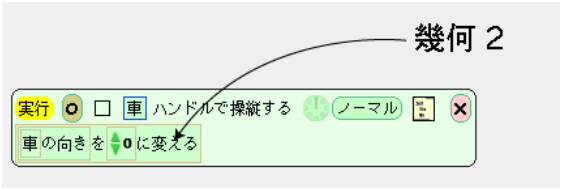
M先生　では，ハンドルを右に切ると車が右を向くようなスクリプトを作ってみよう．

Ｓくん　うっ．どうすればいいんだ？

M先生　ヒントは車の「向き」をハンドルの「向き」に変えるってことかな．

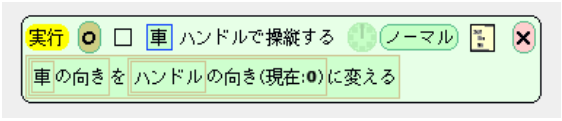
Ｔさん　えっと，「車の向き」を使えばいいんじゃないの？

Ｓくん　まずは車の向きの変更タイルを入れるんだな．



Ｔさん　でも，これでは「車の向きを 0 度に変える」ことを繰り返すプログラムね．これを 0 度じゃなくて，「ハンドルの向き」にすればいいんじゃないかしら．

Ｓくん　そうか，0 のところにハンドルの変数タイルを入れればいいんだね．



M先生　だんだん私の出番がなくなってきたね……．今作ったプログラムを繰り返し実行すれば，車を操縦できると思うよ．

Ｔさん　ハンドルの青ハロを使って，ハンドルを回せばいいんですね．

Ｓくん　できました．

No. 5 – 1　やってみよう！

例題で紹介したプログラムを繰り返し実行しながら，青ハロを使って車を回してみましょう．ハンドルを少し切ってから実験すること！ どのようになるか確認できたら，なぜそうなるのかを説明してみましょう．

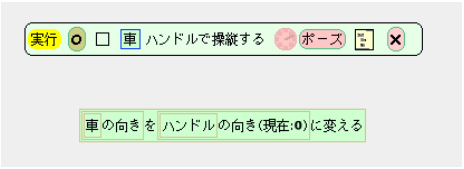
5.2　他の変数を使った現在値の変更（2）– 差分指定

M先生　さて，ハンドルで車の向きが変えられるようになったね．

Sくん　でも，左右に移動できるようにしないと，操縦できているようには見えませんね．

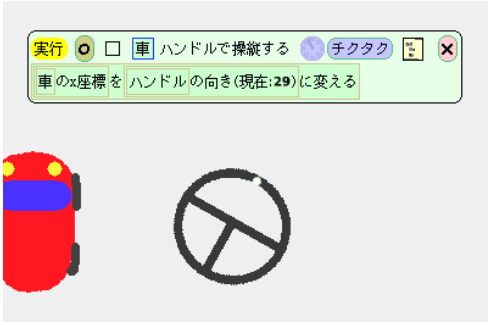
Tさん　そうね．どうすればいいんだろう．

M先生　わかりやすいように，さっき作った「向き」に関するタイルは外しておこう．



Tさん　ハンドルを右に切ったら右に進み続けて，ハンドルを左に切ったら左に進み続けるようにすればいいんじゃない？

Sくん　こうすればいいのかな？



Sくん　あれ，変だな．車が画面の左端に移動してしまったよ．

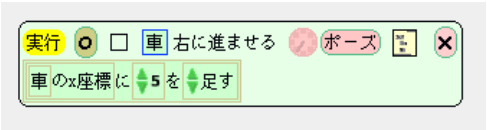
M先生　そのプログラムだと，車の X 座標はいくつからいくつまでの値になるかわかるかな？

Tさん　ハンドルの向きと同じだから，-180 から 180 までですね．

Ｓくん　うーん．

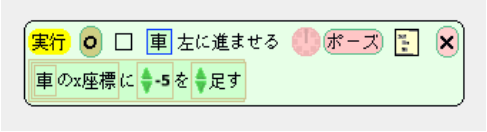
M先生　では，ヒントを出そう．車を右に動かしたいとするね．X 座標を使って，X 座標に数値を足すだけで動かしたいときはどうする？

Ｓくん　x 座標に数値を足せばいいと思います．



M先生　では，同じように左に動かしたいときは？

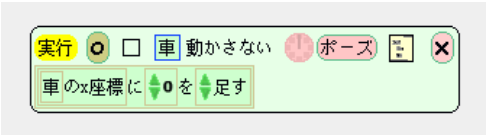
Ｓくん　マイナスを足せばいいのかな．



M先生　では，動かしたくないときは？

Ｓくん　プログラムを作らなければいいんじゃないですか？

Ｔさん　座標に 0 を足すようにすればいいんじゃないかしら．



Ｓくん　あ，そっか．

M先生　次に，ハンドルの向きの値に注目してみてね．ハンドルを右に切ると，ハンドルの向きの値はどうなるかな？

Ｓくん　ハンドルを回す角度によって，増えていきますね．

M先生　逆にハンドルをに左に切ると，どうなるかな？

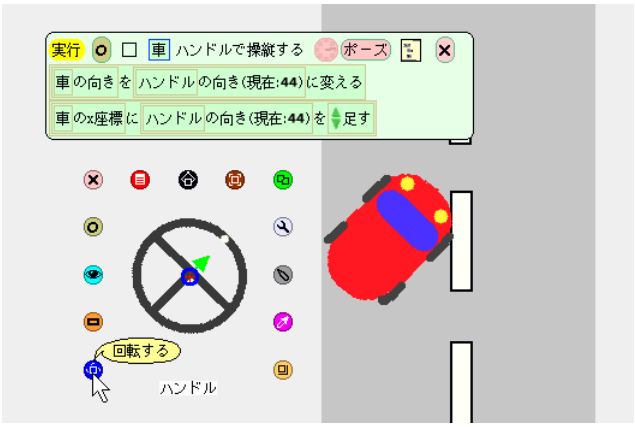
Sくん　マイナスの値になりますね．

Tさん　あ，そっか！ ハンドルの向きの値を車の X 座標の値に足していけばいいんだ！

Sくん　こうかな．

Sくん・Tさん　できたー！

Sくん　はずしておいた方向の操縦用のタイルを戻すと……完成！



No. 5 – 2　考えてみよう！

以下のようなスクリプトを繰り返し実行し，ハンドルを回すと車はどのような動きをするでしょうか．プログラムを実行する前に車の動きを予想してみましょう．

5.3　2 つの変数を使った場合分け

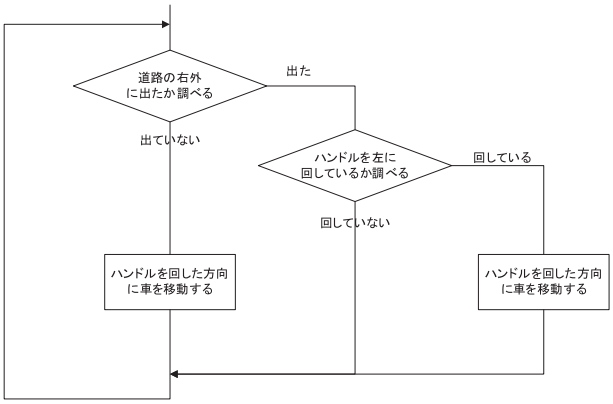
M先生　あと一息で完成だね．それでは最後の仕上げ，ハンドルを切り続けても車が道路の外に出ないようにしてみようか．

Tさん　これもちょっとずつ作ろうよ．

Sくん　じゃあ，まず右方向からやってみようか．

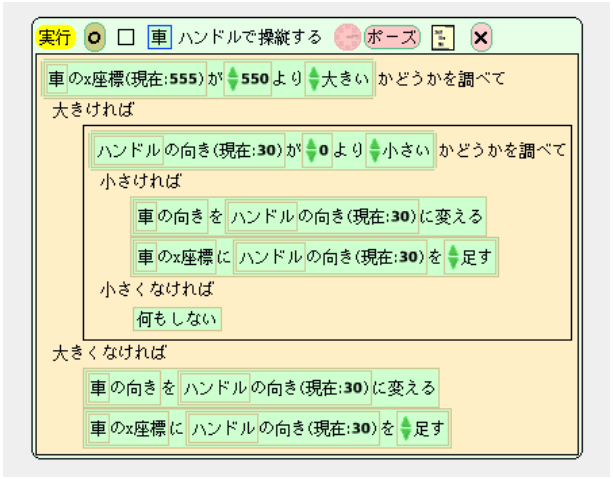
M先生　今回はタイルを組み立てる前にフローチャートを書いてみよう．

Tさん　はい．書けました．



Sくん　なるほど．これでいけそうだね．フローチャートができてしまえば，あとはタイルを組み立てるだけだね．

Sくん　はい．プログラムも完成しました．



M先生　今までは，プログラムを作ってからフローチャートを書いていたけれど，今回は先にフローチャートが書けたね．

Tさん　こうすると，タイルを組み立てるのは楽ですね．ただ，プログラムを作る前にフローチャートを書かなければいけないものなんでしょうか？

M先生　必ずしもそうだというわけではないけど，ちょっと複雑なものは書いてみたほうがいいかな．それがプログラム上達の早道だと思うよ．慣れてくると頭の中でフローチャートが想像できるようになるからね．

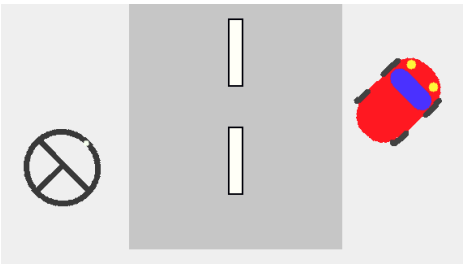
No. 5 – 3　やってみよう！

左右両方の道の外に車が出ないようにしてみましょう．先にフローチャートを書いてから，プログラムを組み立ててみましょう．

M先生　さて，少し意地悪だけど，道を左に移動させても車は外にはみ出さないかな？

Sくん　駄目ですね．

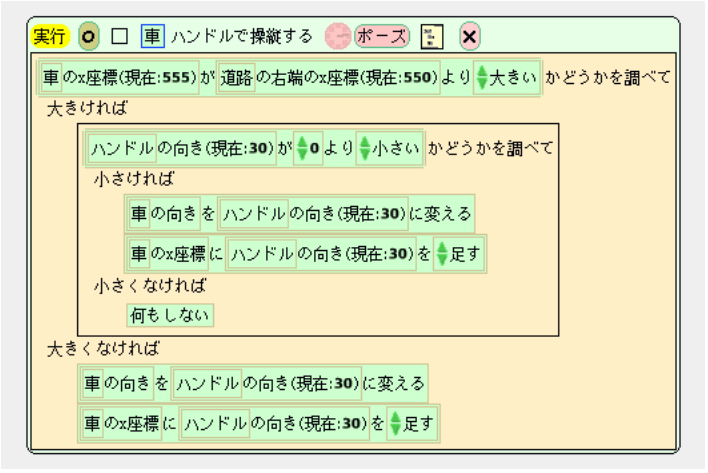
Tさん　場合分けタイルにある道路の X 座標の値（550）は変わらないもんね．



Sくん　じゃあここに変数タイルを入れればいいんじゃないかな．

M先生　そうだね．幾何 1（座標）カテゴリの下の方を探してごらん．

Tさん　あっ，ビューアーをスクロールさせると下の方に道路の「右端の X 座標」っていう変数タイルがあったよ．これを使えばできそうね．



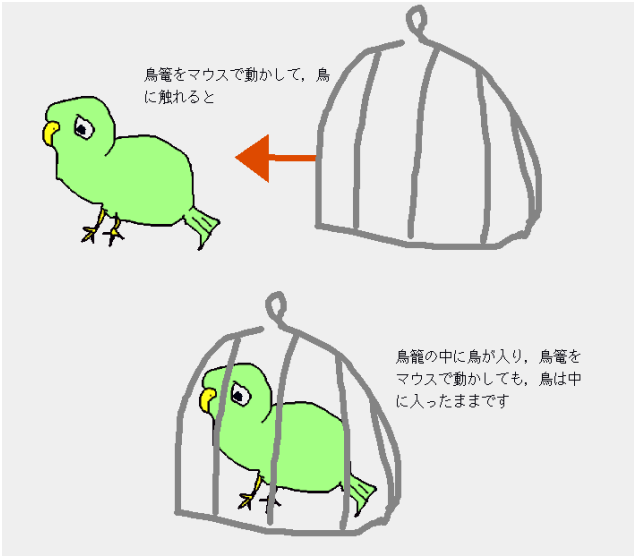
No. 5 – 4　やってみよう！

道路を移動しても左右両方の道の外に車が出ないようにしてみましょう．

練習問題

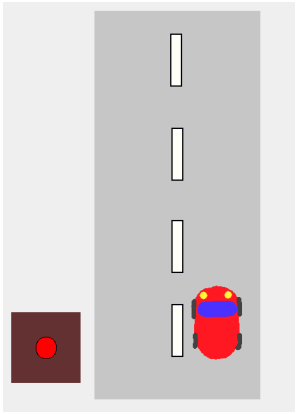
練習問題 5.1

鳥を捕まえられる，鳥かごを作りましょう．



練習問題 5.2

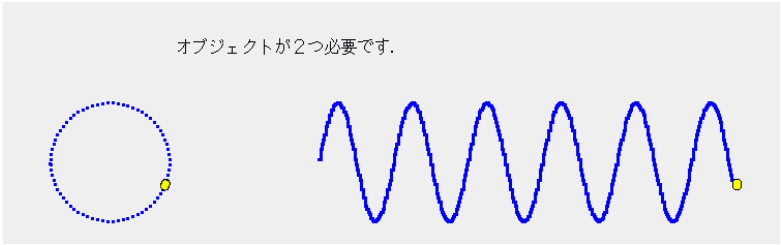
例題を改造して，ハンドルの変わりにジョイスティックを使って車を操縦できるようにしてみましょう．



> ☞ ヒント
>
> ジョイスティックについての解説が P.159（A.11）にあります．

練習問題 5.3

サインカーブを描いてみましょう．

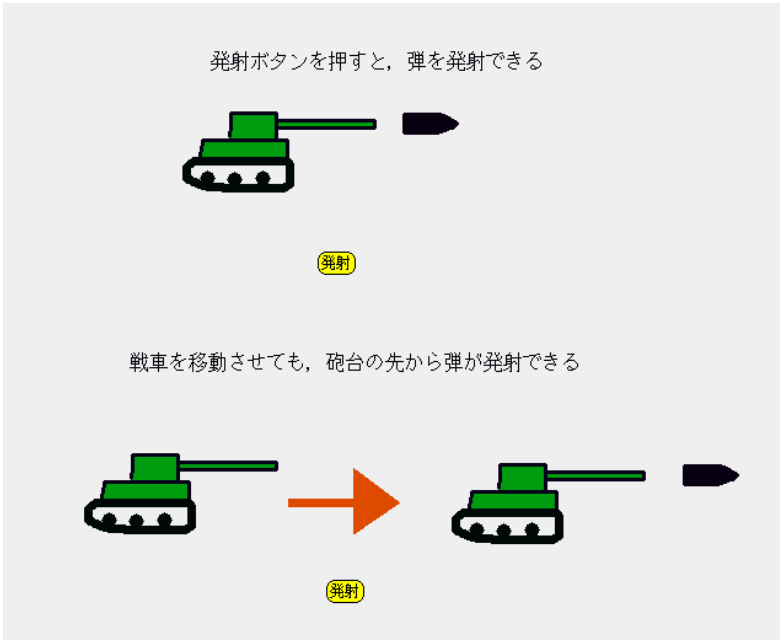


> ☞ ヒント
>
> 円を描くオブジェクトのＹ座標をうまく使うと，サインカーブを描くことができます．サインカーブの性質を良く思い出してみましょう．

練習問題 5.4

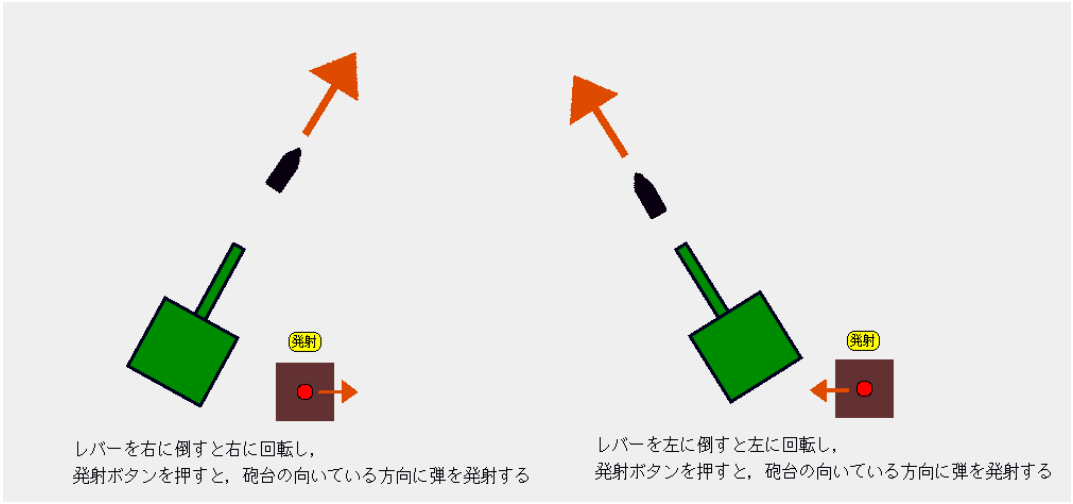
ボタンを押すと，弾を発射する戦車を作りましょう．戦車を動かしても，ちゃんと戦車の砲台の位置から弾が発射されるようにしましょう．

余裕があれば，弾が飛んでいる最中に発射ボタンを押しても発射できないようにしてみましょう．



練習問題 5.5

ボタンを押すと，発射口が向いている方向に弾を発射する砲台を作りましょう．

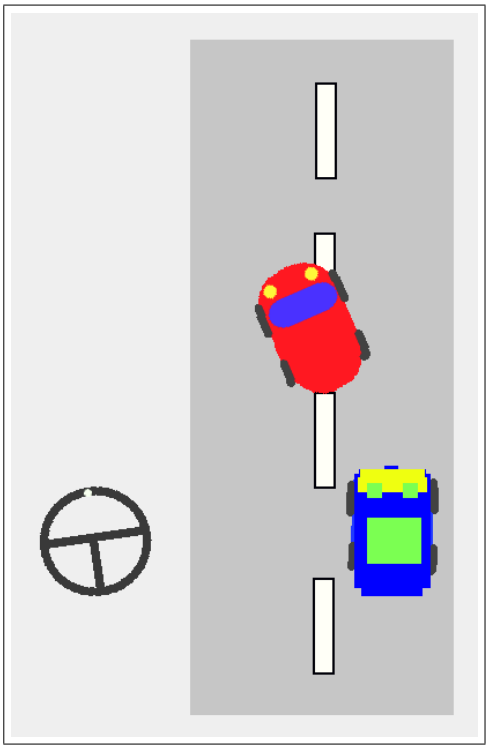


レバーを右に倒すと右に回転し，
発射ボタンを押すと，砲台の向いている方向に弾を発射する

レバーを左に倒すと左に回転し，
発射ボタンを押すと，砲台の向いている方向に弾を発射する

Project 6

レーシングゲームを作ってみよう



このプロジェクトでは，前回のプロジェクトで扱った例題を仕上げ，簡単なレーシングゲームを完成させます．車の操作性を高めるために「計算」を使って，ハンドルの効き具合を調整することに挑戦します．

また，ゲーム性を高めるために敵の車を追加します．敵の車の動きを不規則にするために「乱数」を使ってみましょう．

✎ キーワード

計算，乱数，リモートスクリプティング

6.1　計算

M先生　さて，だいぶゲームらしくなってきたよね．

Sくん　もう少しハンドルのききが悪くてもよいかもしれませんね．ハンドルを切りすぎると車が横を向いてしまうことがあります．

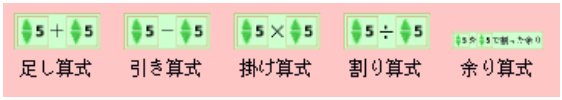
M先生　そうだね．どうしたらいいかな．

Tさん　今はハンドルの向きの値と車の向きの値が同じだからそうなるんだよね．

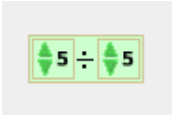
Sくん　ハンドルの向きの値を車の向きの値より小さくできるといいなー．

M先生　それなら“計算”をするようにすればいいね．ハンドルの向きの値を少し割って，車が曲がる量を少なくしてみようか．部品フラップの中に計算式のタイルが入っているよね．

Ｓくん　はい．



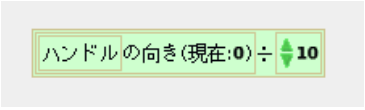
M先生　割り算の「計算式タイル」を取り出してごらん．このタイルには両方の項，つまり数値の部分に変数タイルを入れることができるよ．



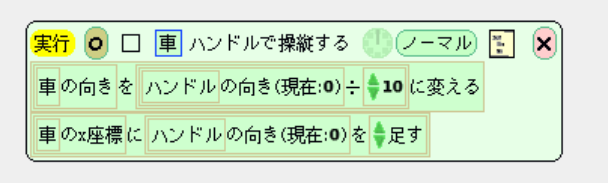
Ｓくん　はい．今回は左項はハンドルの向きですね．

Ｔさん　どのくらいの数で割ればいいんでしょうか？

M先生　まずは 10 で割ってみようか．あとはお好みで調整すればいいよ．



Ｓくん　10 で大丈夫なようですね．ハンドルの効き具合を調整することができました．



Ｔさん　先生，2 つ以上の数値や変数の計算をする場合はどうしたらいいんですか？

M先生　計算式タイルを入れ子にすればいいんだ．

Ｓくん　入れ子ってなんだっけ？

Ｔさん　こういうことだよ．

Ｓくん　あっ．計算式タイルに計算式タイルを入れることができるんだ．場合分けタイルのときと同じだね．

No. 6 – 1　やってみよう！

スライダーを使って，車のスピードを調整するアクセルを作ってください．秒速 0 から 800 ドットの車を作ってみましょう．

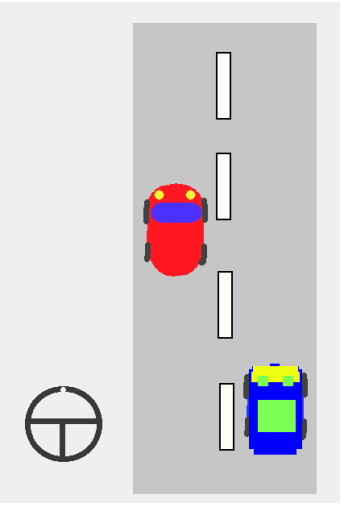
☞ ヒント

スライダーについての解説が P.159（A.12）にあります．スライダーの値は 0 から 1 の範囲をとりますから，計算を使って希望の数値にする必要があります．

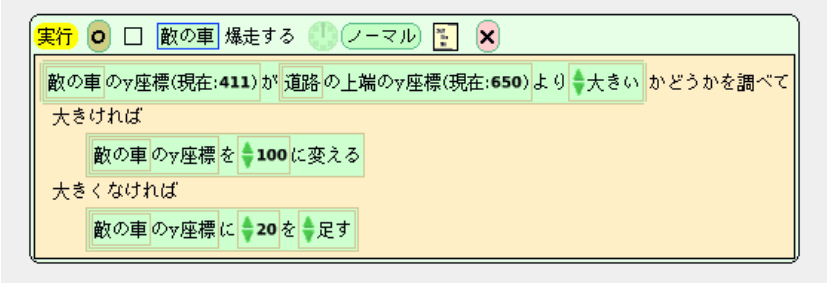
6.2　乱数

M先生　もっとゲームらしくしたいね．

Sくん　ゲームといったら敵キャラですよ．敵の車が襲ってくるというのはどうですか？ 後ろから敵の車が追い抜いてくるのを避けながら運転するのがいいですかね．敵はイカツイ感じの車にしようっと．



Tさん　じゃあ，敵の車のスクリプトはこんな感じでいいかしら．



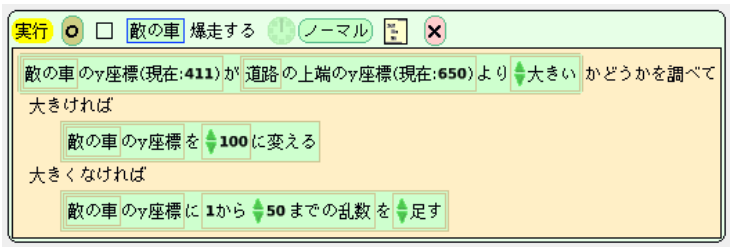
Ｓくん　一定のスピードで敵の車が走ってきては面白くないなー．

M先生　そういう場合は“乱数”を使うといいよ．部品フラップに「乱数タイル」が入っているよね．

Ｔさん　これね．

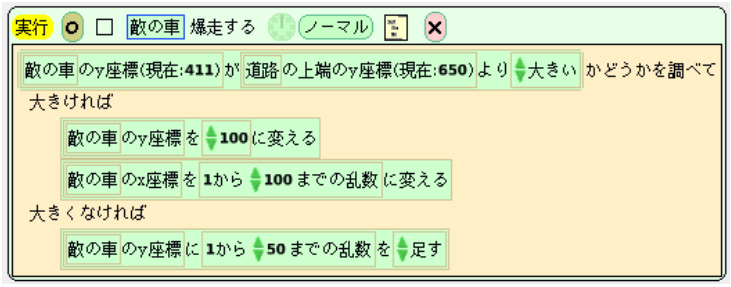
M先生　そう．乱数タイルの数値は，繰り返しのたびに毎回違う値に変更されるんだ．タイルの数値はその値の範囲を示している．この数字が 180 なら，1 から 180 の間のどれかの数値に変更されるということだよ．

Ｓくん　なるほど．じゃあ 50 位に設定して，うわー，結構いやらしい動きになりますね．

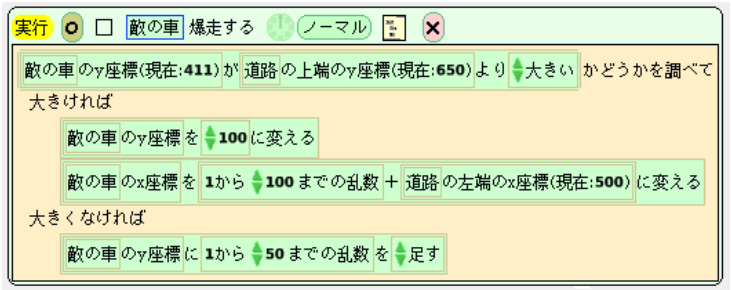


M先生　敵の車が出てくる場所も乱数にしてみたらどうかな？

Ｓくん　ナイスアイデア！ じゃあ，道路の幅は 100 だから，乱数の値を 100 に設定すれば……あれ？ うまくいかないな．



Ｔさん　それだとだめよ．乱数の値に道路の左端の X 座標を足さないと．



M先生　そのとおり，計算と乱数の合わせ技だね．

No. 6 – 2　やってみよう！

敵の車に衝突すると，自分の車がハンドルで操縦できなくなるようにして，ゲームを完成させましょう．

> ☞ ヒント
>
> 車のスクリプトを停止させるためには，リモートスクリプティングカテゴリの命令タイルを使う必要があります．リモートスクリプティングカテゴリの命令の解説が P.156（A.8）にあります．

No. 6 – 3　考えてみよう！

例題では，敵の車が登場するタイミングは一定です．敵の車の速度や走行位置だけでなく，登場するタイミングが不定期になるようにしてみましょう．

練習問題

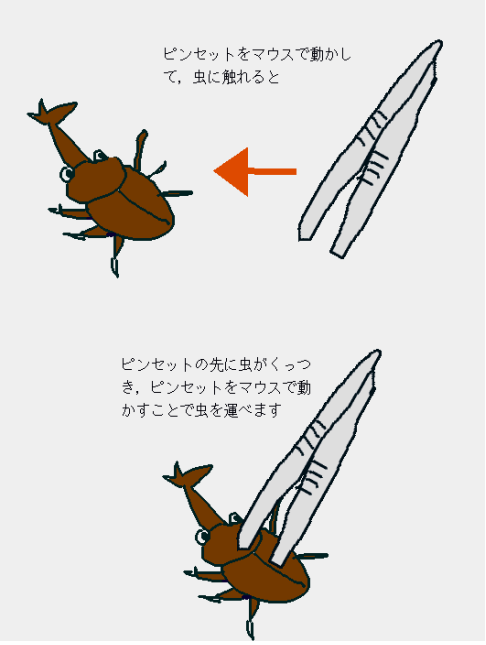
練習問題 6.1

-20 から 20 の範囲の乱数を発生させる方法を考えましょう．

できたら，例題の敵の車をランダムに動かして，邪魔するように改造してみましょう．

練習問題 6.2

練習問題 5.1 と同じ要領で，先端で虫がつまめる虫取りピンセットを作ってください．ピンセットの先端に虫がくっつくようにするのがポイントです．



スタートボタンを押すと，画面内のランダムな位置に虫が登場し，ランダムに逃げ回るというゲームに挑戦してみましょう．

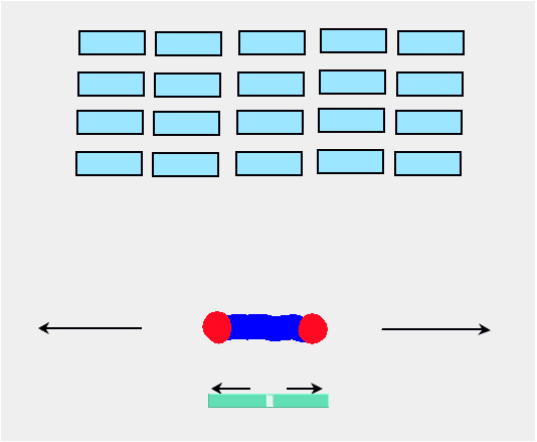
練習問題 6.3

以下のスクリプトを見ただけで，楕円はどんな軌跡を描くのかを予想してみましょう．

※プログラムを組み立てて実行しては駄目ですよ．

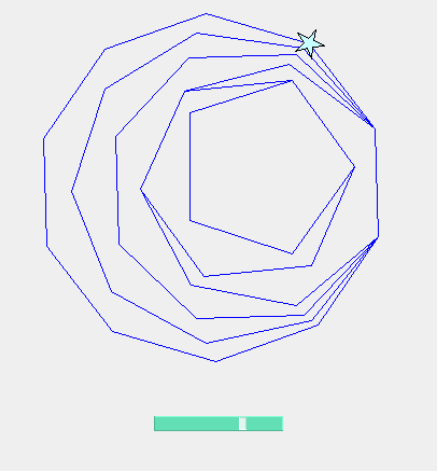
練習問題 6.4

ブロック崩しゲームを作りましょう．まず，下図のようにスライダーとパドルを作り，スライダーにあわせてパドルが動くようにしてください．スライダーの動く幅とパドルの動く幅が違うところに注意しましょう．できたら，ブロックとボール，壁を作り，ブロック崩しゲームを完成させましょう．



練習問題 6.5

正多角形を書くプログラムを作りましょう．正多角形の一つの外角は 360 ÷ N（頂点の数）で求められます．



できたら，スライダーで頂点の数を変えられるようにしてみましょう．
ここで利用した外角の公式が何故成り立つのかも考えてみましょう．

練習問題 6.6

練習問題 4.1 か練習問題 4.2 の流れ星を改造し，星の動きに乱数を取り入れてみましょう．どんな風に乱数を使うと本物の流れ星のような動きになりますか？

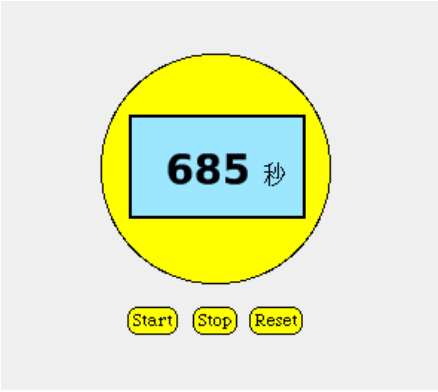
練習問題 6.7

P.62 の岩石に当たると車がワープするという例題を改造して，上から降ってくる岩石に当たると，車がスピンするようにしてみましょう．

> ☞ ヒント
>
> 車が岩石に当たったときに 1 回だけスクリプトを実行してもうまくスピンさせられません．あるスクリプトを繰り返し実行できるようにするためには，リモートスクリプティングカテゴリの命令タイルを使いましょう．リモートスクリプティングカテゴリの解説が P.156（A.8）にあります．

練習問題 6.8

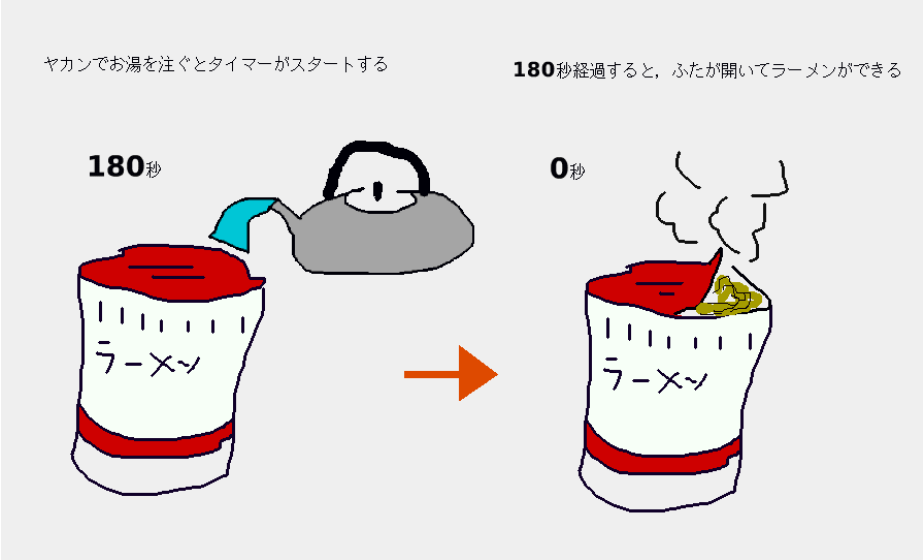
ストップウォッチを作成してみましょう．



> ☞ ヒント
>
> Start ボタンを押して，計測を開始するためにはリモートスクリプティングカテゴリの命令タイルを使う必要があります．また，文字盤にはテキストを使いましょう．テキストの数値に 1 秒ごとに 1 だけ足していけば，時間の計測ができます．P.160（A.13）を参照してください．

練習問題 6.9

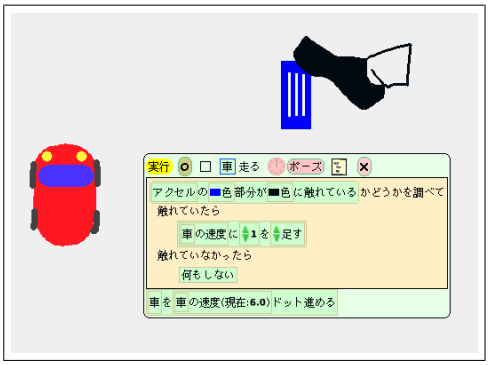
お湯を注いで 3 分間経過すると蓋が開いてでき上がる，カップラーメンを作りましょう．



> ☞ ヒント
>
> タイマーはテキストを使うと簡単です．3 分が経過したら，ラーメンの見た目を変えれば OK です．オブジェクトの画像を変更する方法の解説が P.157（A.9）にあります．

Project 7

車をアクセルで加速できるようにしてみよう



このプロジェクトでは，実世界の車の加速や減速の「シミュレーション」について考察します．実世界の物体の運動をシミュレートするプログラムを作るには，これまで皆さんが勉強してきた物理の法則が役立ちます．車の速度を扱うには，オブジェクトに新しい変数を自分で作成する必要があります．自由に変数が作成できるようになると，Squeak を使って作ることのできる作品の幅がぐんと広がるでしょう．

キーワード

変数の作成，シミュレーション，キーボード入力

7.1　速度の概念と変数の作成

M先生　さて，今回は車にアクセルをつけてみようと思うんだけど．

Ｓくん　踏むと進むということですか？

M先生　本物の車はアクセルを踏むとすぐに速度が上がるかな？

Ｔさん　だんだんと速度が上がっていくのが普通よね．

M先生　そうだね．

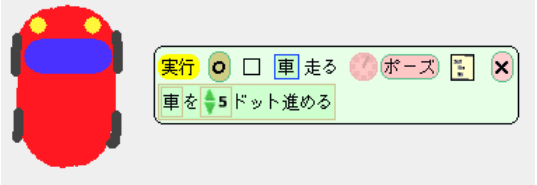
Ｓくん　前のプロジェクトで作った車のゲームを改造するんですか？

M先生　それでもいいけれど，今回は車の方を動かすことにしよう．用意するのは車だけで OK だね．簡単な例で理解できれば，前回作ったゲームに応用もできるからね．

Ｓくん　わかりました．まずは何をすればいいでしょうか？

M先生　まず車を前に進めてみよう．

Ｓくん　はい．



M先生　スクリプトを組むまえに，少し速度について考察してみよう．このスクリプトにある 5 という数値は何を表しているかな？

Sくん　これはスクリプトが 1 回実行されるときに進む距離ですね．

Tさん　これは車の「速度」ということですね．この数値を大きくすれば車は速く進むし，0 にすれば止まったままになると思います．

M先生　そのとおり．

M先生　速度の値をだんだん増やしていければ，加速できるはずだよね．

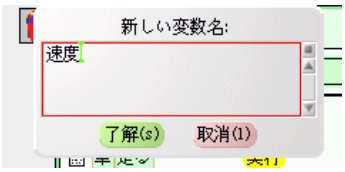
Sくん　うーん．でも速度の値を増やすことはできませんね．変数ならできるんですが．

M先生　じつは，これを変数にできるんだよ．スクリプトと同じように，変数も自分で作ることができるんだ．

Tさん　どうやるんですか？

M先生　車のビューアーを開いて，右上の方を見てごらん．そこにある「新しい変数を作る」と書いたボタンを押すと“変数を作成”することができるよ．

Sくん　押してみよう．あっ，名前を聞かれました．変数の名前は「速度」だな．

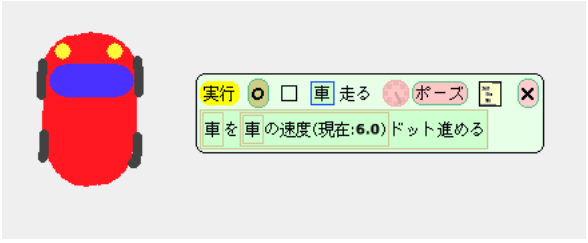


M先生　入力できたら了解を押すと，ビューアーに追加されるはずだよ．

Tさん　一緒に「変数」というカテゴリも追加されましたね．

M先生　そうだね．自分で作成した変数は「変数カテゴリ」に入るようになっているんだ．

Sくん　この速度という変数をさっき作ったスクリプトで使えばいいんだな．確かに速度の値を変えると車の進み方が変わるや．



No. 7 – 1　考えてみよう！

例題で作成した速度という変数の単位（m/s 等）はどのようになるかを考えてみましょう．

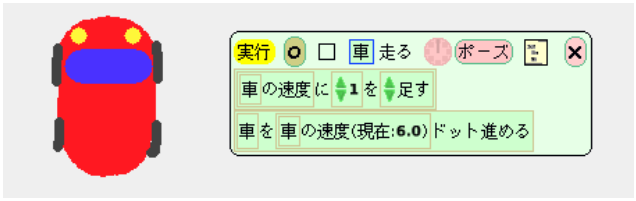
☞ ヒント

単位を考えるには刻み値（1 秒当たりのスクリプトの実行回数）を考慮に入れる必要があります．刻み値の解説は P.154（A.5）にあります．

7.2　加速と減速のシミュレーション

M先生　さて，いよいよ．アクセルを作ろう．車が加速するスクリプトはできているかな？

Tさん　はい．



M先生　OK．ではこの 1 という数値は何を表しているかな？

Sくん　速度の増加量だから，加速度かな．

Tさん　ということは，アクセルを踏んでいる間は速度に加速度を足して，踏んでいないときは足さないようにすればいいのね．

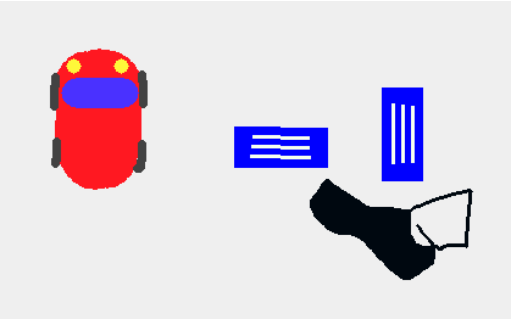
M先生　そういうことだね．

Sくん　こうかな．

M先生　これで実際の車に近い“シミュレーション”ができるようになったね．

No. 7 – 2　やってみよう！

アクセルと同じ要領でブレーキを作ってみましょう．（車は急に止まりませんし，ブレーキを踏み続けても逆走しないことに注意しましょう．）



No. 7 – 3　やってみよう！

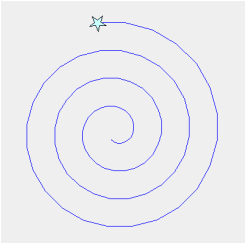
キーボードの「a」を入力するとアクセル，キーボードの「b」を入力するとブレーキで操作できるようにしてみましょう．

> ☞ ヒント
>
> キーボード入力を受け取る方法については，P.161（A.15）に解説があります．

練習問題

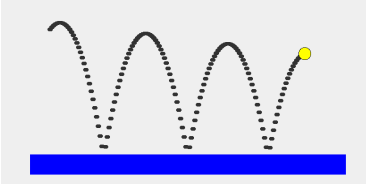
練習問題 7.1

うずまき模様を描いてみましょう．



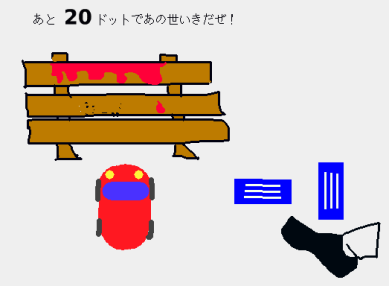
練習問題 7.2

ボールの自由落下運動をシミュレートしてみましょう．



練習問題 7.3

例題を改造して，壁にぶつかるぎりぎりまで車を走らせるチキンレースゲームを作りましょう．



> ☞ ヒント
>
> 壁に激突したら，車がへこむようにしてみましょう．オブジェクトの見た目を変更する方法の解説が P.157（A.9）にあります．壁までの距離を表示しても面白いでしょう．また，ドキドキ感を出すために，ブレーキとアクセルの効きを調整する必要がありますね．

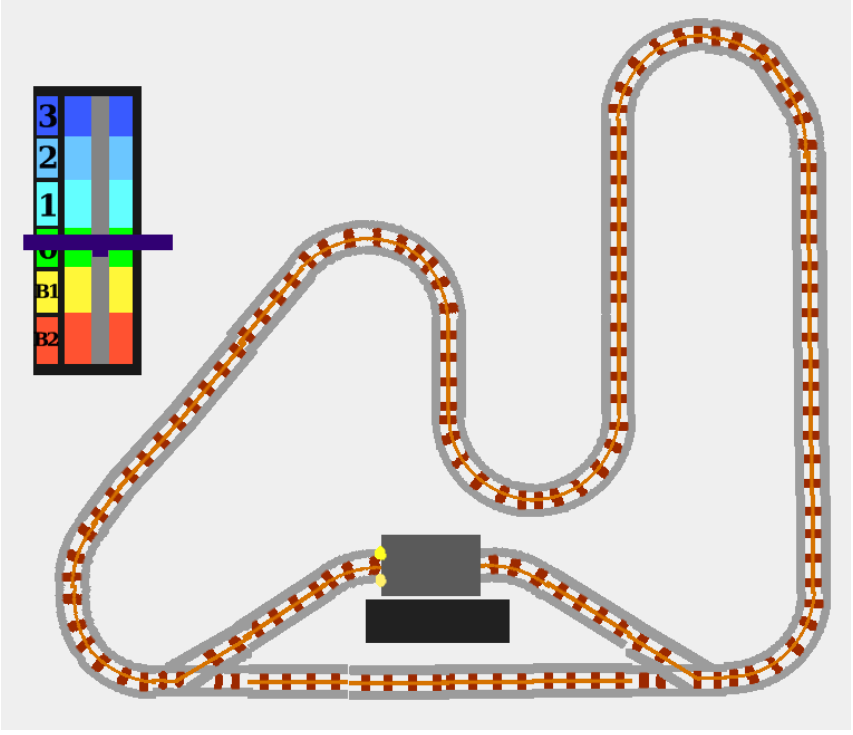
練習問題 7.4

Project 6 で作ったゲームに以下のような改造をしてみましょう．

1. 速度を調整できるようにして，現在の速度を表示できるようにしてみましょう．
2. 一定の距離を走りきったらゲームクリアーになるようにしてみましょう．
3. タイマーをつけ，制限時間内に一定の距離を走らないと，ゲームオーバーになるようにしてみましょう．
4. リセットボタンをつけて，簡単にゲームをリセットできるようにしてみましょう．

練習問題 7.5

電車シミュレータを作ってみましょう．



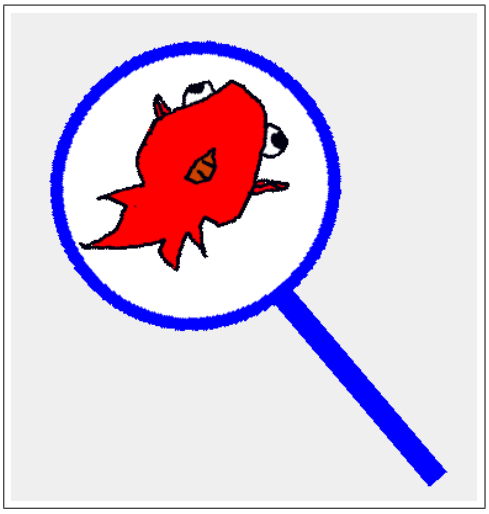
・ Project 3 で学んだセンサーを応用すると，電車がレールから外れないようにできます．
・ レバーを操作するとだんだんと加速し，ブレーキを踏んでもすぐ電車が止まらないようにしましょう．
・ オーバーランせずに，駅に止まると得点が増えるようにしてみましょう．

第 III 部

作品づくりプロジェクト

Project 8

作品づくり



第 III 部では，Squeak を使って，オリジナル作品を作るプロジェクトに挑戦します．まずは，どんな作品を作るかを考える「企画」からスタートです．次に画面の移り変わりや，作品に登場するキャラクターの動きを決めます．この作業を「仕様の策定」といいます．作るべきものが決まったら，作る順番や役割分担を「スケジューリング」します．スケジュールに従って，タイルを組み立ててスクリプトを作っていく「実装」に入ります．

キーワード

企画，仕様，スケジューリング，マイルストーン，実装

8.1　作品作りのプロセス

M先生　さて，いよいよオリジナルの作品作りに挑戦しよう．

Ｔさん　先生，作品とは何ですか？

M先生　Squeak では色々な作品を作ることができる．ゲームやシミュレーション，使う人の操作に反応する絵本みたいなものも作れるね．

Ｓくん　色々できますね．

M先生　では，作品をどのような手順で作っていったらいいのかを紹介するよ．ちょうどいいから，Ｓくんと Ｔさんで協力して１つの作品を作ってみたらどうかな？

Ｔさん　はい…….

Ｓくん　えー，そんなに嫌がるなよー．

M先生　けんかしないようにね．まずは大まかにどのような手順で作品を作っていったらいいかを紹介するよ．作品を作る作業の流れはだいたい以下のようになるね．

1. 企画の立案
2. 仕様の策定
3. スケジューリング
4. 実装
5. プレゼンテーションと評価
6. 報告書の作成

M先生　このプロジェクトでは 4 つめの“**実装**”までを紹介するよ．「プレゼンテーションと評価」，「報告書の作成」については次のプロジェクトで解説するからね．

Ｓくん　わかりました．

M先生　それでは，実装までの各作業で，どのような成果物ができるのかを整理しておこう．

Ｓくん　先生，成果物ってなんですか？

M先生　作業をするとできる「もの」のことだよ．一覧表を書いておいたから，見ておいてね．

作業	成果物
企画の立案	企画書
仕様の策定	仕様書
スケジューリング	作業手順書・スケジュール表
実装	作品

Ｓくん　なるほど，「実装」まで終わると作品ができるんだ．

Ｔさん　企画書，仕様書，作業手順書・スケジュール表は全て文書ですね．

M先生　そうだね．これから，1 つずつ作業を解説していくよ．

8.2　企画の立案

M先生　まずは“**企画**”の立案だね．なにより先に，どんな作品を作るかを決めないといけないよね．どんな作品を作るかを書いたものが「企画書」だよ．

Ｔさん　はい．これはＳくんと相談しながら決めればいいですね．

M先生　この段階では，完成する作品の大まかなイメージを共有することが大事だよ．細かいことはこの次の「仕様の策定」で考えればいいよ．

Ｓくん　作品のコンセプトを決めるってことですね．

M先生　そのとおり．

Ｔさん　具体的にはどんなことを考えればよいのでしょうか？

M先生　企画書を作るためのテンプレートを紹介しよう．大体以下のようなことを決めれば OK だよ．

1. **タイトル・キャッチコピー**：タイトルが長くなるなら，キャッチコピーをつけてタイトルを短くする方法もある．
2. **作者・作成日**：グループの場合は，全員を記入しておくこと．（所属なども忘れずに）
3. **概要**：作品の概略を 100〜200 文字で書く．箇条書きでもよい．
4. **対象**：作品を鑑賞したり，ゲームをプレイしたりするユーザーをなるべく詳しく書く．（対象年齢など）
5. **画面イメージ**：手書きでもいいが，スキャンして張りつけること．

M先生　これに加えて，作成者や日付なんかも入れておくといいかな．

Ｓくん　はい．じゃあ，T さんと相談してみます．

1 時間後…….

Ｓくん　先生，できました．

Tさん　「金魚すくい」を題材にしたゲームを作ろうと思います．

金魚すくいゲーム 企画書

作者　：杉浦学 広瀬智子
作成日：2005/08/30
Ver.　：1.0

タイトル / キャッチコピー

金魚すくい ～夏祭りの夜～

概要

出店の金魚すくいを再現したアクションゲーム
・ポイをマウスで操作し，水槽の中で泳いでいる金魚をすくう
・ポイが破れないようにしながら，どれだけたくさんの金魚をゲットできるかを競う
・ポイを水につけていると，耐久度が減っていき，０になるとポイが破れてゲームオーバー

対象

夏祭りの気分を味わいたい人

画面イメージ

M先生　いいね．企画書の仕上がりをチェックするときのポイントを簡単に整理しておいたから，チェックしてみてね．

- 概要に重要なポイントがもれなく書かれているか，また細かすぎることが書いていないか．
- 対象者がきちんと絞られているか．
- 画面イメージだけで作品の概要がわかるかどうか．（余分なものが描いていないか，大切なものが抜けていないか）

M先生　タイトルやキャッチコピーに関しては，個人のセンスもあるから，一概に良い悪いはいえないね．

Sくん　わかりました．

No. 8 – 1　やってみよう！

作りたい作品を決めて，企画書を書きましょう．

8.3　仕様の策定

M先生　次は“仕様”の策定だね．

Ｓくん　先生，仕様とは何ですか？

M先生　企画だけだと，プログラムが実際にどのように動くのかがわからないよね．例えば金魚すくいの例だと「金魚がどのように動くのか」とかね．だから，少し細かい動きを決めておく必要があるんだ．具体的には，ゲーム全体の流れと，登場するキャラクターの外見や動きを少し細かく考えておくといいね．この作業を「仕様の策定」と呼ぶんだ．仕様を記述したものが「仕様書」だね．

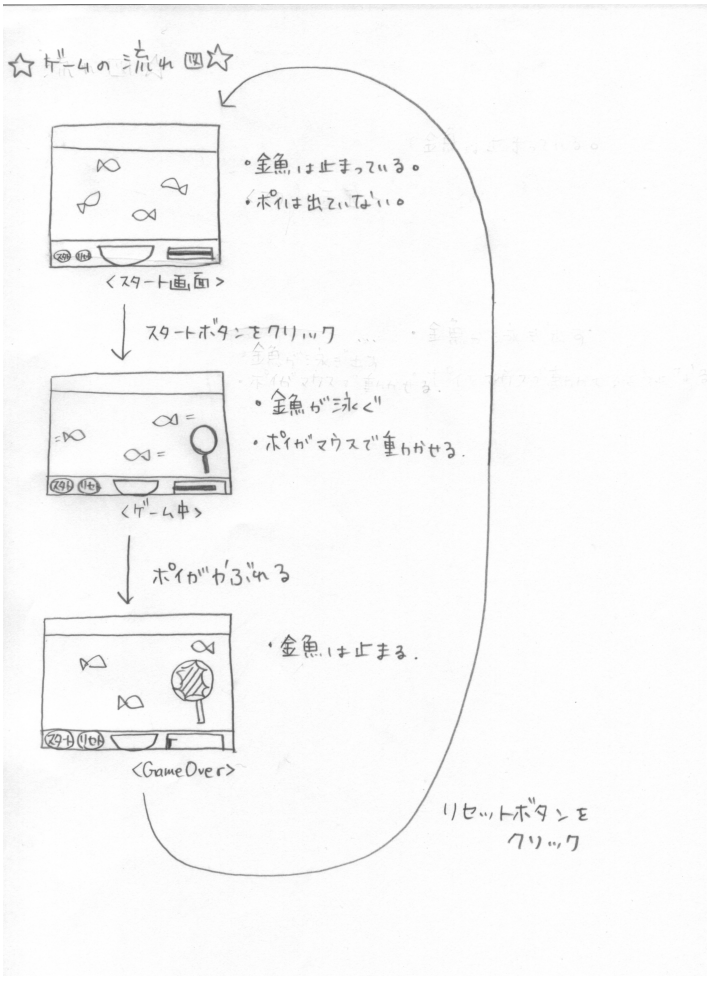
Ｓくん　なるほど．仕様書のテンプレートはありますか？

M先生　作る作品や登場するキャラクターによって一概に決められないので，テンプレートは自由に記述ができるものを用意しておいたよ．工夫して書いてみてね．

Ｓくん　はい．がんばります．

3 時間後…….

Ｔさん　先生，仕様書ができました．ゲームの流れ図も書いてみました．



（ 1枚目 / 4枚中）

オブジェクト名： 金魚 PART I

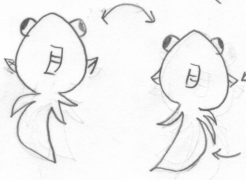
仕様

■ 外観

金魚は上から見た絵にする。

■ 泳ぐ場合

・アニメーション

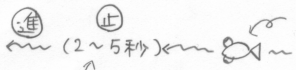


3秒ごとにコマが
ヒカリ変わる。

前ビレは上下する。

尾ビレは左右にふれる。

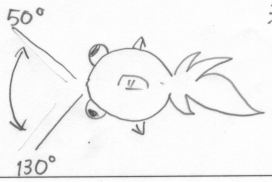
・金魚の泳ぎ方



すぐにつかまらない
程度の速さ

2～5秒間
不定期に止まる。
その間，金魚自体のアニメーションは
止まらない。

・金魚の泳ぐ方向



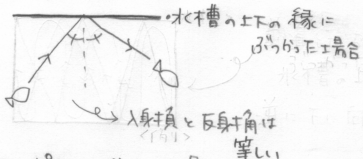
泳ぐ方向は，
0.5秒ずつ変わる。
50°～130°の間でランダムに
方向を変える。

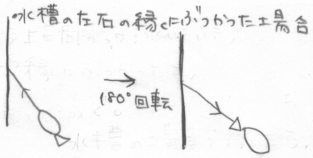
（2 枚目 / 4 枚中）

オブジェクト名：金魚 PARTⅡ

仕様

◦金魚の反射の仕方

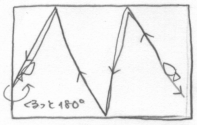




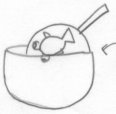
■ ポイにつかまる場合

ポイの上で
金魚自体のアニメーションは継続。

<例>

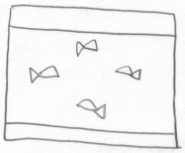


■ 金魚がおけに入る場合



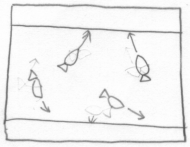
金魚がポイとおけの両方に重なった
場合，自動的におけに入る。
→金魚は消える。

■ スタートボタンが押された場合（初期位置）



金魚はその場で止まっている。
（位置は水槽内で，向きと位置はランダム）

↓スタートボタンをクリック



その場から泳ぎ出す。

（3 枚目 / 4 枚中）

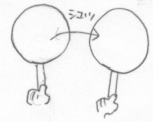
オブジェクト名：ポイ

仕様

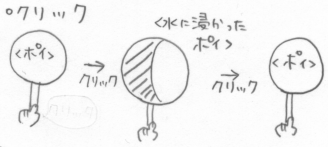
■ 外観見

■ ポイの動き

・ポイはマウスで動かすことができる．

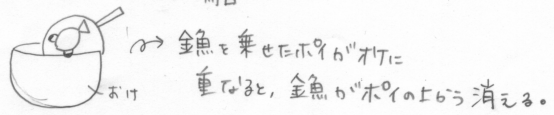


・クリック

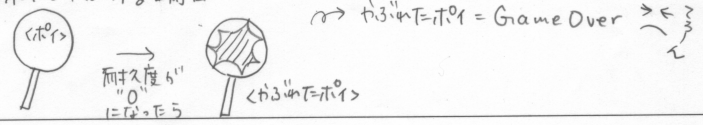


■ 金魚をすくう場合

■ おけに金魚を入れる場合



■ ポイがやぶける場合

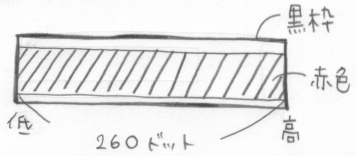


（ 4枚目 / 9枚中）

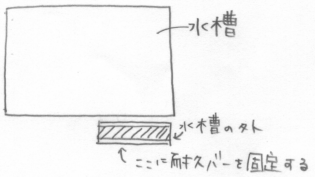
オブジェクト名：耐久バー

仕様

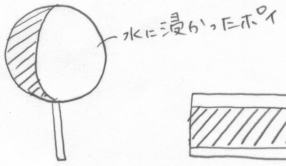
■ 外観見



<位置>



■ 減る場合



ポイが水に浸かっていると，1秒につき1ドットずつ右から左へ減っていく。

M先生　仕様を決めるのは時間のかかる作業だよね．仕様書をチェックするときのポイントを簡単に整理しておいたから，チェックしてみてね．

- 画面がわかりやすく整理されており，ユーザーの使い勝手が考慮されているか
- 仕様の記述に曖昧な点，矛盾点，取りこぼしがないか
- それぞれのキャラクターの仕様が，わかりやすいように分類されているか

M先生　ただ細かく書けばいいというものではなく，ポイントを抑えた記述ができているかが重要だよ．

No. 8 – 2　やってみよう！

作品の仕様書を書きましょう．

8.4　スケジューリング

Sくん　いよいよ作りはじめですか？

M先生　作りはじめてもよいけど“**スケジューリング**”をしておくと，スムーズに作業が進められると思うよ．

Tさん　具体的にどんなことを考える必要がありますか？

M先生　大体次の 3 点を考えておくといいね．

1. 何をどのような手順で作るか？
2. どのくらいの期間で作るか？
3. 誰が作るか？

Sくん　1 人で作品を作る場合は，3 番目は要りませんね．

M先生　まあ，そういうことになるけど，1 番目と 2 番目は 1 人で作品作りをする場合でも重要だよ．

Sくん　何をどのような手順で作るのかについては，どんなことを書けばいいんですか？

M先生　どのような順番で作品作りを進めていけばいいかを考えて，作業を分解していくといいね．ポイントは，ゲームの中心となる部分をなるべく早い段階で作ってしまうことかな．時間が余ったらやりたいなと思うことは後の方に回しておこう．

Tさん　優先順位を考えるってことですね．

M先生　あとは，テストが簡単にできるかについても考える必要がある．スタートボタンやリセットボタンを早めに作っておけば，テストが楽になるよね．

Sくん　なるほど．

M先生　また．スケジュールの大きな区切り目にチェックポイントを設けておくといい．これを“**マイルストーン**”という．

Sくん　なぜそのようなものが必要なのですか？

M先生　作業を進めていくうちに，順調に進んでいるかをチェックする必要があるよね．マイルストーンが立っていれば，そのチェックがしやすくなるんだ．

Tさん　どのように決めればいいのですか？

M先生　そうだね．締切りから逆算してスケジュールを立てるなら，締切りの 1 日前には完成しておきたいから，「ひとまず完成」というマイルストーンは締切り日の一日前になるね．そうやってどんどん逆算していけば，ある程度のスケジュールの骨組みができるね．

Sくん　なるほど．

M先生　あとは分担を決めよう．作業手順が決まれば，同時並行で進められる作業がわかるので，分担を

決めるのも楽なはずだよ．

1 時間後…….

Sくん　先生，できました．

Tさん　作業手順書はこうなりました．

1．オブジェクトの描画

1.1 水槽の絵を描く
1.2 金魚の絵を描く（アニメーション用に 2 種類用意する）
1.3 ポイの絵を描く
1.4 おけの絵を描く

2．プログラミング

2.1．金魚のスクリプト作成
2.1.1 ひれを動かすアニメーションを作る
2.1.2 金魚が泳ぐようにする

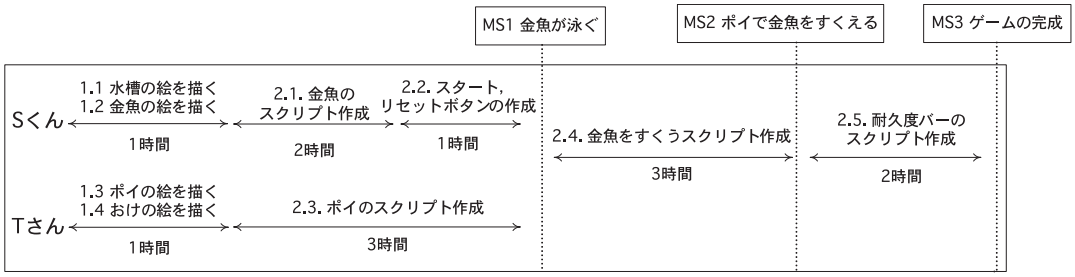
2.2．スタート，リセットボタンの作成
2.2.1 スタートボタンを押すと，ゲームが開始できるようにする
2.2.2 リセットボタンを押すと，ゲームがリセットできるようにする

2.3．ポイのスクリプト作成
2.3.1 ポイをマウスで動かせるようにする
2.3.2 ポイをクリックで水に浸せるようにする

2.4．金魚をすくうスクリプト作成
2.4.1 金魚をポイで捕まえられるようにする
2.4.2 ポイを使っておけに金魚を入れるようにする

2.5．耐久度バーのスクリプト作成
2.5.1 耐久度を表示するバーを描く
2.5.2 ポイが水に使っているときに耐久度を減らすようにする
2.5.3 耐久度が 0 になるとポイが破けるようにする

Sくん　スケジュール表はこうです．真ん中に矢印が描いてある部分は二人で協力して作業する予定です．マイルストーンは 3 つ設定してみました．図で「MS」と書いてあるのがそうです．



M先生　いいね．作業手順を考えるということも広い意味でいうとプログラミングなんだよ．運動会や演奏会で配られるスケジュールのことをプログラムっていうよね．だから，プログラムがうまい人は作業手順を考えるのもうまいんだよ．

Tさん 「手順を考える」のは同じなのね.

M先生 そのとおり．スケジュールを立てるときのポイントを整理しておいたよ.

- 作業手順が正しいかどうか
- 余裕を持った計画であるか
- 計画を全て遂行できなくとも，作品として意味のあるものが早い段階で完成できるように考えられているか
- マイルストーンが細かく設定されすぎていないか，逆に大まかすぎないか
- 同時平行で作業できない仕事を分担していないか

No. 8 – 3 やってみよう！

作品づくりの作業手順書とスケジュール表を作りましょう.

8.5 実装

M先生 あとは“**実装**”を計画通りに進めるだけだね.

Sくん 実装って何ですか？

M先生 Squeak 上でタイルを組み合わせて，実際にプログラムを作る作業のことだよ.

Tさん 何か注意することはありますか？

M先生 仕様書を見ながら作っていくことになると思うけれど，難しくて実装に手間がかかりすぎる部分は，スケジュールも参考にしながら，複雑なものを簡素化してしまう勇気も必要だね.

Sくん わかりました.

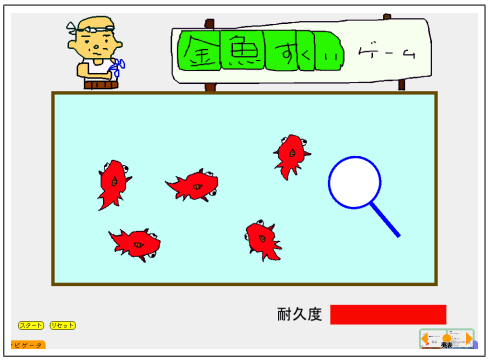
M先生 最後に，実際にかかった作業時間を記録しておくことを忘れないでね．最後の報告書を書くときに使うから．では，がんばってね.

No. 8 – 4 やってみよう！

作品を実装しましょう.

Project 9

まとめと評価



作品が完成したら，みんなに作品を紹介するための「プレゼンテーション」を作成しましょう．Squeak はプレゼンテーションを作るための道具にもなります．作品を紹介できる準備が整ったら，他人に作品を「評価」してもらいましょう．自己満足に終わらないように，評価の結果を参考に改善を行うことも重要です．最後に製作過程をまとめた「報告書」を作成して，作品づくりのプロジェクトが完了します．

キーワード

プレゼンテーション，評価，報告書，スレッド・ナビゲータ

9.1 プレゼンテーションの作成

M先生　作品は完成したかな？

Ｓくん　なんとかできました．でも寝不足です……．

M先生　お疲れ様．では“プレゼンテーション”の準備をしよう．

Ｓくん　プレゼンテーションとは何ですか？

M先生　発表のことだよ．

Ｓくん　発表は得意です．パワーポイント[*1]の使い方はマスターしています．

M先生　いやいや，せっかく Squeak を使っているのだから．Squeak を使って発表しようよ．パワーポイントのスライドショーの途中で，Squeak に切り替えるのはいやだよね．

Ｔさん　Squeak はプレゼンテーションにも使えるんですね！ でもどうやってやるのですか？

M先生　OK．でも，その前に発表のアウトラインを見せてくれる？

Ｓくん　はい．このようにしました．

1. タイトル
2. 対象と使い方
3. デモ
4. 工夫したこと

Ｔさん　4 枚のスライドを作るつもりでした．

M先生　では，Squeak を使って 4 枚のスライドを作ってみようか．

Ｔさん　スライドはどのように作るのですか？

M先生　簡単だよ．プロジェクトで代用する．つまり，1 スライドに対して 1 つのプロジェクトを作れば

[*1] Microsoft PowerPoint

いいんだ．

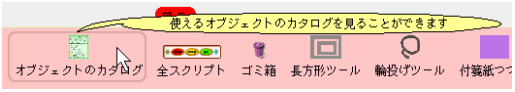
Ｓくん　でも，それじゃ切り替えが面倒じゃ……．

M先生　大丈夫．とりあえず作ってみてね．スライド用のプロジェクトを作っておいて，その中に 4 つのプロジェクトを並べておいておくと，後で整理しやすいよ．

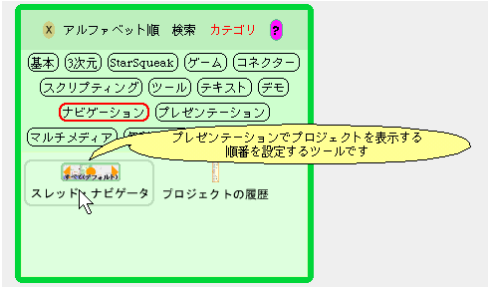
Ｓくん　こんな感じでいいでしょうか？

M先生　OK．では，これらのスライドを一つの流れとしてまとめよう．「ツール」フラップから，「オブジェクトのカタログ」を取り出してくれるかな？

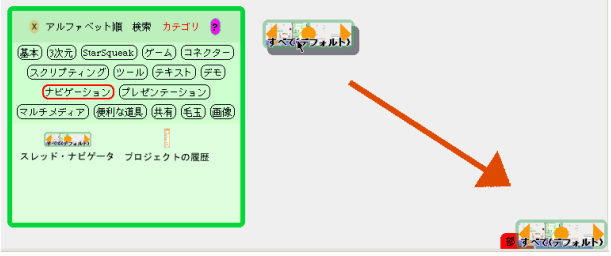
Ｓくん　はい．



M先生　「ナビゲーション」の中に“スレッド・ナビゲータ”というツールがあるでしょ．それを 1 つ取り出してみてね．



Ｓくん　自動的に右下に移動してしまいました．



M先生　それでいいんだよ．スレッド・ナビゲータは自動的に右下に配置されるようになっているんだ．これを使って，プロジェクトを切り替えることができるよ．スレッド・ナビゲータの右側や左側をクリックしてごらん．

Tさん　はい．

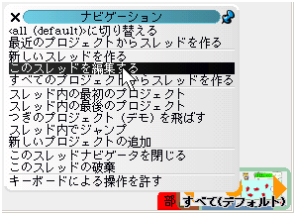
Sくん　あ，プロジェクトが切り替わった！

M先生　これで，スムースに発表ができるね．

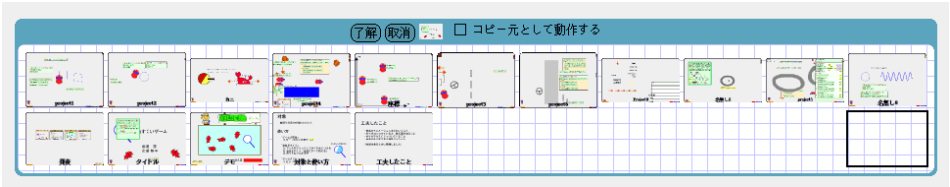
Tさん　そうですね．でも，順番がばらばらなんですが……．

M先生　その場合は，真ん中の矢印の部分をクリックして，

M先生　「このスレッドを編集する」を選ぶといいよ．



Sくん　スライド一覧が出ました．



M先生　これを操作して，スライドの順番を入れ替えることができるよね．できたら「了解」をクリックして完了だ．

Tさん　なるほど．これがあれば，プロジェクトがスライドの代わりになりますね．ところで，要らないスライドはどうしたらいいのですか？

M先生　要らないスライドは一覧から取り出してゴミ箱に捨ててしまっていいよ．スライド一覧の画像を捨ててもプロジェクト自体を捨てたことにはならないから安心してね．

Sくん　わかりました．

M先生　あとは全画面表示にして，フラップを隠すようにすれば完璧だね．全画面表示は「ナビゲータ」タブからできるよ．フラップを隠すには，ワールドメニューの「フラップ」を選んで，そのメニューのチェックを全部外すとできるよ．

Tさん　素敵な発表ができそうね．

M先生　がんばってね．

9.2　作品の評価

M先生　発表はうまくいったかな？

Ｓくん　はい．おかげさまで，うまくいきました．みんなに実際に遊んでもらって，感想を書いてもらいました．

M先生　すばらしいね．どんな感想だった？

Ｔさん　「金魚をすくった数を自動でカウントできるといい」「金魚のすくい方によってポイの耐久度の減り方を変えた方がいい」等ですね．

M先生　なるほどね．そういう“評価”を書きとめておいて，報告書に書けるといいね．

Ｓくん　はい．時間を作って，金魚すくいゲームを更に改造していきたいと思います．

M先生　できあがった作品についての評価だけではなく，企画書や仕様書，スケジューリング等がどうだったかについても評価できると更にいいね．報告書には，実際の作業時間を実績として載せて欲しいな．予定と実績がどう違ったかを分析してみると面白いと思うよ．

Ｓくん　わかりました．

9.3　報告書の作成

M先生　最後にプロジェクトの成果をまとめた“報告書”の作り方について解説しておくね．報告書に是非含めて欲しいものを下にリストアップしておくよ．

- 作業手順やスケジュールの実績
- 実装中に行った設計，実装の解説
- 作品やプロジェクトに対する評価（他人からの評価と自己評価）
- わかったことと感想（複数人の場合は個人で書くこと）

M先生　これ以外にも，企画書や仕様書，作業手順書・スケジュール表などの資料は全て載せておいて欲しい．わかったことや感想は時間をかけて書いて欲しいけれど，新しい資料を作るのにそこまで時間をかける必要はないよ．これまで作ってきた資料を有効に活用してね．

Ｓくん　了解しました．とりあえず二人で目次を作ってみたいと思います．

30 分後…….

Ｔさん　先生，目次ができました．

```
1．プロジェクトの概要
  1.1 メンバー
  1.2 金魚すくいゲームの企画

2．プロジェクトの実績
  2.1 作業手順の計画と実績
  2.2 スケジュールの計画と実績

3．設計と実装
  3.1．金魚のフローチャートと実装の解説
  3.2．ポイのフローチャートと実装の解説

4．感想とわかったこと
  4.1 杉浦
  4.2 広瀬

5．資料
  5.1 クラスの人からの評価結果
  5.2 金魚すくいゲームマニュアル
```

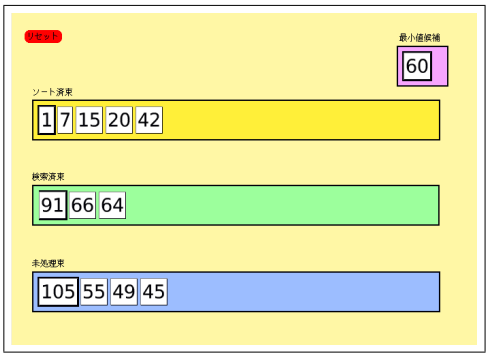
M先生　うん．いいね．素晴らしい報告書になりそうだ．がんばってね．

第 IV 部

アルゴリズムの組み立て

Project 10

並び替えをしてみよう



第 IV 部では「アルゴリズムの組み立て」を扱います．主な作業は，人間が行っている仕事を，コンピュータができる仕事の「手順」へ変換するという作業です．このプロジェクトでは「アルゴリズムの組み立て」の最初の例題として「並び替え」を扱います．アルゴリズムを組み立てる最初の段階として，人間が行っている作業の手順を記述し，それを手作業でやってみることから始めます．次にそれを Squeak を使ってプログラミングをしてみます．この過程で，作業の手順を記述してみることがプログラミングに役立つことがわかるはずです．

✎ キーワード

アルゴリズム，並び替え，最小値選択法，入れ物

10.1　アルゴリズム

M先生　ここからは“アルゴリズム”について考えていこう．

Sくん　アルゴリズムって何ですか？

M先生　アルゴリズムというのは，仕事をする手順のことだ．例えば，ばらばらの数字の書かれたカードがたくさん並んでいたとする．

M先生　これを左から番号の小さい順，つまり昇順に並び替えることができるかな？

Sくん　そんなの簡単ですよ，こんな感じかな．

M先生　じゃあ，どのような手順で並び替えをしたかを説明できるかな？

M先生　うーん．適当に数字の小さいのを選んで……いや，違うな．まずはカード全体を確認するのかな？ あれ？ うまく説明できないや．

M先生　それじゃ，他人やコンピュータに“**並び替え**”の手順を正確に指示することはできないよね．

Tさん　でも先生，人間なら手順を詳しく説明しなくともカードを並び替えることができますよね？ 今のSくんみたいに．

M先生　そうだね．人間なら目的がわかっていれば，手順は「なんとなく」でもできてしまうね．でも，人間ではなく，コンピュータに並び替えをさせる場合，手順を明確にできなければプログラムを組み立てることはできないよね．

Sくん　並び替えの手順なんていままで意識したことがありませんでした．

M先生　人間が手順を意識しないでやっていることは結構たくさんあるんだ．例えば，顔の認識なんかはその1つだね．

Sくん　うーん，たしかに．どうやって人の顔を見分けているんだろう．

M先生　ある仕事をプログラムとして組み立てるためには，その仕事の手順を考えなければいけない．その手順のことをアルゴリズムというんだよ．

Sくん・Tさん　分かりました．

M先生　それでは，並び替えをテーマにして，アルゴリズムについて考えていこう．

10.2　手作業による並び替え

M先生　まずはコンピュータを使わずに，手作業でカードの並び替えをしてみよう．

Sくん・Tさん　はい．

M先生　いきなり並び替えのアルゴリズムを考え出すのは難しいだろうから，アルゴリズムは用意しておいた．名づけて“**最小値選択法**”[*1]だ．並び替えのアルゴリズムはいろいろあるけれど，手作業で理解しやすいものを選んでおいたよ．

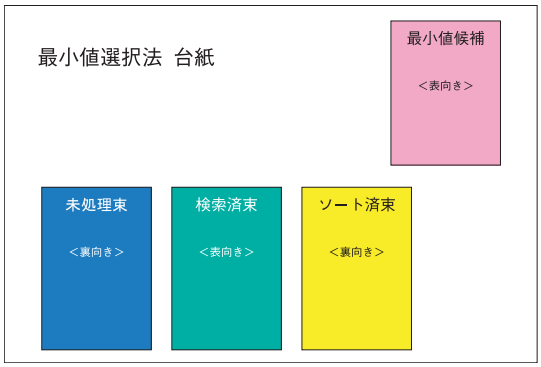
Sくん　初心者用ですね．

M先生　そうだね．では準備をはじめよう．まずは並び替えるカードを用意しよう．名刺大のカードにばらばらの数字を書いたものを40枚用意してくれるかな？ 数字はダブっていても構わないよ．

Sくん　できました．トランプのように，どちらの方向から見ても数字がわかるようにしておきました．

[*1] 一般的には「選択法」と呼ばれているアルゴリズムです．

M先生　いいね．じゃあ，そのカードを置くための台紙を作っておいたから，これを机の上に敷いておいて．



Sくん　なんかゲームみたいで面白そうですね．でも先生，「ソート」って何ですか？

M先生　英語で「並び替え」という意味だよ．だから「ソート済み」は「並び替え済み」という意味だね．

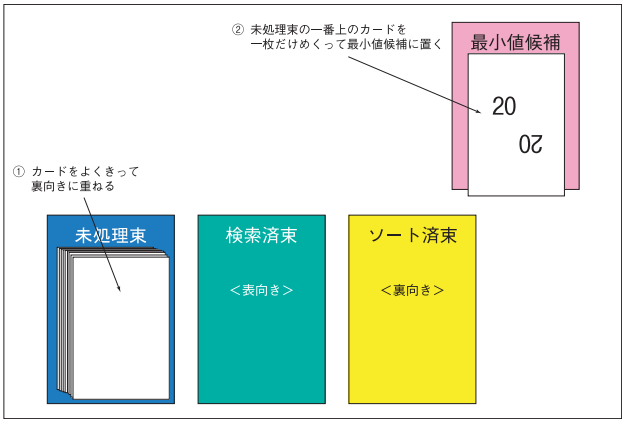
Sくん　なるほど．

M先生　台紙には，未処理束，検索済束，ソート済束，最小値候補という 4 つのカードを置く場所がある．それぞれの場所にカードを置く向きも書いてある．数字が書いてある方が表向き，書いていない方が裏向きだよ．以下にそれぞれの場所についてまとめておいたよ．

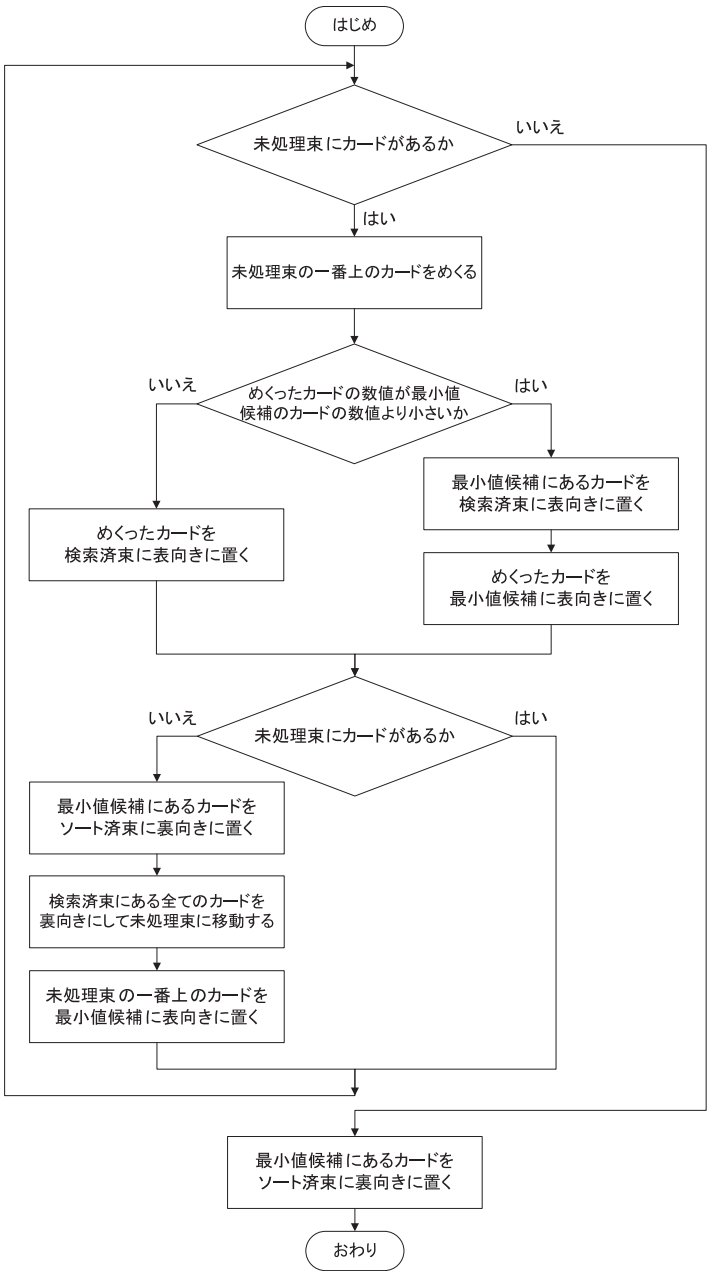
- 未処理束…並び替えが終了していないカードを置いておく場所．並び替えを始める前に，全てのカードをよくシャッフルして置いておく．
- 検索済束…最小値候補との比較が終わったカードを置いておく場所．
- 最小値候補…未処理束の中から一番小さな数字の書かれたカードを選ぶために使う場所．1 枚カードが置けるスペースがあればよい．
- ソート済束…並び替えが終わったカードを置いておく場所．最終的にここに昇順にカードが並ぶ．

M先生　最初にカードをよくきってくれるかな？ いきなり 40 枚を並び替えるのは大変だから，4 枚だけで練習してみようか．4 枚のカードの束を未処理束というところに裏向きにして置くんだ．置いたら一番上のカードを 1 枚だけめくって，最小値候補と書いてあるところに表向きに置く．

Tさん　こうですね．



M先生　これで準備は完了だ．では，最小値選択法のアルゴリズムを説明するよ．アルゴリズムをフローチャートとして書いてみたから，よく読んで確認してみて．



Sくん　フローチャートの手順どおりにやればいいんですね．

M先生　そうだね．フローチャートをよく読んで，その通りにやってみよう．コンピュータになったつもりでね．

Tさん　命令どおりに動作するイメージですね．

Sくん　はい．えっと，まずは未処理束にカードがあるから，未処理束から１枚カードをめくる．えっとめくったカードは「10」ですね．それと最小値候補に置いてあるカードと比較する．最小値候補

にあるカードは「20」だから，めくったカードを検索済束に移動する．

Tさん　次もやってみましょうよ．

Sくん　えっと，未処理束からめくると，「20」だ．あれっ，最小値候補と同じ数字だ．その場合は，検索済束に移動っと．次をめくると「3」だ．めくったカードが最小値候補にあるカードより小さい場合はめくったカードと最小値候補のカードを入れ替えて，最小値候補にあった「20」を検索済束へ移動する．これで未処理束にカードがなくなったから，最小値候補にある「3」をソート済み束に移動する．えっと，次に，検索済み束にある 3 枚のカードを未処理束に移動して，一番上のカードを最小値候補に移動するっと．なるほど，ようやく 1 枚並び替えが終わったんだ．

M先生　最終的にはソート済束というところに昇順にカードが並ぶはずだよ．ふつうにやれば 40 枚のカードの並び替えは 30 分くらいで終わるはずだよ．

Tさん　私もやってみたいわ．

M先生　T さんには計測係という大事な仕事があるから，並び替えは S くんに任せよう．並び替えが終わったら，その効率について考えてみて欲しいから，並び替えにかかった時間を測定してもらいたいんだ．

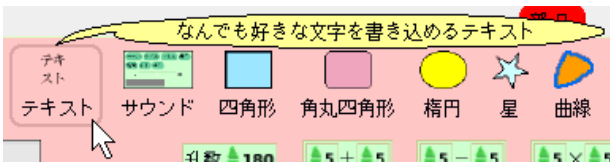
Tさん　どういう風に測定すればいいんですか？

M先生　ソート済束に 1 枚カードが移動するまでの時間を計測しておいてくれるかい？ 40 枚カードがあるから，40 回分の記録がとれるはずだよ．

Tさん　分かりました．えっと，時間をはかるためのストップウォッチが必要ですね．あっ，Squeak で作れるかも．S くんはちょっと並び替えの練習をしておいて．

Sくん　はーい．

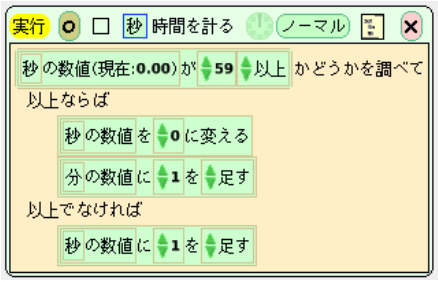
Tさん　分と秒の表示には部品フラップにある，テキスト[*2]を 2 つ用意すれば OK ね．



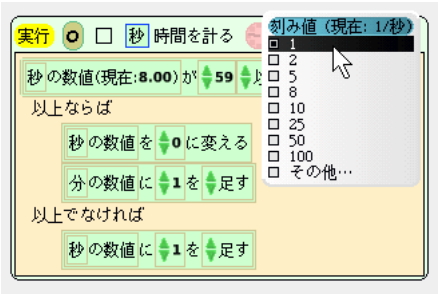
Tさん　取り出したテキストを「秒」と「分」という名前にして．

[*2] テキストの解説は P.160（A.13）にあります．

Tさん　「時間を計る」という名前のスクリプトを作っておいて．

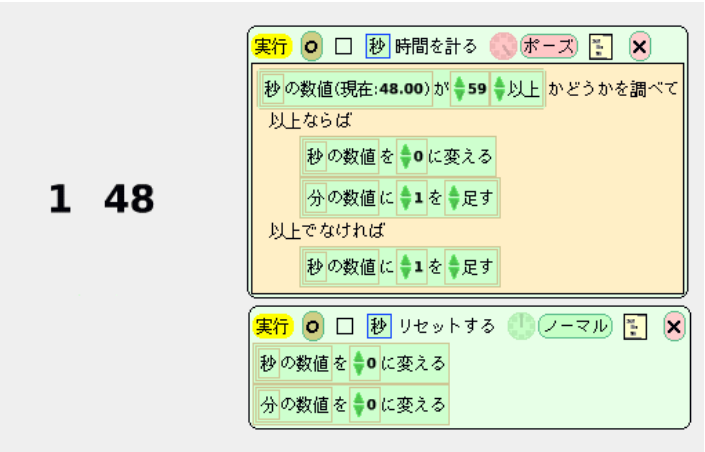


Tさん　スクリプトの刻み値を 1 回/秒にすれば OK ね．



M先生　リセットのためのスクリプトも作っておけば完璧だね．

Tさん　はい．こんな感じでどうでしょう．見た目はちょっと悪いですけど，ちゃんと時間は計れます．



M先生　いいね．じゃあ，並び替えの練習は終わりだ．40 枚のカードを並び替えてみよう．

No. 10 – 1　やってみよう！

最小値選択法で 40 枚のカードを並び替えてみましょう．
詳細は付録 C（P.171）を参照してください．

10.3　コンピュータによる並び替え（1）– カードの準備

M先生　お疲れ様．うまく並び替えはできたかな？

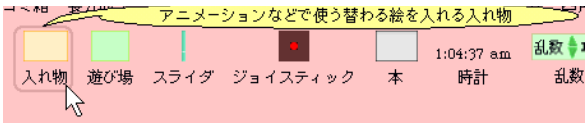
Ｓくん　腕が疲れました．でも，アルゴリズムはちゃんと理解できました．

M先生　では，今度は同じ仕事をコンピュータに指示してみよう．

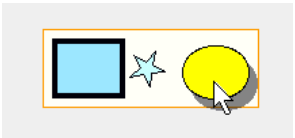
Ｔさん　そんなことができるんですね．すごいっ！

M先生　まず，Squeak 上にカードの束を作るために“入れ物”という部品を使うよ．入れ物を部品フラップから出してみて．

Ｓくん　これですね．

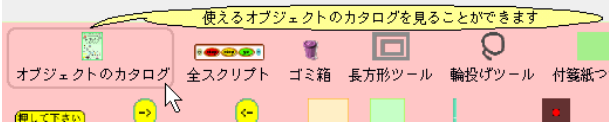


M先生　入れ物にはオブジェクトを入れることができる．部品フラップから星や四角を出して，入れ物オブジェクトの上に重ねてごらん．オブジェクトが入れ物の中に入るはずだよ．



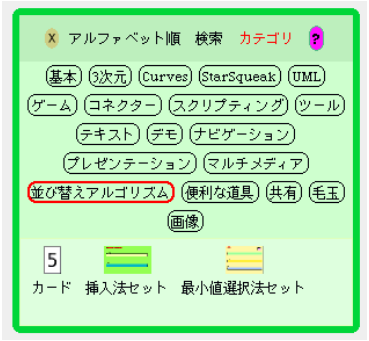
Ｓくん　なるほど．これにカードを入れれば，カード束を作れそうですね．

M先生　その通り．では部品フラップから，オブジェクトのカタログを出してごらん．



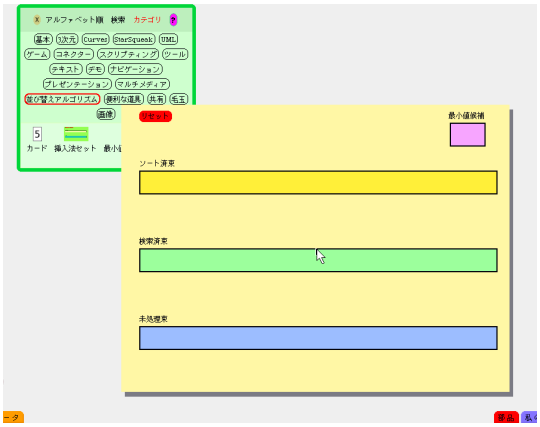
M先生　カテゴリの中から「並び替えアルゴリズム」をクリックしてみよう．

Ｓくん　あ，カードが登録されていますね．あと最小値選択法セットというのがある．



M先生　それを使ってプログラムを組み立ててみよう．

Sくん　まずは，最小値選択法セットを取り出してっと．既にカード束が用意されているんですね．便利だなー．



Tさん　えっと，バラバラの数字が書かれたカードがたくさん必要ね．

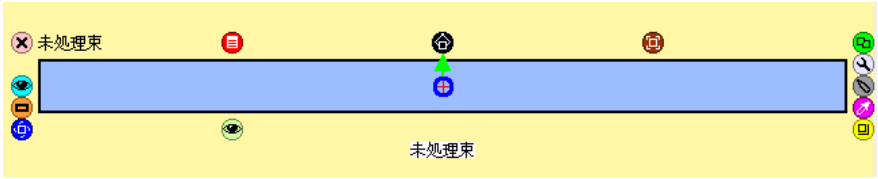
Sくん　40 枚かー．嫌だなー．

M先生　カードを追加するのもプログラムでやってしまおう．まず，未処理束にスクリプトを 1 つ作ってくれるかな？

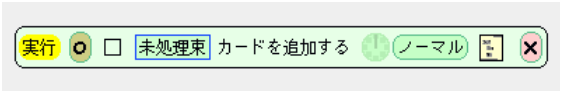
Sくん　未処理束のビューアーが表示できないです．どうすればいいのですか？

M先生　それは未処理束などのカード束が「遊び場」という部品の中に入っているからだね．台紙の代わりに遊び場という部品を使っているんだ．そういう場合，未処理束にマウスカーソルをあわせて，ハロを表示させる動作を 2 回やってごらん．「Alt（Mac の場合は Apple キー）を押しながらオブジェクトをクリック」を 2 回だよ．それで未処理束のハロが表示されるよ．

Sくん　はい．ハロが表示できました．



Sくん　えっと，スクリプトの名前は「カードを追加する」でいいかな．



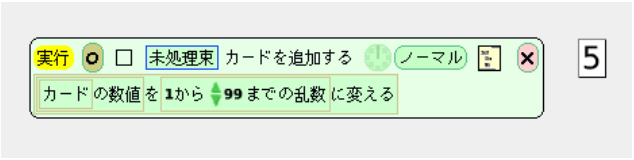
Tさん　ばらばらの数字のカードを作る必要があるから，まずはカードの数字を乱数タイルで変更すればいいんじゃない？

M先生　そうだね．カードはテキスト[*3]の色を変更して，カードのような見た目にしているだけだよ．だから，カードのビューアーを開いて，「テキスト」カテゴリを見てくれる？

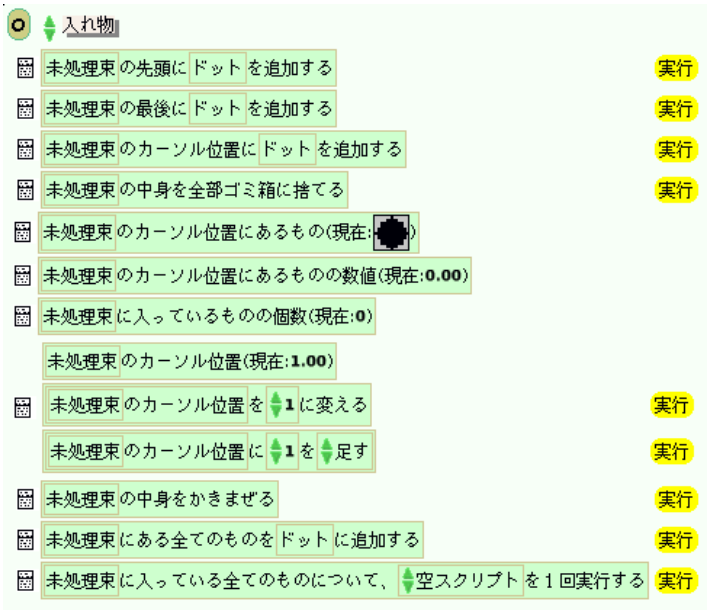
[*3] テキストの解説は P.160（A.13）にあります．

Sくん　カードの数値を変更するためのタイルがありますね．

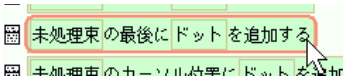
Sくん　となると．プログラムはこうかな．



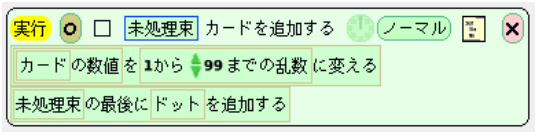
M先生　いいね．そうしたら，未処理束のビューアーを開いて，入れ物カテゴリを表示して，未処理束にカードを追加するためのタイルを取り出そう．



Tさん　入れ物には特別なカテゴリがあるんですね．えっと，未処理束に追加するんだから，これかな？

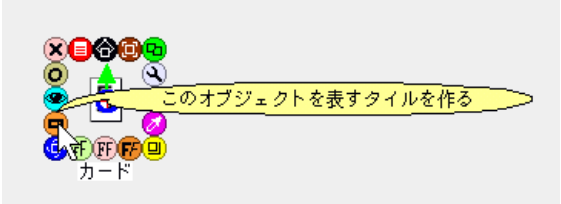


Sくん　よし．あとは追加するカードを指定すればOKかな．

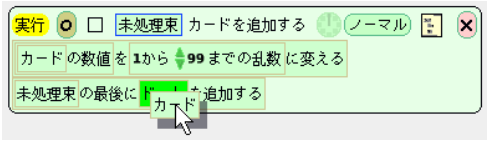


Tさん　先生，タイルにある「ドット」というのはなんですか？

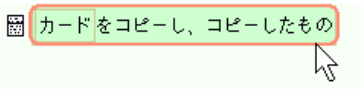
M先生　「ドット」というのは，仮のオブジェクトの名前だよ．「ドット」という部分にはオブジェクトを表現するためのタイルを入れることができる．例えば，カードのハロを表示させてごらん．「四角ハロ」をクリックすると，カードと書かれたタイルが取得できるよ．



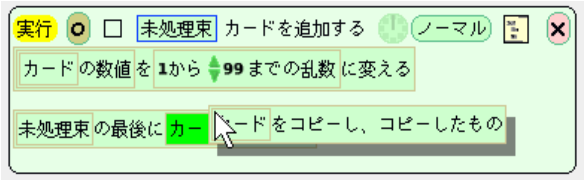
Sくん　じゃあ，この「カード」というタイルをドットのところに入れればいいのかな．でもこれだと，カードが移動するだけで増えないですね．



M先生　そうだね．カードをコピーしたものを追加する必要がありそうだね．カードのビューアーを開いて，「その他」カテゴリを見てみて．「コピーして，コピーしたもの」というタイルがあるね．

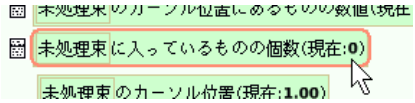


Sくん　なるほど，これを「未処理束の最後にドットを追加する」のドットの部分に入れれば．できたー．

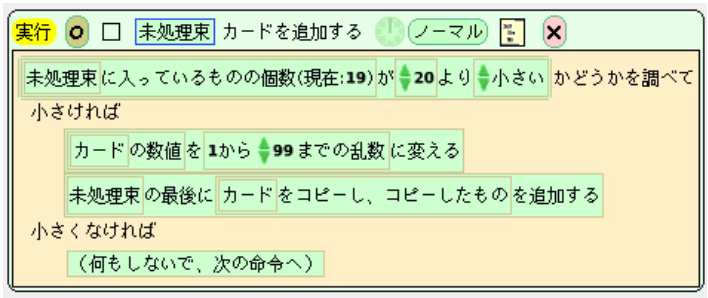


Tさん　でもこれだと何枚追加されたのかよくわからないですね．

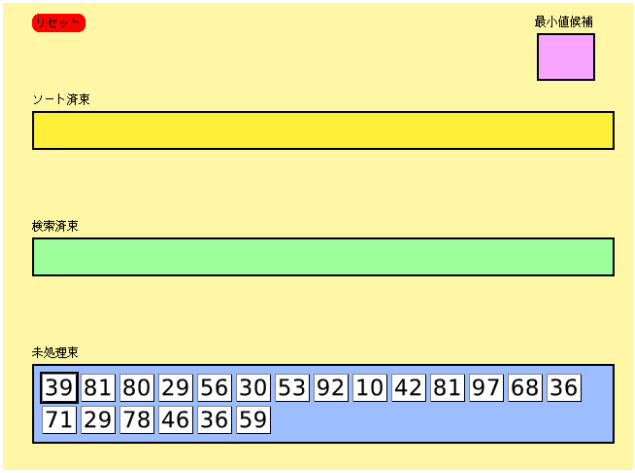
M先生　入れ物に入っているオブジェクトの個数を調べるためのタイルもあるよ．



Sくん　本当だ，じゃあこれをこうして．ひとまずカードを 20 枚追加するプログラムが書けました．

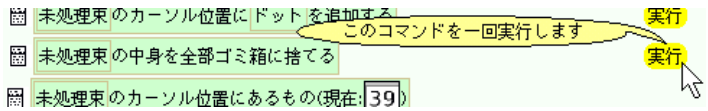


Sくん　よーし．未処理束に 20 枚のカードが追加されたぞ．



Tさん　先生，間違えてカードを追加しちゃった場合はどうすればいいんですか？

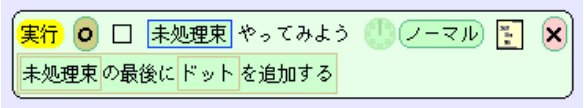
M先生　「中身を全部ゴミ箱に捨てる」という命令タイルを実行すればいいんだ．



Sくん　便利ですね．

No. 10 – 2　やってみよう！

「未処理束の最後にオブジェクトを追加するためのタイル」の「ドット」部分をドットのまま（自分でタイルを入れないで）実行するとどうなるか試してみましょう．



10.4　コンピュータによる並び替え（2）– 並び替えのプログラム

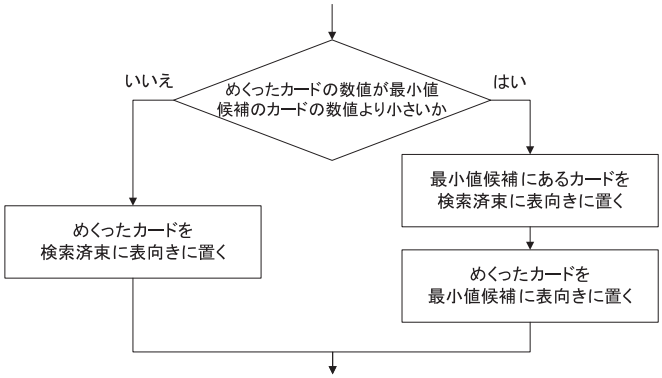
M先生　では，最小値選択法を実際に作っていこう．手作業のときに使ったフローチャートのとおりにプログラムを組み立てていけばできるよ．

Ｓくん　結構たくさんの処理がありますね．

M先生　一度に全部組み立てないで，少しづつ組み立てていこう．一度に全部組み立ててから間違いを探すのは大変だからね．

Ｓくん　そうですね．

M先生　まずはこの部分をやってみたら？

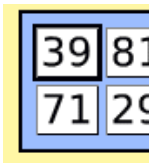


Ｔさん　未処理束に入っているカードの中で最も小さなカードを最小値候補に移動する部分ですね．

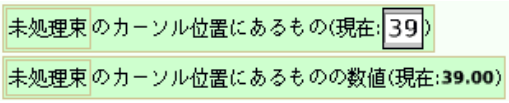
Ｓくん　先生，入れ物の命令タイルにあるカーソルとはなんですか？

M先生　入れ物はカーソルを１つ持っていて，カーソルのあるオブジェクトには太い枠がつくんだ．

Ｓくん　これですね．今は入れ物の先頭の「39」というカードにカーソルがありますね．



M先生　カーソルのあるオブジェクトの情報を調べるためのタイルがあるんだ．



Ｓくん　先生，「カーソル位置にあるもの」と「カーソル位置にあるものの数値」はどう違うんですか？

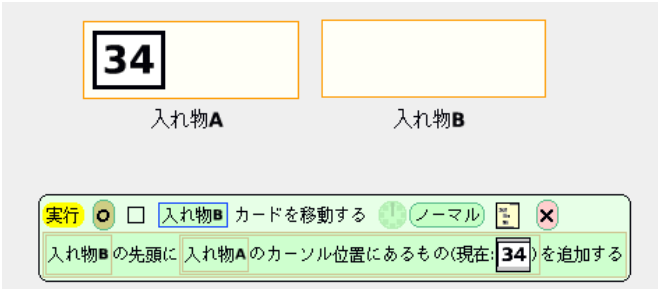
M先生　カードの例でいうと，「カーソル位置にあるもの」は「カード」のことだね．「カーソル位置にあるものの数値」はカードに書かれている数字を示しているんだ．

Ｔさん　場合によって使い分ければいいんですね．

M先生　そうだね．本当はカーソルは移動できるんだけど，最小値選択法の場合は移動する必要はないよ．

Ｓくん　なるほど．あと先生，束から束へカードを移動するときはどうすればいいんですか？

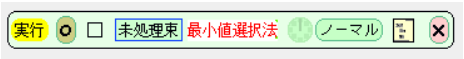
M先生　いい質問だね．ある入れ物に入っているオブジェクトを他の入れ物に追加すると，そのオブジェクトは移動するよ．例えば，入れ物 A と入れ物 B があって，こういうスクリプトを作ったとしようか．



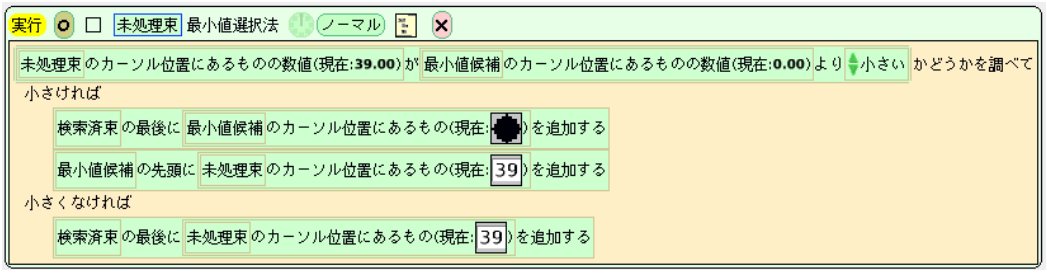
Ｔさん　このスクリプトを実行すると，「34」と書かれたオブジェクトは入れ物 B へ移動するんですね．

M先生　そのとおり．

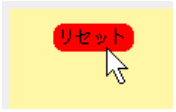
Ｓくん　なるほど．まずは未処理束に最小値選択法というスクリプトを作って．



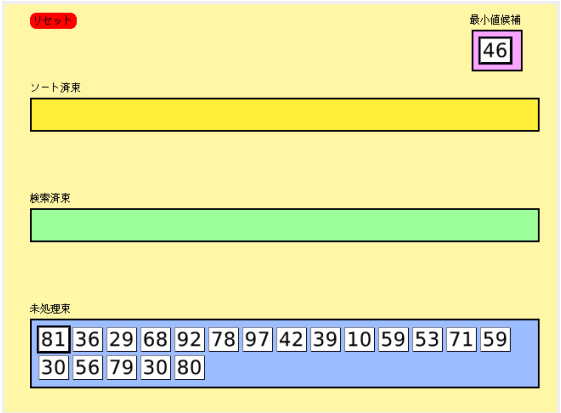
Ｓくん　さっきの部分のプログラムはこんな感じですかね．



M先生　よさそうだね．スクリプトの実行ボタンを何回か押して，期待どおりの動作をするか調べてみるとよい．実行する前にリセットというボタンを押すようにね．このリセットボタンにはカードを最初の位置に戻すための初期化処理がすでにプログラムされているよ．



Tさん　あっ，リセットボタンを押すと，未処理束の中身がシャッフルされて，最小値候補に 1 枚カードが移動したわ．手作業でやったときとまったく一緒ね．



M先生　そのとおり．だから，手作業のフローチャートをそのままプログラムすれば OK だよ．残りは自分達でやってみて．カードを裏にしたり表にしたりする作業はプログラムする必要はないよ．だから「未処理束の一番上のカードをめくる」という処理は必要ないよね．

Sくん　カードは裏返しになってませんもんね．

M先生　では頑張って．

No. 10 – 3　やってみよう！

最小値選択法のプログラムを完成させましょう．

☞ ヒント

入れ物に関する解説は P.163（A.17）にあります．

No. 10 – 4　やってみよう！

「やってみよう！ No. 10 – 3」が完成したら，検索済束を使わず，合計 3 つの入れ物（未処理束，ソート済束，最小値候補）で，最小値選択法をプログラムしてみましょう．
検索済束を使わないようにするためには，カーソルを移動して，未処理束に入っているカードを順番に調べる必要があります．

☞ ヒント

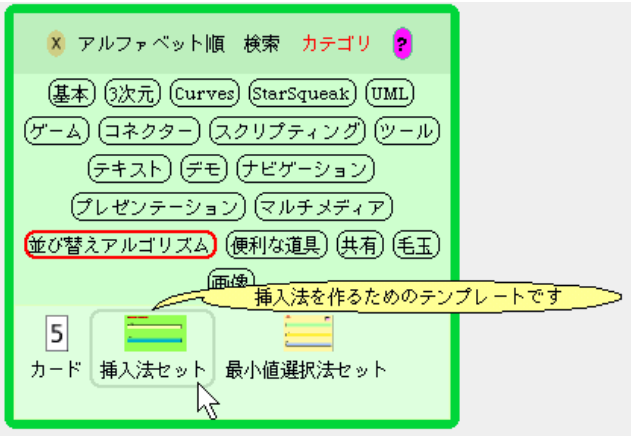
カーソルに関する解説は P.163（A.17）にあります．

練習問題

練習問題 10.1

「挿入法」という「最小値選択法」とは異なるアルゴリズムでカードを並び替えてみましょう．以下の手順で進めるとよいでしょう．

1. 挿入法の動作例は http://www.crew.sfc.keio.ac.jp/squeak/sort から閲覧することができます．動作例を確認して，フローチャートを書きましょう．
2. オブジェクトのカタログにある「挿入法セット」に並び替えるためのカードを追加します．このとき，カードに書かれている数字は 999 以下になるようにします．



3. ソート済束の最後にあらかじめ 999 のカード（未処理束のどのカードより大きな数字，番兵値という）が入っている場合のプログラムを作りましょう（「挿入法セット」の「番兵ありリセット」というボタンを使う）．
4. 3. で作ったプログラムを改造して，並び替えをはじめる前にソート済束にカードがまったくない場合でも動作するプログラムを作りましょう（「挿入法セット」の「番兵なしリセット」というボタンを使う）．

練習問題 10.2

P.120 にある「最小値選択法のフローチャート」には「未処理束にカードがあるか」という処理が 2 回あります．これを 1 回に減らせないか考えて，新しいフローチャートを考えてみましょう．

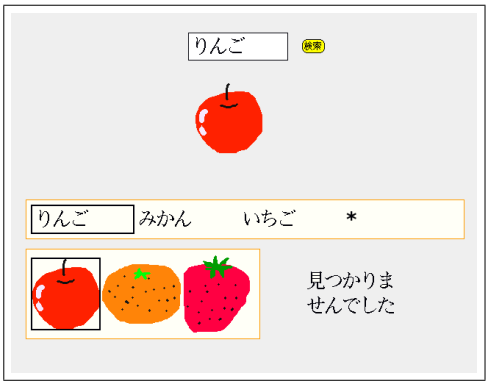
フローチャートができたら，プログラミングに挑戦してみましょう．

練習問題 10.3

最小値選択法や挿入法以外のオリジナルの並び替えアルゴリズムを考えて，プログラミングをしてみましょう．

Project 11

辞書を作ってみよう



最後のプロジェクトでは Project 10 で学んだことを活かして，「絵辞書」のプログラムを作ります．「絵辞書」は単語を入力すると，その単語に対応する絵が表示されるようなプログラムです．辞書のアルゴリズムを作るためには，単語を検索するためアルゴリズムを考える必要があります．このプロジェクトの前半では「検索」のアルゴリズムを考え，後半でそれを応用して「辞書」のアルゴリズムを完成させます．練習問題にも取り組んで，自分でアルゴリズムを考え出すことにも挑戦してください．

キーワード

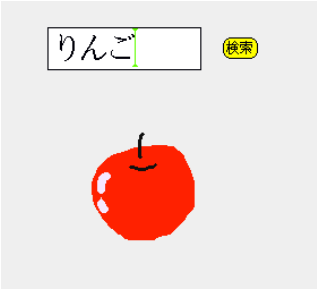
絵辞書，検索，カーソル，番兵

11.1　検索のアルゴリズム

M先生　このプロジェクトでは“絵辞書”を作ってみようと思う．

Sくん　先生，「絵辞書」って何ですか？

M先生　普通の英和辞書だと，「apple」という単語を引くと，「りんご」と出てくるでしょ．絵辞書の場合は「りんご」と引くとリンゴの絵がでてくるようなものだよ．こんな感じかな．



Tさん　面白そうですね．

Sくん　でも複雑で難しそうです．

M先生　一度にやると大変だから，順番に考えていこう．まずは“検索”のアルゴリズムを考えてみようか．

Ｓくん　先生，検索ってなんですか？

M先生　たくさんのものの中から，目的のものを探し出すことだよ．「Web を検索する」なんていうでしょ？

Ｔさん　前のプロジェクトの並び替えでも，小さい数字の書かれたカードを検索していましたよね．

M先生　そうだね．今回は数字の書かれたカードではなく，入れ物の中に検索対象となる単語を入れておいて，その中から目的の単語を探すんだよ．

Ｓくん　なるほど．入れ物の中から使う人が入力した単語，例えば「りんご」を探すんですね？

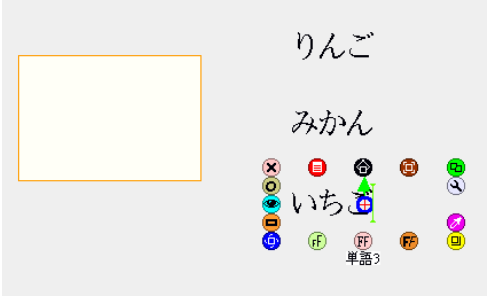
M先生　そのとおり．入力された単語が入れ物のどの位置に入っているのかを探せば良いんだよ．まずは単語を入れた入れ物を１つ使って検索のアルゴリズムを完成させよう．検索のアルゴリズムが完成したら，それを応用して絵辞書を完成させるよ．

Ｓくん　順番に組み立てていくんですね．

M先生　では部品の準備をはじめよう．

Ｓくん　まずは入れ物に単語をいくつか入れますね．入れ物の名前は「単語の入れ物」でいいかな．

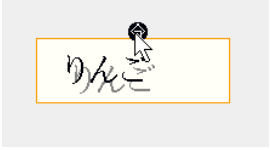
Ｓくん　英語は苦手だから，平仮名で果物の名前にしよう．単語はテキストを使えばいいですよね．とりあえず３つ作っておこう．



Ｓくん　先生，テキストがマウスでつかめません．これじゃ入れ物に入らないですね．

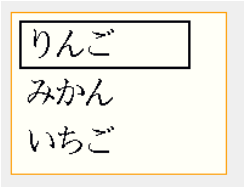
M先生　テキストをつかむ場合は黒ハロを使うとよい．入れ物から取り出す場合も同じだよ．

Ｓくん　はい．うーんでもうまく入らないなー．



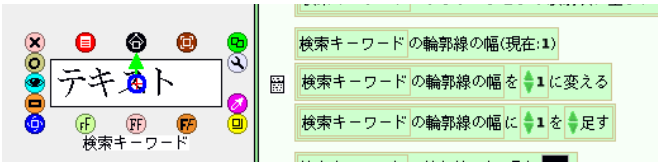
M先生　テキストを入れ物に入れようとしないで，黒ハロを入れようとするとよいよ．

Sくん　入りました．



Tさん　探したい単語を入力するための部品も必要ね．これもテキストでいいんじゃない？

Sくん　なるほど．じゃあ，「検索キーワード」って名前のテキストを用意するよ．背景の色を変えて，枠をつけるとそれっぽく見えるかな．

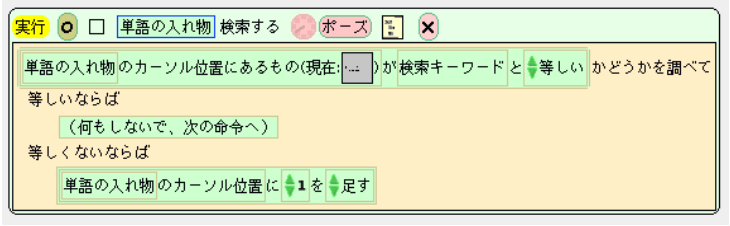


M先生　いいね．それでは，アルゴリズムを考えてみよう．

Sくん　えっと，今回は最小値選択法と違って“カーソル”[*1]を移動させる必要がありそうですね．

Tさん　カーソルを入れ物の先頭から 1 つずつ進めながら，検索キーワードとカーソル位置にある単語の内容を比較すればいいんじゃない？

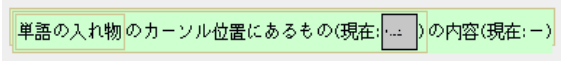
Sくん　そっか．じゃあ，単語の入れ物に「検索する」というスクリプトを作っておいて，こうかな？



M先生　惜しいね．基本的な考え方は OK だけど，これだと「単語の入れ物のカーソル位置にあるオブジェクト」と「検索キーワードのオブジェクト」が同じかどうかを調べていることになる．2 つは違うオブジェクトだよね．これではうまくいかないね．

Tさん　調べたいのは，それぞれの内容が等しいかですね．

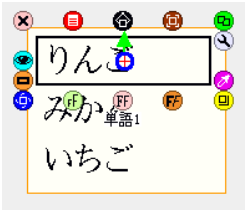
M先生　そういう場合は，既にあるタイルをうまく組み合わせて，こんなタイルを作る必要があるね．



M先生　作るのにちょっと工夫が必要だから説明しよう．単語の入れ物に入っている単語のハロを表示させてみて．どれでもいいよ．

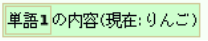
[*1] カーソルに関する詳しい解説が，P.163（A.17.2）にあります．

Ｓくん　じゃあ，先頭の単語でいいかな．できました．



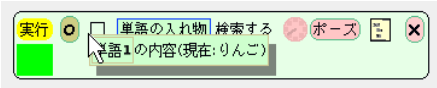
M先生　では，ビューアーを開いて「テキスト」カテゴリを見てくれる．

Ｔさん　「単語１の内容」というタイルがありますね．

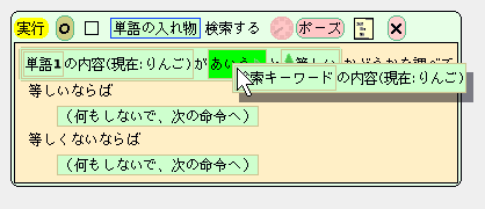


M先生　ひとまず，「単語１の内容」と「検索キーワードの内容」を比較するプログラムを作ってみよう．

Ｓくん　まずは，この「単語１の内容」というタイルをスクリプトに入れて．



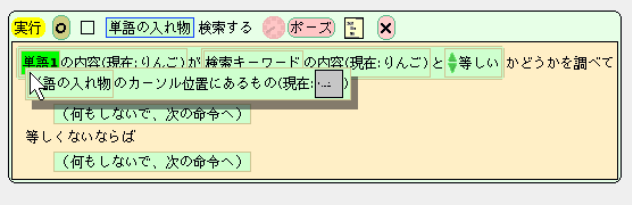
Ｓくん　検索キーワードのビューアーから，「検索キーワードの内容」というタイルを取ってきて比較すると．



M先生　いいね．今比較したいのは「単語１」のオブジェクトの内容ではないよね．

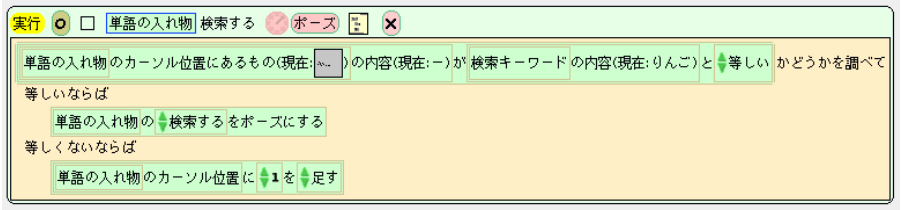
Ｔさん　「単語１」の部分を「カーソル位置にあるもの」にすればいいんじゃない？

Ｓくん　なるほど．こうかな．



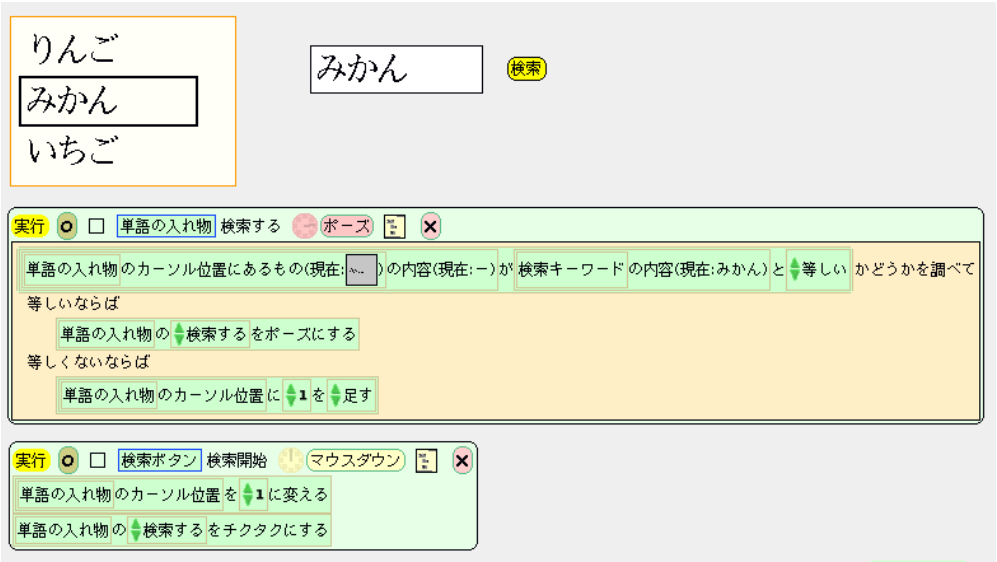
M先生　そのとおり．じゃあ，見つかったらどうすればいいかな？

Sくん　この「検索する」というスクリプトをポーズ[*2]すればOKですね．それ以外はカーソルを1つ進めるから．こうだっ！



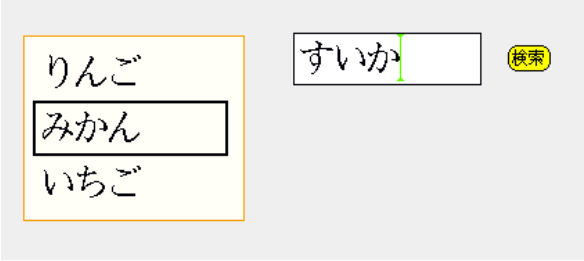
M先生　よいね．検索を始めるためのボタン[*3]を作っておくとよいよ．

Sくん　できました．完璧だー！



M先生　じゃあ，ちょっと意地悪をしよう．単語の入れ物に入っていない単語，例えば「すいか」を検索しようとするとどうなるかな？

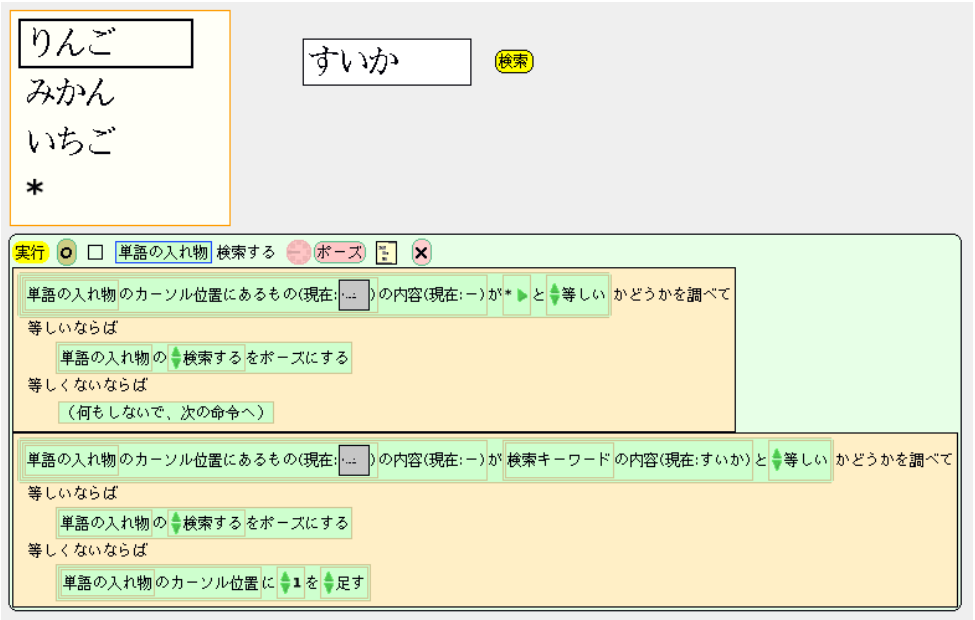
Sくん　プログラムが止まりません．これじゃ困ったなー．



[*2] スクリプトをポーズするためのリモートスクリプティングの解説はP.156（A.8）にあります．
[*3] ボタンの解説はP.158（A.10）にあります．

Tさん　カーソルが単語の入れ物の最後までいったら，プログラムを止めるようにすれば？ 最後に特別なオブジェクトを入れておけばいいのよ．検索することのない単語「＊（アスタリスク）」を入れておくってのはどう？

Sくん　なるほど．検索をするときに，「＊」でないかどうか調べておけばいいから．



M先生　入れ物の最後を示す値を入れておく方法だね．これをプログラミングの用語で“番兵”というよ．色々な場合に使えるアルゴリズムのテクニックだ．

Sくん　なるほど，入れ物の最後を守っているわけですね．

No. 11 – 1　やってみよう！

「＊（アスタリスク）」のような番兵を用いる方法以外で，単語の入れ物に入っていない単語を検索した場合にもうまく動作するようなプログラムを作ってみましょう．

☞ ヒント

カーソルが入れ物の最後にある場合，「カーソルの位置」と「入れ物に入っているものの個数」は等しくなります．また，入れ物に関する解説は P.163（A.17）にあります．

11.2　辞書のアルゴリズム

Ｓくん　先生，検索はできるようになりましたけど，辞書にするにはどうしたらよいのか思い浮かびません．入れ物の 1 番目に「りんご」と「りんごの絵」を両方入れればできる気がするんですけど，入れ物の 1 番目に 2 つのオブジェクトは入らないですよね．

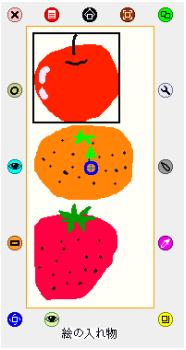
M先生　そうだね．辞書の作り方にも色々あるけれど，最初に説明したとおり，入れ物を 2 つ使う方法が簡単じゃないかな．

Ｔさん　単語の入れ物と絵の入れ物を使うんですね．単語の入れ物の 1 番目には「りんご」，絵の入れ物の 1 番目には「りんごの絵」，単語の入れ物の 2 番目には「みかん」，絵の入れ物の 2 番目には「みかんの絵」のようにしておけばいいんですね．

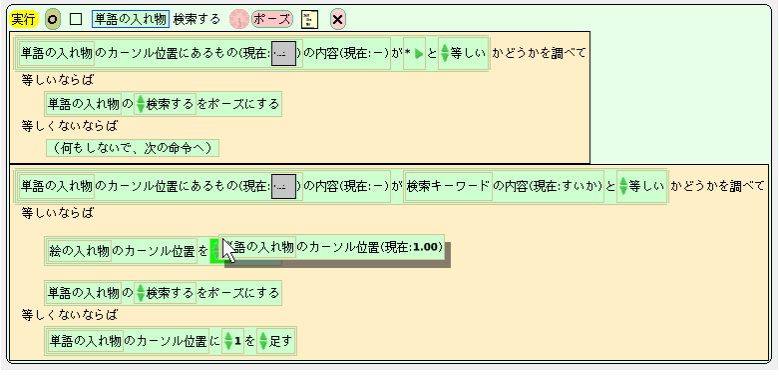
Ｓくん　なるほど．2 つの入れ物の 1 番目に「りんご」と「りんごの絵」の両方が入ることになるんだ．とりあえず，絵の入れ物を「絵の入れ物」という名前で作ってみよう．うーん．単語の入れ物と同じ順番で絵のオブジェクトを入れるのが難しいですね．

M先生　ちょっとテクニックが必要かな．入れ物の先頭にオブジェクトが入りやすいから，入れたい順番と逆順で入れるとうまくいく．つまり，「りんご，みかん，いちご」の順で入れたければ，「いちご，みかん，りんご」の順番でオブジェクトを入れるとよい．

Ｓくん　はい．できました．

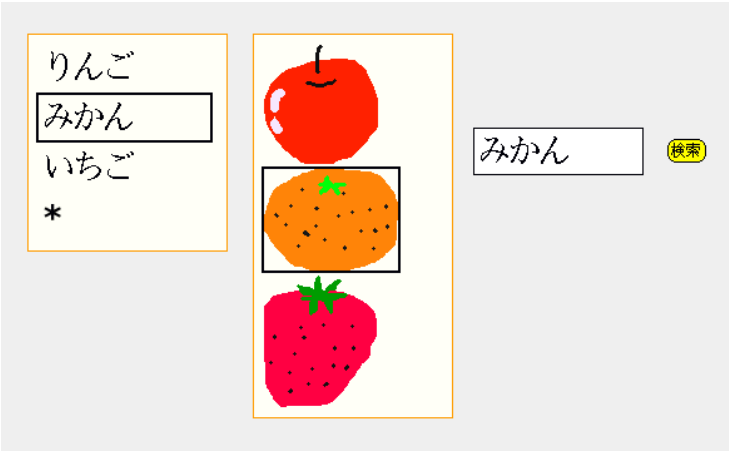


Ｔさん　単語が見つかったら，絵の入れ物のカーソル位置を単語の入れ物のカーソル位置にあわせるんじゃない？　「絵の入れ物のカーソル位置」を「単語の入れ物のカーソル位置」に変えればいいのよ．



Sくん　なるほど，できた．

Sくん　これで見つかった単語のカーソル位置と絵のカーソル位置が同じになるようになったぞ．「みかん」と検索すると，両方の入れ物のカーソルが 2 番目で止まるから OK だね．



M先生　もう一息だね．検索結果を表示するためのオブジェクトを用意しておいて，それを絵の入れ物のカーソル位置にあるオブジェクトに似せればいいんだよ．

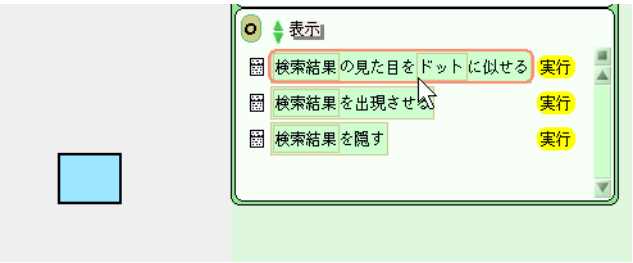
Sくん　そんなことできるんですね．

M先生　まずは，部品フラップから四角形を取り出してごらん．名前は「検索結果」でいいかな．

Sくん　四角形なのに検索結果なんですか？

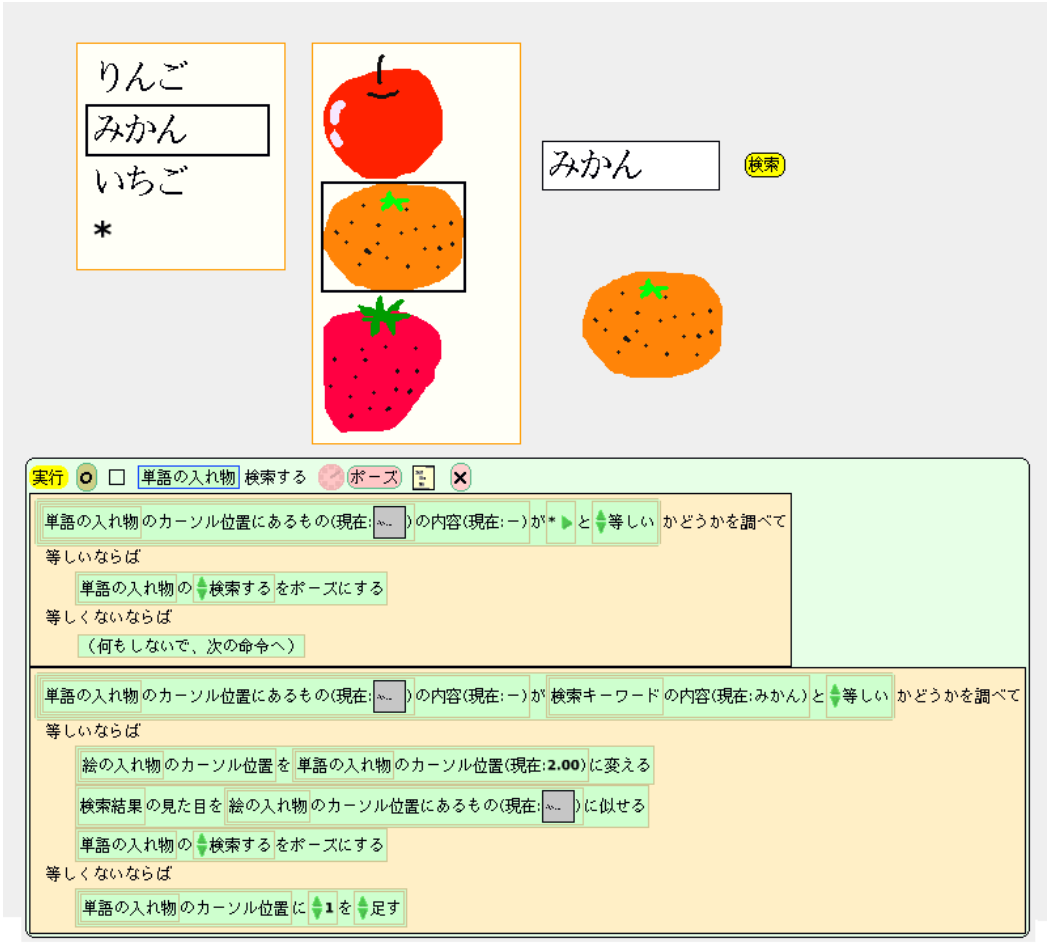
M先生　まあまあ．次に四角形のビューアーを表示させて，「表示」カテゴリを見てみて．

Sくん　あ，「見た目をドットに似せる」っていう．タイルがありますね．



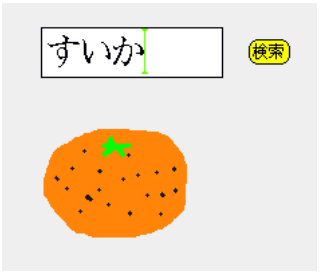
Tさん　絵の入れ物の「カーソル位置にあるもの」に，見た目を似せればいいんじゃない？

Sくん　こうかな．できたー．



M先生　いいね．またちょっと意地悪をするよ．「すいか」を検索するとどうなる？

Tさん　うーん．絵は「みかん」のままですね．これだと検索できたのかできなかったのかわかりにくいですね．

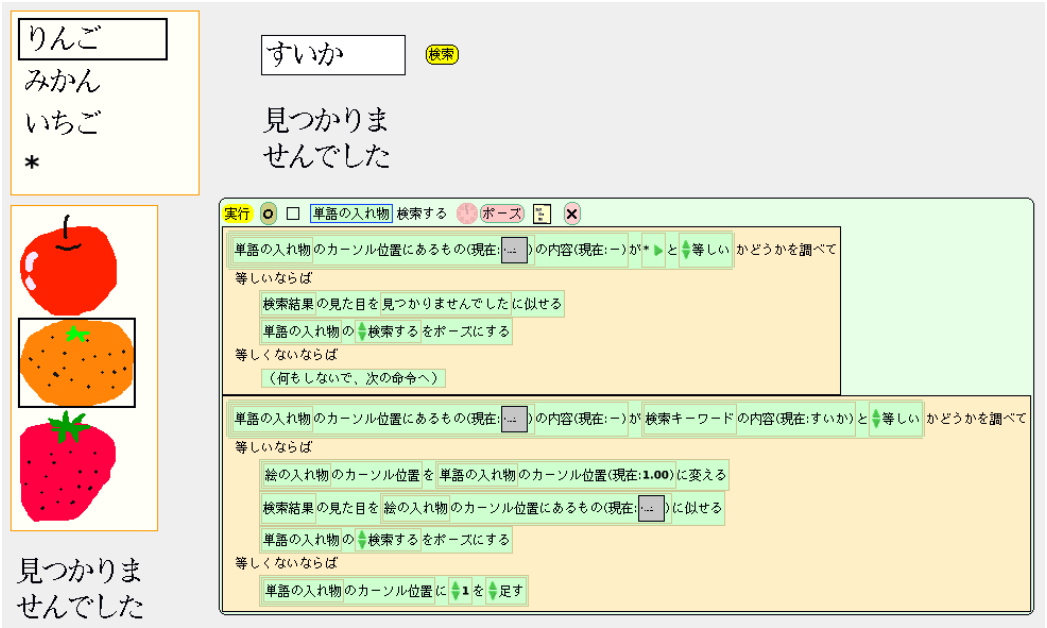


Sくん　「見つかりませんでした」という表示にしたいなー．

M先生　じゃあ，単語の入れ物の検索が番兵まで終わったら「見つかりませんでした」というオブジェクトを表示するようにしておけば？

Sくん　なるほど．見つからなかったときに表示するオブジェクトはテキストでいいかな．

Sくん　プログラムはこうだ．



M先生　さて，絵辞書も完成したね．二人ともお疲れ様．

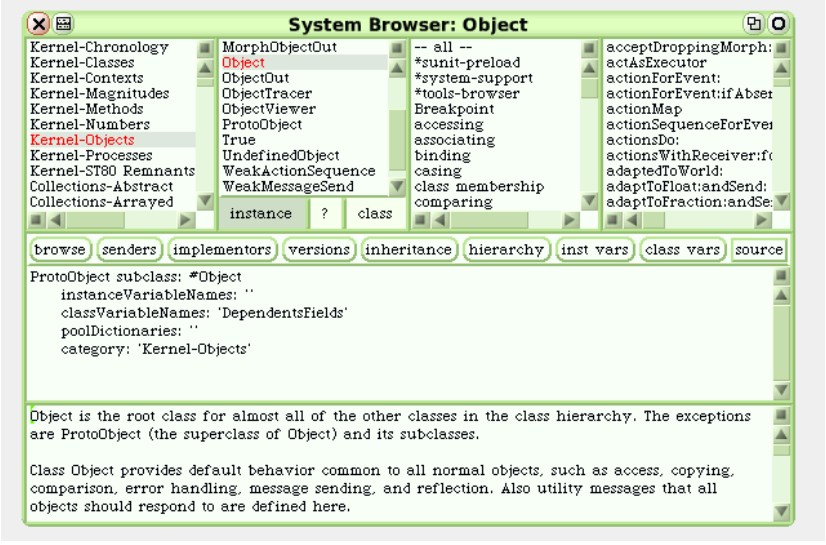
Ｓくん・Ｔさん　先生，ありがとうございました．

M先生　これからもがんばって Squeak を勉強してね．他のプログラミング言語に挑戦してみるのもよいと思うよ．もう二人は立派なプログラマだ．

Ｓくん　照れるなー．

Ｔさん　ありがとうございます．

M先生　今まではタイルでプログラミングをしてきたけれど，Squeak には広大な Smalltalk の世界もあるんだ．ほら．



Ｓくん　えっ？ Smalltalk って何ですか？ しかも英語だらけじゃないですか….

M先生　まあ，ここからが本当のプログラミングの始まりかな．

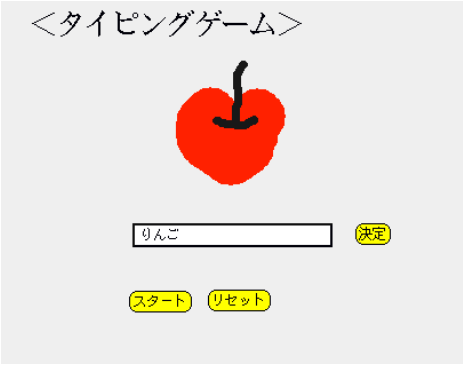
Ｓくん　そんなぁ．

練習問題

練習問題 11.1

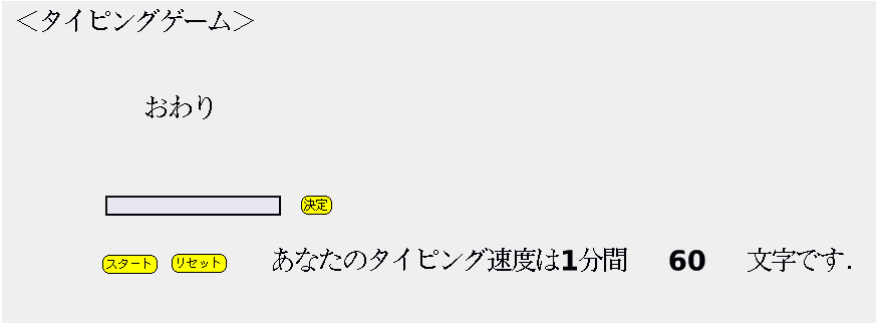
絵辞書を応用して，「タイピングゲーム」を作ってみましょう．

・ スタートボタンを押すと，絵が表示される
・ 表示された絵の単語を入力し，決定ボタンを押す



・ 正しい単語が入力されていれば，次の問題（絵）が表示される
・ 入力した単語が間違えていれば，ボタンを押しても次の問題は表示されない

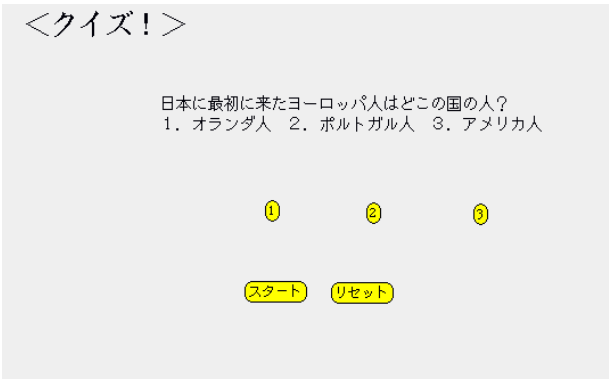
・ 問題の出題が全て終わると，結果（タイピングの速度）を表示する



練習問題 11.2

絵辞書を応用して，「クイズゲーム」を作ってみましょう．

- スタートボタンを押すと，選択式で解答できる問題（歴史に関するクイズでなくともよい）が表示される

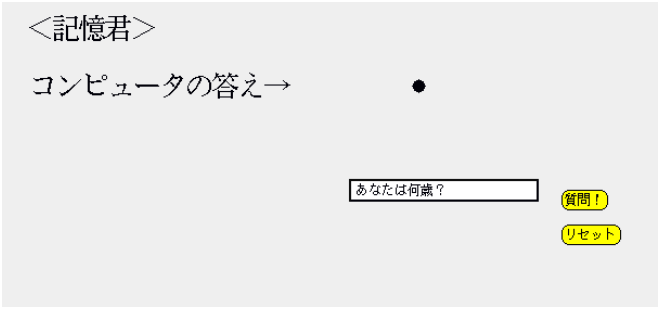


- 答えの番号のボタンを押すと，次の問題が表示される
- 問題の出題が全て終わると，結果（正解数）を表示する

練習問題 11.3

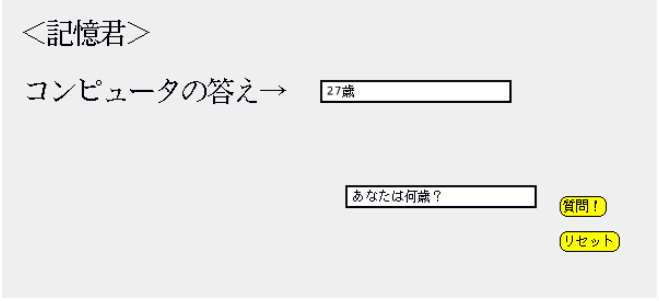
絵辞書を応用して，「記憶君」を作ってみましょう．

・質問を入力し，質問ボタンを押す



・既にその質問の答えが登録されていれば，その答えを表示する
・新しい質問なら，解答を入力するための入力ボックスを表示する

・入力ボックスに解答を入力して，教えるボタンを押すと，質問と解答のペアが登録される（使えば使うほどコンピュータが賢くなる）

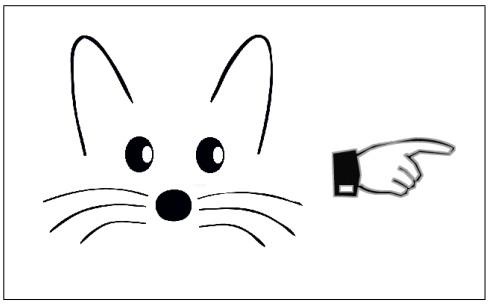


第Ⅴ部

付録

付録 A

Squeak テクニック集

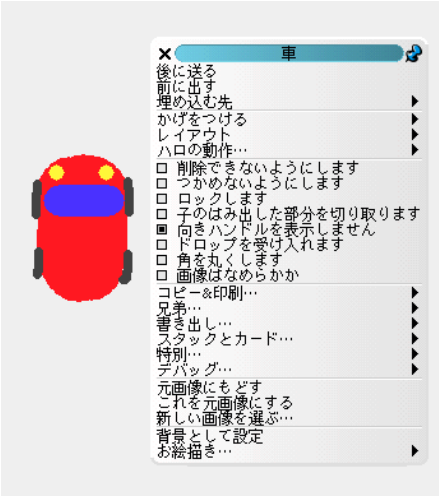


付録 A では，本文で紹介できなかった Squeak の発展的なテクニックをトピックごとに紹介します．本編の「やってみよう！」や「考えてみよう！」などで，関連するトピックへの参照が指示されているので，取り組むときに適宜参照するとよいでしょう．また，このテクニック集を一覧すると，「ことだま on Squeak」にどんな機能があるか概観できます．作品作りを始める前に，一読してみることをオススメします．

A.1　ハロ

A.1.1　赤ハロ

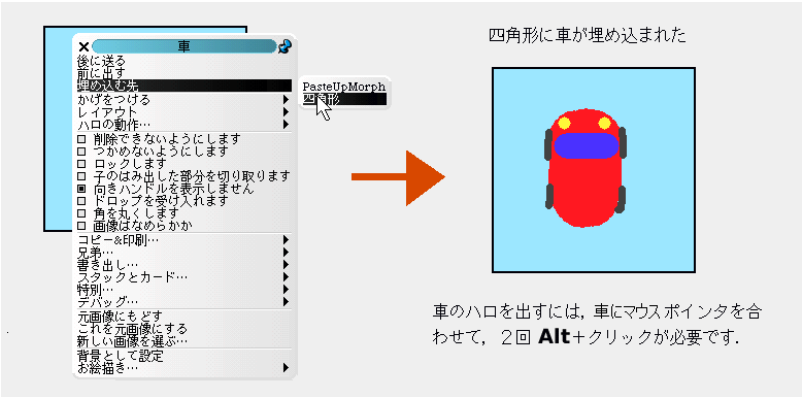
オブジェクトに関する細かい設定や操作をすることができます．



「前に出す」,「後に送る」を使うと複数のオブジェクトが重なっている場合の重なりの順番を変更することができます．

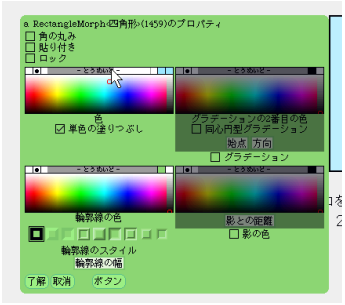
次に説明する，埋め込みを使って一体化させたオブジェクトの重なり順を調整するときにも使えます．

あるオブジェクトを他のオブジェクトに埋め込み，一体化させたい場合は，「埋め込む先」をクリックして，埋め込みたいオブジェクトを選択します．埋め込まれたオブジェクトのハロを表示させるためには，複数回の Alt（Mac の場合は Apple キー）+ クリックが必要です．

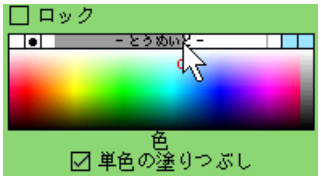


A.1.2　紫ハロ

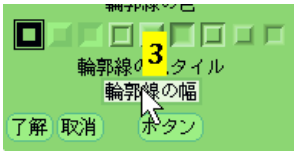
オブジェクトの色を設定することができます．表示されたパレットから設定したい色を選択します．



パレットの上部にあるバーをドラッグすると，オブジェクトの透明度を設定することができます．



輪郭線の幅を変更したいときは，「輪郭線の幅」と書かれている部分をドラッグします．



A.1.3 ○ハロ

オブジェクトをたたむことができます．自分で絵を描いたオブジェクトは画面左上に縮小表示されます．戻すときは縮小表示されているオブジェクトをクリックします．

部品フラップに格納されているオブジェクトはバーのような状態になります．戻す場合はバーの右にある○ボタンをクリックします．

A.1.4 茶ハロと黒ハロ

オブジェクトを移動させるためのハロです．

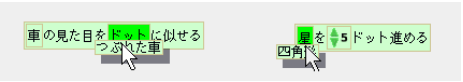
- 茶ハロはオブジェクトを持ち上げずに移動します
- 黒ハロはオブジェクトを一旦持ち上げて移動します

赤ハロを使って埋め込まれたオブジェクトを取り出すときには，黒ハロを使いましょう．

A.1.5　四角ハロ

四角ハロをクリックすると，そのオブジェクトを表すタイルを取得することができます．

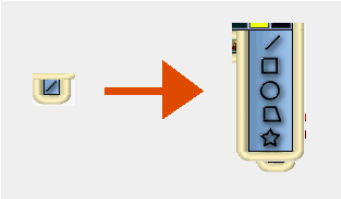
四角ハロで作ったタイルは，「ドット」と書いてある部分や，オブジェクトの名前が書いてある部分に入れることができます．



A.2　お絵かきツール

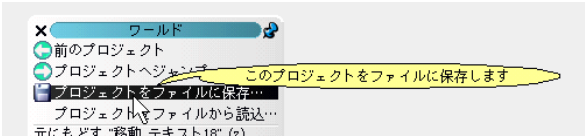
お絵かきツールの使い方は以下の図を参考にしてください．

お絵かきツールの下にある，直線ボタンをクリックすると，様々な図形を簡単に描くことができます．



A.3　プロジェクト単位の保存

ワールドメニューの「プロジェクトをファイルに保存」をクリックすると，現在表示しているプロジェクトだけを保存することができます．

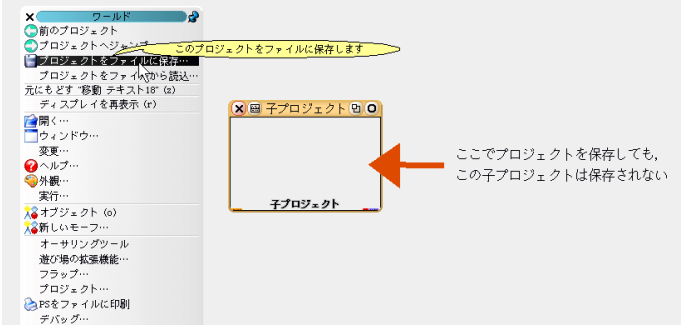


保存の前に，データを保存する場所を確認するウィンドウが表示されます．

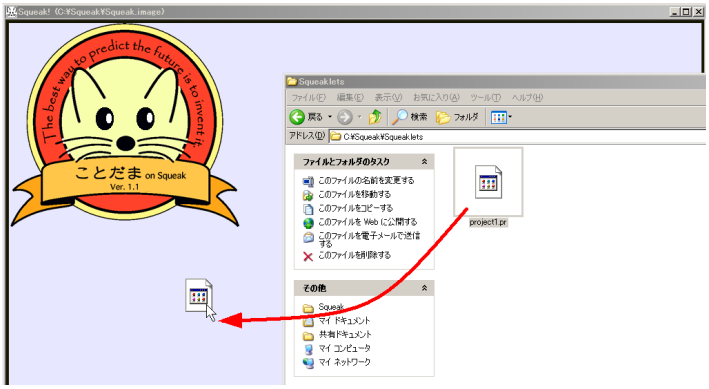
「保存」をクリックすると，ファイルは Squeak フォルダの「Squeaklets」というフォルダと，ウィンドウの下部に表示されているフォルダに保存されます．「手元のディスクにのみ保存」をクリックすると，「Squeaklets」フォルダのみに保存されます．

ファイル名は「プロジェクト名.pr」です．

この方法では，現在表示しているプロジェクトだけが保存できます．プロジェクトが入れ子になっている場合，子プロジェクトは保存されないことに注意してください．

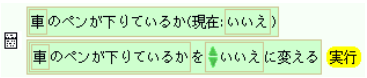


プロジェクトを読み込みたい場合は，Squeak の画面に読み込みたいプロジェクトのファイルをドラッグ＆ドロップしましょう．



A.4　軌跡の描画

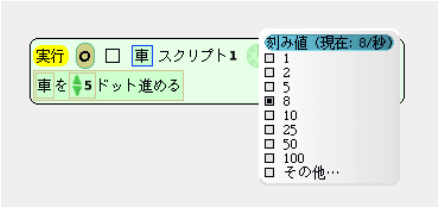
全てのオブジェクトはペンを持っていて，オブジェクトが移動した軌跡を描画することができます．「ペン」カテゴリの「ペンが下りているか」を「はい」にすると軌跡が描画されます．



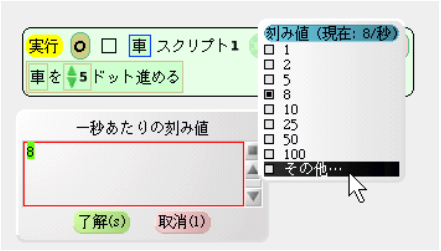
ペンの軌跡を消すには「ペン軌跡を全て消す」を実行します．軌跡を描いたオブジェクトに関係なく，全てのペンの軌跡が消されることに注意してください．

A.5　刻み値

スクリプトが 1 秒間に繰り返す回数のことです．刻み値を変更するには，スクリプトの時計のマークを長くクリックします．刻み値の初期値は毎秒 8 回です．



メニューに表示される以外の刻み値に設定したい場合は，メニューの「その他」をクリックし，設定したい刻み値を入力します．

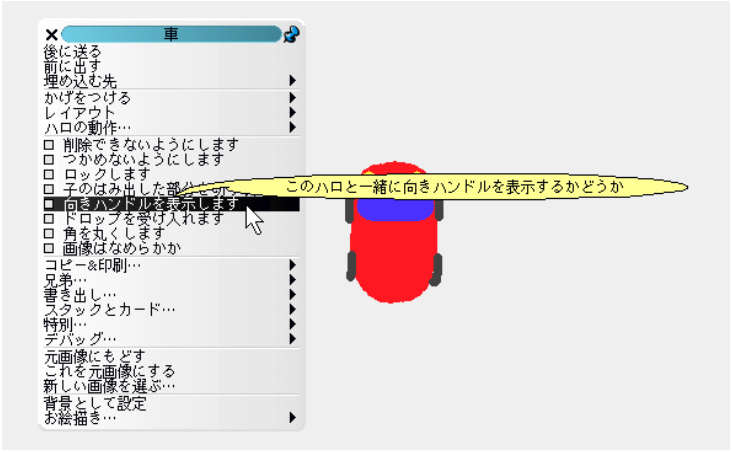


A.6　方向と重心

オブジェクトには「進む」を実行したときに進む方向と，「回す」を実行したときの軸となる重心があります．

ハロを表示したときに一緒に現れる緑の矢印が向きです，青い丸が重心です．方向の矢印はドラッグで，重心の丸はシフト＋ドラッグで変更することができます．

ハロを表示しても方向と重心が表示されない場合，赤ハロのメニューから「向きハンドルを表示します」をクリックしてください．



A.7　全スクリプト

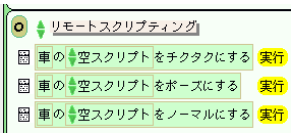
全スクリプトを使うと，全てのオブジェクトの全てのスクリプトを繰り返し実行することができます．部品フラップから取り出せます．

Go をクリックすると，全てのオブジェクトの全てのスクリプトを繰り返し実行します．
Step をクリックすると，全てのオブジェクトの全てのスクリプトを 1 回実行します．
Stop をクリックすると，全てのオブジェクトの全てのスクリプトを停止できます．

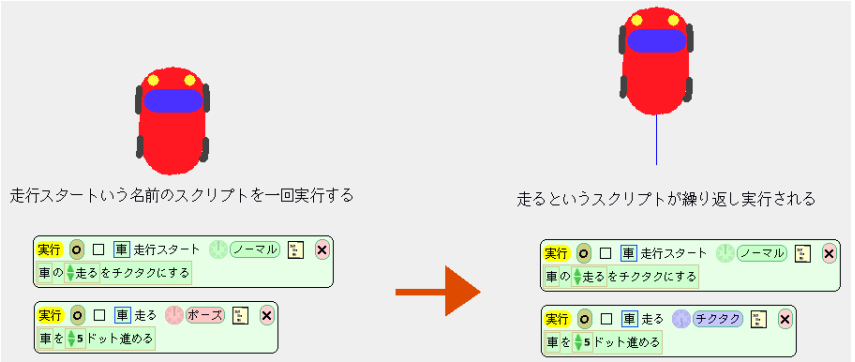
Go と Step をクリックしても，スクリプトの状態が「マウスダウン（A.10 節を参照)」や「ノーマル」のものは実行されません．

A.8　リモートスクリプティング

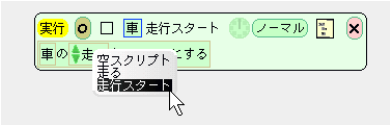
あるスクリプトを（他のスクリプトから）繰り返し実行したり，ストップしたりするための命令タイルがあります．
「リモートスクリプティング」カテゴリの「チクタクにする」,「ポーズにする」という命令タイルを使いましょう．



「走行スタート」のスクリプトを 1 回実行すると，「走る」という名前のスクリプトを繰り返し実行するプログラムは，以下のようになります．



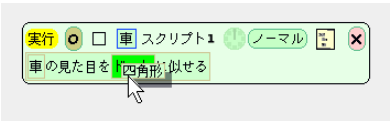
「(空スクリプト)」の部分をチクタク・ポーズにしたいスクリプトの名前へ変更するのを忘れないようにしてください．



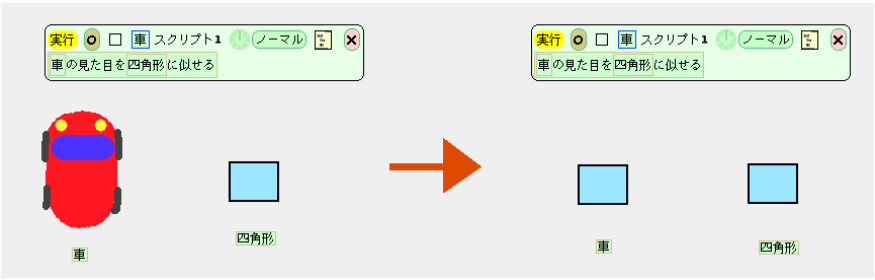
繰り返し実行したいスクリプトがあるオブジェクトのビューアーからタイルを取るようにしましょう．

A.9　見た目を似せる

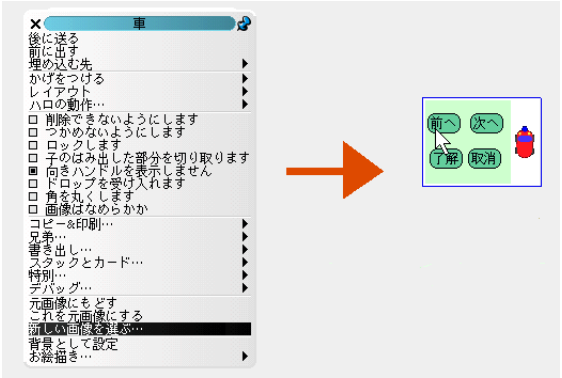
オブジェクトの見た目だけを他のオブジェクトに変更することができます．
「表示」カテゴリの「見た目を似せる」タイルを使います．見た目を似せたいオブジェクトの四角ハロから，オブジェクトを表現したタイルを取ってきて，ドットと描いてある部分に入れましょう．



オブジェクトが入れ替わるわけではなく，見た目だけを変更します．スクリプトは入れ替わりません．



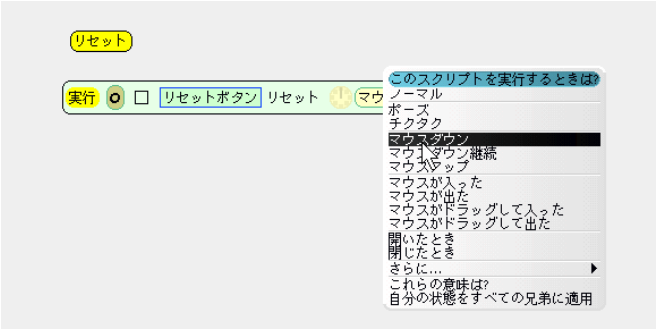
間違えて，見た目を似せてしまって，戻したい見た目のオブジェクトがない場合は，赤ハロの「新しい画像を選ぶ」をクリックし，元に戻したい画像を選択します．



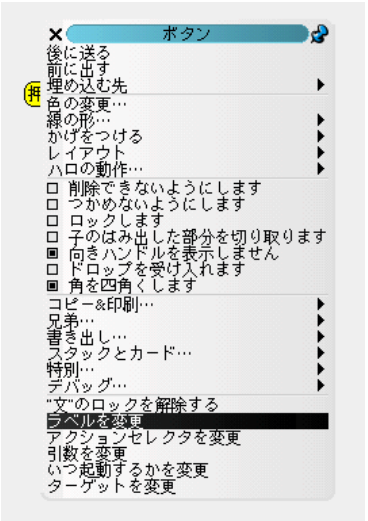
A.10　ボタン

ボタンを簡単に作ることができます．部品フラップから取り出せます．

ボタンはクリックしてもつかめないような設定になっているので，移動は黒・茶ハロで行います．ボタンにスクリプトを作り，時計の右隣の楕円形の部分をクリックすると，スクリプトが実行されるタイミングを選択することができます．「マウスダウン」を選択すると，そのスクリプトはボタンがクリックされたときに一度だけ実行されます．



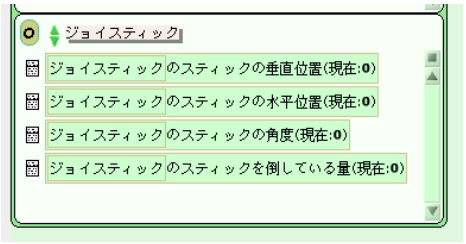
ボタンのラベルはボタンにつけた名前に応じて変わります．「スタートボタン」と名前をつけると，ラベルには「スタート」と表示されます．ボタンの名前とは無関係にボタンのラベルを変更したいときは，赤ハロの「ラベルを変更」をクリックします．



A.11　ジョイスティック

ゲーム等の操作用に便利な部品です．部品フラップから取り出せます．

「ジョイスティック」カテゴリのタイルによって，スティックを倒した方向とその量を取得することができます．



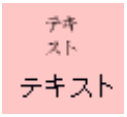
A.12　スライダー

ゲーム等の操作用に便利な部品です．部品フラップから取り出せます．

「スライダー」カテゴリのタイルによって，スライダーのノブに連動したスライダーの数値（0 から 1 まで）を取得することができます．

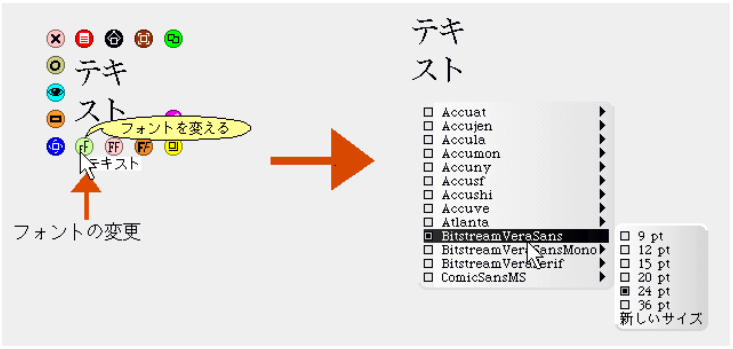
A.13　テキスト

テキストを作ることができます．部品フラップから取り出せます．



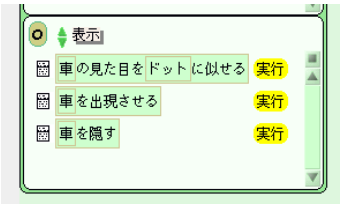
テキストの文字列を操作したい場合は，「テキスト」カテゴリを使います．

フォントや文字の大きさ，書式を変更したいときには，オブジェクトの下に表示される 3 つのハロを使います．



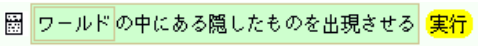
A.14　出現させる・隠す

表示カテゴリにある「出現させる」「隠す」タイルを使うと，オブジェクトを一時的に隠したり，出現させたりすることができます．ゲーム製作等に便利です．



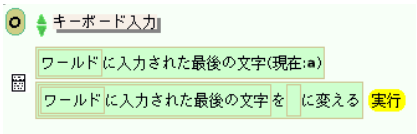
「隠す」を実行すると，オブジェクトが透明になります．「出現させる」を実行すると，透明なオブジェクトを再度出現させることができます．

隠したオブジェクトを表示できなくなってしまった場合は，ワールドのビューアーを開き，「遊び場」カテゴリの「隠したものを出現させる」命令を実行します．

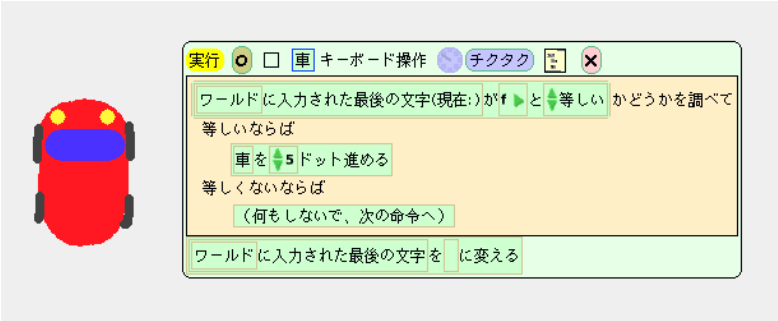


A.15　キーボード入力の受け取り

キーボードの入力を受け取ることができます．ワールドのビューアーの「キーボード入力」カテゴリにあるタイルを使います．



最後に入力されていたキーしか受け取ることができませんので，押しただけ車を進めたい場合は，常に最後に入力されていたキーをリセットし続ける必要があります．

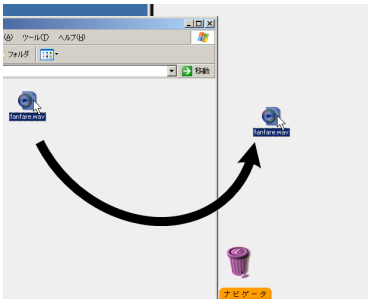


エンターキーは <cr>，矢印キーは <up><down><right><left> といった，半角の < と > で囲んで指定します．

A.16　音楽ファイルの取り込み

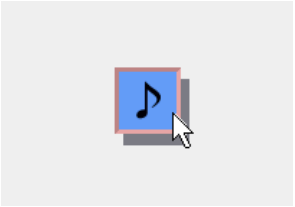
wav ファイル（Windows 標準の音声ファイル）を Squeak に取り込んで，再生や停止といった作業をスクリプトで実行することができます．ゲームに効果音や BGM をつけたいときに便利です．

1. wav ファイル[*1]を Squeak の画面にドラッグ＆ドロップします．

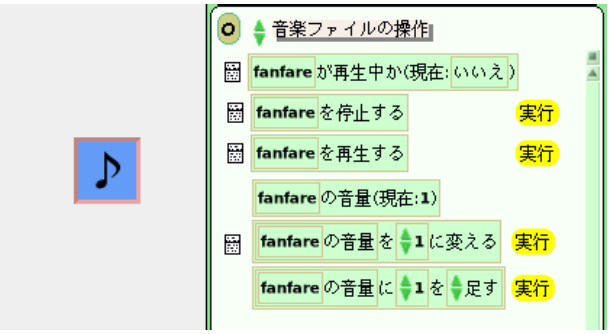


[*1] wav ファイルがあるフォルダのパスに日本語がないか確認してください．また，wav ファイルのファイル名は半角文字でつけてください．

2. Squeak の画面に♪の描かれたオブジェクトが出現します．



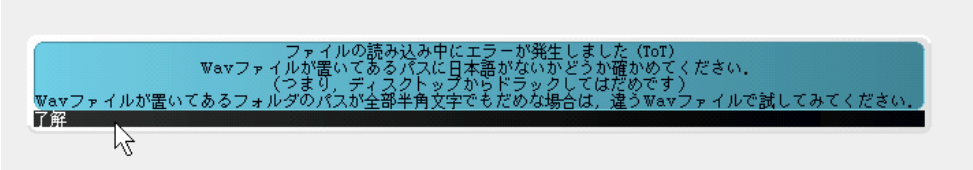
3. ♪の描かれたオブジェクトのビューアを開き，「音楽ファイルの操作」というカテゴリにあるタイルを使ってプログラムを組みます．



- 「再生中か」…音楽が再生中かどうかを調べることができます．
- 「停止する」…音楽の再生を停止します．
- 「再生する」…音楽を再生します（連続して再生したいときは，このタイルを入れたスクリプトをチクタクにしましょう）．
- 「音量」…音楽を再生する音量を調整できます．

wav ファイルを Squeak の画面にドラッグ＆ドロップしたときに，エラーメッセージが表示される場合は，了解を選択し，以下の事柄を確認してください．

- wav ファイルのファイル名は半角文字か？
- wav ファイルがあるフォルダのパスに全角（日本語）がないか？
- wav ファイルが壊れていないか？

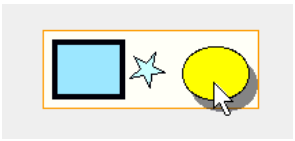


A.17　入れ物

A.17.1　マウスによるオブジェクトの格納・取り出し

入れ物には，複数のオブジェクトを格納することができます．入れ物に格納したいオブジェクトをマウスでつかみ，入れ物の上に重ねると，そのオブジェクトが入れ物に格納されます．

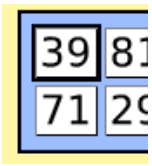
入れ物からオブジェクトを取り出したい場合は，取り出したいオブジェクトをマウスでつかみ，入れ物の外に移動すれば取り出すことができます．



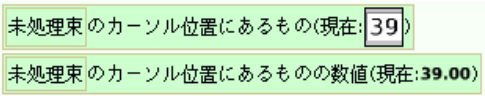
マウスでつかめないオブジェクト（テキストなど）を入れ物に格納したり，取り出したりする場合は，黒ハロを使います．

A.17.2　カーソル

入れ物は「カーソル」を 1 つだけ持っていて，「カーソル」が当たっているオブジェクトには太い枠がつきます．以下の場合，39 と書かれたオブジェクトにカーソルが当たっています．



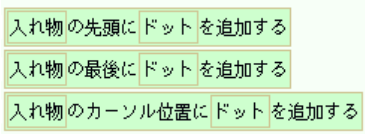
以下のタイルを使うと，カーソルの位置にあるオブジェクトを参照することができます．また，カーソルにあるオブジェクトがテキストの場合，カーソル位置にあるオブジェクトの数値を調べることができます．



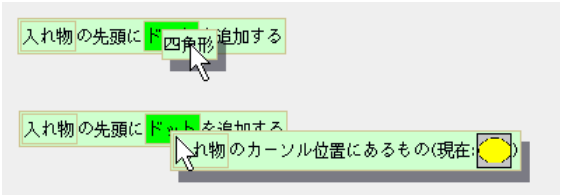
カーソルの位置は移動することができます．カーソルの位置の移動に関係する命令タイルは以下の 3 つです．先頭のカーソル位置は「1」です．入れ物に入っているオブジェクトの個数より，カーソル位置を大きくしようとするとカーソルは自動的に先頭に戻ります．

A.17.3　タイルによるオブジェクトの格納と移動

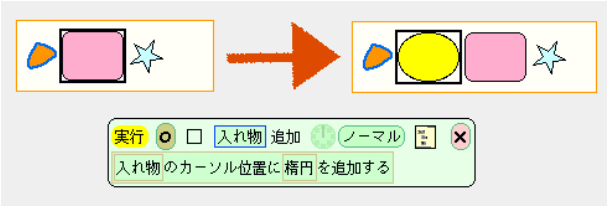
入れ物の先頭，最後，カーソル位置にオブジェクトを追加することができます．以下のタイルが利用できます．



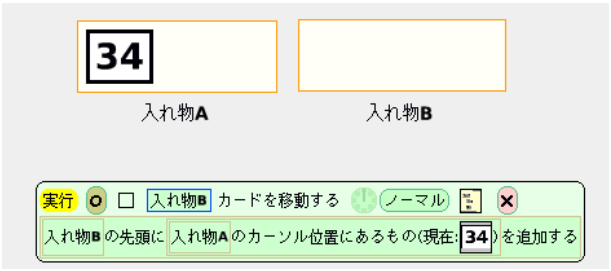
追加のためのタイルの「ドット」(仮のオブジェクトの名前）の部分には，四角ハロで取り出したオブジェクトの名前のタイルや，「入れ物のカーソル位置にあるもの」のタイルを入れることができます．



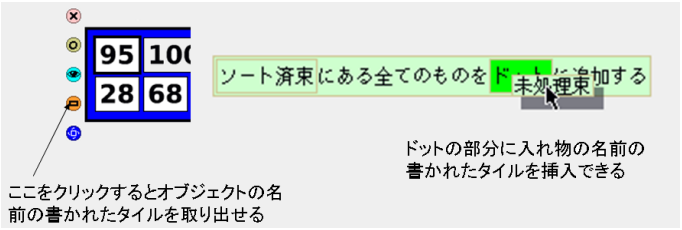
カーソル位置にオブジェクトを追加すると，カーソル位置にあったオブジェクトは新しく追加したオブジェクトの 1 つ後方にシフトされます．以下の図では，楕円のオブジェクトをカーソル位置に追加すると，今までカーソルの位置にあった角丸四角形が後方にずれる様子を示しています．



ある入れ物に入っているオブジェクトを他の入れ物に追加すると，そのオブジェクトは移動します．例えば，以下のプログラムを実行すると，34 と書かれたオブジェクトは入れ物 B に移動します．

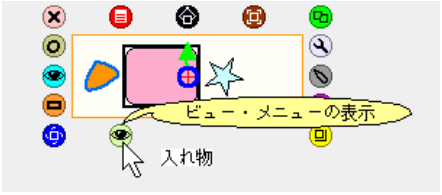


ある入れ物に入っているオブジェクトを他の入れ物に一度に移動させることができます．四角ハロを使うと，入れ物の「名前のタイル」を取得することができます．取得した「名前のタイル」をドットの部分に挿入することができます．



A.17.4　見た目や動作の設定

入れ物は見た目や動作に関する様々な設定ができます．入れ物のハロを表示させ，薄緑色のハロ（目玉のマーク）をクリックすると，設定のためのメニューが表示できます．



A.18　カテゴリレベルの変更

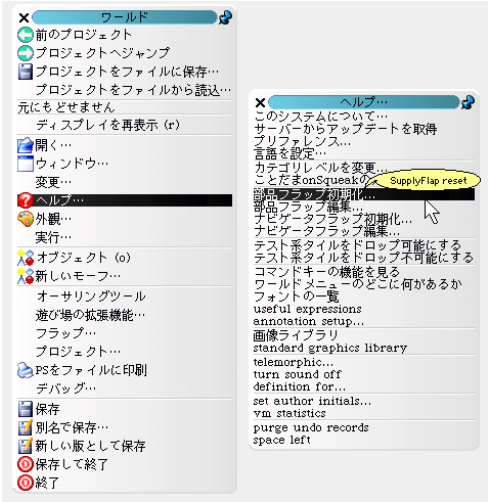
ビューアーに用意されているタイルの種類を変更することができます．ワールドメニューの「ヘルプ」を選択し，「カテゴリレベルの変更」を選択します．

それぞれの設定基準に関しては，メニューの解説を見てください．

A.19　フラップの初期化

「ナビゲータフラップ」や「部品フラップ」を誤って壊してしまったり，削除してしまった場合，フラップを元に戻すことができます．

ワールドメニューの「ヘルプ」を選択し，「部品フラップ初期化」か「ナビゲータフラップ初期化」を選択します．



付録 B

Mac で使うには

ここでは，Mac OS X へ Squeak をインストールする方法と，Squeak を起動・終了する方法を解説します．本編の Project 0 は，Windows へのインストール方法を解説しています．Mac で Squeak を使う場合，Project 0 の代わりにこの付録を参考にしてください．

ここで解説されている作業が終わったら，本編の Project 1 へ進みましょう．

キーワード

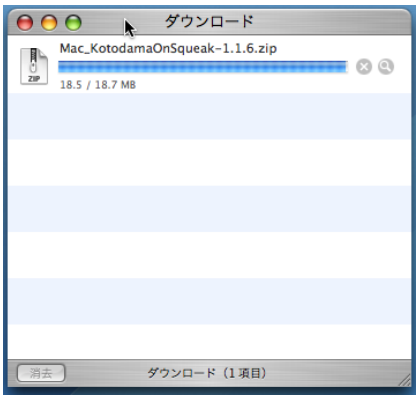
インストール，起動 / 終了，image ファイル，changes ファイル

B.1　インストール

Mac 版「ことだま on Squeak」の“インストール”方法を説明します．

B.1.1　ダウンロードする

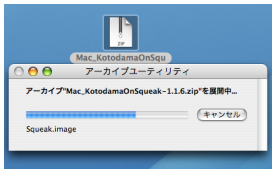
「ことだま on Squeak」の Mac 版をダウンロードします．
URL は http://www.crew.sfc.keio.ac.jp/squeak です．ファイルは zip 形式で圧縮されています[*1]．



[*1] Mac OS X 付属の Safari でダウンロードした場合，デスクトップに自動的に解凍されることがあります．その場合，解凍の作業は不要です．

B.1.2　解凍する

ダウンロードしたファイルを解凍してください．Mac OS X の場合，解凍したいファイルをダブルクリックするとデスクトップに解凍されます．



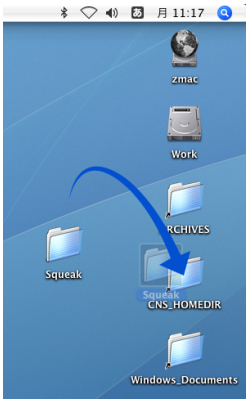
B.1.3　配置する

解凍してできたフォルダの名前を「Squeak」に変更します．テキストの本編ではこのフォルダのことを「Squeak フォルダ」と呼んでいます．

デスクトップ上に Squeak フォルダを置いたままでもとくに支障はありませんが，デスクトップを整理するため，アプリケーションフォルダ等に移動するとよいでしょう[*2]．

慶應義塾大学 SFC の特別教室で利用する場合

Squeak フォルダは CNS_HOMEDIR フォルダに移動しておきましょう．
デスクトップにある「CNS_HOMEDIR」フォルダに，Squeak フォルダをドラッグ＆ドロップしてください．



[*2] アプリケーションフォルダ以外に Squeak フォルダを配置する際には，Squeak フォルダまでのパスに全角文字や空白がない場所に配置するようにしてください．

B.2　起動と終了

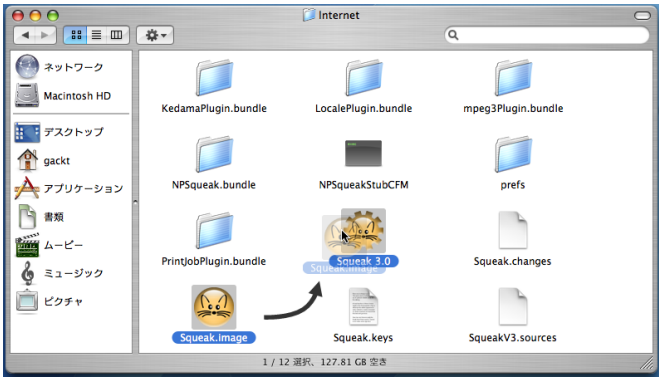
Mac 版「ことだま on Squeak」の“起動”と“終了”について説明します．

B.2.1　起動する

Squeak フォルダをダブルクリックすると，Internet と Plugins という 2 つのフォルダがあります．

Internet フォルダの中に Squeak の本体があります．

Internet フォルダをダブルクリックし，内容を表示させます．Squeak を起動するためには，Squeak.image を Squeak 3.0 という名前のファイルにドラッグ＆ドロップします[*3]．



以下のようなウィンドウが表示されれば，起動は成功です．

[*3] Squeak という名前のファイルをダブルクリックすることでも起動はできますが，Squeak.image ファイルを読み込んで起動します．Squeak.image ファイル以外のファイルを読み込ませたい場合は，読み込ませたい image ファイルを Squeak にドラッグ＆ドロップする必要があります．

B.2.2　終了する

Squeak を終了する場合は，Squeak の画面の左下にある「ナビゲータ」と書いてある部分をクリックします．

ポップアップしたナビゲータの一番右にある終了ボタンを押します．

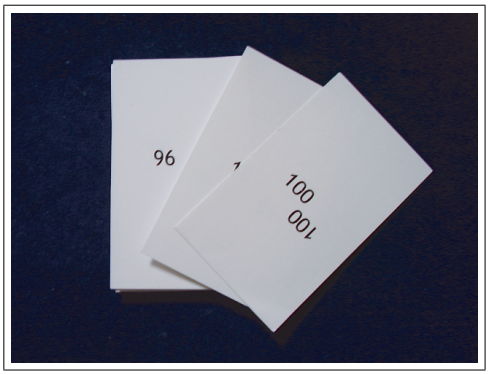
終了ボタンを押すと，メッセージの書かれたメニューが表示されます．「はい」を選択すると Squeak の修正を Squeak.image に保存しないまま終了します．変更を保存する方法は本編の P.18 を参照してください．

Squeak フォルダにある Squeak.changes はプログラムの修正履歴を保存しておくためのファイルです．“changes ファイル”は同じ名前の“image ファイル”とセットで作成されます．誰かに自分の Squeak のデータを渡したいときは image ファイルと changes ファイルの両方を渡すようにしてください．

Mac 用の Squeak で作成したデータを Windows の Squeak で読み込むことも可能です．Squeak.image と Squeak.changes は Mac 用や Windows 用といった区別がなく，共通に利用できます．

付録 C

最小値選択法 ワークシート

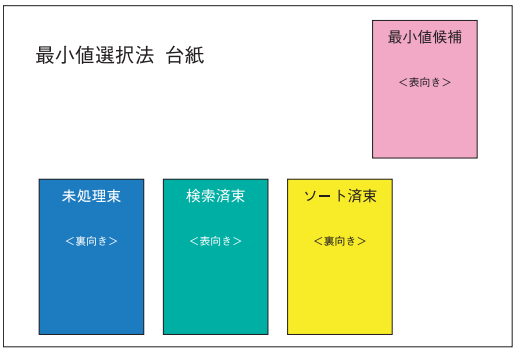


このワークシートは，Project 10 の 10.2 節で紹介した，手作業による並び替えの演習を行うためのものです．この演習は 2 人組のペアを組んで行います．Step1 から Step5 までを順番に実施してください．台紙のファイルやグラフ作成用 Excel ファイルは http://www.crew.sfc.keio.ac.jp/squeak/sort からダウンロードできます．

C.1 Step1. カードの準備

ランダムな数字が書かれたカードを 40 枚用意します．数字が重複していても構いません．

最小値選択法用の台紙[*1]を机の上に敷きます．



40 枚のカードを良くきり，きったカードの束を裏返して，未処理束というところに置きます．次に，未処理束の一番上のカードを 1 枚だけめくって，最小値候補と書いてあるところに表向きに置きます．

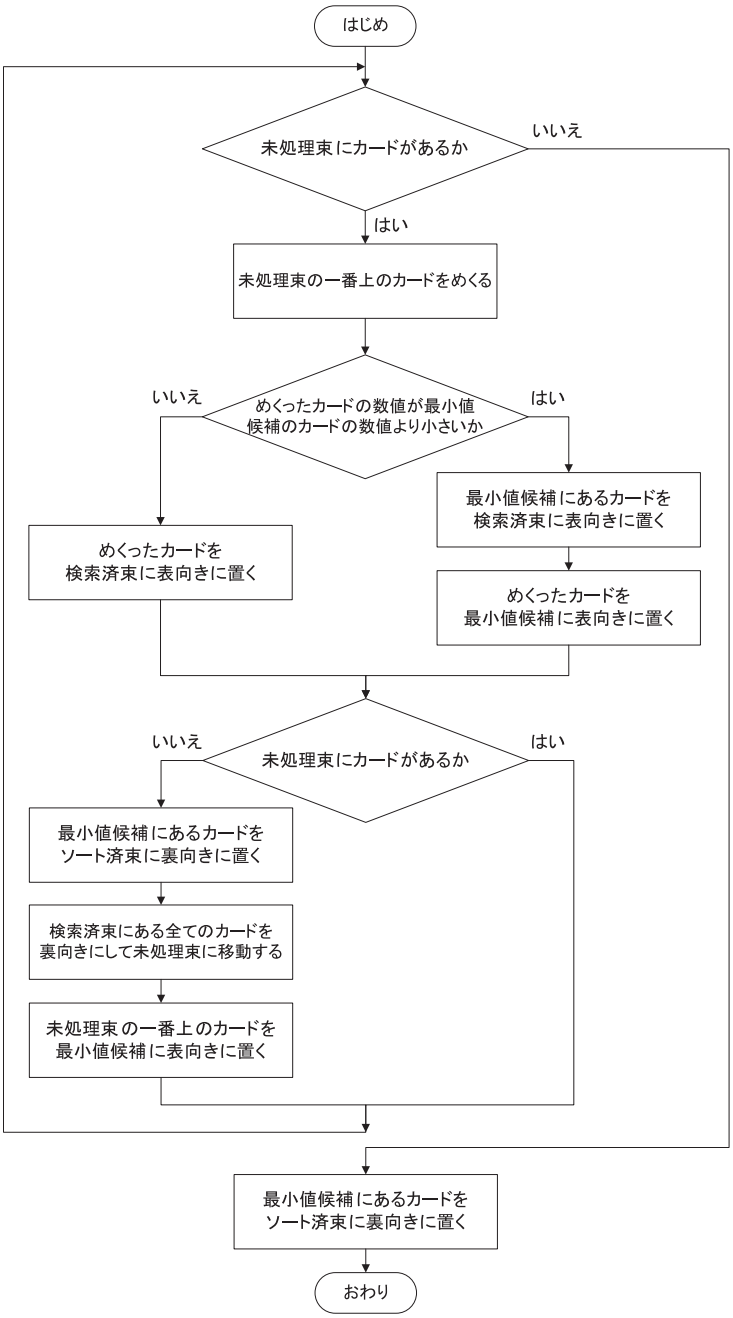
- **未処理束**…並び替えが終了していないカードを置く場所．並び替えを始める前に，全てのカードをよくシャッフルして置く．
- **検索済束**…最小値候補との比較が終わったカードを置く場所．
- **最小値候補**…未処理束の中から一番小さな数字の書かれたカードを選ぶために使う場所．1 枚カードが置けるスペースがあればよい．
- **ソート済束**…並び替えが終わったカードを置く場所．最終的にここに昇順にカードが並ぶ．

[*1] 台紙は http://www.crew.sfc.keio.ac.jp/squeak/sort からダウンロードできます．この付録 C の最後（P.175）にあるものを A3 に拡大コピーして使うこともできます．

C.2　Step2. 並び替えの手順の理解

フローチャートをよく読み，並び替えの手順を理解します．最初は 4 枚程度のカードを使って一通り並び替えをしてみましょう．

カードを並び替えている映像が http://www.crew.sfc.keio.ac.jp/squeak/sort で閲覧することができます．



C.3　Step3. 並び替えと計測

並び替えの作業をする人とタイムを計測する人を決めます．

並び替え係の氏名：＿＿＿＿＿＿＿＿＿＿＿＿

計測係の氏名：＿＿＿＿＿＿＿＿＿＿＿＿

カードが１枚ソート済束に移動したら，作業を始めてからの経過時間を記録[*2]します．

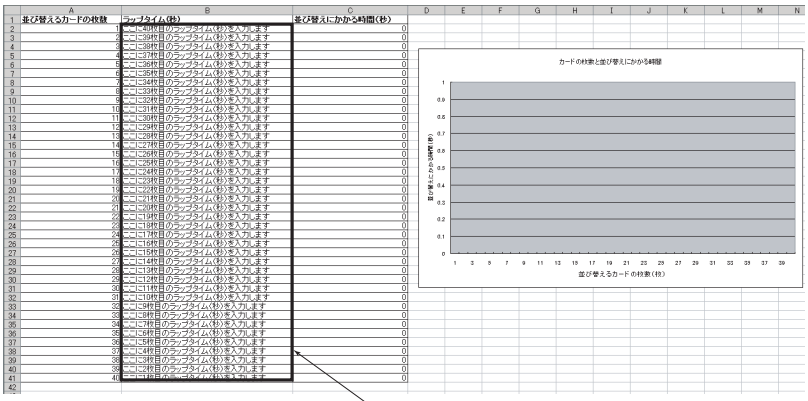
ラップタイムは，１枚のカードを並び替えるのにかかった時間です．経過時間を測定しておけば，ラップタイムは後から計算で求めることができます．

枚目	経過時間	ラップタイム	枚目	経過時間	ラップタイム
1	分　秒	（経過時間と同じ）	21	分　秒	分　秒
2	分　秒	分　秒	22	分　秒	分　秒
3	分　秒	分　秒	23	分　秒	分　秒
4	分　秒	分　秒	24	分　秒	分　秒
5	分　秒	分　秒	25	分　秒	分　秒
6	分　秒	分　秒	26	分　秒	分　秒
7	分　秒	分　秒	27	分　秒	分　秒
8	分　秒	分　秒	28	分　秒	分　秒
9	分　秒	分　秒	29	分　秒	分　秒
10	分　秒	分　秒	30	分　秒	分　秒
11	分　秒	分　秒	31	分　秒	分　秒
12	分　秒	分　秒	32	分　秒	分　秒
13	分　秒	分　秒	33	分　秒	分　秒
14	分　秒	分　秒	34	分　秒	分　秒
15	分　秒	分　秒	35	分　秒	分　秒
16	分　秒	分　秒	36	分　秒	分　秒
17	分　秒	分　秒	37	分　秒	分　秒
18	分　秒	分　秒	38	分　秒	分　秒
19	分　秒	分　秒	39	分　秒	分　秒
20	分　秒	分　秒	40	分　秒	分　秒

[*2] Project 10 の 10.2 節に Squeak を使ったストップウォッチの作り方が掲載されています．

C.4　Step4. グラフ作成

Step3. の計測結果をもとに，並び替えに必要な時間を縦軸，並び替えるカードの枚数を横軸にとったグラフを作成しましょう．Microsoft Excel を使うと簡単にグラフが作成できます．グラフ作成に利用する Excel のテンプレートが http://www.crew.sfc.keio.ac.jp/squeak/sort からダウンロードできます．



Excel のテンプレートには，「並び替えるカードの枚数」，「ラップタイム（秒）」，「並び替えにかかる時間（秒）」という列が用意してあります．

1 枚のカードを並び替えるのに必要な時間は，40 枚目のラップタイムに相当します．2 枚のカードを並び替えるのに必要な時間は，40 枚目と 39 枚目のラップタイムを足した時間です．したがって，ラップタイムの欄には，Step3. の記録を 40 枚目（表の最後から）順番に入力します．

ラップタイムを入力すると，「並び替えにかかる時間（秒）」が自動で計算され，右側のグラフに表示されます．

C.5　Step5. 考察

Step4. の結果から，

- 20 枚のカードを並び替えるのに，10 枚のカードの並び替えの何倍時間が必要かを調べてみよう
- 40 枚のカードを並び替えるのに，20 枚のカードの並び替えの何倍時間が必要かを調べてみよう

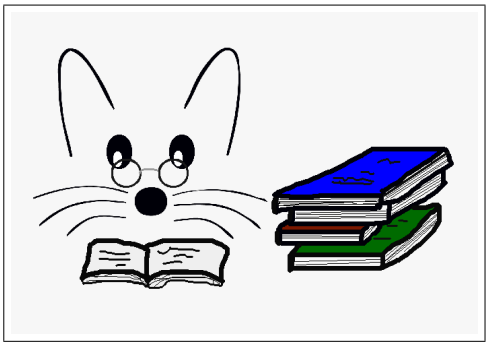
次に，以下の予測問題に答えてみよう．

- 80 枚のカード（実験の 2 倍）を並び替えるには，約 _____ 分かかるだろう
- 160 枚のカード（実験の 4 倍）を並び替えるには，約 _____ 分かかるだろう
- 320 枚のカード（実験の 8 倍）を並び替えるには，約 _____ 分かかるだろう
- 1 日（24 時間）かけると，約 _____ 枚のカードを並び替えることができるだろう
- 1 週間（168 時間）かけると，約 _____ 枚のカードを並び替えることができるだろう

予測問題の結果を一般化し，並び替えに必要な時間を t，並び替えるカードの枚数を N としたとき，t と N の関係を示す式を求めてみよう．

最小値選択法　台紙

参考文献一覧



- Powerful Ideas in the Classroom
  BJ Allen-Conn, Kim Rose 著
  Viewpoints Research Institute，ISBN 0-9743131-0-6，2003

- スクイークであそぼう
  Thoru Yamamoto 著，阿部 和広 監修
  翔泳社，ISBN 4-7981-0480-9，2003

- 実践スクイーク教室
  斉藤 礼美 著
  アカデミア，ISBN 4-902524-00-7，2003

- Squeak 入門 – 過去から来た未来のプログラミング環境
  Mark Guzdial, Kim Rose 著，軋音組 訳
  エスアイビー・アクセス，ISBN 4-434-02947-9，2003

- Squeak プログラミング入門 – オブジェクトランドへの招待
  G.Korienek, T.Wrensch, D.Dechow 著，菅原 一孔，鈴木 元 訳，阿部 和広 監修
  エスアイビー・アクセス，ISBN 4-434-04330-7，2004

- スクイークランド
  http://squeakland.jp/

- アラン・ケイ
  アラン・C・ケイ 著，鶴岡 雄二 訳，浜野 保樹 監修
  アスキー，ISBN 4-7561-0107-0，1992

- マインドストーム　子供，コンピューター，そして強力なアイデア
  S. パパート 著，奥村 喜世子 訳
  未来社，1982

- 自由自在 Squeak プログラミング
  梅澤 真史 著
  ソフト・リサーチ・センター，ISBN 4-88373-203-7，2004

- 先生とパソコン（特集 第 2 部「体験！ Squeak を使ってみよう」）
  阿部 和広，平井 夏児，吉正 健太郎 著
  技術評論社，ISBN 4-7741-2092-8，2004

索引

【あ】

青ハロ 11
赤ハロ 149
遊び場 60
値 56
新しい画像を選ぶ 157
後に送る 149
アプリケーションフォルダ 168
アルゴリズム 117
Etoys 21
image ファイル 5, 6, 169, 170
入れ子 49, 82
入れ物 123, 163
インストール 3, 167
Internet フォルダ 169
wav ファイル 161
埋め込み 150
絵辞書 133
お絵かきツール 14, 152
オブジェクト 10
オブジェクトのカタログ 112
音楽ファイル 161
音楽ファイルの操作 162

【か】

カーソル 128, 135, 163
カーソルの位置 164
解凍 4
描き直しハロ 16
隠す 160
加速度 93
カテゴリ 26
カテゴリレベルの変更 166
空スクリプト 24, 157
キーボード入力の受け取り 161
企画 100
企画書 100
刻み値 154
軌跡を描画 154
起動 5, 169
黄ハロ 11
組み合わせ 28
繰り返し 25
黒ハロ 151
計算 81
計算式タイル 82
ゲームの流れ図 102
検索 133
Go 156
コピーして，コピーしたもの 126
子プロジェクト 153
ゴミ箱 12

【さ】

最小値選択法 118
最小値選択法セット 123
座標 56
四角ハロ 152
実行ボタン 23
実装 100, 109
シミュレーション 93
重心 29, 155
終了 6, 169
縮小表示 151
出現させる 160
順次実行 30
ジョイスティック 159
仕様 100, 102
昇順 117
仕様書 102
書式 160
Squeak 3
Squeak フォルダ 4
Squeaklets 153
スクリプト 24
スケジューリング 107
Step 156
Stop 156
Smalltalk 143
スライダー 159
スライド 111
スレッド・ナビゲータ 112
成果物 100
全画面表示 113
センサー 40
全スクリプト 156
選択法 118
挿入法 131
ソート 119
速度 92

【た】

たたむ 151
changes ファイル 6, 170
チクタクにする 156
茶ハロ 151
直線ボタン 152
テキスト 160
透明度 150
ドット 126, 152, 164

【な】

ナビゲータ 13, 170
並び替え 118

【は】

場合分け 41
発表 111
ハロ 10
番兵 138
.pr 153
ビューアー 22
評価 114
表題を変更 14
フォント 160
部分が色に触れているか 41
フラップ 9
フラップの初期化 166

プレゼンテーション 111
フローチャート 47
プロジェクト 13, 111
プロジェクトをファイルに保存 153
ペン 154
ペンが下りているか 154
ペン軌跡を全て消す 154
変更タイル 57
変数カテゴリ 92
変数タイル 56
方向 25, 155
報告書 114
ポーズにする 156
保存 18
ボタン 158

【ま】

迷子を連れ戻す 60
マイルストーン 107
マウスダウン 158
前に出す 149
Mac OS X 167
Mac 版 167
○ハロ 151
回す 28
水色（目玉）ハロ 22
見た目を似せる 157
緑ハロ 11
向きハンドルを表示します 155
紫ハロ 150
命令タイル 22
桃ハロ 12

【ら】

ラベルを変更 158
乱数 84
乱数タイル 84
リモートスクリプティング 156
輪郭線の幅 150

【わ】

ワールドメニュー 19
私の部品 21

あとがき

このテキストは，千葉商科大学「プログラミング」（担当：松澤），慶應義塾大学「情報技術ワークショップ」（担当：松澤），学習院高等科の「情報 I，II」（担当：杉浦）の講義録が基となり，2005，2006 年 経済産業省 IT クラフトマンシップ・プロジェクトで行なわれる実験授業の教材として作られました．

教材開発協力

川合 尚彦
佐々木 麻里
広瀬 智子
明石 敬
島崎 聡史

ことだま on Squeak 開発協力

藤田 匠

Special Thanks

パイロットテストに参加してくれた皆さん
今まで授業で一緒に学んでくれた皆さん
Viewpoints Research Institute の皆さん
大岩研究室（CreW Project）の皆さん

本テキストの利用実績（2008 年 11 月現在）

慶應義塾大学 SFC 「論理思考とプログラミング」松澤，岡田，杉浦（2007 年度 春学期～）
慶應義塾大学 SFC 「情報技術ワークショップ」杉浦（2006 年度 秋学期～2006 年度 秋学期）
慶應義塾大学 SFC 「情報技術ワークショップ」松澤（2004 年度 秋学期～2005 年度 秋学期）
千葉商科大学 政策情報学部 「プログラミング C（I，II）」松澤（2004 年度 秋学期～2007 年度 秋学期）
千葉商科大学 商経学部 「プログラミング I，II」松澤（2003 年度 秋学期～2007 年度 秋学期）
関西大学 文学部 「プログラミング（旧 アルゴリズムとプログラミング）」本村 康哲 先生（2007 年度～）
獨協大学 「コンピュータ入門」岡田（2006 年度 秋学期～）
帝京平成大学 「コンピュータ演習 I」大岩（2008 年度 春学期）
奈良佐保短期大学 幼児教育科 「教育方法メディア論」中村 恵 先生（2009 年度～）
学校法人 清風明育社 清風情報工科学院 「Squeak プログラミング入門」土井 佳巳 先生（2008 年度～）
学習院高等科 「情報 I，II」杉浦（2005 年度 一学期～）
藤沢市立 村岡中学校「選択授業」北井 淳一 先生（2006 年度 一，二学期）
藤沢市立 村岡中学校「総合の時間」北井 淳一 先生（2005 年度 一学期）

【監修者】

大岩 元

1942 年生まれ　理学博士（東京大学，1971 年）
豊橋技術科学大学 教授，慶應義塾大学 環境情報学部 教授を歴任
2008 年より帝京平成大学 現代ライフ学部 教授
専門は情報教育学，ソフトウェア工学，認知工学

主要著作
情報技術と社会（共著，2005 年，放送大学教育振興会）
みんなの情報 A・B・C（共著，2002 年，オーム社，文部科学省検定済教科書）
情報科教育法（共著，2001 年，オーム社）
二十一世紀 豊かな情報化社会の実現を願って（共著，1999 年，情報処理学会）
ヒューマンインターフェース（共著，1992 年，オーム社）
マイクロエレクトロニクス入門（共著，1984 年，オーム社）
タッチタイプの本（監修，1984 年，エー・アイ・ソフト社発行）

【編著者】

松澤 芳昭

1977 年生まれ　博士（政策・メディア ）慶應義塾大学，2008 年
2002 年 慶應義塾大学大学院 政策・メディア研究科 修士課程 修了
2007 年 慶應義塾大学大学院 政策・メディア研究科 後期博士課程 単位取得退学
静岡大学 情報学部 特任助教，慶應義塾大学 環境情報学部 非常勤講師

杉浦 学

1980 年生まれ
2005 年 慶應義塾大学大学院 政策・メディア研究科 修士課程 修了
慶應義塾大学大学院 政策・メディア研究科 後期博士課程 在学中
学習院高等科 非常勤講師（情報），慶應義塾大学 環境情報学部 非常勤講師

【ことだま on Squeak 開発】

岡田 健

1977 年生まれ
2003 年 慶應義塾大学大学院 政策・メディア研究科 修士課程 修了
慶應義塾大学大学院 政策・メディア研究科 後期博士課程 在学中
慶應義塾大学 環境情報学部 非常勤講師，獨協大学 経済学部 経営学科 非常勤講師

ことだま on Squeak で学ぶ論理思考とプログラミング

2008 年 4 月 8 日　初版 第 1 刷 発行
2008 年 11 月 5 日　　第 2 刷 発行

監修者　大岩 元
編著者　松澤 芳昭，杉浦 学
発行人　原 久太郎
発行所　株式会社 イーテキスト研究所
〒113-0033 東京都文京区本郷 5-1-16
Tel 03-3815-3037　Fax 03-3818-1219
http://www.etext.jp
印刷所　株式会社 加藤文明社

ISBN978-4-904013-01-4

組版 杉浦 学　表紙デザイン 林 健造