

# はじめよう/ ジブン専用パソコン

## 第37回 ビスケットで自動運転車をつくろう

前回、粘土のような不思議なプログラミング言語、「ビスケット (Viscuit)」の基本的な使い方を説明した。車の絵を描いて、「メガネ」で動き方を示したら、車を動かすことができたね。今回はこの車を改造して、障害物にぶつからない自動運転車にしてみよう。実際の自動運転車のしくみもわかるかも？

ジブン専用パソコン特設サイト  
prog.kodomonokagaku.com/jibun/  
index.html

今回の内容は普通のパソコンでも動かせるよ。

## ゲーム 工作 プログラミング ラズベリーパイ Raspberry Pi 大活用



キットの情報は  
KoKa Shop!へ  
定期購読者特典割引あり  
購入ページ



監修・原案／  
青山学院大学大学院 特任教授 阿部和広  
構成・文／塩野祐樹

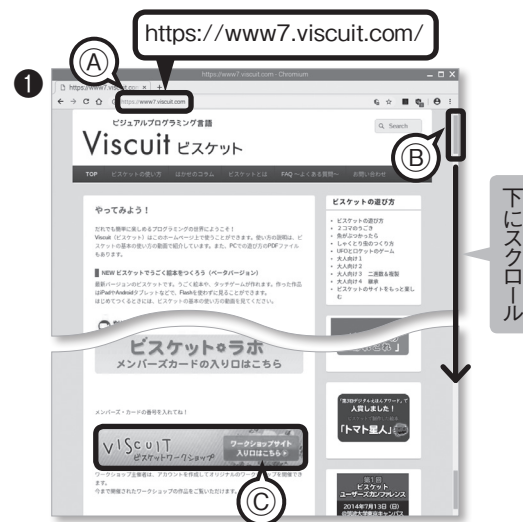
## ビスケットの起動と作品の読み込み

ビスケットはブラウザで動かすんだね。まずはスタートメニューの「インターネット」から

「Chromium ウェブ・ブラウザ」を選ぶか、タスクバーの地球のアイコンをクリックして、クロミウムを起動しよう。

## 作品の読み込みの手順

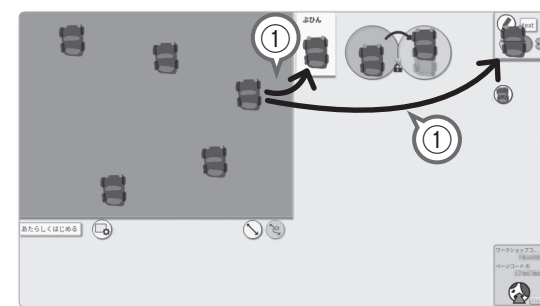
- ① アドレスバーに「https://www7.viscuit.com/」と入力 (A)。ビスケットの旧サイトが開いたら、下にスクロール (B) して、「ワークショップサイト入り口はこちら」 (C) をクリックしよう。
- ② 前回メモしたワークショップコードとページコードを入力して、「直す」 (D) をクリックしよう。
- ③ フラッシュプレイヤーの警告画面が開くので「Get Flash」 (E) をクリックする。許可のダイアログが開いたら、「許可」 (F) をクリックしよう。
- ④ 続いて、フラッシュのバージョンの警告が出るので、「今回は実行する」 (G) をクリックしよう。
- ⑤ 前回保存した作品が開いたね。



起動したら、下の「作品の読み込みの手順」にしたがって、前回保存した作品を開いてね。もし、ワークショップコードとページコードを忘れちゃったら、がんばって前回の内容をもう一度やろう。

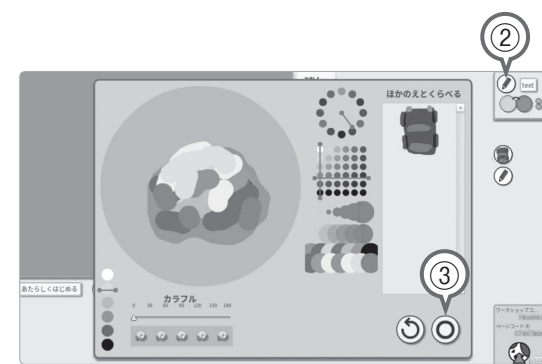
## 障害物を避ける

それでは、前回つくった車が障害物を避けられるようにしよう。でもその前に、動きをわかりやすくするため、ステージの車を1台に減らしておこう。ステージにある車を、右上にある「ぶひん」や「部品置き場」にドラッグ&ドロップ (1) すると消せるよ。



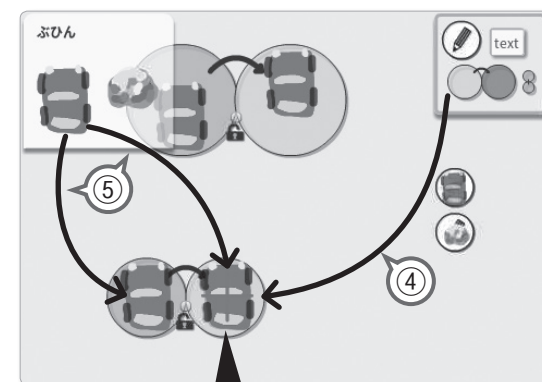
1台だけ残して、あとは消してしまおう。

車が1台になったら、右上の鉛筆のアイコン (2) をクリックして、「お絵かき画面」で障害物となる岩の絵を描こう。描き終わったら「O」 (3) をクリックだ。

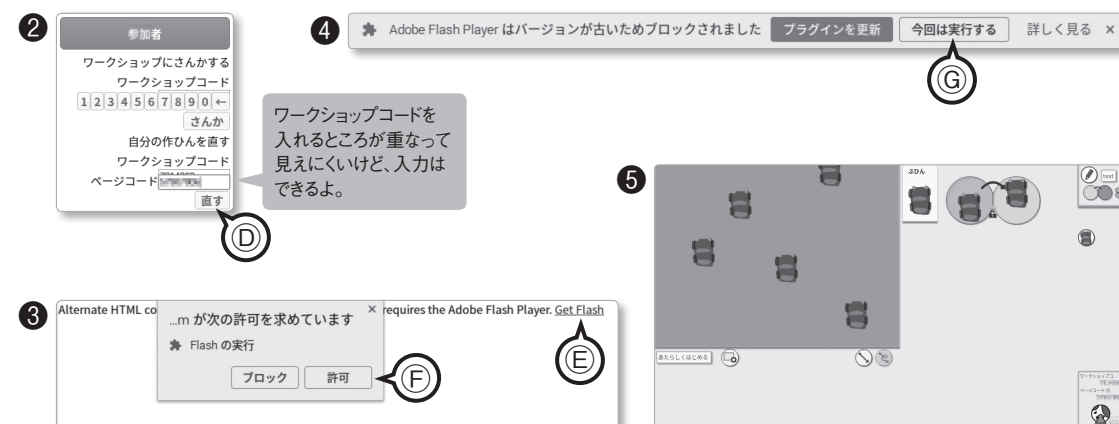


車が岩にぶつかりそうになったときにどうするかを表すために、部品置き場からメガネをドラッグして取り出そう (4)。

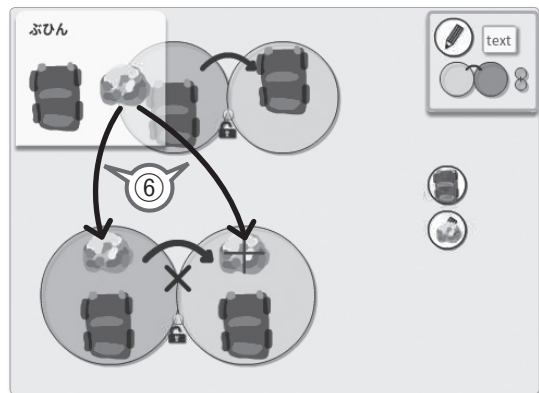
次に、メガネの左右に、「ぶひん」から車をドラッグして入れる (5)。左右の車は同じ場所に置きたいので、赤い十字線が出るのを確認しよう。



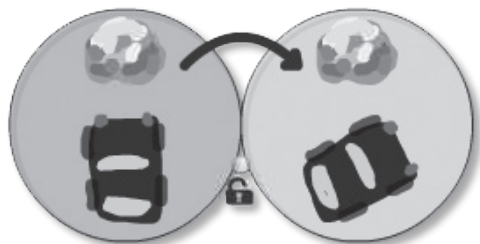
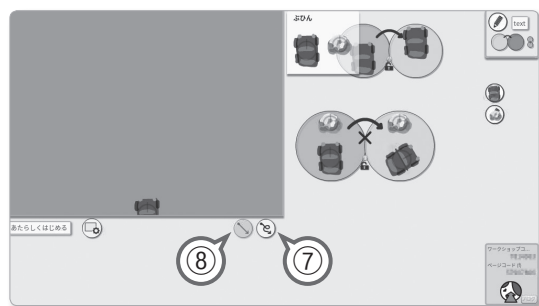
車が赤い場合、十字線が見にくいかもしれない。その場合は、車の色を変えてみよう。



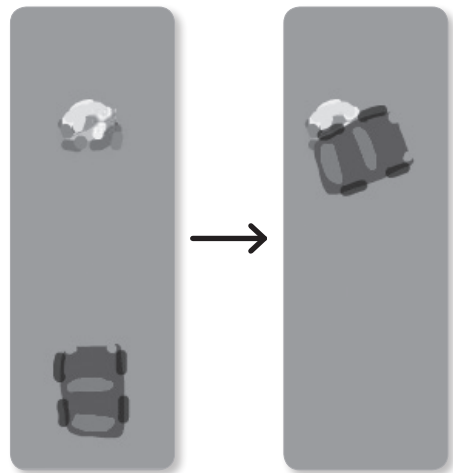
同じように、岩もメガネの左右に入れよう(⑥)。このとき、岩を車のちょっと前に離して置くのがポイントだ。一度、メガネの中に岩をドロップしてから、上にドラッグして位置を調整するとやりやすいぞ。これも左右で同じ場所に置きたいので、赤い十字線を確認してね。



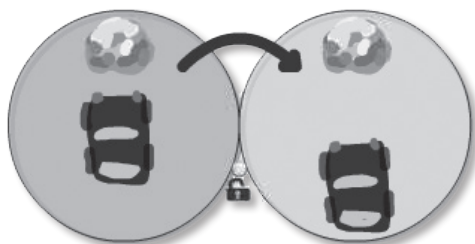
車が岩にぶつかりそうになったら、右に回って避けることにしよう。絵を回転させるには、ステージの右下にある、矢印のついた線がクルっとなっているアイコン(⑦)をクリックする。すると、絵の上に向きを表す線が表示されるので、絵をドラッグすると回転するよ。60°くらい右に傾けよう。回し終わったら、直線の矢印のアイコン(⑧)をクリックして、回らないようにしておこう。



ここまでできたら、「ぶひん」から岩をドラッグして、ステージにある車の前に置いてみよう。どうなるかな？



曲がることはできたけど、岩にぶつかっちゃったし、岩も一緒に回っちゃったね。ぶつからないようにするには、もう一工夫必要みたいだ。例えば、こんなメガネを追加するのはどうかな。



これは、岩にぶつかりそうになったら、ちょっとバックするという意味だ。このメガネを追加したら、ステージに岩を増やして、どうなるか試してみよう。

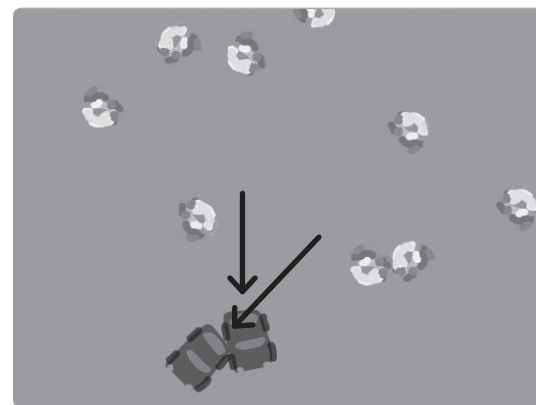


うわー、岩が追いかけてくる! 複数のメガネがある場合、それぞれのメガネで表した条件を満たそうとして、予想しなかった動きになることがあるんだ。みんなはどうなったかな？

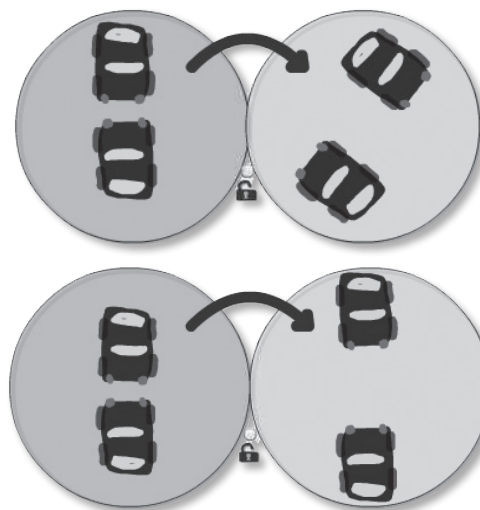
岩は動かないというのが人間の常識だけど、コンピュータにとっては、岩の絵も車の絵も同じ物として扱われるんだね。

## 車同士がぶつからないようにする

続いて、「ぶひん」からステージに車を追加して、どうなるか見てみよう。

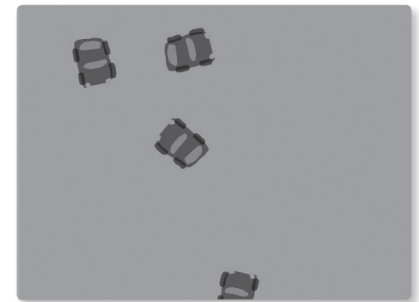


まだ、車と車の関係を表したメガネがないので、お互いにすり抜けてしまっている。では、こんなメガネはどうだろう。



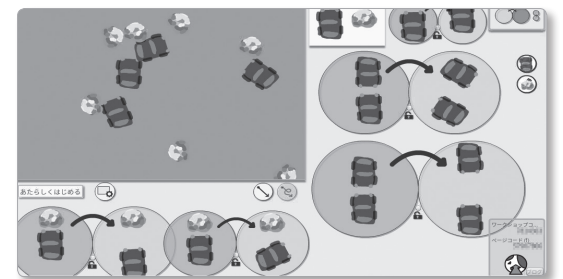
これは、ぶつかりそうになったら、お互いに左に避けることと、ちょっとバックすることを表している。

わかりやすくするために、ステージから岩をどけて車だけにとしてみると、いい感じで衝突を回避しているね。



メガネとしては、さっきの岩と車の関係とほとんど同じなのに、車同士だとうまく動いていると感じるのはおもしろいね。

最後に岩も入れてみよう。



今回作成したプログラムを保存したいときは、右下にある地球と矢印が組み合わさったアイコンをクリックしよう。ワークショップコードとページコードは変わらないので、これは上書き保存になるよ。ビスケットのページには、他にもおもしろい例がたくさん載っているので試してみよう。

2回にわたってビスケットを使ってきたけど、これまでにやってきたプログラミングのやり方と全然違うことがわかったかな。プログラミング言語には、これら以外にも、論理型や関数型、それらのハイブリッド型など、さまざまな考え方でつくられたものがある。1つの言語を極めるのも大切だけど、いろいろな考え方を知るのも重要だ。それぞれの言語の特徴を理解することで、難しい問題にぶつかったとき、別の視点から解決方法を見つけられるかもしれないよ。この連載では、これからもたくさんの言語を紹介していく予定だ。お楽しみに!