

連載

ゼロからはじめるPython

第10回

100行程度でブロック崩しゲームを作ってみよう

クジラ飛行機

2017/08/15 09:30

ここ数回、TkinterというGUIライブラリを利用して、ゲーム制作に挑戦している。今回作るのは、レトロ感溢れるゲーム『ブロック崩し』だ。100行程度のプログラムで実現できるので挑戦してみよう。

ブロック崩しのゲーム (GIFアニメ)

ブロック崩しとは？

ブロック崩しというのは、1970年から1980年にかけて登場したゲームのジャンル。壁にぶつくと反射するボールを、画面下部に落とさないように、バーを左右に移動させるゲームだ。画面にはブロックが配置されており、ボールでブロックを崩していく。

まずはボールを反射させるだけのプログラムから作ってみよう

ボールを画面下部に落とさないようにするという簡単なルールのゲームだが、一度に全部を実装しようとする、どのようにプログラムを組み立てて良いのか悩むかもしれない。ゲームのプログラムを作る場合、その核となる部分を少しずつ組み立てて行くと良い。そこで、まずは、ボールを反射させるだけのプログラムから作ってみよう。



以下はボールが反射して動くだけのプログラムだ。以下のプログラムを「ball-move.py」(ソースコード：[ball-move.lzh](https://news.mynavi.jp/article/zeropython-10/))という名前で保存しよう。

```
# ブロック崩し
from tkinter import *

# ボールを表す辞書型データ --- (*1)
ball = {
    "dirx": 15, # x方向のボールの速さ
    "diry": -15, # y方向のボールの速さ
    "x": 350, # ボールの位置
    "y": 300,
    "w": 10, # ボールの幅
}

# ウィンドウの作成 --- (*2)
win = Tk()
cv = Canvas(win, width = 600, height = 400)
cv.pack()

# 画面を描画する --- (*3)
def draw_objects():
    cv.delete('all') # 既存の描画を破棄
    # ボールを描画
    cv.create_oval(
        ball["x"] - ball["w"], ball["y"] - ball["w"],
        ball["x"] + ball["w"], ball["y"] + ball["w"],
        fill="green")

# ボールの移動 --- (*4)
def move_ball():
    # 仮の変数に移動後の値を記録
    bx = ball["x"] + ball["dirx"]
    by = ball["y"] + ball["diry"]
    # 上左右の壁に当たった?
    if bx < 0 or bx > 600: ball["dirx"] *= -1
    if by < 0 or by > 400: ball["diry"] *= -1
    # 移動内容を反映
    if 0 <= bx <= 600: ball["x"] = bx
    if 0 <= by <= 400: ball["y"] = by
```

```
# ゲームループ --- (*5)
def game_loop():
    draw_objects()
    move_ball()
    win.after(50, game_loop)

game_loop()
win.mainloop() # ゲームウィンドウを表示
```

プログラムを実行するには、PowerShellやターミナルを起動し、プログラムをカレントディレクトリに保存したら、次のコマンドを実行する。Windowsでは、「python」コマンドを、macOSでは「python3」コマンドを実行する。

```
# Windowsの場合
python ball-move.py
# macOSの場合
python3 ball-move.py
```

それでは、プログラムを一つずつ見てみよう。プログラムの(*1)の部分では、ボールを表す辞書型のデータを定義している。これは、辞書型(dict型)の変数だ。辞書型の変数を使うと、複数の値を一つの変数で管理できるのが良い。ここでは、ボールの位置(x,y)、幅(w)、ボールの速さと進行方向(dirx, diry)を一つの変数で管理する。

ゲームの中では、さまざまな変数を利用するので、このように、辞書型変数を使って、特定の目的で利用する変数をまとめておくとう管理しやすくなる。

プログラムの(*2)では、ウィンドウを作成する。ここでは、幅600、高さ400ピクセルのウィンドウを作成する。(*3)では、画面を描画する。ここでは、毎回、全ての描画内容をクリアして、そこに新たに描画内容を書き込むという処理を記述している。cv.delete('all')と記述すると、全ての描画内容をクリアできる。そして、create_oval()メソッドで、(x1,y1,x2,y2)のサイズの楕円を描画する。ここでは、ボールの左上(x1,y1)と右下(x2,y2)の座標を指定する。

プログラムの(*4)では、ボールの座標を移動させる。このとき、X座標とY座標を確認して、上下左右の壁に当たったことが分かれば、ボールの進行方向に-1をかけて、反射させる処理を行う。

そして、プログラムの(*5)の部分で、繰り返し、画面の描画とボールの移動を行うように指定する。

ブロック崩しを完成させよう

そして、ボール移動のプログラムにブロックとバーの描画、そして、マウスの座標を調べて、バーの移動処理を追加すると、ブロック崩しのゲームが完成する。

それらの処理の中で、ブロックに関する扱いを紹介しよう。ブロックの各座標を管理するblocksという配列変数を用意し、その中に各ブロックの座標や色を設定することにしよう。その場合、以下のようなプログラムを用意し、5x8で合計40個のブロックを生成する。

```
blocks = [] # ブロックを管理する配列
# ... 省略 ...
# ブロックを配置する
for iy in range(0, 5):
    for ix in range(0, 8):
        color = "red"
        if (iy + ix) % 2 == 1: color = "blue"
        x1 = 4 + ix * block_size["x"]
```

```
x2 = x1 + block_size["x"]
y1 = 4 + iy * block_size["y"]
y2 = y1 + block_size["y"]
blocks.append([x1, y1, x2, y2, color])
```

ブロックを描画するために、`create_rectangle()`メソッドを使って、以下のように、矩形を描画できる。このようにして、複数のブロックを用意したら、繰り返し構文の`for`を使うなら、数行のプログラムで描画できる。

```
# ブロックを一つずつ描画
for w in blocks:
    x1, y1, x2, y2, c = w
    cv.create_rectangle(x1, y1, x2, y2, fill=c, width=0)
```

あとは、ボールがブロックに当たったかどうか、また、ボールが画面下部に落ちたかどうかを一つずつ確認すればゲームが完成だ。

実際のプログラムは、こちら([block-ball.lzh](#))からダウンロードできる。100行程度のプログラムで、それほど長くないので、ダウンロードしたプログラムを、一行ずつ確認してみたい。基本となる部分は、ボールの反射プログラムとほとんど同じなので、何が付け加えられたのか、見比べながら確認してみると理解しやすいだろう。

なお、プログラムを実行するには、ターミナルから以下のコマンドを実行する。

```
# Windowsの場合
python block-ball.py
# macOSの場合
python3 block-ball.py
```

まとめ

以上、今回は、懐かしのレトロゲームのブロック崩しを作ってみた。ゲームを作ってみると、プログラミング力がぐっとアップする。ゲーム開発は、楽しくプログラミング能力を向上させることができる、素晴らしい題材だ。Pythonの基礎構文が身についたら、ぜひ、ゲーム開発に挑戦してみよう。なお、レトロゲームは、それほど難しい処理や豪華なグラフィック処理が不要なので、オススメだ。

自由型プログラマー。[くじらはんど](#)にて、プログラミングの楽しさを伝える活動をしている。代表作に、日本語プログラミング言語「[なでしこ](#)」、テキスト音楽「[サクラ](#)」など。2001年オンラインソフト大賞入賞、2005年IPAスーパークリエイター認定、2010年 OSS貢献者章受賞。技術書も多く執筆している。

※本記事は掲載時点の情報であり、最新のものとは異なる場合があります。予めご了承ください。

連載一覧

- 第12回 過去の気象情報をもとに運動会の開催日を選んでみよう(その1) - 気象データの入手と簡単な統計
- 第11回 最も身近なオープンデータ「郵便番号データベース」を活用しよう
- 第10回 100行程度でブロック崩しゲームを作ってみよう

2019/6/27

ゼロからはじめるPython(10) 100行程度でブロック崩しゲームを作ってみよう | マイナビニュース

第9回

[生物集団の栄枯盛衰"ライフゲーム"を作ってみよう](#)

第8回

[ゲームで覚えるPythonプログラミング - Tkinterで始めよう](#)

[目次ページへ](#)