

# はじめてのリアルタイム Python プログラミング

佐々木 弘隆

Pythonの強みとして、高度なライブラリを簡単に組み込めることが挙げられます。グラフィックス、入力、音声などの高度な処理が行えるPygameというライブラリ<sup>注1</sup>がありますので、これを使ったゲーム・プログラムを作ります。

## 画像&音声処理ライブラリPygame

### ● 機能

Pygameライブラリの機能は以下です。

画面処理：ウィンドウの制御

図形描画：直線/四角/丸などの描画

画像描画：画像ファイルの画像描画

イベント：キーボードやマウスの読み取り

音声出力：音楽と効果音再生

### ● 準備

#### ▶ Windowsの場合

Pygameを使うには、スクリプトの先頭でPygameをインポートするだけです。Pythonではimport文を使って標準ライブラリやpipでインストールしたパッケージ、自作のパッケージなどをインポートできます。

```
# -*- coding: utf-8 -*-
import pygame
```

なお、1行目はスクリプト・ファイルの文字コード指定です。エディタで使っている文字コードに合わせないと日本語のコメントなどでエラーが出てしまいます。Windowsの場合はシフトJIS(shift\_jis)を使うことが多いのですが、世界的にはUTF8(utf-8)の場合が多いのでエディタで文字コードの指定ができるならUTF8に変更することをお勧めします。

Pythonで対応している各種文字コードの情報は以下のWebページで確認できます。

<https://docs.python.org/3/library/codecs.html#standard-encodings>

#### ▶ Linuxの場合

LinuxなどでPythonを使う場合は次のようなスクリプト・ファイルの記述ができます。

#### リスト1 基本処理ループの作成

```
def main():
    clock = pygame.time.Clock()
    while (1):
        clock.tick(60) # フレーム・レート(60fps)
        # これ以降の処理が60fpsで動く
```

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import pygame
```

1行目の記述は、このファイルを実行するときに起動するプログラムを指定するものです。

## 画面表示の基本プログラム

### ● 基本処理ループの作成

ゲームの基本的な構造はリアルタイム処理になります。処理の間隔は60フレーム/s(1秒間に60回)とします。リアルタイム処理は、一定のタイミングで呼び出す必要があるので、Pygameの時間処理を使ってタイミングを調整した無限ループ構造を作ります(リスト1)。

時間処理を行うにはpygameのtime.Clockの機能を使います。機能を使うたびにpygame.time.Clockと打つのは面倒ですので、これを管理するオブジェクトclockを用意して格納します。以降はclockと書くだけでpygame.time.Clockと書いたのと同じ効果があります。

clockに格納されたtime.Clockのtickには、2つの役割があり、渡した引数(括弧の中の数字)のフレーム・レートと一致するように処理タイミングを待つようになります。

#### ▶ フレーム・レートを60fpsにする理由

リアルタイム処理のベースを構築したので、次はゲームを描画するスクリーンを追加します。ゲームの場合は画面の更新に合わせたリアルタイム処理が必須で、60fps(1秒間に60回)としています。これをフ

注1：現在は後継のPySDL2もありますが、ここではPygameを使います。