

Pygame 入門

@DA研 夏合宿 9/24-9/26

目次

クリックすることでそのページに飛びます

[Pygameをインストールする](#)

[Pygameをインポートする](#)

[よく使うクラス](#)

[ゲーム設定を行って画面を表示する](#)

[ゲームループを定義する](#)

[画面を塗りつぶす](#)

[図形を描画する](#)

[画像を読みこんで表示する](#)

[文字を表示する](#)

[キー入力を受け取る](#)

[キー入力\(長押し\)を受け取る](#)

[参考サイト](#)

Pygameをインストールする

まずはpipをアップデートする.

```
pip install --upgrade pip
```

下記のコマンドを実行し, Pygameをインストールする.

```
pip install pygame
```

Pygameをインポートする

`pygame_test.py` に下記のコードを書き, エラーにならないければOK.

```
import pygame
```

pygameをインストールできていない場合, 以下の画像のようなエラーが出てしまう.

```
import pygame
```

"pygame" にアクセスできません Pylance

AI Fix In Chat Ctrl+Shift+E

`sys` というモジュールもインポートしておく.

```
import sys
```

よく使うクラス

`pygame` でゲームを作成するために必用不可欠なクラスをいくつか紹介するので、以降のページではこれらのクラスを説明なしで使用する。

クラス	概要
<code>Color</code>	画面に表示される色を表すためのクラス。 RGB(赤, 緑, 青)の組合せで指定する。それぞれの値は0~255の範囲の整数で指定する。 黒(0, 0, 0) ~ 白(255, 255, 255)
<code>Rect</code>	矩形(四角形)を表すためのクラス。 物体の位置と大きさを指定するために使われる。位置は矩形の左上の角を基準とする。 (x座標, y座標, 幅, 高さ)で指定する。 <code>Rect.size</code> で(幅, 高さ)を取得できる。
<code>Surface</code>	描画する画像やキャンバスを表すためのクラス。 画面に表示されるすべての要素はこの <code>Surface</code> 上に描画される。 ゲームのメイン画面自体も一つの <code>Surface</code> である。

ゲーム設定を行って画面を表示する

以下の設定は基本的に必須なので、どんなゲームでも記述する.

- `pygame.init()` でpygameを初期化
- `pygame.display.set_mode(Rect.size)` でゲーム画面のサイズを設定
- `pygame.display.set_caption(タイトル)` でゲームのタイトルを設定

```
def main():  
    pygame.init()                # pygameを初期化  
    screen = pygame.display.set_mode((400, 300))  # 400×300のスクリーンを定義  
    pygame.display.set_caption("game title")      # ゲームタイトルを設定  
  
if __name__ == "__main__":      # このファイルが実行されたとき, main関数を呼び出す  
    main()
```

これで実行すると黒いゲームウィンドウが開き、すぐに閉じる.

※次からはmain関数の `while` の中に記述していくので、上のコードは消さずに残しておく.

ゲームループを定義する

今のままでは一瞬でゲームが終了してしまうので、以下のようにゲームループを定義する。

1. `while True` で無限ループを回す
2. `pygame.event.get()` でゲーム内のイベント(マウスイベントやキーボードイベント)を取得
3. 閉じるボタンが押されたとき(`event.type == pygame.QUIT`)に終了

```
pygame.init()
screen = pygame.display.set_mode((400, 300))
pygame.display.set_caption("game title")
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
```

無限ループ
イベントを取得する
閉じるボタンが押されたとき
pygameを終了
このプログラムを終了

画面を塗りつぶす

`screen.fill(Color, Rect)` で画面の指定したところを塗りつぶす。
第2引数を省略すると、画面全体を塗りつぶす。

これだけでは画面は変わらないので、`pygame.display.update()` で画面を更新する必要がある。

```
while True:
    screen.fill((0, 0, 255))          # 画面全体を青色で塗りつぶす
    screen.fill((255, 0, 0), (30, 0, 50, 60)) # 指定したところを赤色で塗りつぶす
    pygame.display.update()          # 画面を更新

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
```


図形を描画する

`pygame.draw.図形名()` で図形を描画する.

図形	コード	説明
直線	<code>line(screen, Color, 始点(x, y), 終点(x, y))</code>	指定した始点 - 終点の直線を描画する
四角形	<code>rect(screen, Color, Rect)</code>	指定した <code>Rect</code> に沿った四角形を描画する
円	<code>circle(screen, Color, 中心(x, y), 半径)</code>	指定した中心・半径の円を描画する

```
while True:
    # 中心(80, 80)で半径60の紫色の円を描画
    pygame.draw.circle(screen, (255, 0, 255), (80, 80), 60)
    pygame.display.update()

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
```

画像を読みこんで表示する

`pygame.image.load(filepath)` で画像を読みこむ.

`pygame.transform.scale(object, size)` で画像(他にもいろんなもの)を指定したサイズにする.

`screen.blit(object, (x, y))` で画像(他にもいろんなもの)を描画する.

```
before = pygame.image.load("./images/test.png")    # test.pngを読みこみ, beforeに代入
after = pygame.transform.scale(before, (200, 200)) # beforeを拡大してafterに代入

while True:
    screen.blit(before, (60, 130))                  # beforeを位置(60, 130)に描画
    screen.blit(after, (180, 50))                   # afterを位置(180, 50)に描画
    pygame.display.update()

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
```

文字を表示する

フォントを `pygame.font.Font(filename, fontsize)` で設定する.

`pygame.font.Font.render(Font, text, antialias, Color)` でテキストを作成する.

最後に `screen.blit(text, (Rect.x, Rect.y))` で指定した位置に表示する.

```
font = pygame.font.Font(None, 50)    # 文字サイズ50pxのフォントを作成
while True:
    # 黒色の"pygame"という文字をtextに代入
    text = pygame.font.Font.render(font, "pygame", True, (255, 255, 255))
    screen.blit(text, (50, 50))        # textを位置(50, 50)に描画
    pygame.display.update()

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
```

※フォントは変更する必要がないので, `while` の外で定義している.

キー入力を受け取る

イベントを取得して, `event.type == pygame.KEYDOWN` の時に様々なキーイベントを処理する.
`event.key` にキーイベントが格納されているので, `if` 文でどのキーが押されたかを確認する.

```
x = 0 # 四角形のx座標を定義
while True:
    screen.fill((255, 255, 255)) # 画面全体を白色で塗りつぶす(重要)
    pygame.draw.rect(screen, (0, 0, 255), (x, 130, 40, 40)) # 横移動する四角形を描画
    pygame.display.update()

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                x -= 10
            if event.key == pygame.K_RIGHT:
                x += 10
```

なにかのキーが押されたとき
「←」が押されたとき
x座標を左に10移動させる
「→」が押されたとき
x座標を右に10移動させる

画面を塗りつぶさなければどうなる？

キー入力(長押し)を受け取る

`pygame.key.get_pressed()` で入力されているキーを取得して長押しできるようにする。

```
x = 0
while True:
    screen.fill((255, 255, 255))
    pygame.draw.rect(screen, (0, 0, 255), (x, 130, 40, 40))
    pygame.display.update()

    pressed_key = pygame.key.get_pressed() # 押されているキーを取得する
    if pressed_key[pygame.K_LEFT]:        # 「←」が押されている時
        x -= 1 / 10                        # x座標を左に10移動させる
    if pressed_key[pygame.K_RIGHT]:        # 「→」が押されている時
        x += 1 / 10                        # x座標を右に10移動させる

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
```

※このやり方では「一度押す」ができないので、状況に応じて使い分ける必要がある。

参考サイト

この資料は以下のサイトを参考にして作成しています.

- [PYGAME Docs](#)

pygameの公式ドキュメント.

使いたいメソッドの引数と返り値, 型や使用方法を知りたいときはここが一番わかりやすい.

- [【Pygame超入門】使い方とサンプルゲームを解説](#)

やりたいこととそのサンプルコードがまとまっているので, 逆引きとして非常に便利.