

プログラミング演習(Python)レポート R04

main

クラス：D

担当教員名：小林郁夫

学籍番号：19K1142

氏名：リ イーセイ

1. 課題

Pygame を用いてゲームを作る。Main のプログラムでは、ピンボールゲームを作った。このレポートでは、ピンボールゲームのプログラムについて詳しく報告する。

今回のプログラムで、(1)スタートボタン、(2)スコアの表示機能、(3)ブロック崩し、(4)スコアにより別の終了画面を実装した。また、ボールがスタートのところから離れると戻れないように設定した。

2. 課題の目的

今回の課題を通し、Pygame の使い方への理解を深め、ゲームを作れるようになる。

3. 方法

Pygame、dataclass、random をインポートする。Pygame の初期化をする。

3.1 初期化

ゲーム進行の Screen とゲーム終了の end_screen を生成する。ゲームのタイトルをセットし、文字のフォントを設定する。また、一秒あたりに 60 回画像を生成するように設定する。このプログラムでは、ボールに重力が働かないと仮定する。

3.2 クラス

Ball、Paddle、Block、Score、Box のクラスを用意する。

Ball のクラスにボールの x、y 座標、横と縦の長さ、x、y 方向の速度と画像の属性を入れる。また、ボールの移動の関数とボールを描く関数を入れる。

Paddle のクラスにパドルの x、y 座標、横と縦の長さ、x 方向の速度の属性を入れる。また、移動と描くの関数を入れる。

Block のクラスにブロックの x、y 座標の属性と描く関数を入れる。

Score のクラスに点数の属性とスコアをスクリーンの特定の場所に置く関数を入れる。

Box のクラスにボックスの大きさを入れる。そして、ゲームに使う関数を書く。

3.3 関数

① ボックスと開始ボタンを描く関数

ボックスの横の長さは 400 であり、縦は 600 である。そして、ボタンは右下にあり、半径 30 の円である。

② 当たる判定の関数

当たる判定は、壁、パドル、ブロックに当たるときの判定とする。パドルとブロックの判定はボールの座標とそれらの座標を比較して、もしボールの座標は障害物の座標に重なると、当たったこととなり、ボールの速度の方向を変えるとよい。一方、壁の判定は、ボールがスタートのところから離れる前と後に分ける。ボールの x 座標が 380 より小さいときにボールがスタートの場所から出たとみなし、新しい壁を作り、ボールが元の場所に戻れなくなる。

③ セット関数

ボール: ボールの画像を導入する。ボールの横と縦の長さを `image.get_rect()` で得る。ボールの初期座標を (405,600) に設定する。Ball クラスを呼び出し、ball の属性を入れる。

パドル: パドルの属性を Paddle クラスに入れる。

ブロック: ブロックのリストを Box の中で作る。Block クラスを n 回呼び出し、座標をランダムに決めて入れる。また作られたブロックをリストに入れる。

スコア: 初期のスコアは 0 とする。Score クラスにスコアを入れる。

④ ゲーム進行の関数

パドルの移動: キー情報を取得する。Left key を押すとき、パドルの x 座標からパドルの速度を減る。Right key を押すとき、パドルの速度を増す。

ボールの y 方向速度: マウス情報を取得する。開始ボタンを押せば、ボールの y 速度をランダムで 3-7 から決められる。x 速度は 5 とする。

ゲーム進行: ボックスと開始ボタンを描く関数を呼び出す。当たる判定に関する関数を入れる。スコアとボールの y 方向速度を表示するテキストを用意する。それぞれをスクリーンに映る。最後、ブロックのリストにブロックが入ってないときは新しいブロックを作る。

ゲーム開始前: マウスの情報を検出し、開始ボタンをクリックしたらゲーム進行の関数を呼び出す。

ゲーム開始後: ゲーム進行の関数を呼び出す。

ゲーム終了: スコアにより、四つの終了画面を用意する。終了画面には違う画像とテキストがある。

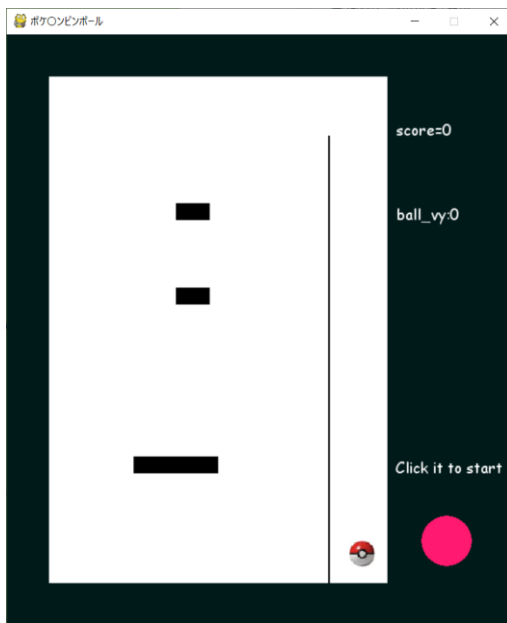
⑤ アニメーション

無限ループに if 文を作り、ボールの位置情報よりゲーム開始前か、ゲーム開始後か、またはゲーム終了かの場合に分ける。

3.4 実行

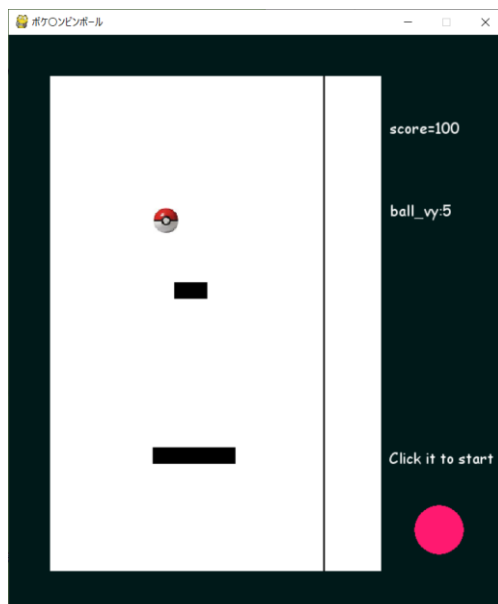
Box クラスに属性情報を入れる。ボール、パドル、ブロック、スコアをセットする関数を呼び出す。

4. 実行結果



(図 1 : 開始前)

図 1 はゲームが開始前の画像である。



(図 2 : ゲーム進行)

図 2 はゲーム進行中の画像である。



(図3：ゲーム終了)

図3はゲーム終了の画像である。

5. 考察

今回のプログラムで、Pygame を用いてピンボールゲームを作った。しかし、当てる判定関数を書くとき、colliderect を使うことができなかった。原因はボールの画像の rect を正確に取得してなかったと想定していたが、調整してもうまくいかなかった。また、ゲーム開始画面を作ってみたが、ゲーム進行の画面と切り替えることができなかった。他の機能は思い通りに動いている。

6. まとめ

今回の課題は最終課題であるが、この課題を通しゲームを作る楽しさを理解し、pygame の機能の使い方も分かるようになった。これからはほかのゲームに挑む。