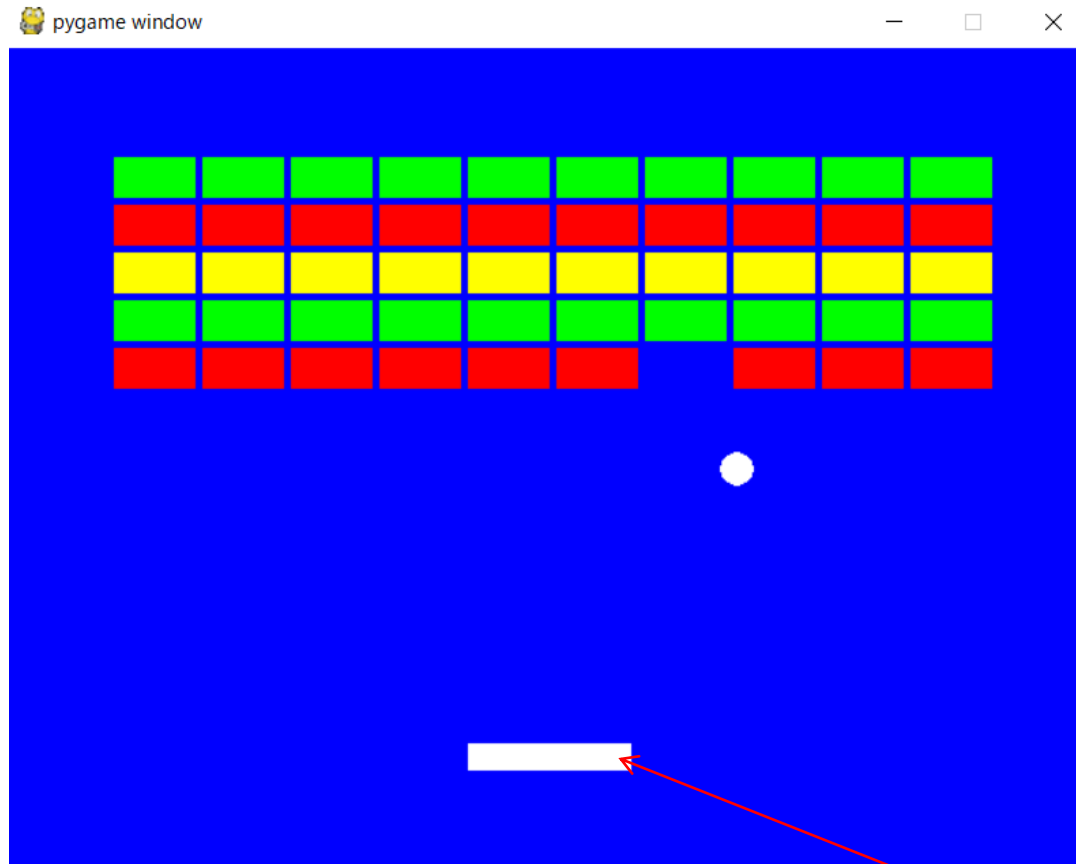


pythonによるブロック崩しゲームのプログラミング



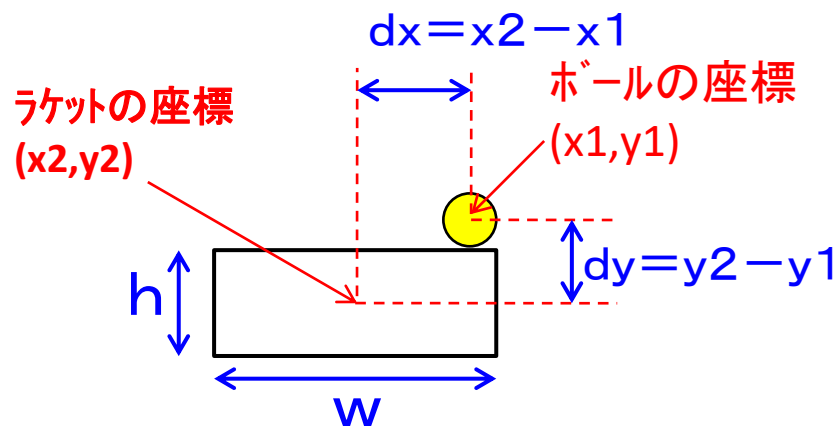
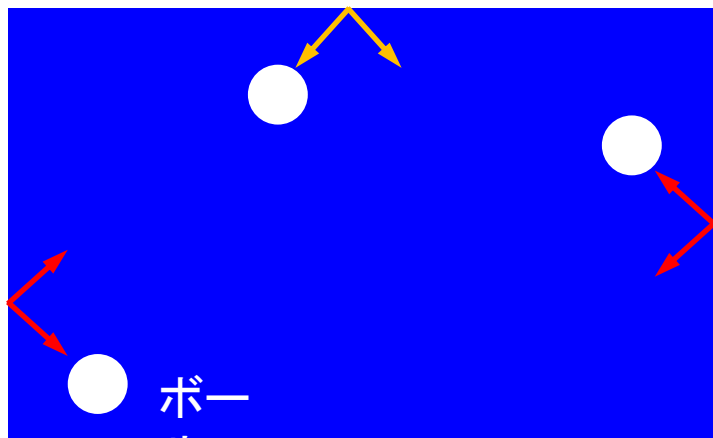
遊び方 パソコンのキーボードにある矢印キーの「←」と「→」を押すとラケットが左右に動きます。ボールを外すと「GAME OVER」の表示になります。

縦枠に当たった時はX方向の速度を反転

$$bx1 = -bx1$$

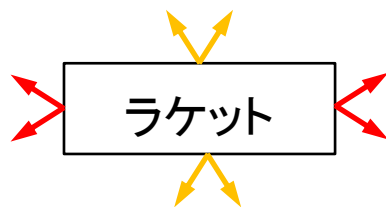
横枠に当たった時はY方向の速度を反転

$$by1 = -by1$$



$abs(dx/dy) > (w/h)$ の場合 X方向の速度を反転 $bx1 = -bx1$

$abs(dx/dy) < (w/h)$ の場合 Y方向の速度を反転 $by1 = -by1$



```
1 import pygame
2 import math
3
4 def sgn(a):
5     return 1 if a > 0 else -1
6
7 WIDTH = 640
8 HEIGHT = 480
9 YELLOW = (255,255,0)
10 WHITE = (255,255,255)
11 RED = (255,0,0)
12 GREEN = (0,255,0)
13 BLUE = (0,0,255)
14 colortable = [RED, YELLOW, GREEN]
15
16 pygame.init()
17 screen = pygame.display.set_mode((WIDTH, HEIGHT))
18 myfont = pygame.font.Font(None, 64)
19 myclock = pygame.time.Clock()
20 br = 10
21 paddlew = 96
22 paddleh = 16
23 blockw = 48
24 blockh = 24
25 endflag = 0
```

br: ボールの半径
paddlew: ラケットの幅
paddleh: ラケットの高さ
blockw: ブロックの幅
blockh: ブロックの高さ

ballx ボールのx座標
bally ボールのy座標

bx1 ボールの速度(x成分)
by1 ボールの速度(y成分)

```
28 while endflag == 0:
29     ballx = WIDTH / 2
30     bally = HEIGHT - 170
31     bx1 = 2
32     by1 = -2.5
33     x = WIDTH / 2
34     y = HEIGHT - 64
35     paddle = pygame.Rect(x - (paddlew / 2),
36     y - (paddleh / 2), paddlew, paddleh)
37     blocks = []
38     for i in range(50):
39         x = (i % 10) * (blockw + 4) + 64
40         y = int(i / 10) * (blockh + 4) + 64
41         blocks.append(pygame.Rect(x, y, blockw,
42         blockh))
43     gameover = 0
```

50個のブロック作成

```

44 while endflag == 0:
45     for event in pygame.event.get():
46         if event.type == pygame.QUIT: endflag = 1
47     press = pygame.key.get_pressed()
48     x = paddle.centerx
49     if (press[pygame.K_LEFT]): x -= 8
50     if (press[pygame.K_RIGHT]): x += 8
51     if x >= (paddlew / 2) &
52     and x <= (WIDTH - (paddlew / 2)):
53         paddle.centerx = x
54     x = ballx + bx1
55     y = bally + by1
56     if x < br or x > (WIDTH - br): bx1 = -bx1
57     if y < br: by1 = -by1
58     if y > HEIGHT: gameover += 1
59     dx = paddle.centerx - x
60     dy = paddle.centery - y
61     if dy == 0: dy = 1
62     if abs(dx) < (paddlew / 2 + br) &
63     and abs(dy) < (paddleh / 2 + br):
64         if abs(dx / dy) > (paddlew / paddleh):
65             bx1 = -bx1
66             ballx = paddle.centerx - sgn(dx) *
67             *(paddlew / 2 + br)
68         else:
69             bx1 = -dx / 10
70             by1 = -by1
71             bally = paddle.centery - sgn(dy) *
72             *(paddleh / 2 + br)

```

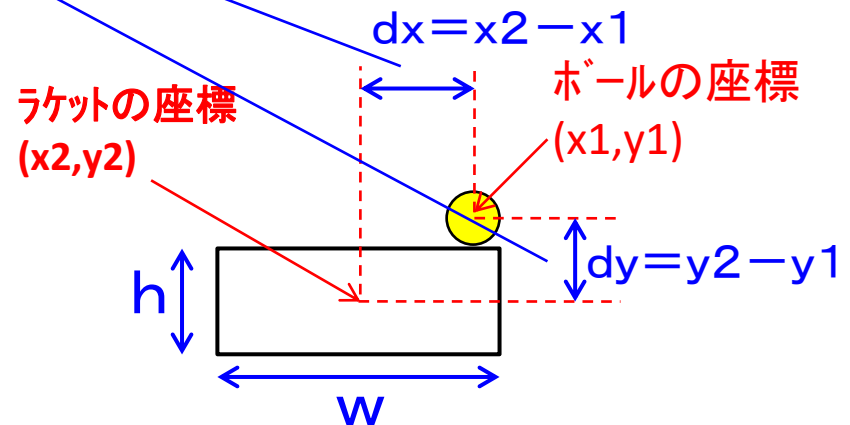
キー入力取得

pygame.K_LEFT
pygame.K_RIGHT

カーソルキー左
カーソルキー右

画面枠での反射

ラケットでの反射

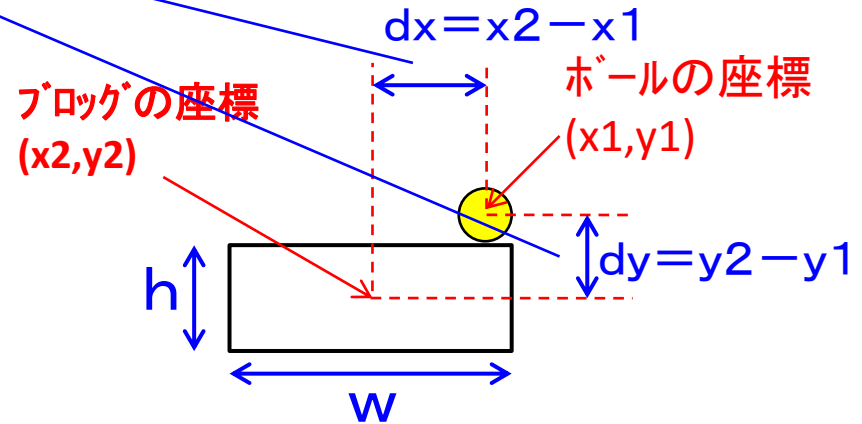


```

73 for block in blocks:
74     dx = block.centerx - x
75     dy = block.centery - y
76     if dy == 0: dy = 1
77     if abs(dx) < (blockw / 2 + br) &
78     and abs(dy) < (blockh / 2 + br):
79         if abs(dx / dy) > (blockw / blockh):
80             bx1 = -bx1
81             ballx = block.centerx - sgn(dx) &
82             * (blockw / 2 + br)
83         else:
84             by1 = -by1
85             bally = block.centery - sgn(dy) &
86             * (blockh / 2 + br)
87         blocks.remove(block)
88         break

```

ブロックでの反射



```

89 ballx += bx1
90 bally += by1

```

← 現在位置に
ボール速度を加算

```

91 screen.fill(BLUE)
92 for block in blocks:
93     color = colortable[int(block.y / 28) % 3]
94     pygame.draw.rect(screen, color, block)
95 pygame.draw.rect(screen, WHITE, paddle)
96 pygame.draw.circle(screen, WHITE, &
97 (int(ballx), int(bally)), br)
98 if gameover > 0:
99     imagetext = &
100     myfont.render("GAME OVER", True, WHITE)
101     screen.blit(imagetext, (180, 300))
102     if gameover > 200: break
103 myclock.tick(60)
104 pygame.display.flip()
105 pygame.quit()

```

ブロックの描画

← ラケットの描画

← ボールの描画

← ゲームオーバー