Assignment 5

學生姓名

— Functionality —

— Style —

+

✓

Style
Comments

Style Comments

- Super clear function comments 👍

- Programs are well-decomposed 👍

- Spaces between functions are consistent 👍

- Indentations and spaces are correct 👍

- Single line comments are precise, enabling users to understand your code easily 👍

Feedbacks is provided as follows:

Style Comments

```
37  def start(_):
38      """
39
40      """
```

這種function還是要寫一下comment

然後header comment要寫！

```
2   File: bouncing_ball.py
3   Name:110613024 呂信吾
4   -------------------------
5   TODO:
```

Case Testing

Problem 1 - draw_line

奇數次點擊畫圓	✓
圓的中心在點擊的位置	✓
偶數次點擊移除圓	✓
偶數次點擊畫線	✓
線的頭尾都在正確的位置	✓

Problem 2 - bouncing_ball

點擊開始動畫	✓	回到初始位置後點擊 球會再次落下	✓
球在下落過程中點擊不會影響	✓	球超出右側視窗三次後 點擊不再下落	✓
球不會黏在地板上	✓	球每次回到初始位置vy會重置	✓
球超出右側視窗會回到初始位置	✓	程式正常結束（exit code: 0）	✓

Functionality Comments

```
43    while True:
44        vy += 1
45        y = vy + (1/2)*1*GRAVITY
46        oval.move(VX, y)
47        pause(DELAY)
48        if oval.y >= window.height-SIZE:
49            vy = (-vy - 2)*0.9
50        if oval.x >= window.width-SIZE+20:
51            window.remove(oval)
52            window.add(oval, x=START_X, y=START_Y)
53            num += 1
54            break
```

```
43    while True:
44        vy += GRAVITY
45        oval.move(VX, vy)
46        pause(DELAY)
47        if oval.y >= window.height-SIZE and vy > 0:
48            vy = (-vy)*0.9
49        if oval.x >= window.width-SIZE:
50            window.remove(oval)
51            window.add(oval, x=START_X, y=START_Y)
52            num += 1
53            break
54    click += 1
```

你在這題加加減減了一些數字，感覺是因為你的球有黏在地板上的問題(？要解決這個問題的話其實只要加上and vy>0這個條件就行了！

然後你可以用click = True跟click = False當開關！

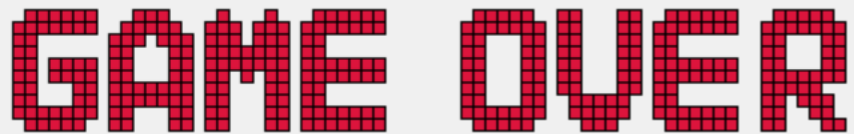# Functionality Comments

```
37    def start(mouse):
```

⬇

```
37    def start(_):
```

因為這題不會用到滑鼠資訊，所以可以把不重要的變數用底線代替就好！

Problem 3 - clear_bricks

視窗最上方隨機生成磚塊 範圍：0 < x < WINDOW_WIDTH – SIZE	✓
磚塊隨機落下速度 範圍：MIN_Y_SPEED～MAX_Y_SPEED	✓
點擊落下中的磚塊會將其移除 並重新從上方掉落一個	✓
磚塊掉出視窗會有新的磚塊從上方落下	✓
磚塊掉出視窗三次後程式會結束 （exit code: 0)	✓

超級喜歡這個！！超強ㄅ！

Functionality Comments