FAUGET HIGH SCHOOL

Pro-Tetris

Python商業應用

CONTENTS

INTRODUCING

MOTIVATION

PROGRAM

SPIN

```
#-------------------------------------------------------------------
# 變換方塊-上.
elif event.key == pygame.K_UP and game_mode == 0:
    # 在右邊界不能旋轉.
    if (container_x == 8):
        break
    # 判斷磚塊N1、N2、I.
    if (brick_id == 1 or brick_id == 2 or brick_id == 7):
        # 長條方塊旋轉例外處理.
        if (brick_id == 7):
            if (container_x < 0 or container_x == 7):
                break
        # 旋轉方塊.
        brick_state = brick_state + 1
        if (brick_state > 1):
            brick_state = 0
        # 轉換定義方塊到方塊陣列.
        transformToBricks(brick_id, brick_state)
        # 碰到磚塊.
        if (not ifCopyToBricksArray()):
            brick_state = brick_state - 1
            if (brick_state < 0):
                brick_state = 1
    # 判斷磚跨L1、L2、T.
    elif (brick_id == 3 or brick_id == 4 or brick_id == 5):
        # 旋轉方塊.
        brick_state = brick_state + 1
        if (brick_state > 3):
            brick_state = 0
        # 轉換定義方塊到方塊陣列.
        transformToBricks(brick_id, brick_state)
        # 碰到磚塊.
        if (not ifCopyToBricksArray()):
            brick_state = brick_state - 1
            if (brick_state < 0):
                brick_state = 3
#-------------------------------------------------------------------
```

DROP

```
# 遊戲中.
if (game_mode == 0):
    # 處理磚塊下降.
    if(time_now >= brick_down_speed):
        # 往下降.
        container_y = container_y + 1;
        # 碰到磚塊.
        if (not ifCopyToBricksArray()):
            #產生新塊.
            brickNew()
        # 轉換定義方塊到方塊陣列(bricks).
        transformToBricks( brick_id, brick_state)
        # 清除時脈.
        time_now = 0
```

GENARATE

```
def brickNew():
    global game_over, container_x, container_y, brick_id, brick_next_id, brick_state
    global lines_number, game_mode

    # 判斷遊戲結束.
    game_over = False
    if (container_y < 0):
        game_over = True

    # 複製方塊到容器內.
    container_y = container_y - 1
    copyToBricksArray()

    #-----------------------------------------------
    # 判斷與設定要清除的方塊.
    lines = ifClearBrick() / 10;
    if (lines <= 3):
        # 消除連線數量累加.
        lines_number =  lines_number + lines *100
        # 修改連線數量.
        #modifyLabel(linesNumber, fontLinesNumber)
        # 1:清除磚塊.
        game_mode = 1
    elif (lines == 4):
        lines_number =  lines_number + lines *200
        game_mode = 1
    # 初始方塊位置.
    container_x = 3
    container_y = -3

    # 現在出現方塊.
    brick_id = brick_next_id

    # 下個出現方塊.
    # 方塊編號(1~7).
    brick_next_id = random.randint( 1, 7)

    # 初始方塊狀態.
    brick_state = 0
```

NEXT CUBE

```
def updateNextBricks(brickId):
    global bricks_next

    # 清除方塊陣列.
    for y in range(4):
        for x in range(4):
            bricks_next[x][y] = 0

    # 取得磚塊索引陣列.
    pBrick = getBrickIndex(brickId, 0)

    # 轉換方塊到方塊陣列.
    for i in range(4):
        bx = int(pBrick[i] % 4)
        by = int(pBrick[i] / 4)
        bricks_next[bx][by] = brickId

    # 更新背景區塊.
    background_bricks_next.update()

    # 更新磚塊圖.
    pos_y = 52
    for y in range(4):
        pos_x = 592
        for x in range(4):
            if(bricks_next[x][y] != 0):
                bricks_next_object[x][y].rect[0] = pos_x
                bricks_next_object[x][y].rect[1] = pos_y
                bricks_next_object[x][y].update()
            pos_x = pos_x + 28
        pos_y = pos_y + 28
```

REMOVE

```
def ifClearBrick():
    pointNum = 0
    lineNum = 0
    for y in range(20):
        for x in range(10):
            if (bricks_array[x][y] > 0):
                pointNum = pointNum + 1
            if (pointNum == 10):
                for i in range(10):
                    lineNum = lineNum + 1
                    bricks_array[i][y] = 9
        pointNum = 0
    return lineNum
```

GAME OVER

```
# 判斷遊戲結束.
game_over = False
if (container_y < 0):
    game_over = True
```

PLAYTIME

Division

Arad

曾竑竣 10844227

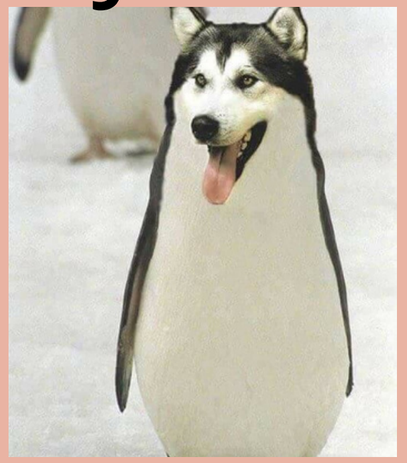
PPT Design

Victor

王柏凱 10844242

PPT Design

Jack



呂允丞 10844243

Program Design

Nick

楊子璿 10844244

Program Design

Thank You!

Any question?