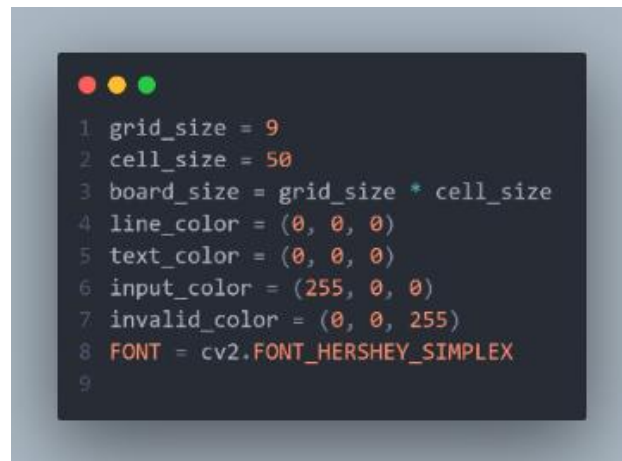


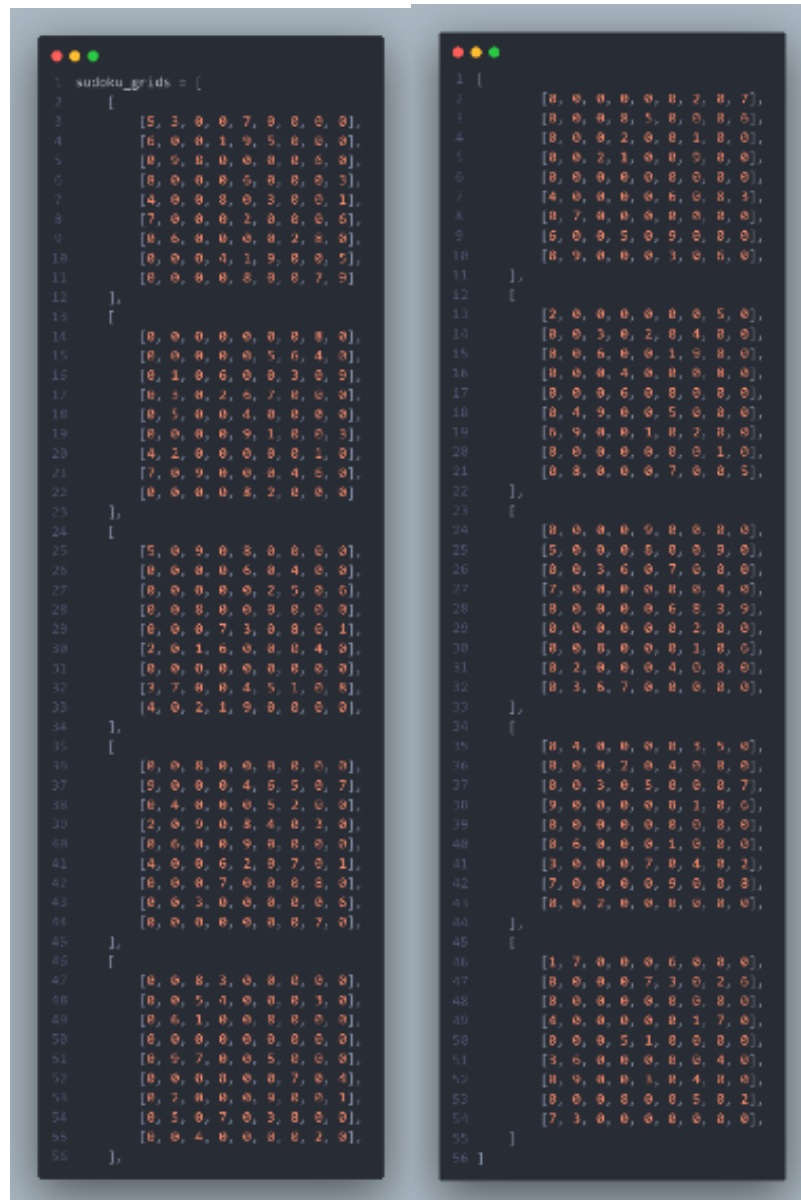
ภาคผนวก



รูปที่ 1 Import binary



รูปที่ 2 กำหนดตัวแปรของตารางและสี



รูปที่ 3 ด้านของเกม



รูปที่ 4 ตัวแปรข้อมูลของเกม

```

1 def is_valid_move(grid, row, col, num): # เช็คค่าแนวตั้ง แนวนอน ตาราง 3*3
2     for i in range(grid_size):
3         if grid[row][i] == num or grid[i][col] == num:
4             return False
5     start_row, start_col = 3 * (row // 3), 3 * (col // 3)
6     for i in range(3):
7         for j in range(3):
8             if grid[start_row + i][start_col + j] == num:
9                 return False
10    return True

```

รูปที่ 5 ฟังก์ชันตรวจสอบการใส่ตัวเลข

```

1 def draw_grid(img, grid, input_cells, invalid_cells, selected_cell): # สร้างตาราง
2     for i in range(grid_size + 1):
3         thickness = 2 if i % 3 == 0 else 1
4         cv2.line(img, (0, i * cell_size), (board_size, i * cell_size), line_color, thickness)
5         cv2.line(img, (i * cell_size, 0), (i * cell_size, board_size), line_color, thickness)
6
7     if selected_cell is not None:
8         sel_row, sel_col = selected_cell
9         cv2.rectangle(img, (sel_col * cell_size, sel_row * cell_size),
10                        ((sel_col + 1) * cell_size, (sel_row + 1) * cell_size),
11                        (255, 26, 0), 2)
12
13     for row in range(grid_size):
14         for col in range(grid_size):
15             if grid[row][col] != 0:
16                 text = str(grid[row][col])
17                 x = col * cell_size + cell_size // 4
18                 y = row * cell_size + int(cell_size * 0.75)
19                 color = text_color if (row, col) in default_cells else input_color
20                 cv2.putText(img, text, (x, y), FONT, 1, color, 2)
21
22     for (row, col), value in input_cells.items():
23         x = col * cell_size + cell_size // 4
24         y = row * cell_size + int(cell_size * 0.75)
25         color = invalid_color if (row, col) in invalid_cells else input_color
26         cv2.putText(img, str(value), (x, y), FONT, 1, color, 2)
27

```

รูปที่ 6 ฟังก์ชันวาดตารางซูโดกุ

```

1 def mouse_click(event, x, y, flags, param): # รับค่าจากเมาส์
2     global selected_cell
3     if event == cv2.EVENT_LBUTTONDOWN:
4         row = y // cell_size
5         col = x // cell_size
6         if (row, col) not in default_cells:
7             selected_cell = (row, col)

```

รูปที่ 7 ฟังก์ชันรับค่าจากเมาส์

```

1 def check_win(grid, input_cells): # เช็ควิน
2     for row in range(grid_size):
3         for col in range(grid_size):
4             if (row, col) not in default_cells:
5                 if (row, col) not in input_cells or not is_valid_move(grid, row, col, input_cells[(row, col)]):
6                     return False
7     return True

```

รูปที่ 8 ฟังก์ชันตรวจสอบว่าผู้เล่นชนะเกม

```

1 def start_new_level(): # เริ่มต้นด่าน
2     global level_index, sudoku_grid, default_cells, input_cells, invalid_cells, incorrect_input_count, selected_cell
3     level_index += 1
4     if level_index < len(sudoku_grids):
5         sudoku_grid = sudoku_grids[level_index]
6         default_cells = {(row, col) for row in range(grid_size) for col in range(grid_size) if sudoku_grid[row][col] != 0}
7         input_cells = {}
8         invalid_cells = set()
9         incorrect_input_count = 0
10        selected_cell = None
11
12        img[:] = 255
13        draw_grid(img, sudoku_grid, input_cells, invalid_cells, selected_cell)
14        cv2.imshow("Sudoku", img)
15

```

รูปที่ 9 ฟังก์ชันสำหรับเปลี่ยนด่านของเกม

```

1 img = np.ones((board_size, board_size, 3), dtype=np.uint8) * 255
2 draw_grid(img, sudoku_grid, input_cells, invalid_cells, None)
3
4 cv2.imshow("Sudoku", img)
5 cv2.setMouseCallback("Sudoku", mouse_click)
6

```

รูปที่ 10 การแสดงหน้าต่างของเกม

```

1 while True: # เริ่มเกม
2     cv2.imshow("Sudoku", img)
3     key = cv2.waitKey(1)
4
5     if key in [ord(str(i)) for i in range(1, 10)] and selected_cell:
6         row, col = selected_cell
7         num = int(chr(key))
8
9         if is_valid_move(sudoku_grid, row, col, num):
10             input_cells[(row, col)] = num
11             invalid_cells.discard((row, col))
12         else:
13             if (row, col) not in invalid_cells:
14                 incorrect_input_count += 1
15                 invalid_cells.add((row, col))
16                 input_cells[(row, col)] = num
17
18         img[:] = 255
19         draw_grid(img, sudoku_grid, input_cells, invalid_cells, selected_cell)
20         selected_cell = None
21
22     if check_win(sudoku_grid, input_cells):
23         cv2.rectangle(img, (0, board_size // 2 - 50),
24                       (board_size, board_size // 2 + 100), (200, 200, 200), -1)
25
26         win_text = "You win!!"
27         incorrect_text = f"Incorrect inputs: {incorrect_input_count}"
28
29         win_text_size = cv2.getTextSize(win_text, FONT, 2, 3)[0]
30         incorrect_text_size = cv2.getTextSize(incorrect_text, FONT, 1, 2)[0]
31
32         win_text_x = (board_size - win_text_size[0]) // 2
33         win_text_y = board_size // 2
34
35         incorrect_text_x = (board_size - incorrect_text_size[0]) // 2
36         incorrect_text_y = win_text_y + 50
37
38         cv2.putText(img, win_text, (win_text_x, win_text_y), FONT, 2, (0, 255, 0), 3)
39         cv2.putText(img, incorrect_text, (incorrect_text_x, incorrect_text_y), FONT, 1, (0, 0, 255), 2)
40         cv2.imshow("Sudoku", img)
41         cv2.waitKey(10000)
42         start_new_level()
43
44     if key == 8 and selected_cell:
45         row, col = selected_cell
46         if (row, col) in input_cells:
47             del input_cells[(row, col)]
48             invalid_cells.discard((row, col))
49
50         img[:] = 255
51         draw_grid(img, sudoku_grid, input_cells, invalid_cells, selected_cell)
52         selected_cell = None
53
54     if key == 27: # ESC key to exit
55         break
56
57     if cv2.getWindowProperty("Sudoku", cv2.WND_PROP_VISIBLE) < 1:
58         break
59
60 cv2.destroyAllWindows()

```

รูปที่ 11 ฟังก์ชันการทำงานของเกม sudoku

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

รูปที่ 12 การวาดตาราง

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

รูปที่ 13 การเลือกช่อง

5	3			7				
6			1	9	5			
1	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

รูปที่ 14 การกรอกข้อมูลตัวเลข 1-9

5	3			7				
6	1		1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

รูปที่ 15 การแสดงผลข้อมูลที่ผิดพลาด

5	3	4		7				
6		2	1	9	5			
1	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

รูปที่ 16 การลบข้อมูล

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
9	8	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

รูปที่ 17 การตรวจสอบการชนะ

							8	
					5	6	4	
	1		6			3		9
	3		2	6	7			
	5			4				
				9	1			3
4	2						1	
7		9				4	6	
				8	2			

รูปที่ 18 การเปลี่ยนระดับความยาก