PROGRAM:

```
import random

def start_game():
    mat = []
    for i in range(4):
        mat.append([0] * 4)
    print("Commands are as follows : ")
    print("'W' or 'w' : Move Up")
    print("'S' or 's' : Move Down")
    print("'A' or 'a' : Move Left")
    print("'D' or 'd' : Move Right")
    add_new_2(mat)
    return mat

def add_new_2(mat):
    if all(all(cell != 0 for cell in row) for row in mat):
        return

    empty_cells = []
    for r in range(4):
        for c in range(4):
            if mat[r][c] == 0:
                empty_cells.append((r, c))

    if empty_cells:
        r, c = random.choice(empty_cells)
        mat[r][c] = 2

def get_current_state(mat):
    for i in range(4):
        for j in range(4):
            if mat[i][j] == 2048:
                return 'WON'
    for i in range(4):
        for j in range(4):
            if mat[i][j] == 0:
                return 'GAME NOT OVER'
    for i in range(3):
        for j in range(3):
            if mat[i][j] == mat[i + 1][j] or mat[i][j] == mat[i][j + 1]:
                return 'GAME NOT OVER'
    for j in range(3):
        if mat[3][j] == mat[3][j + 1]:
            return 'GAME NOT OVER'
    for i in range(3):
        if mat[i][3] == mat[i + 1][3]:
            return 'GAME NOT OVER'
    return 'LOST'

def compress(mat):
    changed = False
    new_mat = []
    for i in range(4):
        new_mat.append([0] * 4)
    for i in range(4):
        pos = 0
        for j in range(4):
            if mat[i][j] != 0:
                new_mat[i][pos] = mat[i][j]
                if j != pos:
                    changed = True
                pos += 1
    return new_mat, changed

def merge(mat):
    changed = False
    for i in range(4):
        for j in range(3):
            if mat[i][j] == mat[i][j + 1] and mat[i][j] != 0:
                mat[i][j] = mat[i][j] * 2
                mat[i][j + 1] = 0
                changed = True
    return mat, changed

def reverse(mat):
    new_mat = []
    for i in range(4):
        new_mat.append(mat[i][::-1])
    return new_mat

def transpose(mat):
    new_mat = []
    for i in range(4):
        new_mat.append([])
        for j in range(4):
            new_mat[i].append(mat[j][i])
    return new_mat

def move_left(grid):
    new_grid, changed1 = compress(grid)
    new_grid, changed2 = merge(new_grid)
    changed = changed1 or changed2
    new_grid, _ = compress(new_grid)
    return new_grid, changed

def move_right(grid):
    new_grid = reverse(grid)
    new_grid, changed = move_left(new_grid)
    new_grid = reverse(new_grid)
    return new_grid, changed

def move_up(grid):
    new_grid = transpose(grid)
    new_grid, changed = move_left(new_grid)
    new_grid = transpose(new_grid)
    return new_grid, changed

def move_down(grid):
    new_grid = transpose(grid)
    new_grid, changed = move_right(new_grid)
    new_grid = transpose(new_grid)
    return new_grid, changed

def print_grid(mat):
    print("-" * 25)
    for row in mat:
        print(' '.join([f'{num:5}' for num in row]))
    print("-" * 25)

if _name_ == '_main_':
    mat = start_game()
    print_grid(mat)

    while True:
        x = input("Press the command: ")

        move_made = False

        if x.lower() == 'w':
            mat, move_made = move_up(mat)
        elif x.lower() == 's':
            mat, move_made = move_down(mat)
        elif x.lower() == 'a':
            mat, move_made = move_left(mat)
        elif x.lower() == 'd':
            mat, move_made = move_right(mat)
        else:
            print("Invalid Key Pressed")
            continue

        if move_made:
            add_new_2(mat)

        print_grid(mat)
        status = get_current_state(mat)

        if status != 'GAME NOT OVER':
            print(f"You {status}!")
            break
```