```python
class VacuumCleaner:
    def __init__(self, grid, start):
        self.grid = grid
        self.x, self.y = start
        self.rows = len(grid)
        self.cols = len(grid[0])
    def display(self):
        for row in self.grid:
            print(row)
        print()
    def clean(self):
        print("Starting Vacuum Cleaner Simulation...\n")
        self.display()
        while True:
            dirty_cells = [(i, j) for i in range(self.rows)
                           for j in range(self.cols) if self.grid[i][j]
                               == 'D']
```

```
Starting Vacuum Cleaner Simulation...

['C', 'D', 'C', 'C']
['C', 'C', 'D', 'C']
['D', 'C', 'C', 'C']
['C', 'C', 'C', 'D']

Moved Right to (0, 1)
Cleaned cell (0, 1)
['C', 'C', 'C', 'C']
['C', 'C', 'D', 'C']
['D', 'C', 'C', 'C']
['C', 'C', 'C', 'D']

Moved Down to (1, 1)
Moved Right to (1, 2)
Cleaned cell (1, 2)
```

```
            if not dirty_cells:
                print("All dirty cells cleaned!")
                break
            dx, dy = dirty_cells[0]
            self.move_to(dx, dy)
            self.grid[dx][dy] = 'C'
            print(f"Cleaned cell ({dx}, {dy})")
            self.display()
    def move_to(self, dx, dy):
        while self.x < dx:
            self.x += 1
            print(f"Moved Down to ({self.x}, {self.y})")
        while self.x > dx:
            self.x -= 1
            print(f"Moved Up to ({self.x}, {self.y})")
        while self.y < dy:
            self.y += 1
```

```
            print(f"Moved Right to ({self.x}, {self.y})")
        while self.y > dy:
            self.y -= 1
            print(f"Moved Left to ({self.x}, {self.y})")
environment = [
    ['C', 'D', 'C', 'C'],
    ['C', 'C', 'D', 'C'],
    ['D', 'C', 'C', 'C'],
    ['C', 'C', 'C', 'D']
]
start_position = (0, 0)
vacuum = VacuumCleaner(environment, start_position)
vacuum.clean()

```