# Assessment 1: Implementing a Simple Reflex Agent for a Grid-Based Environment

**Step-by-Step Instructions with Comments in Python Code**

# Step 1: Define the grid environment

class GridEnvironment:

def \_\_init\_\_(self, rows, cols, dirt\_positions):

self.rows = rows

self.cols = cols

# Initialize grid; 0 = clean, 1 = dirt

self.grid = [[0 for \_ in range(cols)] for \_ in range(rows)]

for (x, y) in dirt\_positions:

self.grid[x][y] = 1

self.agent\_position = (0, 0) # Start at top-left corner

def get\_percept(self):

"""Return the percept: 1 if current cell has dirt, else 0."""

x, y = self.agent\_position

return self.grid[x][y]

def perform\_action(self, action):

"""Update the environment based on the agent's action."""

x, y = self.agent\_position

if action == "Suck":

self.grid[x][y] = 0 # Clean the current cell

elif action == "Move Up" and x > 0:

self.agent\_position = (x - 1, y)

elif action == "Move Down" and x < self.rows - 1:

self.agent\_position = (x + 1, y)

elif action == "Move Left" and y > 0:

self.agent\_position = (x, y - 1)

elif action == "Move Right" and y < self.cols - 1:

self.agent\_position = (x, y + 1)

def is\_clean(self):

"""Check if all cells in the grid are clean."""

return all(cell == 0 for row in self.grid for cell in row)

# Step 2: Define the Simple Reflex Agent

class SimpleReflexAgent:

def \_\_init\_\_(self):

pass # No state needed for a reflex agent

def get\_action(self, percept):

"""Return an action based on the current percept."""

if percept == 1:

return "Suck"

# If the current cell is clean, move to explore the environment

# Priority: Right > Down > Left > Up

return "Move Right"

# Step 3: Simulate the environment and agent

def simulate():

# Initialize a 4x4 grid with dirt at specific positions

dirt\_positions = [(0, 1), (1, 3), (2, 2), (3, 0)]

environment = GridEnvironment(rows=4, cols=4, dirt\_positions=dirt\_positions)

agent = SimpleReflexAgent()

steps = 0

while not environment.is\_clean():

percept = environment.get\_percept()

action = agent.get\_action(percept)

environment.perform\_action(action)

steps += 1

print(f"Step {steps}: Action={action}, Position={environment.agent\_position}")

print("\nAll cells are clean!")

# Step 4: Run the simulation

simulate()