1. Week 2 Lab Plan: Setting Up AIMA Python Environment and Agent Testing

1.1. Overview

This lab session will guide students through setting up the AIMA Python environment and working with intelligent agents using the provided `modelling_cs118.ipynb` notebook. Students will learn to run agent simulations and make modifications to understand how agents interact with their environments.

1.2. Learning Objectives

By the end of this lab, students will be able to:

- Set up the AIMA Python environment using conda
- Run Jupyter notebooks for AI agent simulations
- Understand basic agent-environment interactions
- Modify agent behaviors and environment parameters
- Experiment with 2D grid world environments

1.3. Prerequisites

- Basic Python knowledge
- Understanding of AI agent concepts from lectures
- Laptop with internet connection

1.4. Lab Setup Instructions

1.4.1. 1. Environment Setup (20 minutes)

1.4.1.1. Step 1: Install Git

Ensure Git is installed on your system:

Windows: Download and install from git-scm.com. Use the defaults for everything you are not sure about. If a beginner I recommend using the Github Desktop GUI for a simpler experience.

Linux (Debian/Ubuntu):

```
sudo apt update && sudo apt install git
```

MacOS:

```
brew install git # Requires Homebrew: https://brew.sh/
```

1.4.1.2. Step 2: Install Conda/Miniconda

Python installation and libraries will be handled by conda. Conda allows for the dependencies to be handled automatically.

- Install conda (Miniconda recommended)

Windows Notes:

- To access conda from anywhere you can add it to your path but this might cause collisions with other python installations. Alternatively you can open anaconda prompt powershell instead from the start menu.
- If you have a space in your windows username this can cause issues with installation. Instead use WSL or change your username.

1.4.1.3. Step 3: Clone the Repository

Using Git (CLI):

```
git clone https://github.com/abeljim/aima-python-eecs118-fall-25.git
cd aima-python-eecs118-fall-25
```

Using GitHub Desktop (Beginner Friendly):
- Open https://github.com/abeljim/aima-python-eecs118-fall-25 in your browser
- Click the green "Code" button
- Select "Open with GitHub Desktop"
- Choose a local path to clone the repo
- Click "Clone" — the repository will be downloaded to your machine

1.4.1.4. Step 4: Create and Activate Conda Environment

```
conda env create -f environment.yml
conda activate aima-python
```

1.4.1.5. Step 5: Fetch Datasets

You also need to fetch the datasets from the aima-data repository:

```
git submodule init
git submodule update
```

Wait for the datasets to download, it may take a while.

1.4.1.6. Step 6: Test Installation

Run the tests to ensure everything is working:

```
py.test
```

1.4.1.7. Step 7: Launch Jupyter Notebook

```
jupyter notebook
```

1.4.2. 2. Exploring the Agents Notebook

1.4.2.1. Understanding the Basic Components

Students will work through the `agents.ipynb` notebook to understand:

1. **Agent Class Structure**
   - Review the `Agent` base class
   - Understand agent properties: `alive`, `bump`, `holding`, `performance`, `program`
2. **Environment Class Structure**
   - Review the `Environment` base class
   - Understand key methods: `percept()`, `execute_action()`, `is_done()`
3. **Simple Agent Example - BlindDog**
   - Run the BlindDog simulation in 1D park
   - Observe how the agent moves and interacts with food/water
   - Understand the agent program logic
4. **2D Environment - Park2D**
   - Run the EnergeticBlindDog simulation
   - Observe 2D movement and visual representation
   - Understand direction handling and boundary detection

1.4.3. 3. Hands-On Exercises

1.4.3.1. Exercise 1: Modify Grid World Size

Objective: Change the park dimensions and observe behavior

Task: In the Park2D example, modify the park size from (5,5) to (8,8)

```
# Find this line in the notebook:
park = Park2D(5,5, color={'EnergeticBlindDog': (200,0,0), 'Water': (0, 200, 200),
'Food': (230, 115, 40)})

# Change to:
park = Park2D(8,8, color={'EnergeticBlindDog': (200,0,0), 'Water': (0, 200, 200),
'Food': (230, 115, 40)})
```

Questions for Students:
- How does the larger environment affect the dog's ability to find food and water?
- Does the random movement strategy become less efficient?

1.4.3.2. Exercise 2: Change Initial Agent Position

Objective: Experiment with different starting positions

Task: Modify the dog's starting position

```
# Find this line:
park.add_thing(dog, [0,0])

# Try different starting positions:
park.add_thing(dog, [4,4])  # Center of 8x8 grid
# or
park.add_thing(dog, [7,7])  # Corner position
```

Questions for Students:
- How does starting position affect the agent's performance?
- Which starting position seems most efficient for finding resources?

1.4.3.3. Exercise 3: Implement Barriers/Walls

Objective: Add obstacles to make the environment more challenging

Task: Create a new `Wall` class and add barriers to the environment

```
class Wall(Thing):
    pass

# Add walls to the environment
wall1 = Wall()
wall2 = Wall()
park.add_thing(wall1, [2,2])
park.add_thing(wall2, [3,3])

# Update the color dictionary
park = Park2D(8,8, color={
    'EnergeticBlindDog': (200,0,0),
    'Water': (0, 200, 200),
    'Food': (230, 115, 40),
    'Wall': (100, 100, 100)  # Gray color for walls
})
```