

# FIT5222 Laboratory #1: Installation and Setup

# Laboratory Classes

In this unit we use laboratory classes to complement your learning with to test your understanding of the lecture material.

Each week's lab will focus on **previous week's lecture**.

Each week we will release a lab worksheet. The worksheet includes:

- Content Questions: we will answer a set of open questions together
- Programming Exercises: we will implement and test different algorithmic strategies.

**Laboratories are marked in two ways** (2% total weekly marks):

- 1) Content questions are useful preparation for lecture and lab material quizzes** (1% per quiz)
- 2) Submission of programming exercises as a written report** (1% per worksheet)

# Environment

The installations on this page are **recommendations**. We will be able to help best with these tools, but feel free to use your code editor and virtual environment manager of your choice

- Language: Python 3
- Tools: Piglet, Flatland, Pacman capture the flat, ...
- Code Editor: You also can use any code editor you like. If you have no preference, we recommend using Visual Studio Code.
- Others: Command line interface, git version control.

## Hint for Windows users:

Select add to Path Variable (or similar term if have) during the installation of git and anaconda.

## Hint for Users have python but not using anaconda:

Install miniconda through:

<https://docs.conda.io/en/latest/miniconda.html>

1. Install python with anaconda:  
<https://docs.anaconda.com/anaconda/install/>  
(To avoid PATH issues, we recommend checking “add Anaconda to system’s PATH environment variable”)
2. Install Visual Studio Code:  
<https://code.visualstudio.com/download>
3. Install git:  
<https://www.atlassian.com/git/tutorials/install-git>  
(use Visual Studio Code as Git’s default editor)  
(Git for Windows includes Mintty as its default terminal for Git Bash. Installing Git for Windows will automatically install Mintty as well.)

# Required Skills

- Programming with python.
- Using git to manage codes.
- Using command line interface.
- Interacting with existing libraries

If you have never used Terminal and Git before, this reference is extremely useful!

- Python introduction:  
[https://www.w3schools.com/python/python\\_intro.asp](https://www.w3schools.com/python/python_intro.asp)
- Git introduction:  
<https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud>
- Linux/Mac CLI cheat sheet:  
<https://cheatography.com/davechild/cheat-sheets/linux-command-line/>
- Windows CMD cheat sheet:  
<http://www.cs.columbia.edu/~sedwards/classes/2015/1102-fall/Command%20Prompt%20Cheatsheet.pdf>
- The missing semester in computer science:  
<https://missing.csail.mit.edu/>

# Initial Setup (Using the Command Line)

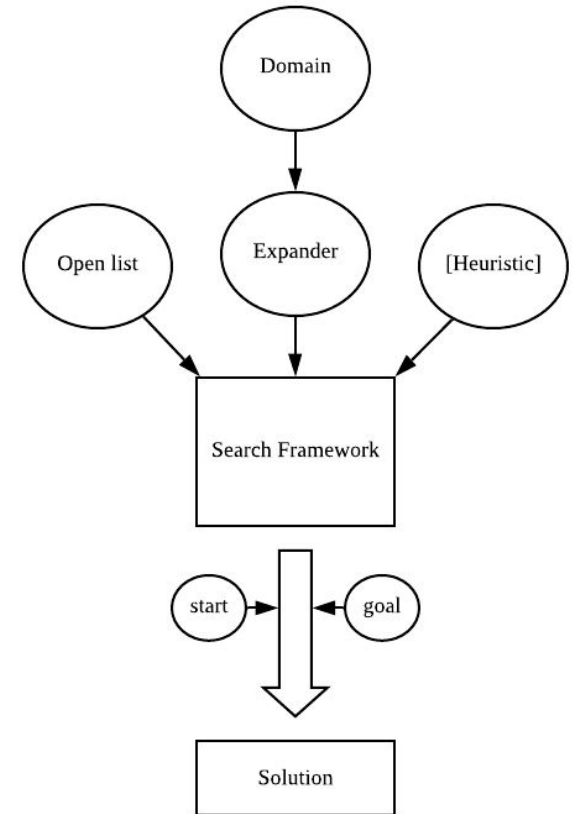
Skip this page if you are familiar with using the command line

- 1) Open Terminal (Unix) or Anaconda Powershell Prompt (Windows)
- 2) Type “ls” (Unix) or “dir” (Windows) to list the contents of the current directory
  - By default, you will be in your HOME directory. We want to navigate to whatever directory/folder you will want to keep your code for this class. We recommend creating a FIT5222 directory.
- 3) Navigate to your preferred FIT5222 directory using “cd path/to/folder” (use backslash for Windows)
  - E.g., “cd Documents/Uni”, then use “mkdir FIT5222” to create the folder, then move to it “cd FIT5222”

You should now have a FIT5222 folder ready to clone: Piglet, Flatland, and Pacman in this lab.

# Piglet

- Piglet is a search algorithm library developed for FIT5222.
- We will use this library for programming exercises.

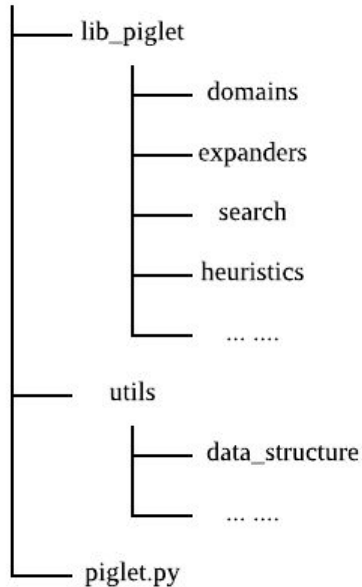


# Install Piglet

- Clone piglet to your machine:

```
git clone https://bitbucket.org/fit5222/piglet-public.git
```

## Piglet



1. **piglet.py** : The piglet command-line interface.
2. **domains**: This folder contains codes for describing and loading different types of problems.
3. **expanders**: Each type of problem has a corresponding expander which is responsible for generating new states based on available actions.
4. **search**: This folder contains a variety of search algorithm frameworks.
5. **heuristics**: This folder contains a variety of heuristics for informed search.
6. **data\_structure**: This file provides a variety of data structures, which include binary heap, stack, and queue.



# How to use Piglet

- Piglet is used through the command-line interface. It directly solve problems.
- By importing modules from piglet, you can use it as a library and build your own algorithm and program.
- We will remove parts of Piglet and let you implement them.

# Piglet Command-line

-h help  
-p /Path/to/scenario\_file  
-f graph/tree/iterative  
-s uniform/depth/breadth  
-t 30 (Time limit)  
--solution (Print solution)

Example:

```
python3 piglet.py -h
```

```
python3 piglet.py -p ./example/example_8_puzzle.scen -f graph -s uniform -t 30
```

## Troubleshooting tip:

- Error: some version of 'python3' is not recognized as an internal or external command, operable program or batch file.
- If you have already installed Python, your machine may have python3 as default, common on newer Windows machines. Instead, replace `python3` with `python` for all commands
- You can run `python --version` to verify it points to some version of Python 3

Make sure you are running this command from the directory in which [piglet.py](#) has been installed

# How to add Python Path to windows

<https://realpython.com/add-python-to-path/#how-to-add-python-to-path-on-windows>

1. Locate the directory in which your target Python executable lives.
2. Open the Start menu and search for the *Edit the system environment variables* entry, which opens up a *System Properties* window. In the Advanced tab, click on the button *Environment Variables*. There you'll see User and System variables, which you'll be able to edit
3. In the section entitled *User Variables*, double-click on the entry that says *Path*. Another window will pop up showing a list of paths. Click the *New* button and paste the path to your Python executable there. Once that's inserted, select your newly added path and click the *Move Up* button until it's at the top.

Also make sure you have only one Python executable chosen in Settings->Apps->Advanced app settings->App execution aliases

# Piglet Library

Now, we will guide you walk through some important files in Piglet.

# Flatland



Flatland is an open-source toolkit for developing Multi-Agent Pathfinding algorithms in grid rail worlds. It provides a two-dimensional grid environment in which many agents can be placed, and each agent must solve one or more navigational tasks in the grid world. The library was developed by SBB, Alcrowd, and numerous contributors and Alcrowd research fellows from the Alcrowd community.

In this unit, we will run a Flatland challenge for FIT5222 and you will implement algorithms you learn during the unit to win the competition.

# Flatland

## Python Virtual Environment

- Using a python virtual environment can help us better manage python packages and different versions of python. The packages you installed in one virtual environment won't influence other python projects.
- If you are not using Anaconda and have the python environment already installed on your machine, install miniconda: <https://docs.conda.io/en/latest/miniconda.html>

# How to Install Flatland

1. Clone [Flatland source code](https://bitbucket.org/fit5222/flatland.git) to your machine with:

```
git clone https://bitbucket.org/fit5222/flatland.git
```

2. If you are windows, open git bash or anaconda command prompt or CMD or powershell. If you are using unix/mac os, open terminal.

3. Go to the 'flatland' folder (the root folder of the source code):

```
cd flatland
```

4. **If you are** an arm-64 mac users (e.g. macbook with m1/m2/m3 chips):

```
conda config --env --set subdir osx-64
```

5. Type following commands to create a virtual environment:

```
conda create -n flatland-r1 python=3.8
```

and activate it:

```
conda activate flatland-r1
```

6. Install flatland (this step may take a long time for the first time installation.):

```
python setup.py install
```

7. Test installation:

```
flatland-demo
```

or navigate a train using the arrow keys:

```
python examples/keyboard.py
```

8. Exit virtual environment (If you want):

```
conda deactivate
```

## Hint for Mac users:

If an OpenGL framework error occurs running flatland-demo, try perform following operations:

1. Open a new tab/window in Terminal
2. Go to ~/.matplotlib folder by:  

```
cd ~/.matplotlib
```
3. Create a new file named matplotlibrc by  

```
nano matplotlibrc
```
4. Type:  

```
backend: TkAgg
```
5. Press control + x to exit, and press y to confirm.
6. Press enter to confirm write to matplotlibrc .
7. Return to previous tab/window

## Additional Troubleshooting:

- If python is an unrecognised command, try using python3 instead. Verify that Python points to a version of Python3 using 

```
python --version
```
1. Inside your flatland directory, open *requirements\_dev.txt* to view each required package and version
  2. Make sure your versions are up to date by:  

```
pip show [package_name]
```

    - a. (Perhaps try msgpack & pytest first)

# Windows note:

IF you see this message after step 6, please give it time:  
the process can often take a few minutes to complete.

```
!!
Error: setuptools.installer and fetch_build_eggs are deprecated.
```

```
!!
```

```
*****
```

```
Requirements should be satisfied by a PEP 517 installer.
```

```
If you are using pip, you can try `pip install --use-pep517`.
```

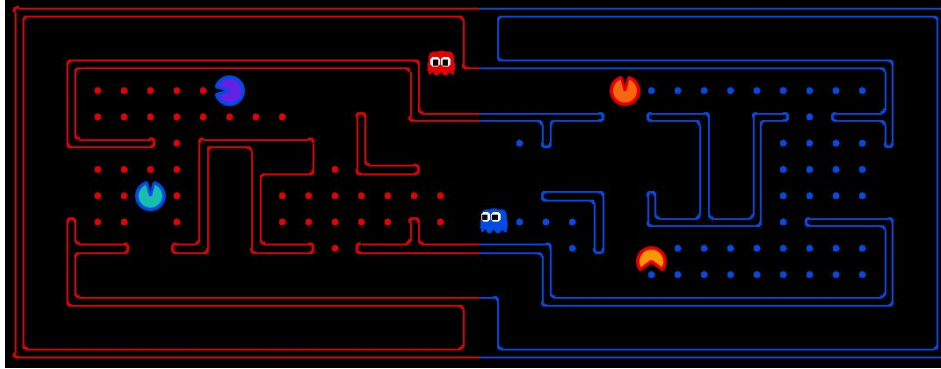
```
*****
```

```
!!
```

```
dist.fetch_build_eggs(dist.setup_requires)
```



# Pacman: Capture The Flag



This environment will be relevant for the second assessment.

Pacman Capture the Flag is developed by UC Berkeley CS188. It is a multiplayer capture-the-flag variant of Pacman. It consists of two teams, blue and red, which compete with each other. Each team needs to eat the food on the far side of the map while defending their own food.

We will hold a Pacman Capture the Flag competition at the end of this semester.

# How to Install Pacman

1. Clone the source code of capture the flag:

```
git clone https://bitbucket.org/fit5222/pacman-public.git
```

1. Go to the root folder of pacman library.
2. You could activate your virtual environment for flatland before running pacman.
3. Run example:

```
python3 capture.py
```

Play with your keyboard:

```
python3 capture.py --keys0
```

Specify red team and blue team:

```
python3 capture.py -r berkeleyTeam.py -b berkeleyTeam.py
```

# Homework

Refer to the Laboratory #1 worksheet and complete the specified exercises. You must submit your answers to Moodle **by next week**.