

# 24304 Artificial Intelligence

## *Universitat Pompeu Fabra*

### Lab projects

#### Introduction

We'll be using Pacman (and maybe other games we've created) for the lab projects. The Pacman projects were created by University of California, Berkeley and are a great example of gamification.

Everything will be developed in python 2.7 and corrected in Linux.

For every project, you'll be able to modify some functions and parts of the code. These are explicitly indicated, any other file or given function/code cannot be modified.

Make sure to use only the allowed information e.g. you can't access RAM information if you've been told to work only with screen output (pixels).

#### Submissions

- I. Projects will be done in groups of 2.
- II. Every submission will be a compressed file .zip:
  - a) Name of the file: Px\_NIA1\_NIA2.zip (where x is the number of the assignment; 1,2 or 3)
  - b) Include all .py **required** files. Only the ones you have to modify!
  - c) Also a Px\_NIA1\_NIA2.pdf containing the names of each participant and the answers to:
    - i. What works?
    - ii. What doesn't work?
    - iii. Which problems did you have doing the exercise?
    - iv. Specific questions for each assignment.

#### Evaluation

- Submissions every other session. 6 lab sessions in total, 3 submissions.
- Private interviews with the members of the team. To make sure you understand what your agent is doing, and how the algorithm works.
- I'll look at the grader and the code as well. Make sure to use meaningful data structures and efficient implementations.
- **Plagiarism is not allowed**, don't copy answers from the internet and **don't share your code with other teams**. You can discuss with your classmates, but don't share answers. Please, don't share your answers through the internet as well, be respectful with UC Berkeley Pacman project creators will.
- Non-working code will not be corrected (graded as a 0!!), make sure it works in your own computer before submitting it.

## Extra

For **windows users**: although not necessary, you may want to use WSL (windows subsystem for linux): enable required options, then install ubuntu app. Use Xming X server or similar for the screen renders to work...

**Code editors**: emacs, sublime, visual studio, notepad ++... Make sure it follows python syntax if you use a different one.

**Python review** (recommendations from last year's course, but there are many other good ones):

<https://wiki.python.org/moin/BeginnersGuide/Programmers>

<https://www.afterhoursprogramming.com/tutorial/coldfusion/introduction-coldfusion/>

### Python hints

- Be careful with spaces and indentation.
- `a = b` ties both variables, it's not a copy but a reference to the same object. Copy using `a=b[:]` or `copy`, `deepcopy` commands.
- Use `virtualenv` or the like to create personalized python + packages installs. We'll be using different configurations for the seminar's programming assignments.

### Design hints

- Imagine how everything should work before starting to program. Having a clearer idea will help making a better implementation.
- Keep trying and using the autograder to see how your code is doing.