

# Overview of git

# What is git for?

- store the history of your code (version control)
- work in parallel with others

# Basic Concepts

(git vocab)

these are git commands

# Repository (repo)

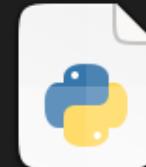
- A folder with all your code
- metadata (history)



.git



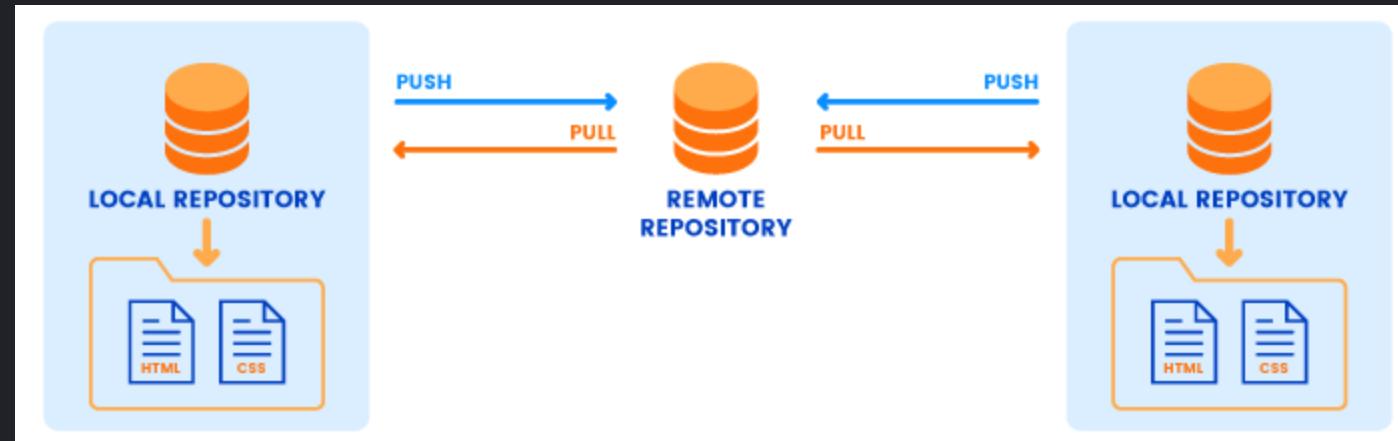
README.md



the-code.py

# Local vs Remote

- Local: code on your computer
- Remote: code on (usually) Github
- clone download repo remote -> local



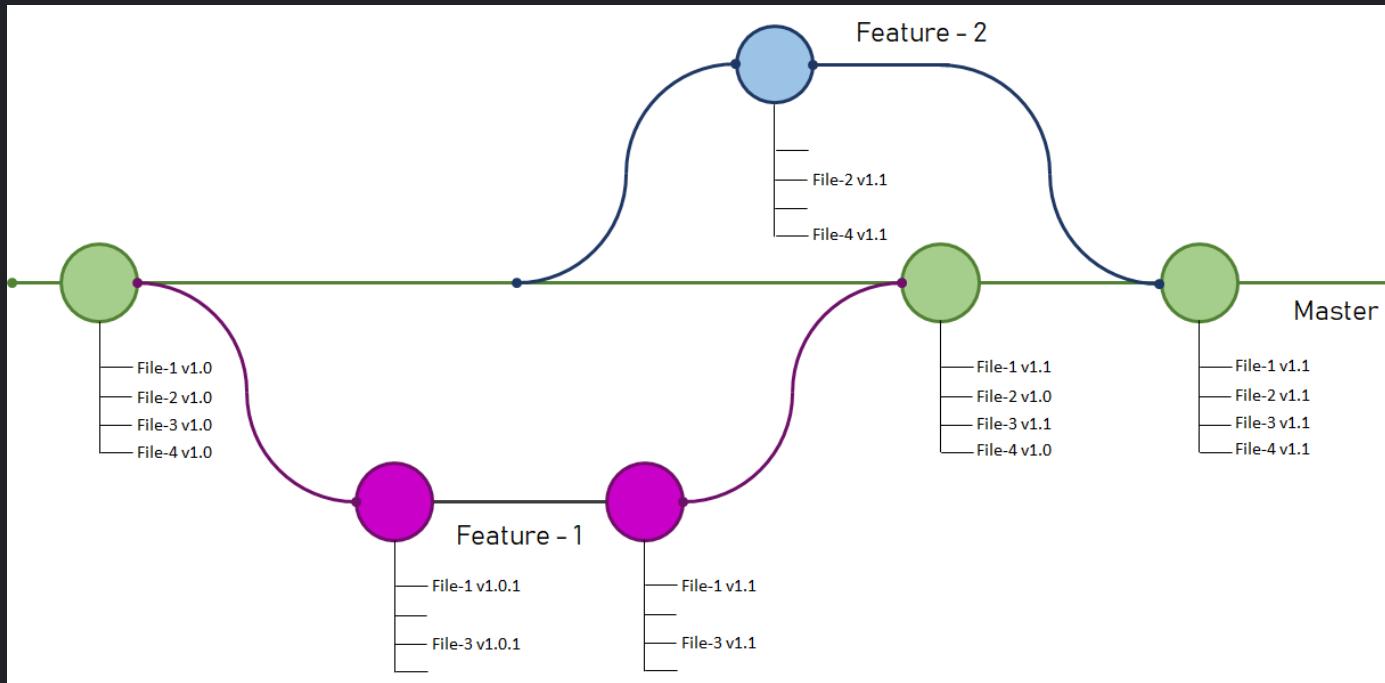
<https://rubygarage.org/blog/most-basic-git-commands-with-examples>

# Commit

- A set of changes made to the code
- Who made those changes
- Short description

# Branches

- Branches contain sequences of commits
- Work in parallel, then **merge** branches



# Basic git workflow

0. Get latest code ( `clone` / `pull` )
1. Make local changes
2. Choose which changes to share ( `stage` / `add` )
3. Add a description of the changes ( `commit` )
4. Send the changes to everyone ( `push` )

# A pragmatic git walkthrough

- Using VSCode + Github
- Use the terminal if you prefer
- You'll find *lots* of answers online

# You probably already have git

```
# check if it's installed  
git --version
```

Otherwise: <https://git-scm.com/downloads>

# 1. Create a repo on github

**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

*Required fields are marked with an asterisk (\*).*

**Repository template**

No template ▾

Start your repository with a template repository's contents.

---

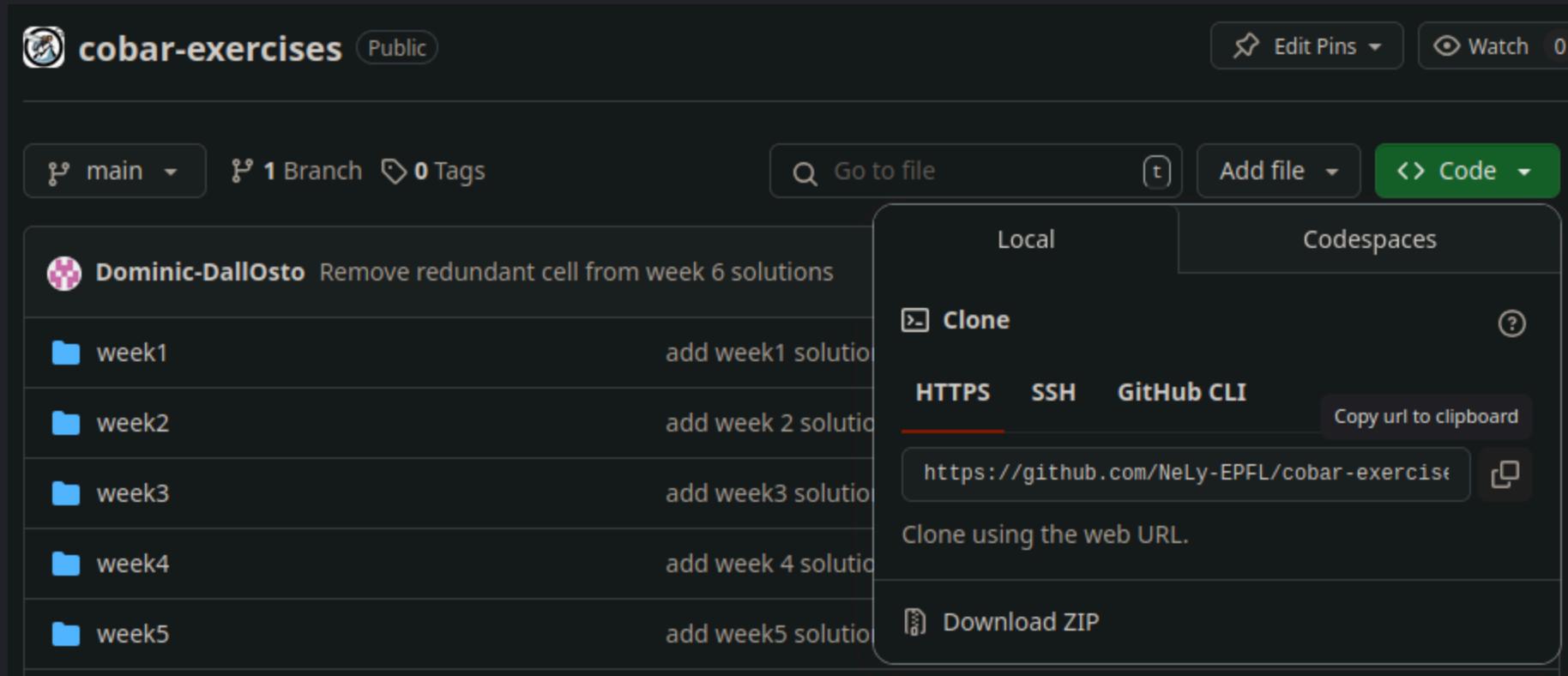
**Owner \***  Dominic-DallOsto /

Great repository names are short and memorable. Need inspiration? How about [supreme-broccoli](#) ?

**Description (optional)**

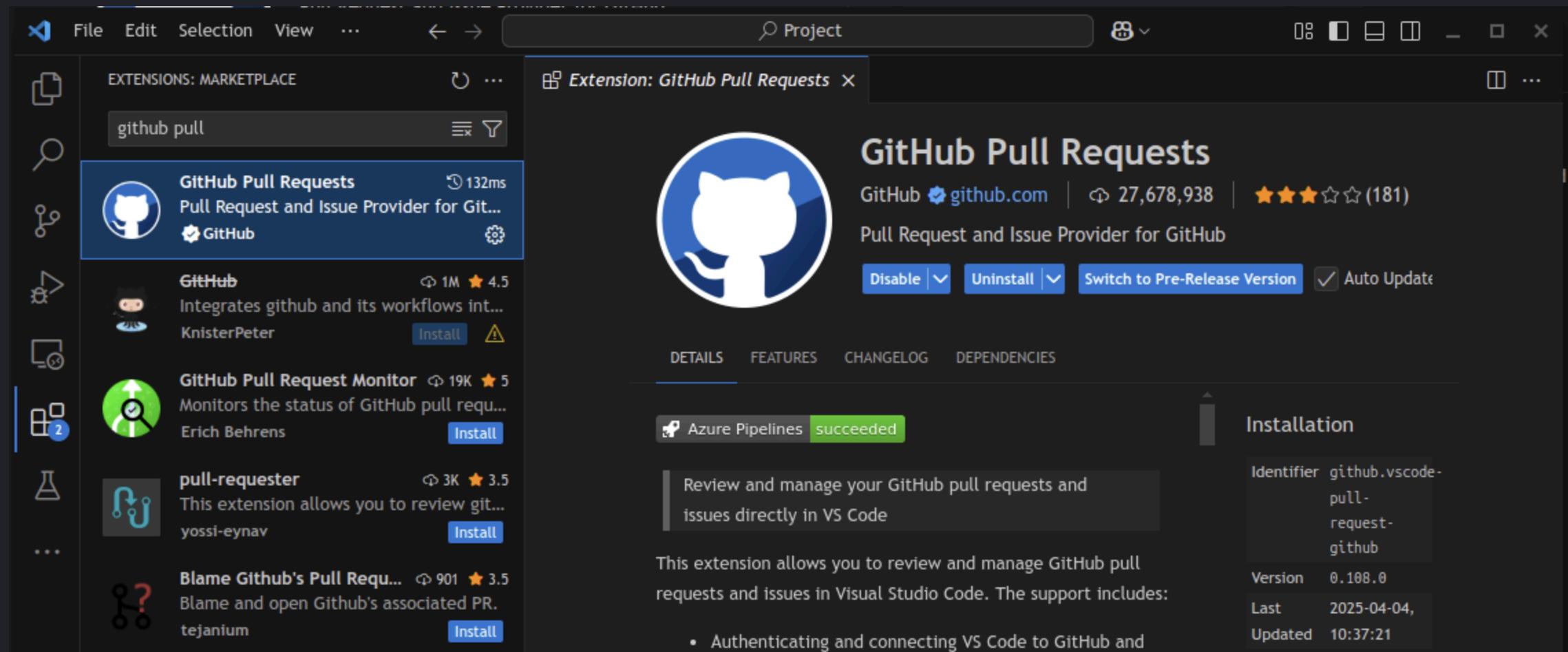
<https://github.com/new>

## 2. Clone it locally

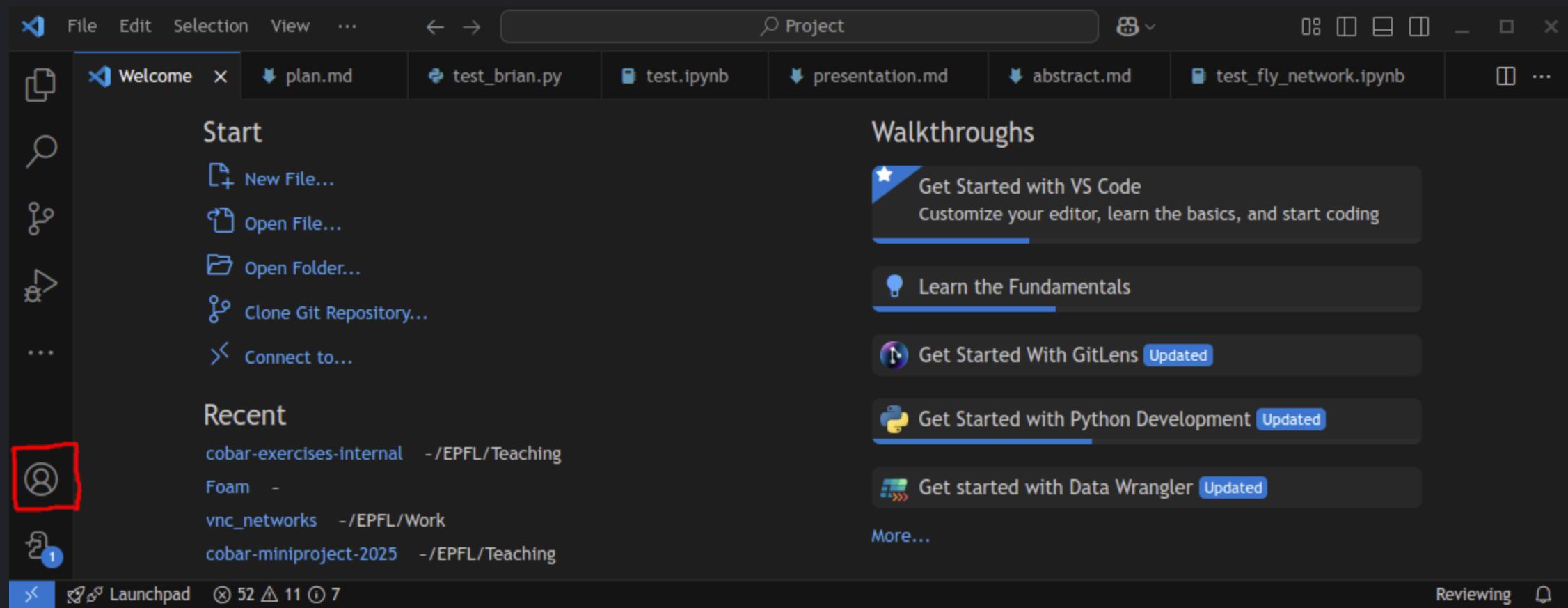


```
git clone https://github.com/NeLy-EPFL/cobar-exercises
```

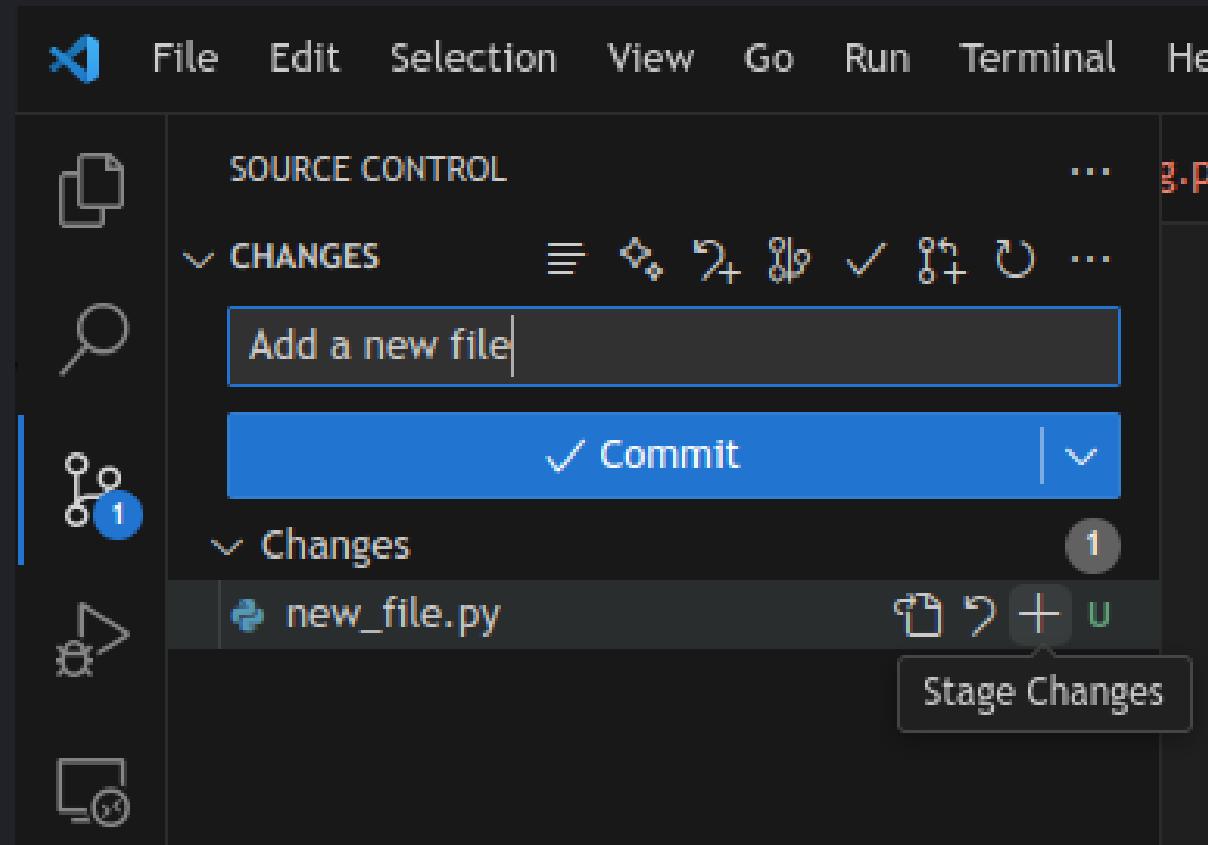
### 3. Open VSCode to that folder, install the github extension



# 4. Sign into github in VSCode



5. Add a new file to your repo, stage (+), add a message, then commit, push



If you get an error

```
git config --global user.name "Your name"  
git config --global user.email "your.email@mail.com"
```

# Other things to do

- Create a new branch (Ctrl+Shift+P > Create branch)
- make a change
- commit
- merge into main (Ctrl+Shift+P > Merge...)

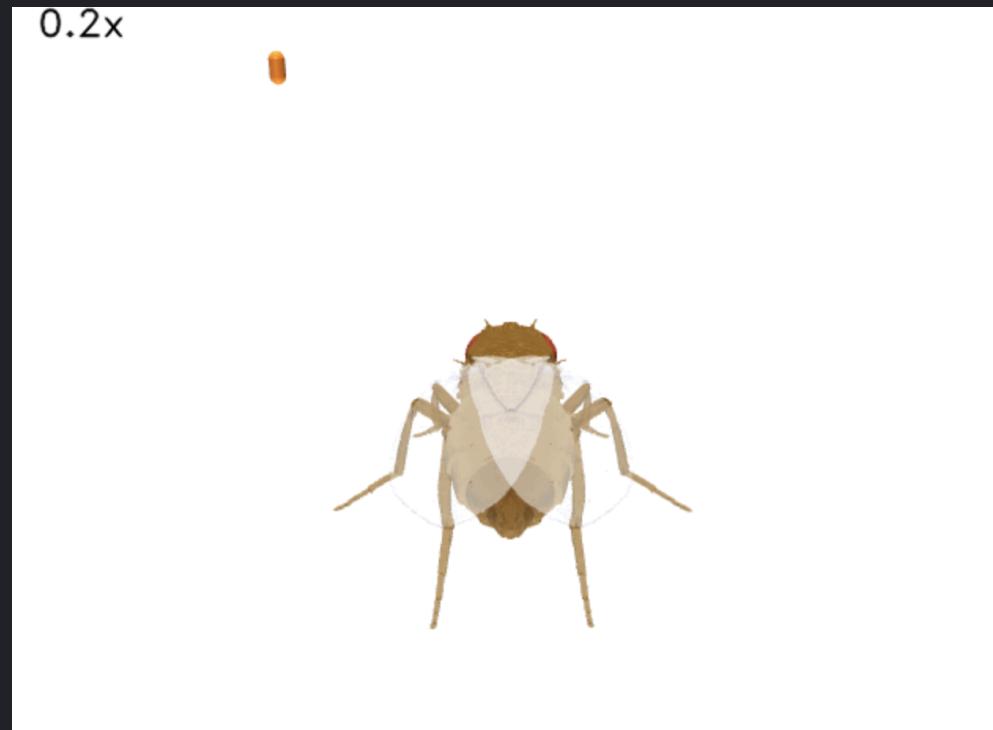
Why is this useful for you?

- Work in parallel on different levels, then combine
- Reset the code to an old (working) version
- Everyone uses it

# Project presentation

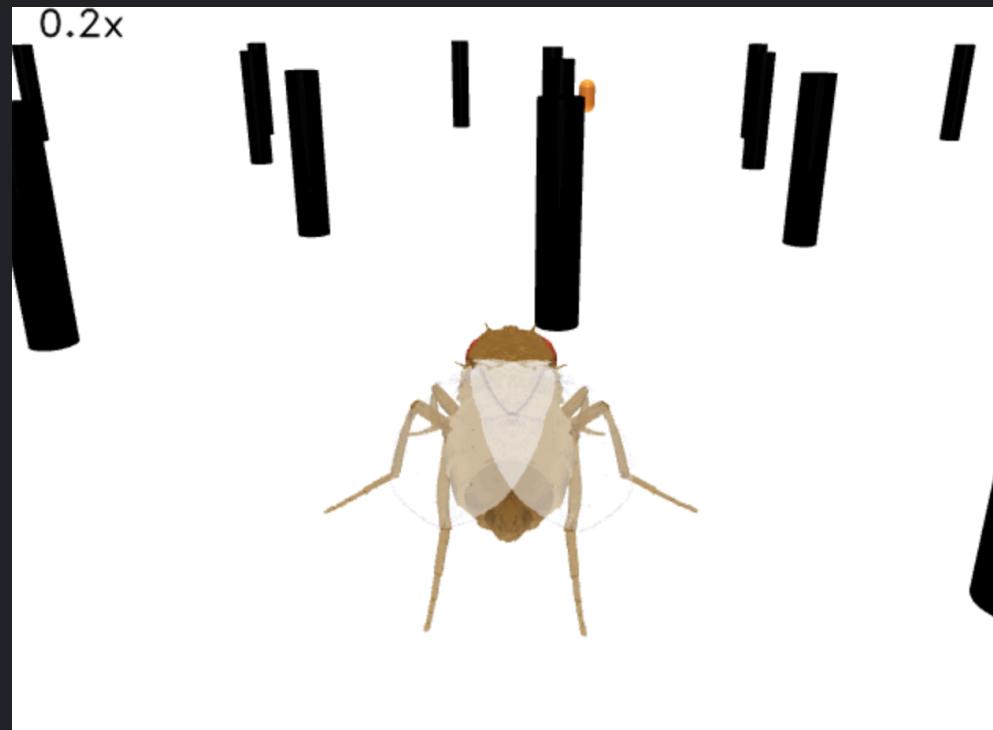
- 5 levels
- Using multiple senses

# Level 1 - Empty Odour Arena



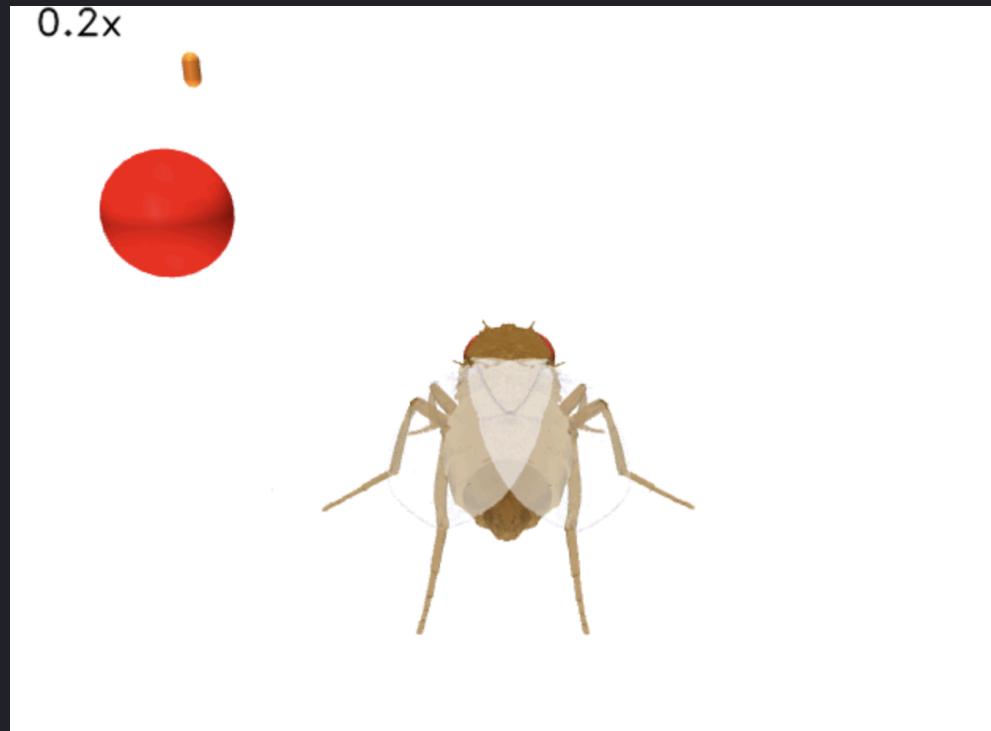
Goal: get to the odour in an empty environment

# Level 2 - Scattered Pillars



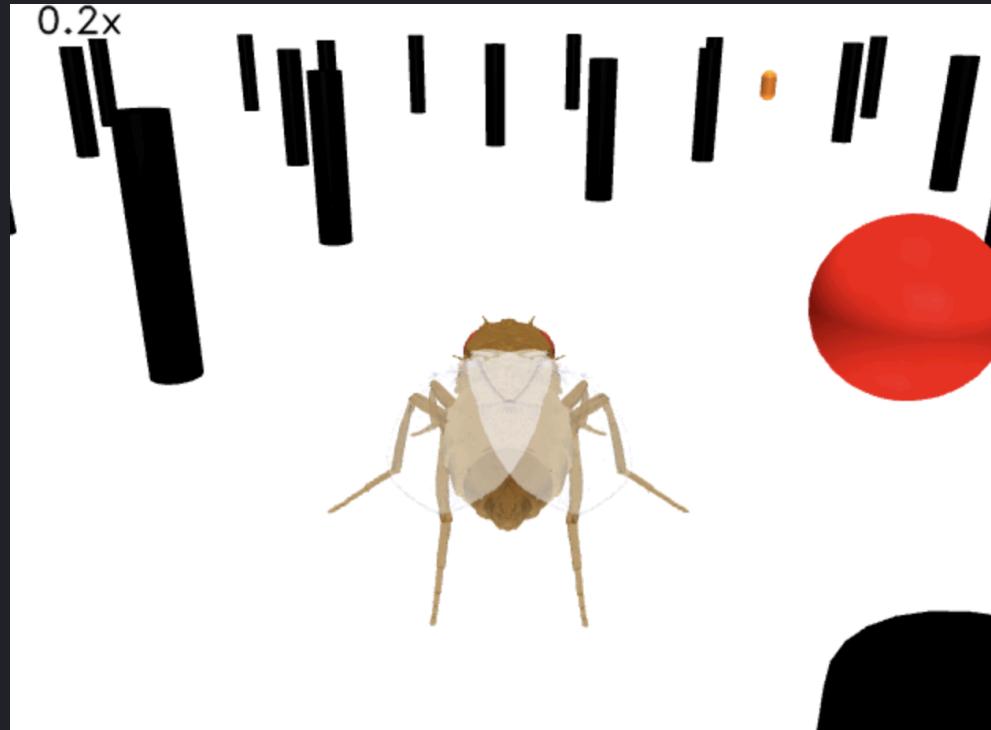
Goal: get to the odour while avoiding the pillars

# Level 3 - A threat approaches



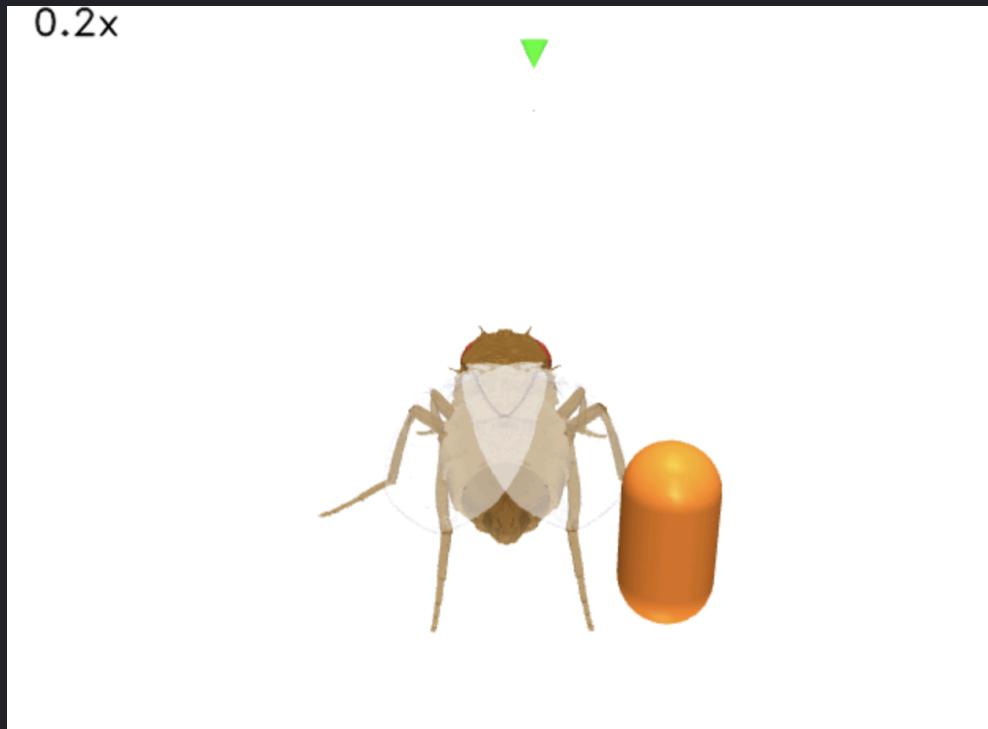
Goal: get to the odour while avoiding the balls coming towards the fly

# Level 4 - Pillars + threats



Goal: get to the odour, avoiding both the pillars and attacking balls

# Level 5 - Return to Nest



Goal: same as level 4, but return to the fly's spawn point after collecting the odour

# Assessment guidelines

- Performance
- Report
- Bonuses (fly-like implementation, presentation)

More specific details to come!

# Goal of today

- Explore the environments
- Discuss strategies
- Organise the work

# Demo time!