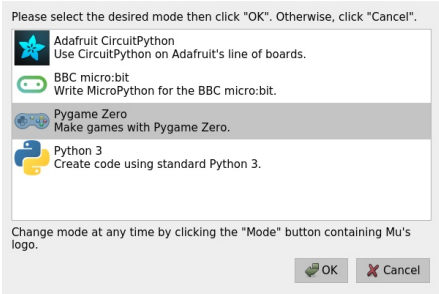
Your first program

When you start the Mu-Editor, you can pick a mode: choose "PyGame Zero".



If you have already chosen a different mode, click on the "Mode" button in the toolbar to get it again.

In the editor's window, you can remove the placeholder

```
1  # Write code here :-)
```

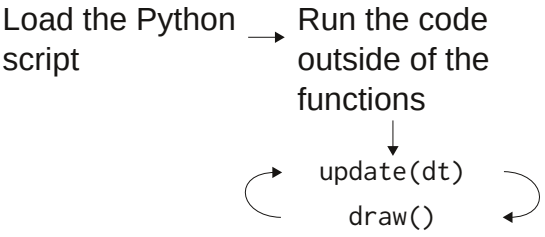
Type your first code:

```
1  WIDTH = 640
2  HEIGHT = 480
3
4  def update(dt):
5      pass
6
7  def draw():
8      screen.fill((128, 0, 0))
```

Click on the "Play" button: save the file as `main.py` in a new directory named after your project.
As soon as you have clicked on "Save", you should see a dark red window popping up: it's your program running!

You can close it by clicking on the stop button in the Mu-editor toolbar.

What does the code you just wrote?




```
1  WIDTH = 640
2  HEIGHT = 480
```

The two variable `WIDTH` and `HEIGHT` are defined outside of all functions: they are global variables that can be accessed from everywhere.
The two variables contain the width and height of the window and are"magically" used by PyGame to set the size of the game.

```
4  def update(dt):
5      pass
```

The `def` keyword starts the definition of a function. A function is a set of actions wrapped together. The `update(dt)` function

has the name `update` and receives the argument `dt`. It gets called very often by PyGame Zero: you will write in there the code that modifies the state of your game. Currently, our game does "nothing", so we just write there the void command `pass` (because in Python a function cannot be left empty).

```
7  def draw():
8      screen.fill((128, 0, 0))
```

The `draw` function is run just after the `update(dt)` function. In this first program, it simply calls the `fill()` function from the PyGame Zero module called `screen` to set the background of the window to a dark red: the argument we are passing to the `screen.fill` function is a tuple with the red, green and blue composant of the color.