Getting Started with the Project

Here are some additional clarifications about the game and the code that might help you find your way around the code that has been provided.

About the game

Some quick clarifications about some potentially ambiguous points:

- When your robot finds a plant, the "status" of the plant can be either **nutritious**, **poisonous**, or **unknown.** In either case, you can ask for an image (or in fact as many as you want) of the plant, which will cost you 1 unit of life.
- There is some uncertainty in your movements, in the sense that if you decide to move in a certain direction, you have some probability of going in another direction. That being said, you can always ask for your exact position (free of charge!).
- If your robot takes more than 1 second to make its move, then the game will pick a random direction for it, and if it happen to be in a cell with a plant, it will **not** make it eat the plant.

About the code

The minimal skeleton for your code is as follows:

```
import time
import random
import game_interface

def get_move(view):

  # If there is a plant in this location, then try and eat it.
  if view.GetPlantInfo() == game_interface.STATUS_NUTRITIOUS_PLANT:
     eat = True
  elif view.GetPlantInfo() == game_interface.STATUS_UNKNOWN_PLANT:
     eat = random.choice([True,False])
  else:
     eat = False

  # Choose a random direction
  direction = random.choice([game_interface.UP,
                             game_interface.DOWN,
                             game_interface.LEFT,
                             game_interface.DOWN])
  time.sleep(0.1)
  return (direction, eat)
```

This simple agent does two things: - he checks if there is a nutritious plant on his cell. If yes, he eats it. If the status of the plant is unknown, he flips a coin to decide what to do. If it is poisonous, he doesn't eat it. - he then picks a direction at random, and goes with it.

From this example, you can see the 3 main components you interact with, that is the **game_interface** module, the **view** object, and your **return values**. The following sections give more informations on each element.

The game_interface module

This module mainly gives aliases to the values encountered in the game. You will only have to worry about the following:

- game_interface.UP, game_interface.DOWN, game_interface.LEFT, game_interface.RIGHT, which are the possible directions you can take.
- game_interface.STATUS_NUTRITIOUS_PLANT, game_interface.STATUS_POISONOUS_PLANT, game_interface.STATUS_UNKNOWN_PLANT, game_interface.STATUS_NO_PLANT.

So if you want to check wether there is a plant in your cell, you would use:

```
if view.GetPlantInfo() != game_interface.STATUS_NO_PLANT:
   ... do something ...
```

if you want to treat the specific case where the status is unknown, you would do:

```
if view.GetPlantInfo() != game_interface.STATUS_UNKNOWN_PLANT:
   ... do something ...
```

NOTE: Asking for the status of a plant doesn't cost you anything.

The view object

The view object is passed by the game to the **get_move** function that you are implementing. It contains the informations you have access to at each round:

- view.GetImage() : if there is a plant on the cell, returns a list of length 6x6 = 36, containing zeros and ones. **This operation will cost you 1 unit of life.** If there is no plant on the cell, it will return an empty list, and that won't cost you anything.
- view.GetLife() : this returns an integer representing the amount of life units you have remaining.
- view.GetRound() : this returns an integer representing the round being played
- view.GetPlantInfo() : this returns an integer in [0,1,2,3] representing the status of the plant. You don't have to manipulate the integers directly, but should rather use the game_interface.STATUS_{NO,UNKNOWN,POISONOUS,NUTRITIOUS}_Plant variables. Note that behind the hoods, these variables are simply mapped to integer in [0,1,2,3]

- view.GetXPos() : the position of your robot on the x axis
- view.GetYPos() : the position of your robot on the y axis

What you should return

Your function **get_move** should return a tuple (direction, eat), where: - direction is one of the four directions defined in game_interface (see above) - eat is a boolean value: True if you want to eat the plant, False if there is no plant, or if you don't want to eat the plant in your cell