0HV120 2022 Assignment 2

The context of assignment 2 is robotics. Please refer to the slides of dr. R. Cuijpers on Canvas for an introducton in this field.

Preparation

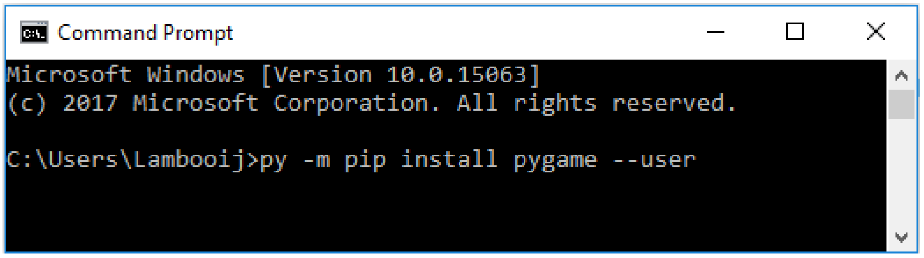
Download the following starting files into one directory:

name	comment
assignment2.py	This file contains the code that you will submit in Canvas
create_level.py	This file allows you to define custom start positions
main.py	Start this file (in IDLE) to start the robot world, do not change
definitions.py	helper file, do not change
objects.py	helper file, do not change
robot_world.py	helper file, do not change
robot.py	helper file, do not change

Installing Pygame in Windows 10

Open a Command Prompt window (press the Windows button and type 'cmd' followed by <enter>) and type on the command line:

```
py -m pip install pygame --user
```



```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Lambooij>py -m pip install pygame --user
```

You can see an example in figure 1.

If Windows 10 does not recognize the 'py' command or 'pip' reports SSL errors try the procedure in Appendix 1 or 2.

Installing Pygame on an Apple

Instructions for installing pygame on OSX can be found here:

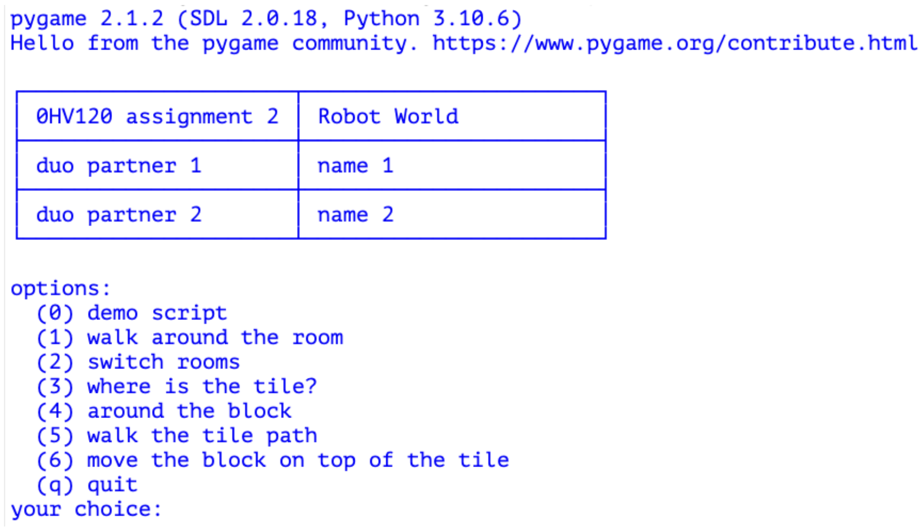
https://www.pygame.org/wiki/MacCompile

Run the starter code

Once the Pygame install has succesfully completed,

open file main.py (in IDLE) and start it.

You should see a menu similar to this:



```
pygame 2.1.2 (SDL 2.0.18, Python 3.10.6)
Hello from the pygame community. https://www.pygame.org/contribute.html

 ┌────────────────────┬───────────────────────┐
 │ 0HV120 assignment 2 │ Robot World           │
 ├────────────────────┼───────────────────────┤
 │ duo partner 1       │ name 1                │
 ├────────────────────┼───────────────────────┤
 │ duo partner 2       │ name 2                │
 └────────────────────┴───────────────────────┘

options:
  (0) demo script
  (1) walk around the room
  (2) switch rooms
  (3) where is the tile?
  (4) around the block
  (5) walk the tile path
  (6) move the block on top of the tile
  (q) quit
your choice:
```

Figure 1: Output of starter code of assignment2.

If you see the output above, you are ready to start assignment 2.

Submission rules

- Solutions must be submitted as a duo, so choose someone to collaborate with.
- Your solution must be submitted in the Canvas course of 0HV120 in a single file called **assignment2.py**.
- Solutions in any other form will not be accepted and graded with mark 1.

General guidelines

- Your code should be runnable, e.g. if you could not implement a part, it should give a message and not crash.
- Your code should be readable (clear names for variables, comments if necessary)
- Your code should be yours (The TU/e has a Canvas plug-in for detection of plagiarism).

Grading

In programming it is hard to give absolute rules for grading, e.g. code can be working but unreadable, or working only in most cases, etc. For this assignment we will have the following rough grading guideline:

- If you complete parts 1 - 4 with working (and readable) code you can obtain a sufficient mark.
- Parts 5 - 6 can improve your mark, but they cannot fully compensate for parts 1 - 4 being insufficient.
- Well-written code improves your mark. Examples:
  - Good naming of variables
  - Good use of the DRY principle (“don’t repeat yourself”)
  - Good use of self defined functions (a.k.a. “helper functions”).

Good luck!

The Robot World

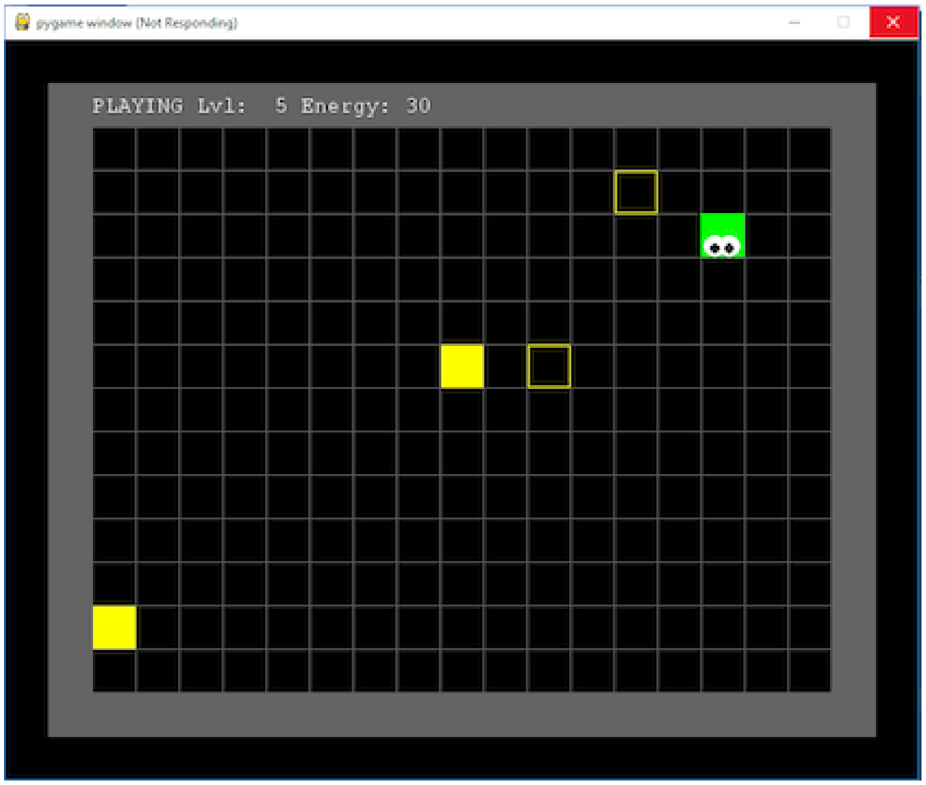


Figure 2: Example of the robot world.

The assignment parts takes place in an imaginary robot world like the one in figure 2. The worlds have the following rules:

- Each part starts with a new world with dimensions *robot.width* and *robot.height*.
- Each world has different starting conditions (you can think of different dimensions, starting positions of objects, etc.).
- Each world is bounded by a solid outer wall and sometimes inner walls.
- There is one robot in the world.
- The are 0 or more of the following objects: ‘Wall’, ‘Block’ and ‘Tile’.
- The robot can touch walls and blocks but can not walk into/over them.
- The robot can grab a block, but only when it faces (and touches) the block.
- The robot can push/drag a block forward/backward (after grabbing).
- The robot can walk over tiles (no influence on robot movement).
- The robot cannot move from its position when the energy is down to 0.

The robot has certain available actions and available scans. They are all listed in the two tables below. Both actions and scans are methods of object robot. None of the methods have any parameters. As an example: to let the robot take one step forward in its current direction you need this in your program: `robot.step_forward()`

Table 1. Actions of the robot

action	robot action
step_forward	take one step forward in the current direction, if no wall or block is in the way
step_back	take one step backward from the current direction, if no wall or block is in the way
turn_right	turns right (e.g. ‘UP’ -> ‘RIGHT’), if it has not grabbed a block
turn_left	turns left (e.g. ‘UP’ -> ‘LEFT’), if it has not grabbed a block
grab_release_block	grabs or releases a block faced by the robot

Table 2. Scans of the robot

scan	return value
scan_direction	current direction (‘UP’, ‘DOWN’, ‘LEFT’ or ‘RIGHT’)
scan_energy	remaining energy before robot cannot move anymore
scan_object_ahead	object first encountered with repeated step_forward (can be a ‘Wall’, ‘Tile’ or ‘Block’)
scan_steps_ahead	number of steps that robot can take before touching an object (‘Wall’, ‘Tile’, or ‘Block’

Assignment

Duo names: Replace the names name1 and name2 in function show_duo_names() with your own names.

0. Demo Level (no submission)

Select option '0'. The given menu code will create a new world with a robot in it at a random position.

After selecting the world window and pressing a key function `script_0_demo()` is called.

- The code in `script_0_demo()` has been given to demonstrate the effects of calling the robot methods.
- Run the demo and change the code of function `script_0_demo` as you like. This part will be NOT be looked at in grading.
- Examine the possibilities until you feel confident to start on part 1 - 6.
- The `robot.xxxxx_yyyyy()` methods are enough to solve all parts of assignment 2.
- The window can be closed with the mouse.

1. Walk around the World

Select option '1'. The given code will create a world of varying size with a robot in it at a random position.

Click on the world window and press a key, function `script_1()` is now called. Write your commands `robot.xxxx_yyyyyy()` etc. in `script_1()` such that:

- the robot first goes to one of the corners.
- the robot next walks around the World along the outer wall and stops in the corner where it started.
- the robot does not crash into a wall

When the robot is done print the number of steps it took to walk 'around the world'. Here is an example of the print-out for one run:

```
It takes 48 steps to walk around the world.
```

You may assume that the world will not contain any blocks or tiles.

2. Switch Rooms

Select option ‘2’. The given code will create a world (of varying size) with a vertical wall and a robot at a random position (so it can start left or right).

Click on the world window and press a key, function `script_2()` is now called.

Write your commands `robot.xxxx_yyyyyy()` etc. in `script_2()` such that:

- the robot walks from the room where it was created to the other room.
- the robot has to stop moving/turning in the other room (the exact location is not important).
- the robot does not crash into a wall

You may assume that the world does not contain any blocks or tiles.

3. Where is the Tile?

Select option ‘3’. The given code will create a world (of varying size) with tile and a robot both at a random positions.

Click on the world window and press a key, function `script_3()` is now called.

Write your commands `robot.xxxx_yyyyyy()` etc. in `script_3()` such that:

- the robot finds the tile and ends up on top of the tile.
- the robot does not crash into a wall

Print the (x, y) coordinates of the tile. Here is an example print-out:

```
    The tile is at position (x, y) = (6, 4)
```

You may assume that the world does not contain any blocks.

4. Walk around the Block

Select option ‘4’. The given code will create an empty world (of varying size) with a robot and one.

Click on the world window and press a key, function `script_4()` is now called.

Write your commands `robot.xxxx_yyyyyy()` etc. in `script_4()` such that:

- the robot first walks to the blue block until they touch.
- the robot then walks around the block in clock-wise direction (while touching the block) completing exactly one full round.
- the robot does not crash into a wall or the block

You may assume that the block does not touch any of the walls.

5. Follow the Tile Path

Select option ‘5’. The given code will create an empty world (of varying size) with a ‘path’ of yellow tiles from left to right. A robot is also created at the left-most tile of the path.

Click on the world window and press a key, function `script_5()` is now called. Write your commands `robot.xxxx_yyyyyy()` etc. in `script_5()` such that:

- the robot walks over the yellow tiles until the robot touches the right wall.
- the robot does not leave the path
- the robot does not crash into a wall

6. (challenge) Move the block on top of the Tile

Select option ‘6’. The given code will create an empty world (of varying size) with a robot, a block and a tile.

Click on the world window and press a key, function `script_6()` is now called. Write your commands `robot.xxxx_yyyyyy()` etc. in `script_6()` such that:

- the robot finds the block and pushes/drags it until the block is over the tile
- the robot does not crash into a wall or a block

*** * * * * END OF ASSIGNMENT 2 * * * * ***

Appendix 1: Adding Python to your environment variables (in case Windows 10 does not recognize the ‘py’ command)

The location of your Python 3.10 must be a part of the environment variables before you can install Pygame. If you have trouble running the command line please re-run the Python/IDLE installer/setup (see the installation guide of the 0HV120 exercises):

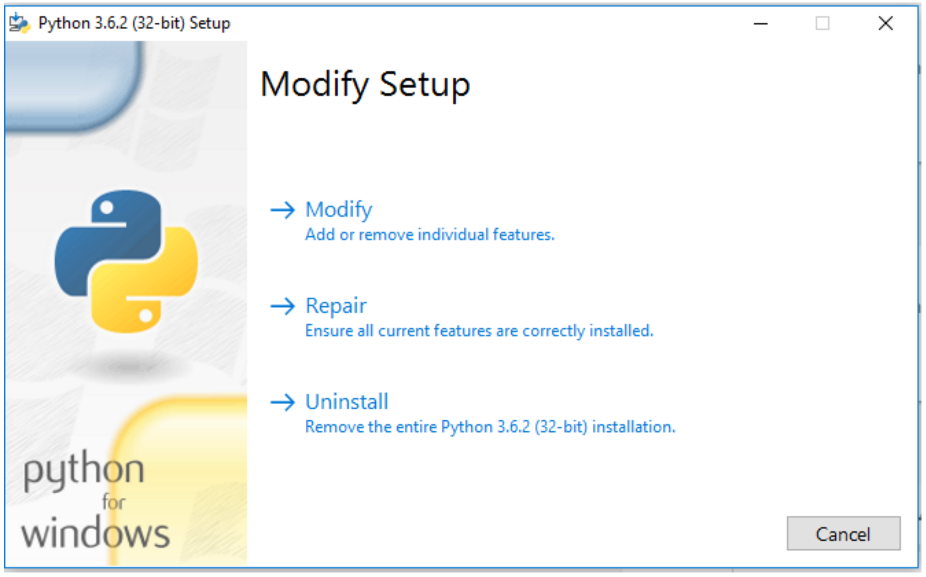
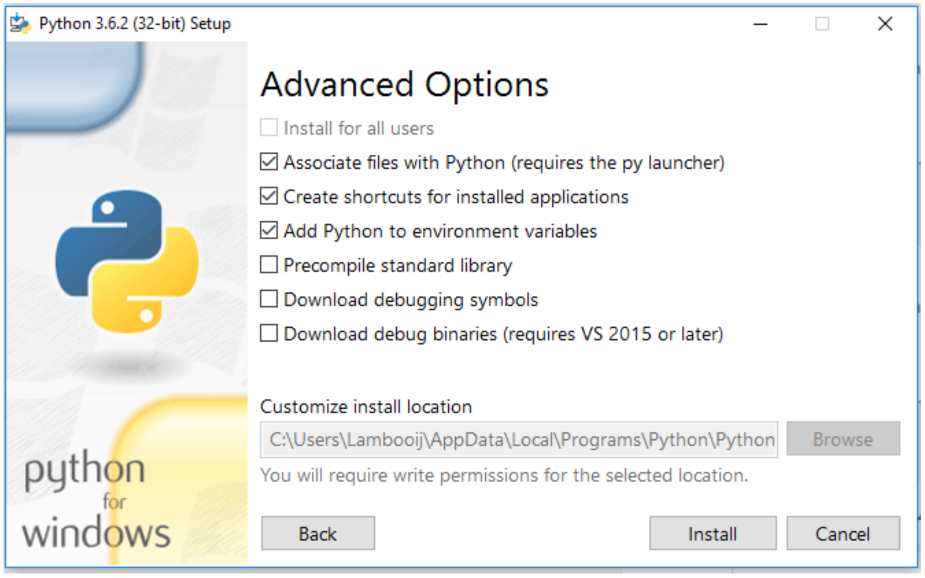

Python 3.6.2 (32-bit) Setup
Modify Setup
Modify
Add or remove individual features.
Repair
Ensure all current features are correctly installed.
Uninstall
Remove the entire Python 3.6.2 (32-bit) installation.
Cancel
python for windows

Figure 3: Command Prompt

Choose ‘Modify’ (figure 2) and in the next window (figure 3) check the box ‘add Python to the environment variables’.

After succesful completion you can redo the installation on page 1.



Advanced Options

☐ Install for all users

☑ Associate files with Python (requires the py launcher)

☑ Create shortcuts for installed applications

☑ Add Python to environment variables

☐ Precompile standard library

☐ Download debugging symbols

☐ Download debug binaries (requires VS 2015 or later)

Customize install location

C:\Users\Lambooij\AppData\Local\Programs\Python\Python

You will require write permissions for the selected location.

Figure 4: Adding Python to the environment variables

Appendix 2: Starting python.exe directly from the python folder (in case ‘pip’ reports SSL problems)

Sometimes another python version will prevent proper installation of pygame (e.g. if you also have an Anaconda installation). In that case you are likely to get reports of ‘SSL’ errors during the installation. You can circumvent this by running ‘python.exe’ directly from the folder where it is installed. The command should be:

```
python.exe -m install pygame --user
```

Ask your teacher or one of the student assistants if you do not know how to do this. After succesful completion you should be able to run the starter code of the assignment.