

# Labyrinthe à bille

## Créer une scène 3D interactive

# Tutoriel 3 : Interfacer avec Arduino avec pyFirmata



Philippe Roy <[philippe.roy@ac-grenoble.fr](mailto:philippe.roy@ac-grenoble.fr)>

<https://forge.aeif.fr/blender-edutech/blender-edutech-tuto>

# Objectif

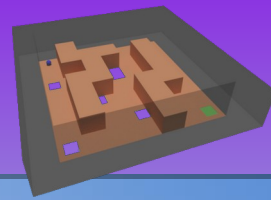


L'objectif de ce tutoriel est faire interagir les objets de la scène 3D (des objets virtuelles) à partir d'actions physiques mesurées par des capteurs. Les **capteurs** sont ici reliés à un **micro-contrôleur Arduino** par la **connectique Grove** et le **protocole Firmata**. La guidance de ce tutoriel a pour pré-requis la réalisation des deux tutoriels précédents (Tutoriel 1 : Ma première scène, Tutoriel 2 : Passage au Python).

Le tutoriel se décompose en 6 étapes :

- [1. Installation de la bibliothèque pyFirmata](#)
- [2. Préparer la carte Arduino](#)
- [3. Déplacer le plateau avec 4 boutons binaires](#)
- [4. Déplacer le plateau avec un joystick analogique](#)
- [5. Détecter automatiquement le micro-contrôleur](#)
- [6. Distribuer l'exécutable](#)

# 1. Installation de la bibliothèque pyFirmata



Le codage se fait par un éditeur de texte. Il en existe beaucoup et tout éditeur peut convenir. Pour ce tutoriel j'utilise **Emacs**, il est open source, très efficace et polyvalent mais peu intuitif. Le choix de l'éditeur est souvent très personnel, sans préférence je vous conseille **Spyder**, il est open source et complet.

- **Emacs** ce trouve à cette adresse : <https://www.gnu.org/software/emacs>
- **Spyder** ce trouve à cette adresse : <https://www.spyder-ide.org>

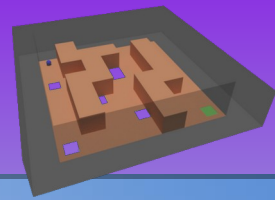
```
1 import bge # Bibliothèque Blender / Game Engine (BGE)
2
3 #####
4 # labyrinthe.py
5 # @title: Commandes pour le tutotiel Labyrinthe
6 # @project: Blender-EduTech
7 # @lang: fr
8 # @authors: Philippe Roy <philippe.roy@ac-grenoble.fr>
9 # @copyright: Copyright (C) 2021 Philippe Roy
10 # @license: GNU GPL
11 #
12 # Commandes déclenchées par UPBGE pour le tutoriel Labyrinthe
13 #
14 #####
15
16 # Récupérer la scène 3D
17 scene = bge.logic.getCurrentScene()
18 # print("Objets de la scene : ", scene.objects)
19
20 # Constantes
21
22 JUST_ACTIVATED = bge.logic.KX_INPUT_JUST_ACTIVATED
23 JUST_RELEASED = bge.logic.KX_INPUT_JUST_RELEASED
24 ACTIVATE = bge.logic.KX_INPUT_ACTIVE
25 JUST_DEACTIVATED = bge.logic.KX_SENSOR_JUST_DEACTIVATED
26
27 #####
28 # Gestion du clavier
29 #####
30
31 # Flèches pour tourner le plateau
32 def clavier(cont):
33     # obj = cont.owner
34     obj = scene.objects["Plateau"]
35     keyboard = bge.logic.keyboard
36     resolution = 0.01
37
38     # Up
39     if (ACTIVATE == keyboard.events[bge.events.UPARROWKEY]):
40         obj.applyRotation((-resolution,0,0), False)
41
42     # Down
43     if (ACTIVATE == keyboard.events[bge.events.DOWNARROWKEY]):
44         obj.applyRotation((resolution,0,0), False)
```

Emacs

```
5 # labyrinthe.py
6 # @title: Commandes pour le tutotiel Labyrinthe
7 # @project: Blender-EduTech
8 # @lang: fr
9 # @authors: Philippe Roy <philippe.roy@ac-grenoble.fr>
10 # @copyright: Copyright (C) 2021 Philippe Roy
11 # @license: GNU GPL
12 #
13 # Commandes déclenchées par UPBGE pour le tutoriel Labyrinthe
14 #
15 #####
16 # Récupérer la scène 3D
17 scene = bge.logic.getCurrentScene()
18 # print("Objets de la scene : ", scene.objects)
19
20 # Constantes
21
22 JUST_ACTIVATED = bge.logic.KX_INPUT_JUST_ACTIVATED
23 JUST_RELEASED = bge.logic.KX_INPUT_JUST_RELEASED
24 ACTIVATE = bge.logic.KX_INPUT_ACTIVE
25 JUST_DEACTIVATED = bge.logic.KX_SENSOR_JUST_DEACTIVATED
26
27 #####
28 # Gestion du clavier
29 #####
30
31 # Flèches pour tourner le plateau
32 def clavier(cont):
33     # obj = cont.owner
34     obj = scene.objects["Plateau"]
35     keyboard = bge.logic.keyboard
36     resolution = 0.01
37
38     # Up
39     if (ACTIVATE == keyboard.events[bge.events.UPARROWKEY]):
40         obj.applyRotation((-resolution,0,0), False)
41
42     # Down
43     if (ACTIVATE == keyboard.events[bge.events.DOWNARROWKEY]):
44         obj.applyRotation((resolution,0,0), False)
```

Spyder

## 2. Préparer la carte Arduino



Le codage se fait par un éditeur de texte. Il en existe beaucoup et tout éditeur peut convenir. Pour ce tutoriel j'utilise **Emacs**, il est open source, très efficace et polyvalent mais peu intuitif. Le choix de l'éditeur est souvent très personnel, sans préférence je vous conseille **Spyder**, il est open source et complet.

- **Emacs** ce trouve à cette adresse : <https://www.gnu.org/software/emacs>
- **Spyder** ce trouve à cette adresse : <https://www.spyder-ide.org>

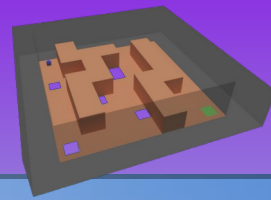
```
1 import bge # Bibliothèque Blender / Game Engine (BGE)
2
3 #####
4 # labyrinth.py
5 # @title: Commandes pour le tutoriel Labyrinthe
6 # @project: Blender-EduTech
7 # @lang: fr
8 # @authors: Philippe Roy <philippe.roy@ac-grenoble.fr>
9 # @copyright: Copyright (C) 2021 Philippe Roy
10 # @license: GNU GPL
11 #
12 # Commandes déclenchées par UPBGE pour le tutoriel Labyrinthe
13 #
14 #####
15
16 # Récupérer la scène 3D
17 scene = bge.logic.getCurrentScene()
18 # print("Objets de la scène : ", scene.objects)
19
20 # Constantes
21
22 JUST_ACTIVATED = bge.logic.KX_INPUT_JUST_ACTIVATED
23 JUST_RELEASED = bge.logic.KX_INPUT_JUST_RELEASED
24 ACTIVATE = bge.logic.KX_INPUT_ACTIVE
25 JUST_DEACTIVATED = bge.logic.KX_SENSOR_JUST_DEACTIVATED
26
27 #####
28 # Gestion du clavier
29 #####
30
31 # Flèches pour tourner le plateau
32 def clavier(cont):
33     # obj = cont.owner
34     obj = scene.objects["Plateau"]
35     keyboard = bge.logic.keyboard
36     resolution = 0.01
37
38     # Up
39     if (ACTIVATE == keyboard.events[bge.events.UPARROWKEY]):
40         obj.applyRotation((-resolution,0,0), False)
41
42     # Down
43     if (ACTIVATE == keyboard.events[bge.events.DOWNARROWKEY]):
44         obj.applyRotation((resolution,0,0), False)
```

Emacs

```
5 # labyrinth.py
6 # @title: Commandes pour le tutoriel Labyrinthe
7 # @project: Blender-EduTech
8 # @lang: fr
9 # @authors: Philippe Roy <philippe.roy@ac-grenoble.fr>
10 # @copyright: Copyright (C) 2021 Philippe Roy
11 # @license: GNU GPL
12 #
13 # Commandes déclenchées par UPBGE pour le tutoriel Labyrinthe
14 #
15 #####
16 # Récupérer la scène 3D
17 scene = bge.logic.getCurrentScene()
18 # print("Objets de la scène : ", scene.objects)
19
20 # Constantes
21
22 JUST_ACTIVATED = bge.logic.KX_INPUT_JUST_ACTIVATED
23 JUST_RELEASED = bge.logic.KX_INPUT_JUST_RELEASED
24 ACTIVATE = bge.logic.KX_INPUT_ACTIVE
25 JUST_DEACTIVATED = bge.logic.KX_SENSOR_JUST_DEACTIVATED
26
27 #####
28 # Gestion du clavier
29 #####
30
31 # Flèches pour tourner le plateau
32 def clavier(cont):
33     # obj = cont.owner
34     obj = scene.objects["Plateau"]
35     keyboard = bge.logic.keyboard
36     resolution = 0.01
37
38     # Up
39     if (ACTIVATE == keyboard.events[bge.events.UPARROWKEY]):
40         obj.applyRotation((-resolution,0,0), False)
41
42     # Down
43     if (ACTIVATE == keyboard.events[bge.events.DOWNARROWKEY]):
44         obj.applyRotation((resolution,0,0), False)
```

Spyder

# 3. Déplacer le plateau avec 4 boutons binaires



Le codage se fait par un éditeur de texte. Il en existe beaucoup et tout éditeur peut convenir. Pour ce tutoriel j'utilise **Emacs**, il est open source, très efficace et polyvalent mais peu intuitif. Le choix de l'éditeur est souvent très personnel, sans préférence je vous conseille **Spyder**, il est open source et complet.

- **Emacs** ce trouve à cette adresse : <https://www.gnu.org/software/emacs>
- **Spyder** ce trouve à cette adresse : <https://www.spyder-ide.org>

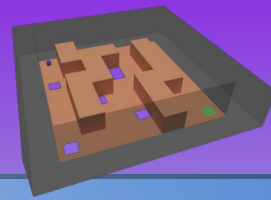
```
1 import bge # Bibliothèque Blender / Game Engine (BGE)
2
3 #####
4 # labyrinth.py
5 # @title: Commandes pour le tutotiel Labyrinthe
6 # @project: Blender-EduTech
7 # @lang: fr
8 # @authors: Philippe Roy <philippe.roy@ac-grenoble.fr>
9 # @copyright: Copyright (C) 2021 Philippe Roy
10 # @license: GNU GPL
11 #
12 # Commandes déclenchées par UPBGE pour le tutoriel Labyrinthe
13 #
14 #####
15
16 # Récupérer la scène 3D
17 scene = bge.logic.getCurrentScene()
18 # print("Objets de la scene : ", scene.objects)
19
20 # Constantes
21
22 JUST_ACTIVATED = bge.logic.KX_INPUT_JUST_ACTIVATED
23 JUST_RELEASED = bge.logic.KX_INPUT_JUST_RELEASED
24 ACTIVATE = bge.logic.KX_INPUT_ACTIVE
25 JUST_DEACTIVATED = bge.logic.KX_SENSOR_JUST_DEACTIVATED
26
27 #####
28 # Gestion du clavier
29 #####
30
31 # Flèches pour tourner le plateau
32 def clavier(cont):
33     # obj = cont.owner
34     obj = scene.objects["Plateau"]
35     keyboard = bge.logic.keyboard
36     resolution = 0.01
37
38     # Up
39     if (ACTIVATE == keyboard.events[bge.events.UPARROWKEY]):
40         obj.applyRotation((-resolution,0,0), False)
41
42     # Down
43     if (ACTIVATE == keyboard.events[bge.events.DOWNARROWKEY]):
44         obj.applyRotation((resolution,0,0), False)
```

Emacs

```
5 # labyrinth.py
6 # @title: Commandes pour le tutotiel Labyrinthe
7 # @project: Blender-EduTech
8 # @lang: fr
9 # @authors: Philippe Roy <philippe.roy@ac-grenoble.fr>
10 # @copyright: Copyright (C) 2021 Philippe Roy
11 # @license: GNU GPL
12 #
13 # Commandes déclenchées par UPBGE pour le tutoriel Labyrinthe
14 #
15 #####
16 # Récupérer la scène 3D
17 scene = bge.logic.getCurrentScene()
18 # print("Objets de la scene : ", scene.objects)
19
20 # Constantes
21
22 JUST_ACTIVATED = bge.logic.KX_INPUT_JUST_ACTIVATED
23 JUST_RELEASED = bge.logic.KX_INPUT_JUST_RELEASED
24 ACTIVATE = bge.logic.KX_INPUT_ACTIVE
25 JUST_DEACTIVATED = bge.logic.KX_SENSOR_JUST_DEACTIVATED
26
27 #####
28 # Gestion du clavier
29 #####
30
31 # Flèches pour tourner le plateau
32 def clavier(cont):
33     # obj = cont.owner
34     obj = scene.objects["Plateau"]
35     keyboard = bge.logic.keyboard
36     resolution = 0.01
37
38     # Up
39     if (ACTIVATE == keyboard.events[bge.events.UPARROWKEY]):
40         obj.applyRotation((-resolution,0,0), False)
41
42     # Down
43     if (ACTIVATE == keyboard.events[bge.events.DOWNARROWKEY]):
44         obj.applyRotation((resolution,0,0), False)
```

Spyder

# 4. Déplacer le plateau avec un joystick analogique



Le codage se fait par un éditeur de texte. Il en existe beaucoup et tout éditeur peut convenir. Pour ce tutoriel j'utilise **Emacs**, il est open source, très efficace et polyvalent mais peu intuitif. Le choix de l'éditeur est souvent très personnel, sans préférence je vous conseille **Spyder**, il est open source et complet.

- **Emacs** ce trouve à cette adresse : <https://www.gnu.org/software/emacs>
- **Spyder** ce trouve à cette adresse : <https://www.spyder-ide.org>

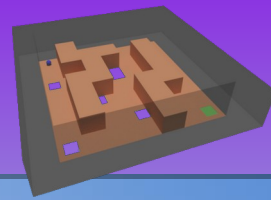
```
1 import bge # Bibliothèque Blender / Game Engine (BGE)
2
3 #####
4 # labyrinth.py
5 # @title: Commandes pour le tutotiel Labyrinthe
6 # @project: Blender-EduTech
7 # @lang: fr
8 # @authors: Philippe Roy <philippe.roy@ac-grenoble.fr>
9 # @copyright: Copyright (C) 2021 Philippe Roy
10 # @license: GNU GPL
11 #
12 # Commandes déclenchées par UPBGE pour le tutoriel Labyrinthe
13 #
14 #####
15
16 # Récupérer la scène 3D
17 scene = bge.logic.getCurrentScene()
18 # print("Objets de la scene : ", scene.objects)
19
20 # Constantes
21
22 JUST_ACTIVATED = bge.logic.KX_INPUT_JUST_ACTIVATED
23 JUST_RELEASED = bge.logic.KX_INPUT_JUST_RELEASED
24 ACTIVATE = bge.logic.KX_INPUT_ACTIVE
25 JUST_DEACTIVATED = bge.logic.KX_SENSOR_JUST_DEACTIVATED
26
27 #####
28 # Gestion du clavier
29 #####
30
31 # Flèches pour tourner le plateau
32 def clavier(cont):
33     # obj = cont.owner
34     obj = scene.objects["Plateau"]
35     keyboard = bge.logic.keyboard
36     resolution = 0.01
37
38     # Up
39     if (ACTIVATE == keyboard.events[bge.events.UPARROWKEY]):
40         obj.applyRotation((-resolution,0,0), False)
41
42     # Down
43     if (ACTIVATE == keyboard.events[bge.events.DOWNARROWKEY]):
44         obj.applyRotation((resolution,0,0), False)
```

Emacs

```
5 # labyrinth.py
6 # @title: Commandes pour le tutotiel Labyrinthe
7 # @project: Blender-EduTech
8 # @lang: fr
9 # @authors: Philippe Roy <philippe.roy@ac-grenoble.fr>
10 # @copyright: Copyright (C) 2021 Philippe Roy
11 # @license: GNU GPL
12 #
13 # Commandes déclenchées par UPBGE pour le tutoriel Labyrinthe
14 #
15 #####
16 # Récupérer la scène 3D
17 scene = bge.logic.getCurrentScene()
18 # print("Objets de la scene : ", scene.objects)
19
20 # Constantes
21
22 JUST_ACTIVATED = bge.logic.KX_INPUT_JUST_ACTIVATED
23 JUST_RELEASED = bge.logic.KX_INPUT_JUST_RELEASED
24 ACTIVATE = bge.logic.KX_INPUT_ACTIVE
25 JUST_DEACTIVATED = bge.logic.KX_SENSOR_JUST_DEACTIVATED
26
27 #####
28 # Gestion du clavier
29 #####
30
31 # Flèches pour tourner le plateau
32 def clavier(cont):
33     # obj = cont.owner
34     obj = scene.objects["Plateau"]
35     keyboard = bge.logic.keyboard
36     resolution = 0.01
37
38     # Up
39     if (ACTIVATE == keyboard.events[bge.events.UPARROWKEY]):
40         obj.applyRotation((-resolution,0,0), False)
41
42     # Down
43     if (ACTIVATE == keyboard.events[bge.events.DOWNARROWKEY]):
44         obj.applyRotation((resolution,0,0), False)
```

Spyder

# 5. Détecter automatiquement le micro-contrôleur



Le codage se fait par un éditeur de texte. Il en existe beaucoup et tout éditeur peut convenir. Pour ce tutoriel j'utilise **Emacs**, il est open source, très efficace et polyvalent mais peu intuitif. Le choix de l'éditeur est souvent très personnel, sans préférence je vous conseille **Spyder**, il est open source et complet.

- **Emacs** ce trouve à cette adresse : <https://www.gnu.org/software/emacs>
- **Spyder** ce trouve à cette adresse : <https://www.spyder-ide.org>

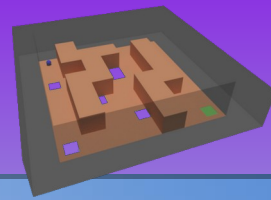
```
1 import bge # Bibliothèque Blender / Game Engine (BGE)
2
3 #####
4 # labyrinth.py
5 # @title: Commandes pour le tutotiel Labyrinth
6 # @project: Blender-EduTech
7 # @lang: fr
8 # @authors: Philippe Roy <philippe.roy@ac-grenoble.fr>
9 # @copyright: Copyright (C) 2021 Philippe Roy
10 # @license: GNU GPL
11 #
12 # Commandes déclenchées par UPBGE pour le tutoriel Labyrinth
13 #
14 #####
15
16 # Récupérer la scène 3D
17 scene = bge.logic.getCurrentScene()
18 # print("Objets de la scene : ", scene.objects)
19
20 # Constantes
21
22 JUST_ACTIVATED = bge.logic.KX_INPUT_JUST_ACTIVATED
23 JUST_RELEASED = bge.logic.KX_INPUT_JUST_RELEASED
24 ACTIVATE = bge.logic.KX_INPUT_ACTIVE
25 JUST_DEACTIVATED = bge.logic.KX_SENSOR_JUST_DEACTIVATED
26
27 #####
28 # Gestion du clavier
29 #####
30
31 # Flèches pour tourner le plateau
32 def clavier(cont):
33     # obj = cont.owner
34     obj = scene.objects["Plateau"]
35     keyboard = bge.logic.keyboard
36     resolution = 0.01
37
38     # Up
39     if (ACTIVATE == keyboard.events[bge.events.UPARROWKEY]):
40         obj.applyRotation((-resolution,0,0), False)
41
42     # Down
43     if (ACTIVATE == keyboard.events[bge.events.DOWNARROWKEY]):
44         obj.applyRotation((resolution,0,0), False)
```

Emacs

```
5 #####
6 # labyrinth.py
7 # @title: Commandes pour le tutotiel Labyrinth
8 # @project: Blender-EduTech
9 # @lang: fr
10 # @authors: Philippe Roy <philippe.roy@ac-grenoble.fr>
11 # @copyright: Copyright (C) 2021 Philippe Roy
12 # @license: GNU GPL
13 #
14 # Commandes déclenchées par UPBGE pour le tutoriel Labyrinth
15 #
16 #####
17
18 # Récupérer la scène 3D
19 scene = bge.logic.getCurrentScene()
20 # print("Objets de la scene : ", scene.objects)
21
22 # Constantes
23 JUST_ACTIVATED = bge.logic.KX_INPUT_JUST_ACTIVATED
24 JUST_RELEASED = bge.logic.KX_INPUT_JUST_RELEASED
25 ACTIVATE = bge.logic.KX_INPUT_ACTIVE
26 JUST_DEACTIVATED = bge.logic.KX_SENSOR_JUST_DEACTIVATED
27
28 #####
29 # Gestion du clavier
30 #####
31 # Flèches pour tourner le plateau
32 def clavier(cont):
33     # obj = cont.owner
34     obj = scene.objects["Plateau"]
35     keyboard = bge.logic.keyboard
36     resolution = 0.01
37
38     # Up
39     if (ACTIVATE == keyboard.events[bge.events.UPARROWKEY]):
40         obj.applyRotation((-resolution,0,0), False)
41
42     # Down
43     if (ACTIVATE == keyboard.events[bge.events.DOWNARROWKEY]):
44         obj.applyRotation((resolution,0,0), False)
```

Spyder

# 6. Inclure pyFirmata dans la distribution de l'exécutable



Le codage se fait par un éditeur de texte. Il en existe beaucoup et tout éditeur peut convenir. Pour ce tutoriel j'utilise **Emacs**, il est open source, très efficace et polyvalent mais peu intuitif. Le choix de l'éditeur est souvent très personnel, sans préférence je vous conseille **Spyder**, il est open source et complet.

- **Emacs** ce trouve à cette adresse : <https://www.gnu.org/software/emacs>
- **Spyder** ce trouve à cette adresse : <https://www.spyder-ide.org>

```
Emacs: 1-1-labyrinthe.py (/home/phroy/Bureau/seriousgames/blender-edutech/git/blender-edutech-tuto/labyrinthe/1-scene3D/1-1-labyrinthe)
File Edit Options Buffers Tools Python Virtual Envs Elpy Flymake YASnippet Help
1 import bge # Bibliothèque Blender Game Engine (BGE)
2
3 #####
4 # labyrinthe.py
5 # @title: Commandes pour le tutotiel Labyrinthe
6 # @project: Blender-EduTech
7 # @lang: fr
8 # @authors: Philippe Roy <philippe.roy@ac-grenoble.fr>
9 # @copyright: Copyright (C) 2021 Philippe Roy
10 # @license: GNU GPL
11 #
12 # Commandes déclenchées par UPBGE pour le tutoriel Labyrinthe
13 #
14 #####
15
16 # Récupérer la scène 3D
17 scene = bge.logic.getCurrentScene()
18 # print("Objets de la scene : ", scene.objects)
19
20 # Constantes
21
22 JUST_ACTIVATED = bge.logic.KX_INPUT_JUST_ACTIVATED
23 JUST_RELEASED = bge.logic.KX_INPUT_JUST_RELEASED
24 ACTIVATE = bge.logic.KX_INPUT_ACTIVE
25 JUST_DEACTIVATED = bge.logic.KX_SENSOR_JUST_DEACTIVATED
26
27 #####
28 # Gestion du clavier
29 #####
30
31 # Flèches pour tourner le plateau
32 def clavier(cont):
33     # obj = cont.owner
34     obj = scene.objects["Plateau"]
35     keyboard = bge.logic.keyboard
36     resolution = 0.01
37
38     # Up
39     if (ACTIVATE == keyboard.events[bge.events.UPARROWKEY]):
40         obj.applyRotation((-resolution,0,0), False)
41
42     # Down
43     if (ACTIVATE == keyboard.events[bge.events.DOWNARROWKEY]):
44         obj.applyRotation((resolution,0,0), False)
45
46     # Left
47     if (ACTIVATE == keyboard.events[bge.events.LEFTARROWKEY]):
48         obj.applyRotation((0,-resolution,0), False)
49
50     # Right
51     if (ACTIVATE == keyboard.events[bge.events.RIGHTARROWKEY]):
52         obj.applyRotation((0,resolution,0), False)
53
54     # Quit
55     if (ACTIVATE == keyboard.events[bge.events.ESC]):
56         bge.logic.setScene3D("1-1-labyrinthe.blend")
57
58     # End
59     return True
60
61 #####
62 # README.md
63 #####
64
65 1 ~/Bureau/seriousgames/blender-edutech-tuto/labyrinthe
66 2 /git/blender-edutech-tuto/labyrinthe
67 3 /1-scene3D/
68 4 [+] 1 - Ma première scene.odp
69 5 [+] 1 - Ma première scene.pdf
60 6 [+] 1 - Ma première scene.pdf
61 7 [+] 1-1-labyrinthe.blend
62 8 [-] 1-1-labyrinthe.py
63 9 > clavier (def)
64 10 > joystick (def)
65 11 > init (def)
66 12 > applyRotationTo (def)
67 13 > restart (def)
68 14 > gagne (def)
69 15 [+] 2 - Passage au Python.odp
70 16 [+] Blender_poster_3.3.jpg
71 17 [+] README.md
```

Emacs

```
Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
/home/phroy
~/Bureau/seriousgames/blender-edutech/git/blender-edutech-tuto/labyrinthe/1-scene3D/1-1-labyrinthe.py
1-1-labyrinthe.py X
1 # labyrinthe.py
2 # @title: Commandes pour le tutotiel Labyrinthe
3 # @project: Blender-EduTech
4 # @lang: fr
5 # @authors: Philippe Roy <philippe.roy@ac-grenoble.fr>
6 # @copyright: Copyright (C) 2021 Philippe Roy
7 # @license: GNU GPL
8 #
9 # Commandes déclenchées par UPBGE pour le tutoriel Labyrinthe
10 #
11 # Récupérer la scène 3D
12 scene = bge.logic.getCurrentScene()
13 # print("Objets de la scene : ", scene.objects)
14
15 # Constantes
16
17 JUST_ACTIVATED = bge.logic.KX_INPUT_JUST_ACTIVATED
18 JUST_RELEASED = bge.logic.KX_INPUT_JUST_RELEASED
19 ACTIVATE = bge.logic.KX_INPUT_ACTIVE
20 JUST_DEACTIVATED = bge.logic.KX_SENSOR_JUST_DEACTIVATED
21
22 #####
23 # Gestion du clavier
24 #####
25
26 # Flèches pour tourner le plateau
27 def clavier(cont):
28     # obj = cont.owner
29     obj = scene.objects["Plateau"]
30     keyboard = bge.logic.keyboard
31     resolution = 0.01
32
33     # Up
34     if (ACTIVATE == keyboard.events[bge.events.UPARROWKEY]):
35         obj.applyRotation((-resolution,0,0), False)
36
37     # Down
38     if (ACTIVATE == keyboard.events[bge.events.DOWNARROWKEY]):
39         obj.applyRotation((resolution,0,0), False)
40
41     # Left
42     if (ACTIVATE == keyboard.events[bge.events.LEFTARROWKEY]):
43         obj.applyRotation((0,-resolution,0), False)
44     # Right
45     if (ACTIVATE == keyboard.events[bge.events.RIGHTARROWKEY]):
46         obj.applyRotation((0,resolution,0), False)
47
48     # Quit
49     if (ACTIVATE == keyboard.events[bge.events.ESC]):
50         bge.logic.setScene3D("1-1-labyrinthe.blend")
51
52     # End
53     return True
54
55 #####
56 # README.md
57 #####
```

Spyder



## 2. Déplacer le plateau



Le fichier Blender de départ est le fichier résultat du tutoriel 1 sans les briques logiques ni les propriétés. Il est disponible dans le répertoire du tutoriel sous le nom « **2-labyrinthe-debut.blend** ».

Pour la gestion du clavier, le principe est de créer un **boucle infinie** qui exécute la fonction **clavier** à chaque **tic logique (logic tick)**.

Briques logiques de **Plateau**

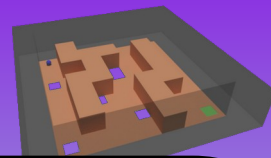
### 1 : Créer la boucle infinie

- **Ajouter** un **Capteur Toujours**
- **activer** le **Pulse True Level (▲)**
- **renommer** le capteur avec **Clavier**

### 2 : Appeler la fonction

- **Ajouter** un **Contrôleur Python**
- **définir** le **Module** avec la fonction **2-labyrinthe.clavier**

## 2. Déplacer le plateau



Le module Python est le fichier « **2-labyrinthe.py** ».

```
import bge # Bibliothèque Blender Game Engine (UPBGE)

#####
# 2-labyrinthe.py
#####

# Récupérer la scène 3D
scene = bge.logic.getCurrentScene()

# Constantes
JUST_ACTIVATED = bge.logic.KX_INPUT_JUST_ACTIVATED
JUST_RELEASED = bge.logic.KX_INPUT_JUST_RELEASED
ACTIVATE = bge.logic.KX_INPUT_ACTIVE

#####
# Gestion du clavier
#####

# Flèches pour tourner le plateau
def clavier(cont):
    obj = cont.owner # obj est l'objet associé au contrôleur donc 'Plateau'
    keyboard = bge.logic.keyboard
    resolution = 0.01

    # Flèche haut - Up arrow
    if keyboard.inputs[bge.events.UPARROWKEY].status[0] == ACTIVATE:
        obj.applyRotation((-resolution,0,0), False)

    # Flèche bas - Down arrow
    if keyboard.inputs[bge.events.DOWNARROWKEY].status[0] == ACTIVATE:
        obj.applyRotation((resolution,0,0), False)

    # Flèche gauche - Left arrow
    if keyboard.inputs[bge.events.LEFTARROWKEY].status[0] == ACTIVATE:
        obj.applyRotation((0, -resolution,0), False)

    # Flèche droit - Right arrow
    if keyboard.inputs[bge.events.RIGHTARROWKEY].status[0] == ACTIVATE:
        obj.applyRotation((0, resolution,0), False)
```

**3 : Créer le fichier Python**  
Ouvrir votre éditeur et créer le fichier **2-labyrinthe.py**

**4 : Créer le fonction clavier**  
Copier-coller le code

**5 : Tester la scène [P]**

Les pages de l'**API Python de UPBGE** les plus utilisées sont

- **GameObject**
- **Game Logic**