

Romain DUSSOT

Utilisation de UnityVibes et PyUnityVibes

1

D'où ca vient ?

- Conçu lors de l'UV 5.7 de l'année dernière
- Développé par Rémi RIGAL et Noëlie RAMUZAT

Unity-Vibes : Rémi Rigal, Noëlie Ramuzat

L'objectif est d'afficher en 3D rapidement une animation impliquant des robots en interface avec Python. Une classe Animation pour UnityVibes a été créée. Le package Python est sur pip.

<https://www.ensta-bretagne.fr/jaulin/cordeliere.html>

Pourquoi
utiliser cette
bibliothèque
?

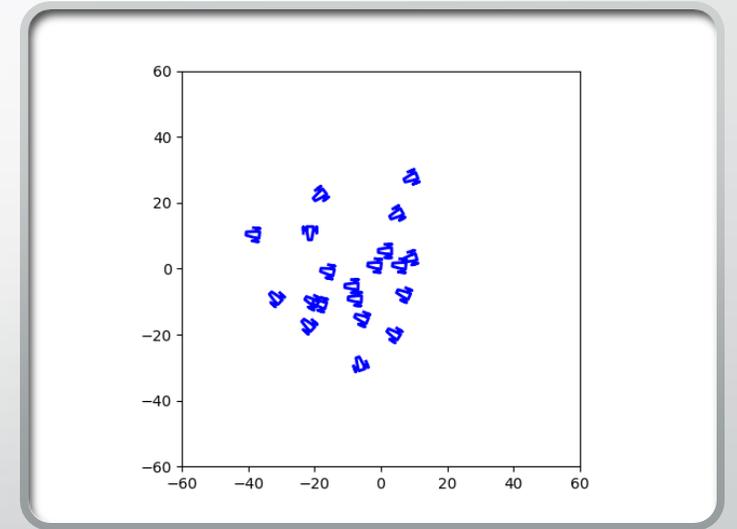
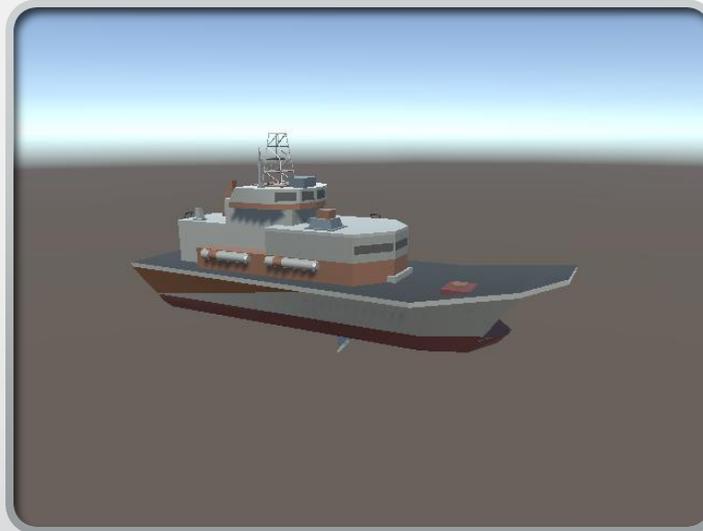
Avoir une interface graphique en 3d
rapidement lié à Python

Arrêter d'utiliser *pylab* et sa fonction
Axis3D qui nécessite une machine de
guerre pour fonctionner en temps réel

Améliorer le rendu pour avoir un visuel
plus "professionnel"

Différence de rendu

- On peut facilement afficher des objets 3D
- L'affichage est fluide
- Si les calculs sont faits préalablement, le résultat sera fluide.



Comment ça marche et pourquoi est-ce si fluide ?

- 2 étapes dans le programme:
- La boucle sur le temps:
 - Calculatoire : calcul des évolutions des vecteurs d'états, etc.
 - Stockage des données dans des *frame*
- Affichage de l'animation

Calcul

Affichage

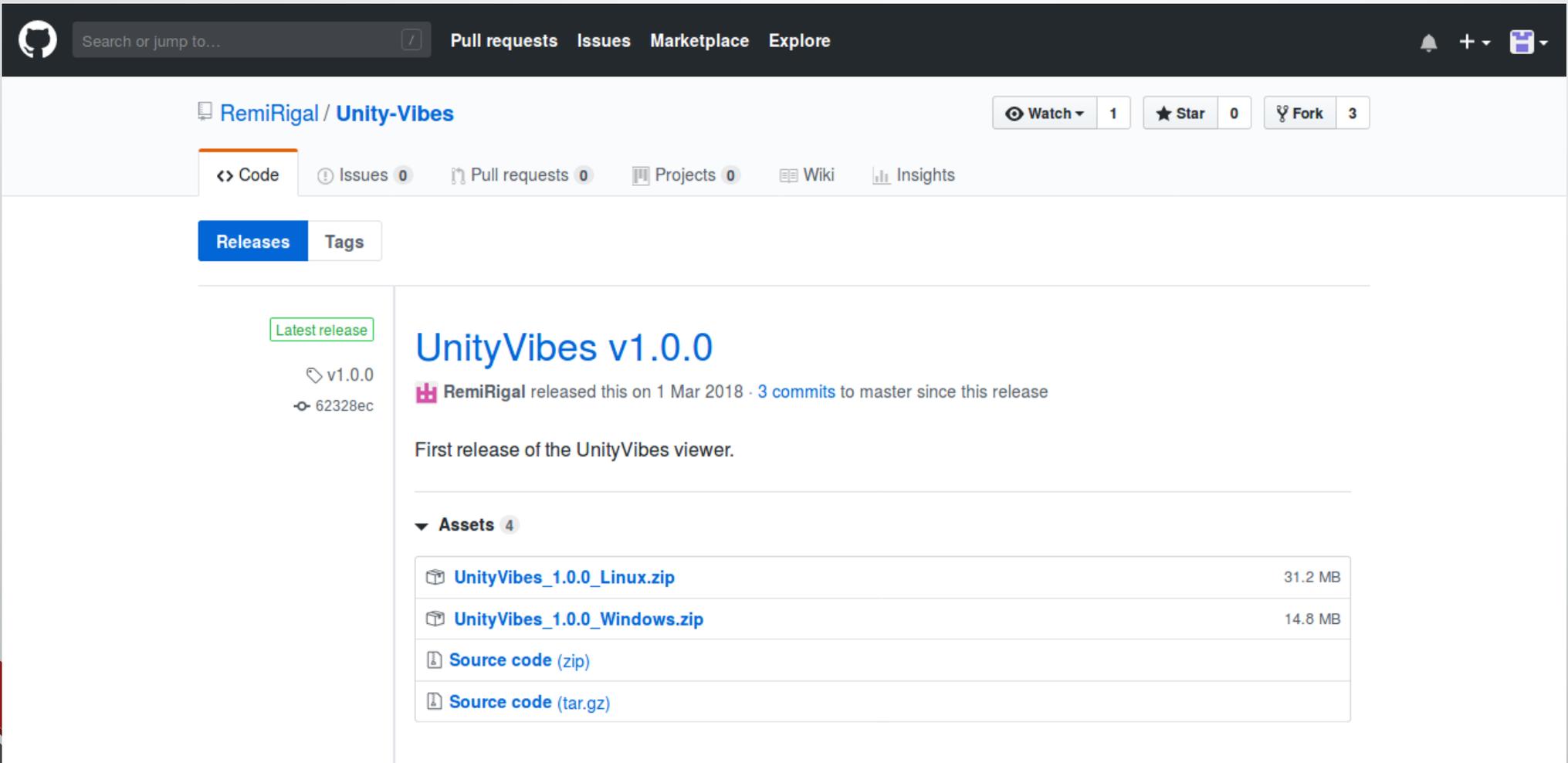
Inconvénient

- L'affichage n'est pas un modèle physique
 - Toutes les équations doivent être définies avant
- Il est quasiment impossible de faire du dynamique
 - Un seul socket pour la communication

Installation

- Pour PyUnityVibes
 - Paquet disponible sur *pip* : `$ pip3 install PyUnityVibes`
 - Puis importation dans python : `>> import PyUnityVibes`
- Pour UnityVibes
 - Exécutable à télécharger sous le github de Rémi
<https://github.com/RemiRigal/Unity-Vibes/releases>

Installation



Search or jump to... Pull requests Issues Marketplace Explore

RemiRigal / UnityVibes Watch 1 Star 0 Fork 3

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights

Releases Tags

Latest release

v1.0.0
62328ec

UnityVibes v1.0.0

RemiRigal released this on 1 Mar 2018 · 3 commits to master since this release

First release of the UnityVibes viewer.

Assets 4

UnityVibes_1.0.0_Linux.zip	31.2 MB
UnityVibes_1.0.0_Windows.zip	14.8 MB
Source code (zip)	
Source code (tar.gz)	

8

Structure du code – version Initiale

Importation de la bibliothèque

```
from PyUnityVibes.UnityFigure import UnityFigure
```

Création de l'objet *UnityFigure*

```
### Initialisation de Unity ###  
figure = UnityFigure(UnityFigure.FIGURE_3D, UnityFigure.SCENE_EMPTY)  
time.sleep(1)
```

Ajout des figures à afficher

```
boat = figure.create(UnityFigure.OBJECT_3D_BOAT, 0, 0, 0)  
time.sleep(1)
```

Création des frames dans la boucle temporelle

```
boat.updatePosition(1, 1, 1)  
time.sleep(1)  
  
boat.updateRotation(45, 90, 0)  
time.sleep(1)
```

Structure du code – version Animation

Importation de la bibliothèque

```
from PyUnityVibes.UnityFigure import UnityFigure
```

Création de l'objet *UnityFigure* et de l'animation

```
### Initialisation de Unity ###  
figure = UnityFigure(UnityFigure.FIGURE_3D, UnityFigure.SCENE_EMPTY)  
time.sleep(1)  
anim = figure.createAnimation(dt)  
time.sleep(1)
```

Ajout des figures à afficher

```
### Creation des AUVs ###  
AUVs = []  
for ind_auv in range(N):  
    AUVs.append(figure.create(UnityFigure.OBJECT_3D_SUBMARINE, 0, -0.4, 0, dimX=5, dimY=5, dimZ=5))  
    anim.addObject(AUVs[ind_auv])  
    time.sleep(0.1)
```

Création des frames dans la boucle temporelle

```
### Calcul des frames successives ###  
for t in np.arange(0, 70, dt):  
    for ind_auv, auv in enumerate(AUVs):  
        anim.appendFrame(auv, x=2*ind_auv, y=-0.4, z=t, rx=0, ry=0, rz=0)  
    time.sleep(1)
```

Comment lancer un programme ?

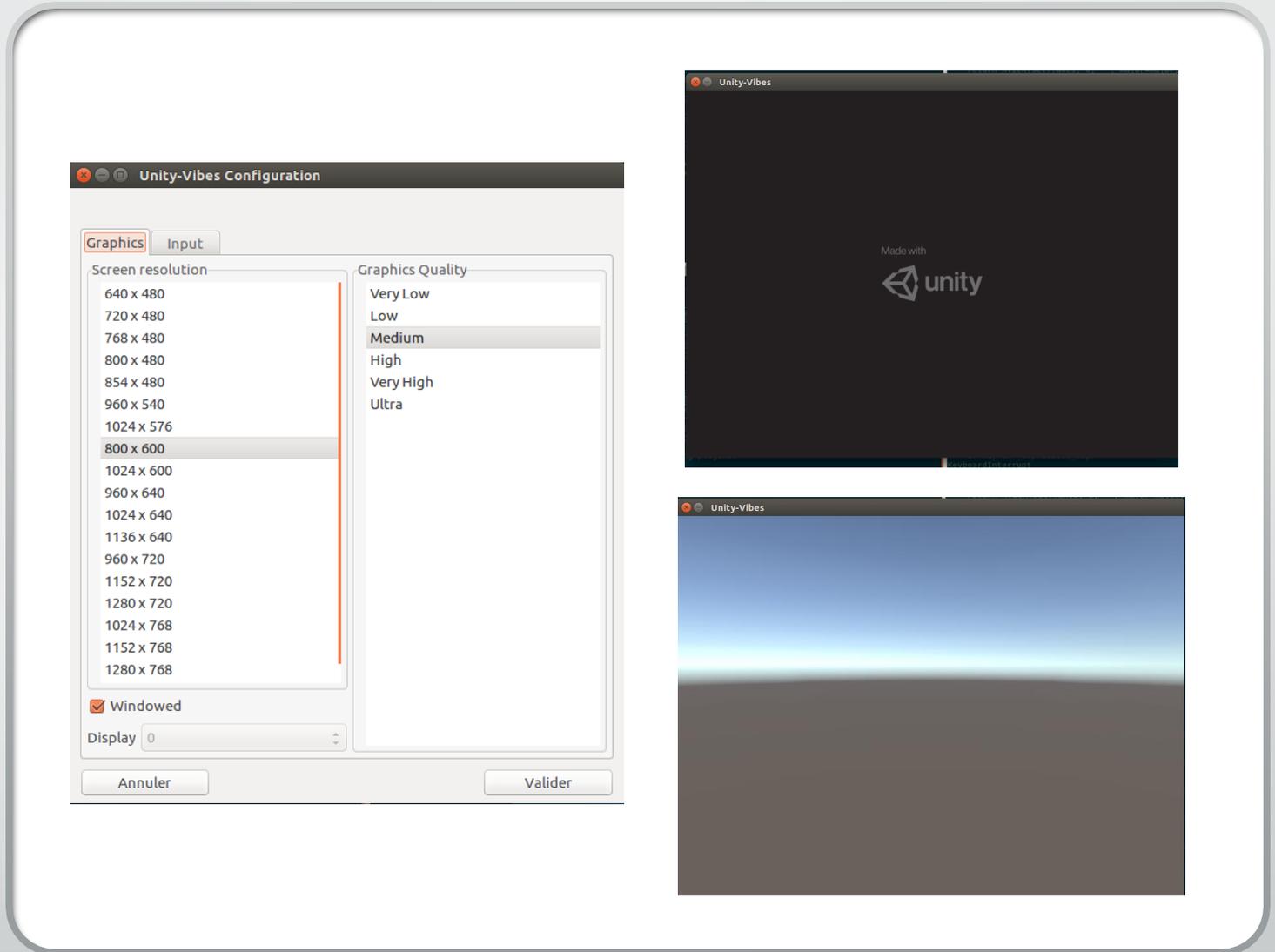
- Fonctionnement similaire à *Vibes* (KALMOOC)
- Exécution de la fenêtre avant de lancer le programme.

```
x deegs@Pedeiche ~/Téléchargements/Linux ls
UnityVibes_Data UnityVibes.x86 UnityVibes.x86_64
deegs@Pedeiche ~/Téléchargements/Linux ./UnityVibes.x86_64
Set current directory to /home/deegs/Téléchargements/Linux
Found path: /home/deegs/Téléchargements/Linux/UnityVibes.x86_64
Mono path[0] = '/home/deegs/Téléchargements/Linux/UnityVibes_Data/Managed'
Mono config path = '/home/deegs/Téléchargements/Linux/UnityVibes_Data/Mono/etc'
Preloaded 'ScreenSelector.so'
Unable to preload the following plugins:
    ScreenSelector.so
Logging to /home/deegs/.config/unity3d/DefaultCompany/Unity-Vibes/Player.log
```

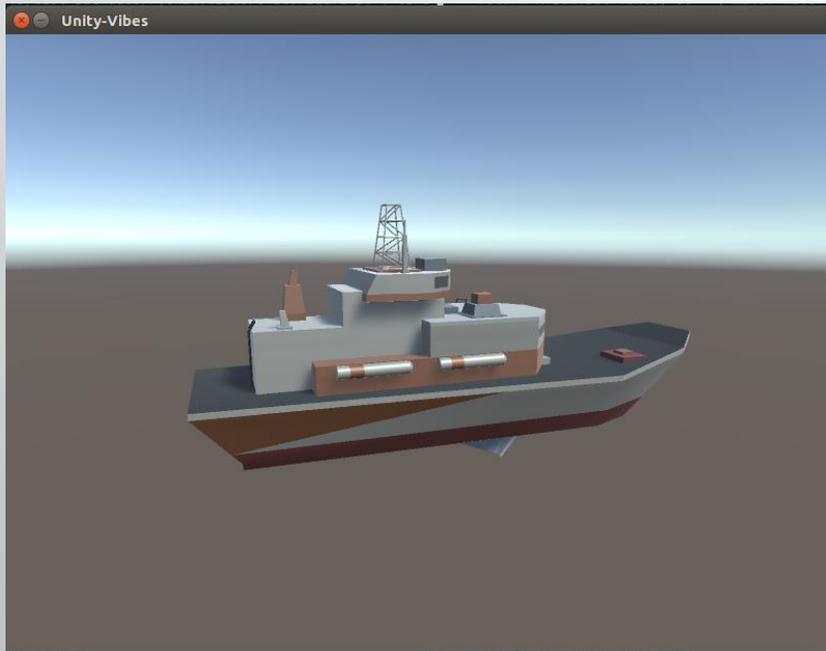
- Exécution du programme python

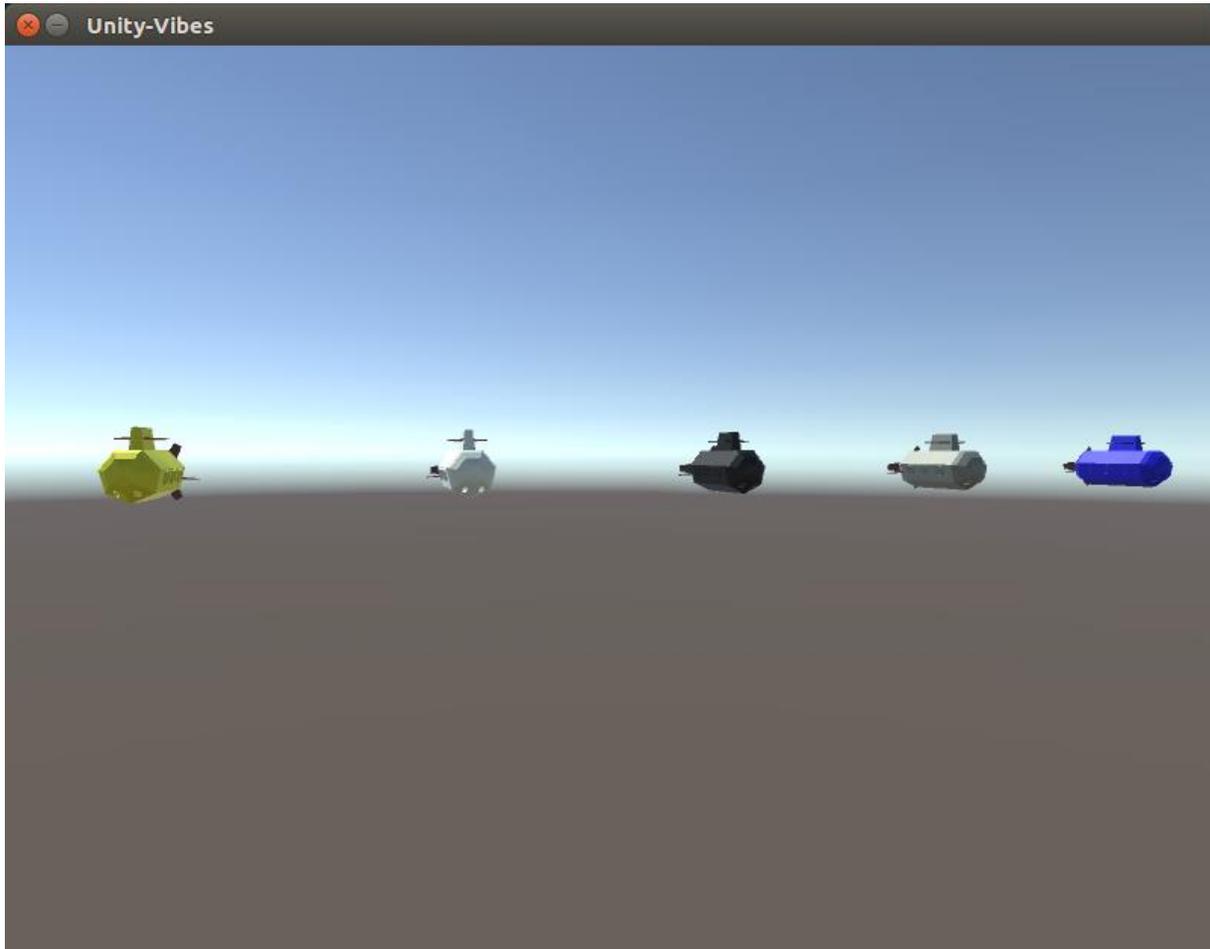
Visualisation

- Une fenêtre de paramétrage
- Une fenêtre d'affichage



Possibilités implémentées





Possibilités
implémentées



Maintenant

Place aux démonstrations !