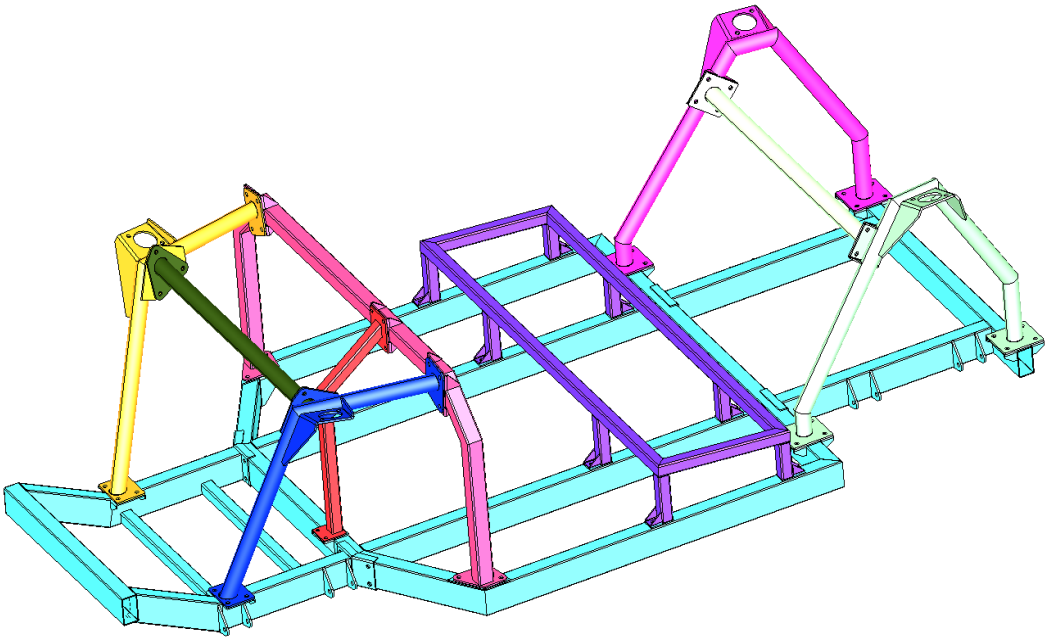
PyOSV

Modélisation libre d'objets complexes

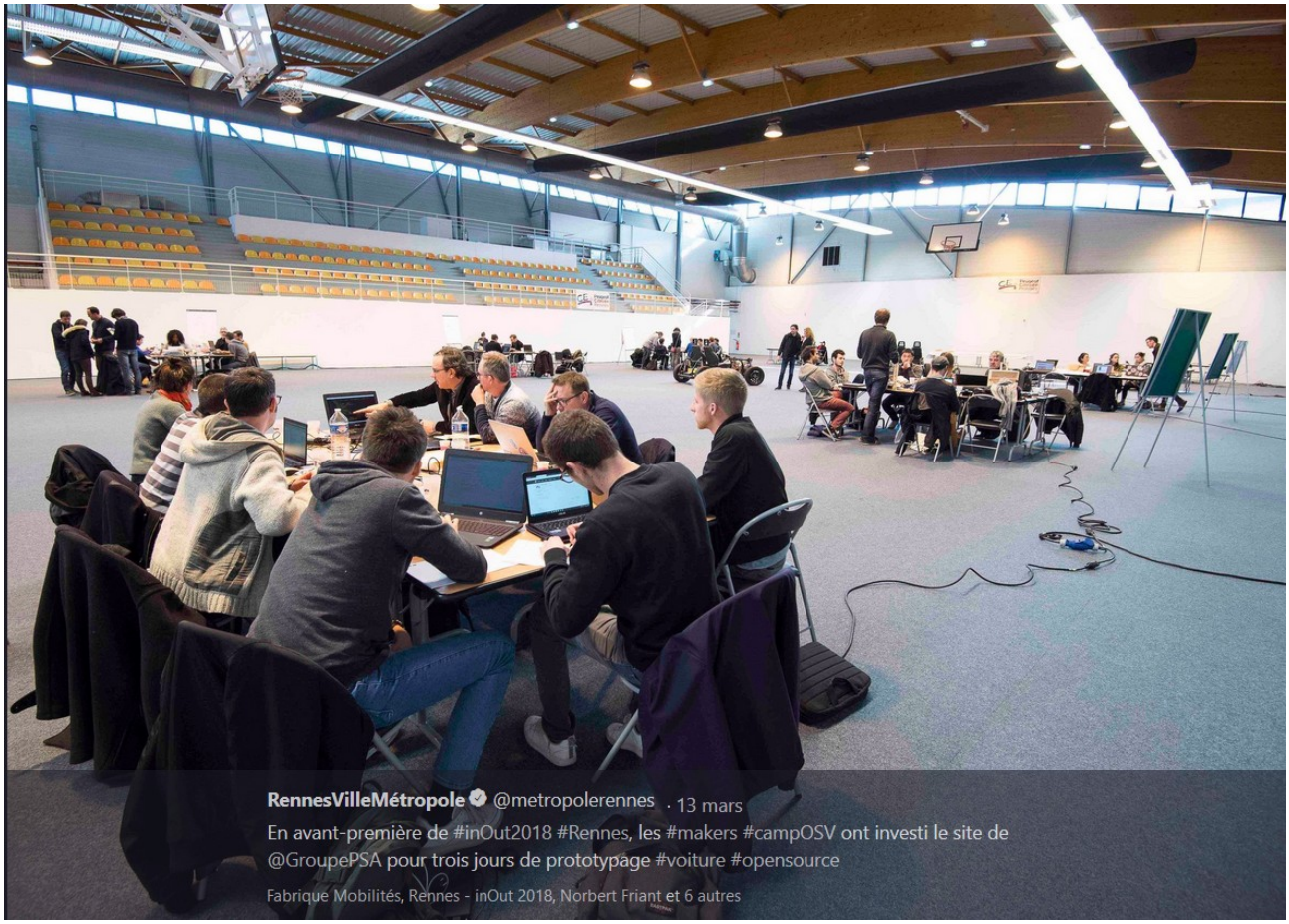


Newsletter #1

Mars 2018

Auteurs : Guillaume Florent (florentsailing-at-gmail.com), Bernard Uguen, Valerie Guichon
Repositories : https://github.com/osv-team

CampOSV 2018



Du 13 au 15 mars se déroulait à Rennes (Site de PSA / La Janais) l'édition 2018 de CampOSV, dans le cadre plus large de inOut2018 / Les Mobilités Numériques. Au programme, des conférences et des ateliers autour de l'Open Source Vehicle. L'atelier PyOSV a réuni 11 participants qui, après avoir pris en main les concepts et le code issu de la phase 1 du projet, ont apporté leur réflexion et leurs idées pour la suite. Les échanges ont principalement tourné autour des interfaces web du projet et de leur prototypage. Les modèles économiques envisageables ont également été évoqués.

Un grand merci donc à Julie, Julien, Alexandre, Mathieu, Adrien, Bernard, Laurent, Emile, Vivien et François d'avoir mis leur enthousiasme et leur intelligance au service du projet pendant ces 3 journées.

Les présentations et documents de CampOSV2018 pour PyOSV sont disponibles à l'URL:
https://github.com/osv-team/pyosv/tree/master/CampOSV_2018

Phase 2 – La roadmap

Suite à CampOSV, la 'core team' PyOSV (Thomas Paviot, Bernard Uguen, Guillaume Florent) s'est réunie afin d'intégrer les nouveau éléments à la Roadmap de la phase 2 du projet.

Communication / communauté

- Tutoriel détaillé pour la création des librairies de pièces
- Rédaction d'un lexique autour du projet
- Liste des tâches avec temps estimé et degré de difficulté (afin de faciliter l'intégration de nouveaux contributeurs)
- Documentation !

Spécifications

- Cas d'utilisations (aka Use Cases)

Qualité du projet

- Redaction d'un guide de contribution, des normes et standards de développement
- Python 3.6 +
- Packages installables via conda
- Images Docker des environnements
- Intégration Continue (Travis, Appveyor)
- Qualité de code (Codacy)

Fonctionnalités

PyOSV

- 2eme vecteur pour les ancres
- Lien entre 'Constraints' et liaisons mécaniques possibles.
- Nouveaux types de noeuds du graphe de modélisation.
- 'IDE' de conception
- Procédure d'extraction du site 'concepteur' vers le site 'fabricant'.
- Intégration FreeCAD

Web

- Site 'concepteur' / pateforme collaborative
- Site 'fabricant' / instructions de fabrication

Les modules du projet

PyOSV est un projet modulaire, à la fois dépendant de librairies externes (mais pas d'inquiétude, leur créateur et mainteneur, Thomas Paviot, fait partie du projet) et créant ses propres librairies de modélisation. Voivi un petit récapitulatif des 'briques' du projet PyOSV :

Librairies 'externes' sur lesquels PyOSV est basé

https://github.com/tpaviot/oce

Wrappers Python pour OpenCascade

https://github.com/tpaviot/pythonocc-core

https://github.com/tpaviot/pythonocc-*

Cet ensemble de librairies maintenues par Thomas Paviot permet de représenter de la géométrie 3D dans la navigateur, il permet également de manipuler cette géométrie dans le jupyter notebook (htttp://jupyter.org).

Librairies 'internes' développés pour PyOSV

https://github.com/osv-team/ccad

ccad est un module développé initiallement par Charle Sharman. Il permet de **scripter de la géométrie en Python** de façon simplifée par rapport à PythonOCC qui fait appel aux très complètes et très riches fonctions natives d'O**penCascade**. Un solide décrit par **ccad** peut être exporté vers les principaux formats de CAO. Une pièce décrite sous la forme d'un script Python est extrèmement compacte et contient **l'intention de conception,** dont la forme géométrique de l'objet n'est qu'un sous-produit.

https://github.com/osv-team/osvcad

Ce module décrit les assemblages sous la forme d'un graphe dont les noeuds sont des solides ou des **assemblages**. Les solides et les asssemblages sont liés entre eux par des **contraintes** définies via des **ancres**. Ce format est encore en cours de défintion en particulier en ce qui concerne la représentation des 11 liaisons mécaniques. De cette structure de données pourront dériver des **extractions de notice de montage-démontage** et la **BOM** (Bill of Materials ou en français Nomenclature)

https://github.com/osv-team/party

Ce module permet la création et l'utilisation des librairies de pièces. Il ne s'agit pas de remodéliser l'ensemble des pièces normalisées, il s'agit de progressivement enrichir la plateforme en construisant des librairies de pièces et **d'enrichir ainsi les catalogues accessibles.**

https://github.com/osv-team/reversy

Ce module vise à analyser des fichiers STEP existant en les décomposant en solides et assemblages élémentaires. Il s'agit également de retrouver par inférence le graphe de l'assemblage sous-jacent afin de l'importer dans les formats **osvcad.** Ce module est encore très expérimental.

Les librairies de pièces

La richesse et la viabilité du projet PyOSV passent par l'abondante présence de librairies de pièces standards (vis, écrous, roulements à bille ...). Aussi si PyOSV vous intéresse, si vos étudiants s'y intéressent, si vous cherchez un nouveau projet prometteur, si vous aimez le logiciel libre, la création de librairies de pièces est un excellent point d'entrée. Un tutoriel détaillé, en complément de la documentation *readthedocs*, sera bientôt rédigé afin de vous aider à concevoir les librairies de manières inédite et innovante. Une fois la méthode maitrisée, la création d'un librairie prend entre 1 et 3 jours, en fonction de sa complexité , et représente un pas supplémentaire vers un système libre de conception d'objets complexes.