

# Résumé TPI

Anthony Jaccard SI-CA2a 2018-2019

## Situation de départ :

### Objectif :

Créer une application munie d'une interface graphique et permettant de piloter un robot sur un réseau local et d'afficher des images prises par ce dernier

### Outils :

- Programmation : Python, tkinter, subprocess
- Logiciel : OpenSSH, Visual Studio Code (+ extension Python), Github Desktop, Word, Excel
- Matériel : Ordinateur CPNV, Robot, Carte Wifi USB, Point d'accès Wifi, Chargeur, Multimètre

## Mise en œuvre :

### Approche :

Afin de simplifier le développement et plus tard l'installation de l'application, j'ai décidé de n'utiliser que les modules standard de python. L'interface est donc entièrement programmée à l'aide du toolkit tkinter dont j'ai utilisé certaines classes pour en dériver les éléments constituant de mon interface. La communication se fait à l'aide des fonctions du module subprocess qui fait appel aux commandes « ssh » et « scp » installées à l'aide d'openSSH. Les classes Setting et View (ainsi que les classes dérivées) dérivent respectivement des classes Frame et LabelFrame de tkinter et composent l'interface de l'application. Les classes NetControls, NetCamera, SettingSaver et Logger regroupent à elles quatre les fonctionnalités d'arrière-plan de l'application, à savoir piloter le robot, ordonner la prise de vue au robot, sauvegarder les paramètres sur le bon périphérique et sauvegarder les logs générés par le logiciel.

### Problèmes rencontrés :

- Carte Wifi USB fait planter l'ordinateur quand branchée sur un port USB3
- Type d'ouverture de session précis nécessaire afin de charger les bons PATH
- Pas de connexion persistante avec les subprocess
- Gestion des formats d'image entre tkinter et l'API de la caméra du robot limitée
- Connexion lente
- Temps de chauffe de la caméra

## Résultat :

L'application est fonctionnelle même si elle ne fonctionne pas aussi bien qu'elle le devrait : des latences conséquentes ainsi que quelques problèmes d'ergonomie persistent. Néanmoins, le pilotage et la prise de vue ainsi que l'affichage de l'image dans l'application sont possibles à condition d'avoir un peu de patience. Quelques améliorations tant au niveau de la structure du code ainsi qu'au niveau de ses fonctionnalités sont envisageables voire nécessaires mais l'expérience acquise au cours de ce TPI permet d'avoir déjà quelques pistes quant à la manière d'approcher ces améliorations et quels outils utiliser et, moyennant un peu plus de temps, seraient parfaitement réalisables.