

---

## **Journal de travail**

last modified by Forestier Robin

on 2022/04/04 20:53

---

# Table of Contents

Semaine 1 .....	3
Lundi 7 mars 2022 .....	3
Mardi 8 mars 2022 .....	3
Mercredi 9 mars 2022 .....	4
Semaine 2 .....	4
Lundi 14 mars 2022 .....	4
Mardi 15 mars 2022 .....	4
Mercredi 16 mars 2022 .....	5
Semaine 3 .....	5
Lundi 21 mars 2022 .....	5
Mardi 22 mars 2022 .....	5
Mercredi 23 mars 2022 .....	6
Semaine 4 .....	6
Lundi 28 mars 2022 .....	6
Mardi 29 mars 2022 .....	6
Mercredi 30 mars 2022 .....	6
Semaine 5 .....	7
Lundi 4 avril 2022 .....	7
Mardi 5 avril 2022 .....	7

## Contents

- [Semaine 1](#)
  - [Lundi 7 mars 2022](#)
  - [Mardi 8 mars 2022](#)
  - [Mercredi 9 mars 2022](#)
- [Semaine 2](#)
  - [Lundi 14 mars 2022](#)
  - [Mardi 15 mars 2022](#)
  - [Mercredi 16 mars 2022](#)
- [Semaine 3](#)
  - [Lundi 21 mars 2022](#)
  - [Mardi 22 mars 2022](#)
  - [Mercredi 23 mars 2022](#)
- [Semaine 4](#)
  - [Lundi 28 mars 2022](#)
  - [Mardi 29 mars 2022](#)
  - [Mercredi 30 mars 2022](#)
- [Semaine 5](#)
  - [Lundi 4 avril 2022](#)
  - [Mardi 5 avril 2022](#)

# Semaine 1

## Lundi 7 mars 2022

### Matin:

Création du planning ([Planning\\_TPI\\_FOR\\_v1](#)) et répartition du temps pour les différentes activités.  
Planning à jour: [Planning\\_TPI\\_FOR\\_v2](#)  
Création du GitLab avec la structure de fichier de l'atelier. [GitLab - TPI\\_Gestion\\_Lumiere\\_KNX](#)  
Création des deux pages XWiki: **Journal de travail** et [Documentation](#)

### Après-midi:

J'ai réalisé toutes les installations nécessaires sur mes 3 Raspberry Pi. (OpenCV / GitLab / ...)  
J'ai aussi modifié les paramètres de gunicorn et de nginx sur le Raspberry Pi serveur pour que les liens vers le nouveau dossier de travail soit correct.  
Gunicorn : /etc/systemd/system/lightcontrol.service  
NginX : /etc/nginx/sites-available/lightcontrol

J'ai aussi commencé la programmation, en réalisant la sélection des portes.  
Le programme est normalement fonctionnel mais pas complètement commenté.

## Mardi 8 mars 2022

### Matin:

J'ai commencé par commenter complètement le fichier Select\_door.py.  
J'ai réalisé un script pour la détection des portes grâce au "matching template". Ce code pourra être utile si la caméra bouge de sa position initiale.

J'ai ensuite commencé la détection des personnes. Pour détecter les personnes, je réalise une soustraction d'images avec l'algorithme (createBackgroundSubtractorKNN). Mes résultats actuels sont fiables et utilisables mais peuvent être améliorés.

### Après-midi:

Création d'un code plus structuré et documenté (personne\_detect.py) et création de la class PersonneDetect.  
J'ai aussi pu réaliser des tests pour le calcul de l'angle de déplacement permettant de savoir la trajectoire de la personne en mouvement.

## Mercredi 9 mars 2022

### Matin:

Avec les tests d'hier après-midi, j'ai pu créer le code `personne_tracking.py` et la class `PersonneTracking`. Je peux dès à présent calculer l'angle de déplacement d'un objet en mouvement ou d'une personne. Le tracking n'est pas efficace à 100% mais suffit pour ce projet.

### Après-midi:

J'ai connecté les Raspberry Pi sur les serveurs NTP Suisse ([ch.pool.ntp.org](http://ch.pool.ntp.org)). J'ai réalisé quelques améliorations sur le tracking des personnes.

J'ai pu remarquer plusieurs problématiques. Premièrement, si deux personnes se trouvent proches, le système ne les différenciera pas. Deuxièmement, je dois encore réaliser un tri des résultats car on trouve des faux positifs dans la détection.

J'ai aussi décelé plusieurs améliorations possibles au système, comme un positionnement des caméras au-dessus des portes regardant directement le sol. Cela simplifierait la détection et permettrait un comptage précis des personnes.

Après discussion avec Monsieur Dupertuis, je vais terminer proprement la détection des personnes et commencer l'interface graphique. Je vais aussi simplifier un peu la détection et s'il me reste du temps à la fin de mon TPI, je reviendrai dessus.

## Semaine 2

### Lundi 14 mars 2022

#### Matin:

J'ai optimisé le code `entry_detect.py` en réalisant un tri des informations dans la class `info`. Pour valider la sortie ou l'entrée d'une personne, il faut au minimum trois fois des informations de détection identiques. Cela vient simplement diminuer le nombre d'erreurs et augmente le pourcentage de détections correctes.

#### Après-midi:

J'ai terminé l'optimisation du code `entry_detect.py`. Il est maintenant fonctionnel. Il me faudra encore tester les caméras avec la vision nocturne.

J'ai discuté avec Monsieur Dupertuis pour clarifier deux points de mon cahier des charges:

1. Sur le tableau, les cases bleues représentent une extinction automatique (sans prendre en compte les caméras).
2. L'accès aux settings depuis l'écran tactile doit se faire avec un mot de passe. Mais ce mot de passe doit pouvoir être rentré sans clavier externe.

Pour ce deuxième point, il faudra que l'utilisateur puisse taper le mot de passe sur un clavier visible sur l'écran.

### Mardi 15 mars 2022

#### Matin:

Pour répondre à la problématique du mot de passe pour l'écran tactile, j'ai créé la page login 2. Cette page est uniquement accessible si on se connecte depuis le serveur. Cette page affiche un clavier avec les chiffres de 0 à 9 avec un bouton "effacer" et un bouton "submit". Création de la page HTML settings est gestion des autorisations pour l'utilisateur "local" (écran tactile).

#### Après-midi:

Amélioration de la page settings et ajout des différentes sections.

- Tableau 1.

- Tableau 2.
- Ajout des heures.
- Mode manuel.
- Heures manuelles.
- Caméras.
- Modification des mots de passe.

## Mercredi 16 mars 2022

### Matin:

Implémentation de l'heure et de la date manuelle. Pour cela je viens récupérer uniquement le jour de la semaine et la différence de temps entre l'heure réelle et l'heure entrée par l'utilisateur. Quand le système récupère l'heure pour l'afficher il reçoit l'heure plus la différence.

J'ai aussi réalisé la réinitialisation du système à minuit.

### Après-midi:

Sous la section **caméras** dans la page settings, les champs **Online** et **Offline** se mettent à jour automatiquement. Pour le savoir, je réalise un "ping" sur l'adresse IP de chacune des caméras. Cette action est réalisée dans la fonction run\_job.

J'ai aussi commencé la communication entre les caméras et le serveur WEB. La communication se fait grâce à une request http envoyée sur une page non accessible par les utilisateurs. Pour reconnaître une caméra, je viens vérifier son adresse MAC (après discussion avec Monsieur Dupertuis) et non son adresse IP.

## Semaine 3

### Lundi 21 mars 2022

#### Matin:

J'ai commencé l'implémentation de la lecture des tables horaires. J'ai pu ajouter l'extinction automatique et l'allumage automatique des lumières suite à une requête des caméras. Mais un problème de Wi-Fi est arrivé et a stoppé mon avance. Je vais donc réaliser de la documentation jusqu'au retour du Wi-Fi.

#### Après-midi:

Après rétablissement de la connexion Wi-Fi, j'ai pu reprendre et j'ai enfin passé le serveur sur le Raspberry Pi. Je me suis heurté à un problème. Le système n'arrivait pas à exécuter les commandes Ping et Arp.

### Mardi 22 mars 2022

#### Matin:

Après plus amples vérifications j'ai découvert que le problème venait de la configuration de mon serveur Unicorn.

Après cette modification le programme était de nouveau fonctionnel.

J'ai enfin pu tester l'entièreté de mon projet.

J'ai remarqué quelques petits bugs qui ont été corrigés.

1. La sauvegarde des portes ne se faisait pas dans le bon dossier.
2. L'affichage online/offline des caméras ne fonctionnait pas.

Après quelques tests, j'ai remarqué que mon projet fonctionnait selon mon cahier des charges, mais la détection n'est pas à la hauteur pour les deux portes les plus éloignées des caméras. Je vais d'abord essayer de trouver une autre place pour cette caméra.

J'ai marqué plus haut qu'un comptage était possible avec les caméras positionnées directement au-dessus des portes, je discuterai de cela avec Monsieur Dupertuis demain pour proposer cette amélioration au vu de mon avance sur le planning.

**Après-midi:**

Après discussion avec Monsieur Dupertuis, il m'a conseillé de solidifier mon projet et assurer qu'il répond au cahier des charges avant de me lancer dans l'amélioration.

Mise à jour de l'entièreté des commentaires dans mes différents codes.

## Mercredi 23 mars 2022

**Matin:**

Explication détaillée du fonctionnement du système dans ma documentation.

**Après-midi:**

Explication de la programmation du système et du calcul des angles dans ma documentation.

## Semaine 4

### Lundi 28 mars 2022

**Matin:**

J'ai commencé la réalisation de l'amélioration.

Pour cela, j'ai filmé plusieurs personnes entrer et sortir avec la caméra placée au-dessus de la porte.

J'ai pu réutiliser mes codes: `personne_detect`, `personne_tracking` et `entry_detect`. Mais j'ai supprimé tous les calculs d'angles et je les ai simplifiés.

La détection marche très bien, il faudra encore la tester en condition réelle mais les résultats sont pour le moment excellents.

Après discussion avec Monsieur Dupertuis, je vais tester cette nouvelle méthode de détection sur deux des trois portes, car pour la porte rouge, il faudra réaliser des travaux pour la fixation de la caméra, cela sera réalisé après le TPI si la détection est fonctionnelle.

**Après-midi:**

J'ai fixé les deux caméras sur du bois pour leur ajouter du poids et j'ai ensuite réalisé 2 rallonges pour les raccorder.

Après plusieurs tests concluants, j'ai ajouté un comptage des personnes.

Pour réaliser un test plus réel de mon système, je vais le laisser fonctionner et regarder demain matin si les personnes ont été comptées.

### Mardi 29 mars 2022

**Matin:**

Le test que j'ai voulu réaliser n'a pas fonctionné car le programme de la caméra a "freeze" au milieu de la nuit. Je pense que le problème vient de la température du Raspberry. Après redémarrage le comptage a bien fonctionné.

**Après-midi:**

J'ai réalisé le branchement de la deuxième caméra à travers le faux plafond.

J'ai pu tester le système avec les deux caméras en fonction.

J'ai aussi ajouté la vérification du comptage avant extinction automatique.

### Mercredi 30 mars 2022

**Matin:**

J'ai de nouveau laissé les caméras activées toutes la nuit, le même phénomène s'est produit. Après un certain temps les caméras "freez".

J'ai donc essayé plusieurs modifications pour trouver d'où vient le problème.

Premièrement, j'ai affiché la température en temps réel. Au moment du "freez" la température était encore de 80 °C et n'avait pas varié depuis longtemps. J'ai ensuite essayé de lancer mon programme en "nohup", en arrière-plan. Sans changement.

#### **Après-midi:**

J'ai poursuivi les tests en vérifiant l'utilisation de la mémoire et du CPU.

J'ai aussi lancé le programme en mode debug.

Le programme semble toujours se stopper à la demande de l'image à la caméra.

Je vais ventiler mon Raspberry Pi et la caméra pour faire descendre la température.

## **Semaine 5**

### **Lundi 4 avril 2022**

#### **Matin:**

Après tous les tests de la semaine dernière, je pense que la température de la caméra devient après une longue période de temps, trop élevée. Ce qui stoppe celle-ci.

Pour le tester, je vais utiliser un Fluke et un thermocouple pour mesurer sa température.

Avec une température ambiante à 24 °C, ma caméra monte en moins d'une heure à 65 °C et les LED infrarouge à 70 °C. Je vais donc ajouter des dissipateurs thermiques et refaire ces mesures.

#### **Après-midi:**

Avec les dissipateurs thermiques, la température n'augmente pas au-dessus de 55 °C environ.

J'ai créé les deux images de mes Raspberry Pi.

Liens:

OMVSERVER\formation\TPI\TPI\_Forestier\_Gestion\_Lumiere\_KNX, **RPi\_server.img** et **RPi\_camera.img**.

### **Mardi 5 avril 2022**

#### **Matin:**

Finalisation de la documentation.

#### **Après-midi:**

Impression des documents et des codes.

Rendu du projet, fin du TPI.