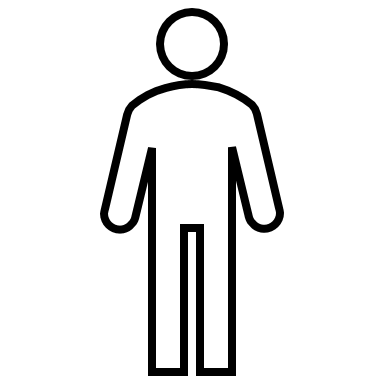
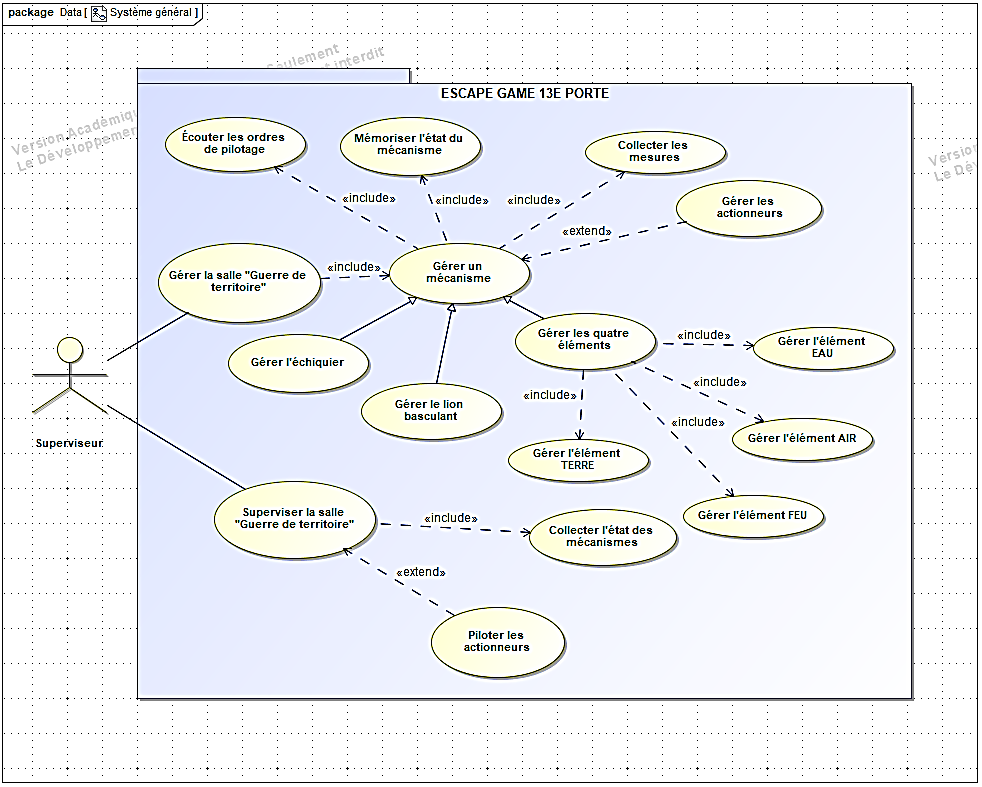
Revue de Projet n°1

***Présentation du projet***

La société *La 13EME PORTE* est une enseigne d’Escape Game. Les gérants de la société souhaitent que le système technique d’une salle actuellement en place soit recréé entièrement afin de corriger les erreurs de conception et les différents bugs existants. De plus il est demandé d’ajouter plusieurs fonctionnalités permettant d’améliorer le travail du superviseur.  
  
Le système de la salle s’appuie sur 9 mécanismes et il devra associer à chacun des mécanismes un Arduino dédié.

Un mécanisme pourra collecter des mesures par l’intermédiaire d’actionneur (capteurs, boutons, etc…) placé sur la carte Arduino et devra écouter les ordres de pilotage.  
De plus chaque mécanisme transmettra régulièrement par liaison I2C à la Raspberry les valeurs de ses différentes entrées.   
Chaque information reçue par la Raspberry sera enregistrée en base de données et installée sur un serveur positionné dans le bâtiment par l’intermédiaire du poste de supervision.  
Sur ce poste de supervision on pourra visualiser l’état de la salle et piloter les actionneurs.

***Diagramme de cas d’utilisation***

***Répartition des tâches***

* Etudiant 1 : Constantin
* Gérer le Mécanisme 4
* Gérer le Mécanisme 8
* Gestion du bus I2c entre Arduino et Raspberry :

Envoie des ordres de pilotage de la Raspberry à l’Arduino et les traiter

Envoi des mesures et des états des actionneurs sur la Raspberry et les traiter

* Etudiant 2 : Corentin
* Créer la base de données
* Installer les serveurs Apache et MySql
* Gérer le Mécanisme 6
* Gérer le Mécanisme 7
* Récupérer la valeur des capteurs et l’état des actionneurs de la Raspberry au serveur
* Etudiant 3 : Joshua
* Configuration réseau du matériel
* Création d’une application Web sur l’affichage de l’état de la salle
* Gérer le Mécanisme 1
* Gérer le Mécanisme 2
* Gérer le Mécanisme 3
* Etudiant 4 : Thomas
* Création d’une application Web de pilotage des actionneurs
* Gérer le Mécanisme 5
* Gérer le Mécanisme 9
* Envoie des ordres de pilotage du PC de supervision à la Raspberry

Thomas

Joshua

Corentin

******

Constantin

Constantin

Thomas

Joshua

Corentin

**Application Web de pilotage des actionneurs**

**Application Web sur l’affichage de l’état de la salle**

***Matériels utilisés***



* Raspberry Pi 3B :

**Procédure d’installation :**

## *Installer Raspbian à partir d’une carte SD vierge*

**Etape 1 : Insérez votre carte SD dans votre ordinateur et lancez Etcher**

**Etape 2 : Décompresser l’archive**

**Etape 3 : Télécharger Etcher**

Etcher est un logiciel qui va permettre d’installer Raspbian (ou n’importe quel système d’exploitation) sur la carte SD et de le rendre directement bootable.

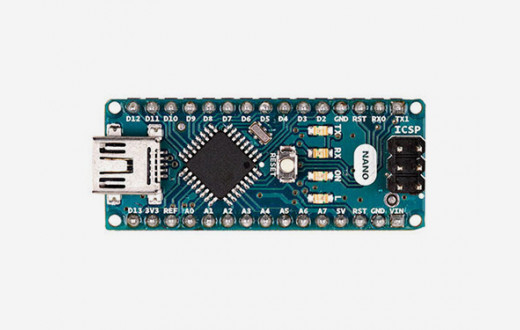
**Etape 4 : Insérer votre carte SD dans votre ordinateur et lancez Etcher**

**Etape 5 : Flasher la carte SD avec Raspbian**

**Programmation sur Raspberry :**

Langage : Python

Nous utiliserons l’émulateur de Terminal PuTTY afin de pouvoir programmer directement depuis notre PC sur la Raspberry

* Arduino :

**Matériel Arduino pour la salle de jeu**

* Neuf Arduino Nano
* Quatre capteurs à effet hall A3144
* Module de commutateur de clé 3 pin 1NO1NC
* Afficheur 7 segments
* Capteur d’humidité
* Interrupteur à bascule
* Sept capteurs photosensibles
* Boutons poussoir
* Leds
* Capteur de fin de course
* Capteur de poids HX711

**Programmation sur Arduino :**

Langage : Arduino

IDE : Arduino

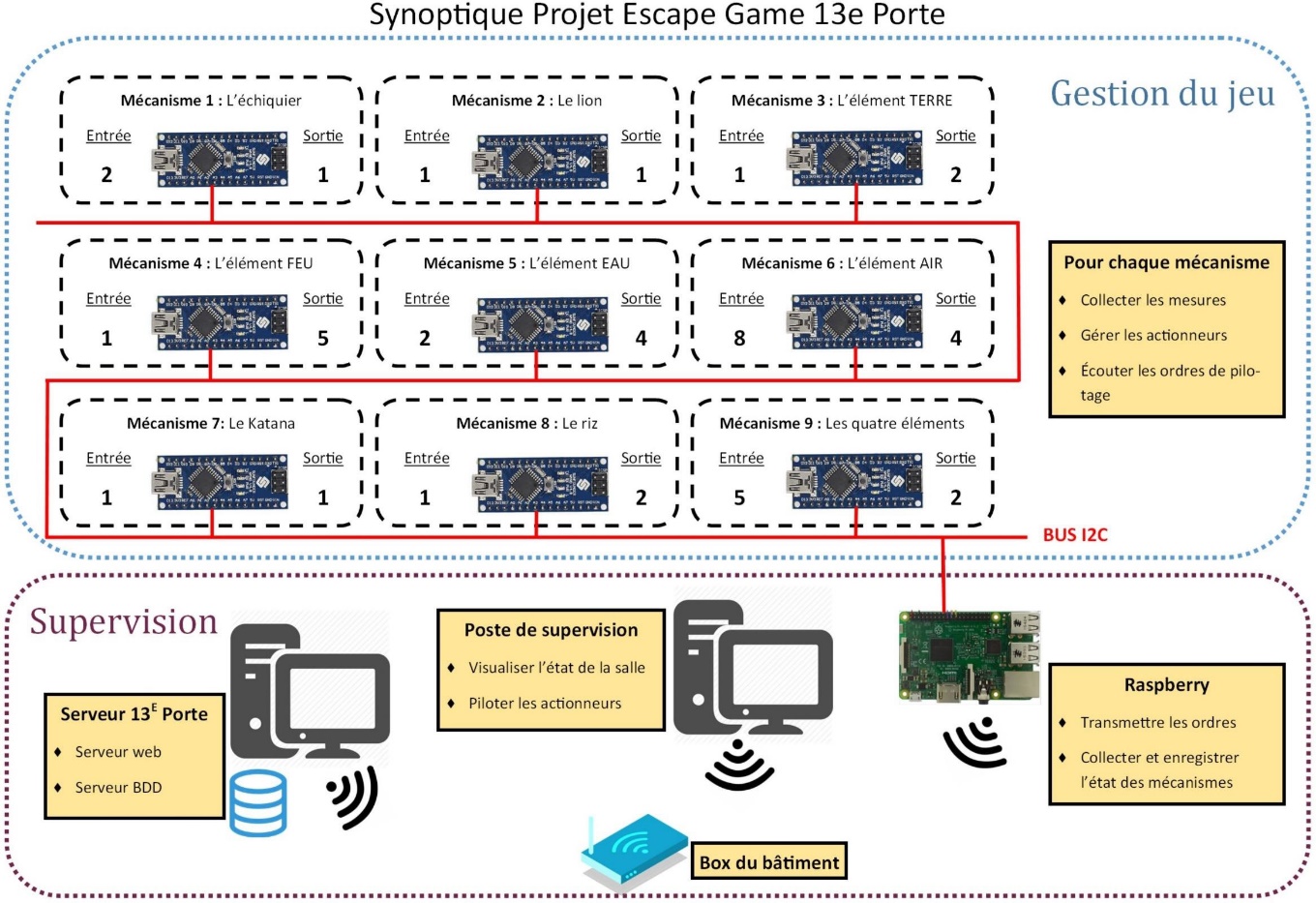
Nous utiliserons l’émulateur de Terminal PuTTY afin de pouvoir programmer directement depuis notre PC sur l’Arduino

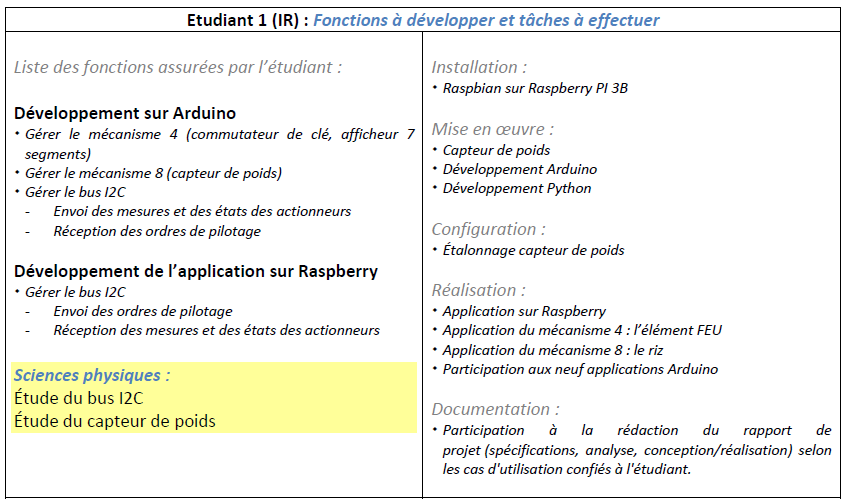
• Un poste informatique pour le serveur *13e Porte*

• Un poste informatique de supervision (éventuellement le même poste que le serveur)

***Ma tâche***

Ma tâche





***Liaison I2c***

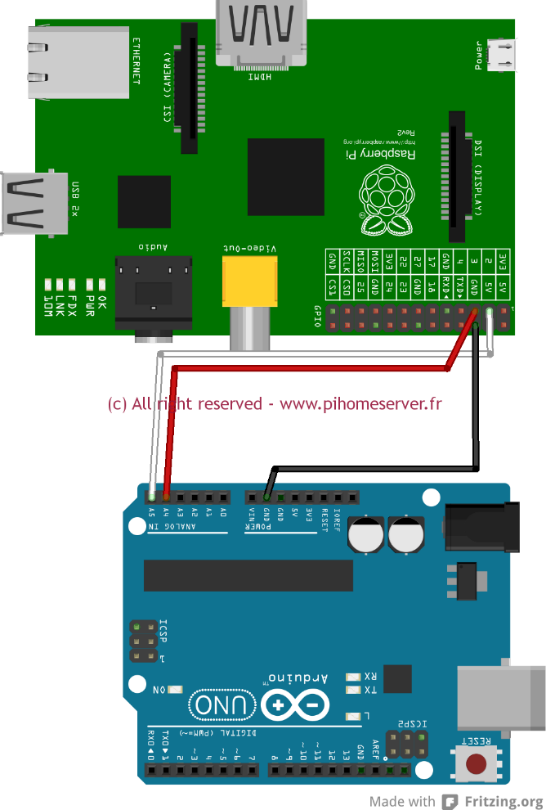
Pour lier les Arduino à la Raspberry nous allons passer par le protocole i2C :   
- pour envoyer l’état des actionneurs et les valeurs des capteurs des Arduino à la Raspberry   
- pour envoyer les ordres de pilotage du PC de supervision de la Raspberry aux Arduino.   
  
Plusieurs raisons d’utiliser l’i2c :  
- ne pas utiliser le port USB, le Raspberry Pi étant limité  
- parce qu’il y a beaucoup de Pin, assez pour faire communiquer 9 Arduino avec la Raspberry

Le protocole I2C, créé par Philips avec une orientation domotique, permet à plusieurs composants de dialoguer entre eux de manière bidirectionnelle mais en half-duplex uniquement. Et grâce à seulement trois fils : Un signal de donnée (SDA), un signal d’horloge (SCL), et un signal de référence électrique (Masse).

Les données sont transmises en série à 100Kbits/s en mode standard et jusqu’à 400Kbits/s en mode rapide.

Chaque périphérique sur le bus I2C est adressable, avec une adresse unique pour chaque périphérique du bus

Il y a un principe de maitre et esclave. Ici nous utiliseront le Raspberry Pi en tant que maitre et les Arduino en tant qu’esclave.   
Le maitre (la Raspberry) est le composant qui initialise un transfert, génère le signal d’horloge et termine le transfert. Dans notre cas il sera récepteur et émetteur.  
Les esclaves (Les Arduino) sont les composants adressés par un maître. Dans notre cas ils seront récepteurs et émetteurs.

Une image contenant capture d’écran

Description générée automatiquement

Une image contenant capture d’écran

Description générée automatiquement***Description des mécanismes***

* Mécanisme 4 :

Gestion de la clé (Elément FEU)

Description du Sous-système : Un interrupteur à clef est caché dans le décor, les joueurs doivent trouver la clef et la tourner dans l’interrupteur

Résultat :

1. Une LED témoin s’allume sur le tableau de contrôle
2. Une « Tête de Dragon » sort du plafond (S2). En effet, la désactivation de l’électroaimant a pour effet de libérer une trappe au plafond
3. Une machine à fumée (220 Volt) est allumée via un relais. Cette machine à fumée est dissimilée dans la « tête de dragon » afin que ce dernier crache de la fumée lorsqu’il apparait aux joueurs
4. 5 LED blanches s’allument afin au-dessus afin d’éclairer cette dernière
5. L’élément FEU (LED) s’allume sur la tablette à destination des joueurs

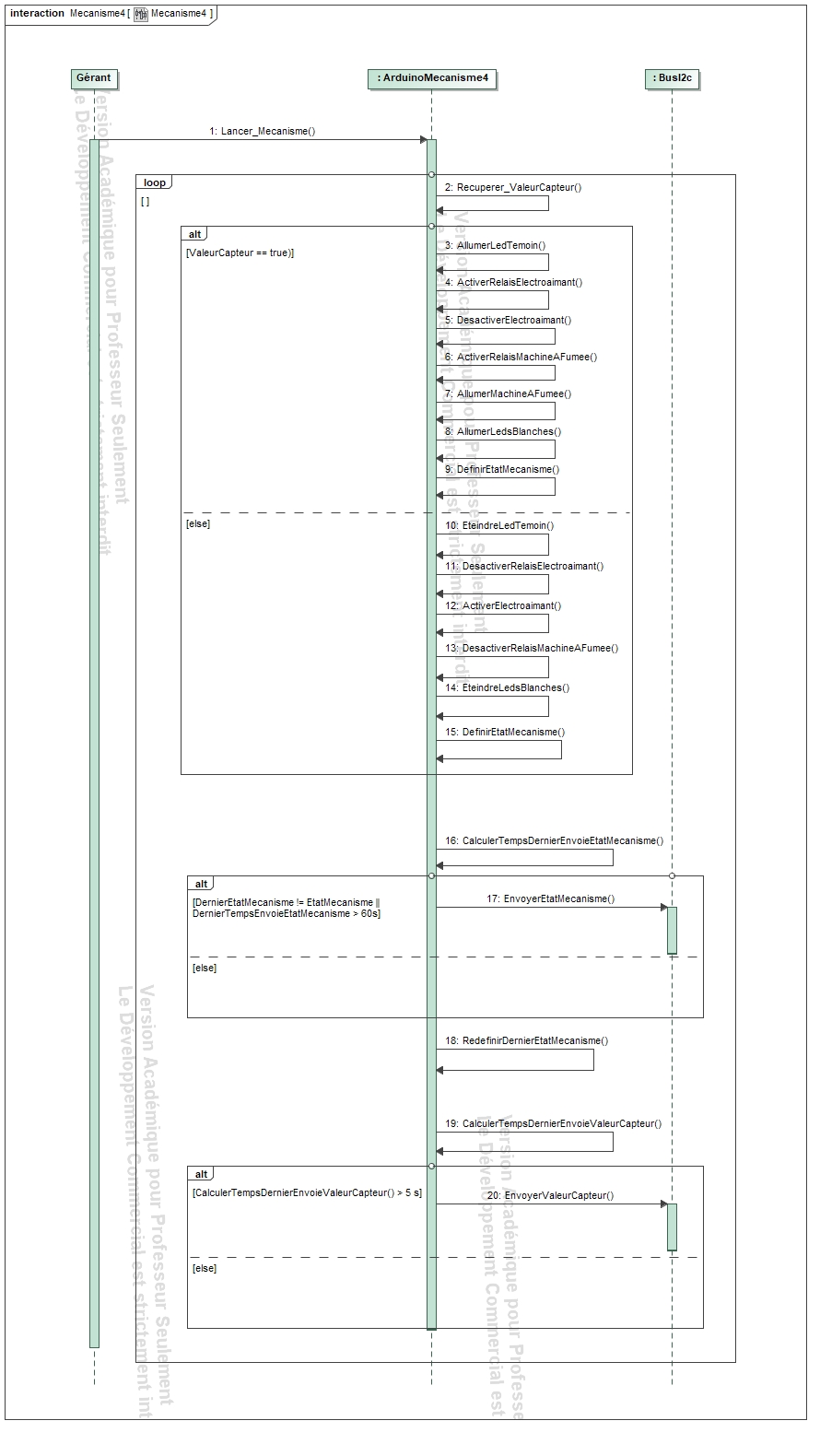
## Entrées :

* E1
  + Type de capteur : Interrupteur à clef
  + Référence : Commutateur de clé 3 pin 1NO1NC
  + Signal : numérique
  + Voltage : 0-5V

## Sorties :

* m\_dragon
  + Rôle : Activer/désactiver un électroaimant via un relais + LED de contrôle
  + Condition : Si E1 est à 1 alors activer la sortie, sinon, désactiver la sortie
  + Voltage : 0-5V
* m\_fumee
  + Rôle : Activer/désactiver une machine à fumée (220Volt) via un relais
  + Condition : Si E1 est à 1 alors activer la sortie, sinon, désactiver la sortie
  + Voltage : 0-5V
* m\_led
  + Rôle : Activer/désactiver 5 LED blanche câblées en parallèles
  + Condition : Si E1 est à 1 alors activer la sortie, sinon, désactiver la sortie
  + Voltage : 0-5V (5 Volts + résistance de 220 ohms)

* m\_feu
  + Rôle : entrée de la gestion des quatre éléments



* Mécanisme 8 :

# Gestion de la balance de riz et du tableau

Description du Sous-système : les joueurs doivent faire « une offrande » de 50 grammes de riz sur coupelle posée sur un capteur de poids. La condition est validée avec une précision de +/- 2 grammes.

Résultat :

1. La LED rouge s’éteint et la LED verte s’allume
2. Une LED témoin s’allume au panneau de contrôle.
3. Un électroaimant (12V) est désactivé via un relais (5V) libérant ainsi la chute d’un tableau (indice)

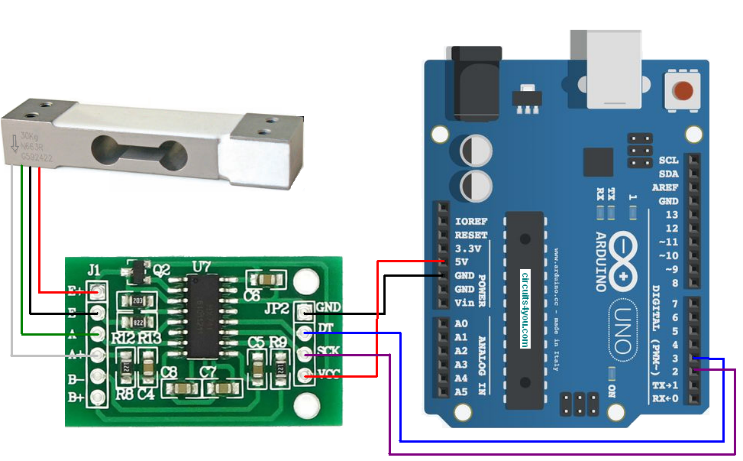
## Entrées :

* E1
  + Type de capteur : capteur de poids
  + Référence : HX711
  + Signal : numérique
  + Voltage :  0-5V

## Sorties :

* S\_Tableau
  + Rôle : activer/désactiver une électro aimant via un relais (tableau)
  + Condition : Si E1 est compris entre 48 grammes et 52 grammes alors activer la sortie, sinon, désactiver la sortie
  + Voltage : 0-5V
* S\_Led
  + Rôle : Désactivation LED rouge et activation LED verte (Indicateur de réussite)
  + Condition : Si E1 est compris entre 48 grammes et 52 grammes alors éteindre la LED rouge et allumer la LED verte, sinon c’est l’inverse
* Voltage : 0-5V

## Remarques :

  
Schéma de branchement du capteur de poids HX711

