**SFL5 : Projet Escape Game**

**Sommaire**

[I. Introduction 2](#_Toc39604475)

[Rappel du cahier des charges 2](#_Toc39604476)

[Répartition des tâches 4](#_Toc39604477)

[II. Analyse 5](#_Toc39604478)

[Mécanisme N°1 : L’Échiquier 5](#_Toc39604479)

[Diagramme de classe 5](#_Toc39604480)

[Mécanisme N°2 : Le lion basculant 7](#_Toc39604481)

[Mécanisme N°3 : L’élément Terre 9](#_Toc39604482)

[Application Web de supervision 11](#_Toc39604483)

[III. Réalisation du projet 13](#_Toc39604484)

[Rappel de la tâche de l’étudiant 13](#_Toc39604485)

[Mécanisme N°1 : L’Échiquier 14](#_Toc39604486)

[Mécanisme N°2 : Le lion basculant 14](#_Toc39604487)

[Mécanisme N°3 : L’élément Terre 14](#_Toc39604488)

[Application Web de supervision 15](#_Toc39604489)

[Partie réseau 20](#_Toc39604490)

# Introduction

## Rappel du cahier des charges

### Objectifs du projet

Mme. Sterenn LE GOFFIC et M. Jean-Charles Douguet, les deux gérants de la société, souhaitent que le système technique actuellement en place soit recréé entièrement afin de corriger les erreurs de conception et les différents bugs existants. Il est également demandé l’ajout de plusieurs fonctionnalités permettant d’améliorer le travail du superviseur.

### Système de gestion de la salle

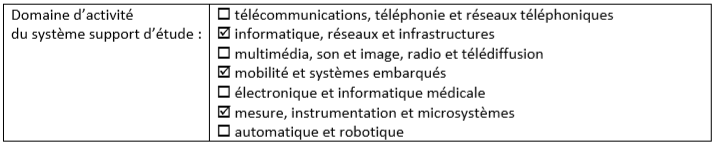
Actuellement, les nombreux mécanismes du jeu dans la salle sont gérés par quatre Arduino UNO. Chaque Arduino a la charge de plusieurs mécanismes distincts. Les gérants souhaitent revoir ce découpage dans le nouveau système en associant à chacune des neuf étapes du jeu un Arduino dédié. Une étape sera appelée mécanisme dans la suite de ce document. Les différents mécanismes du jeu sont nommés ci-dessous.

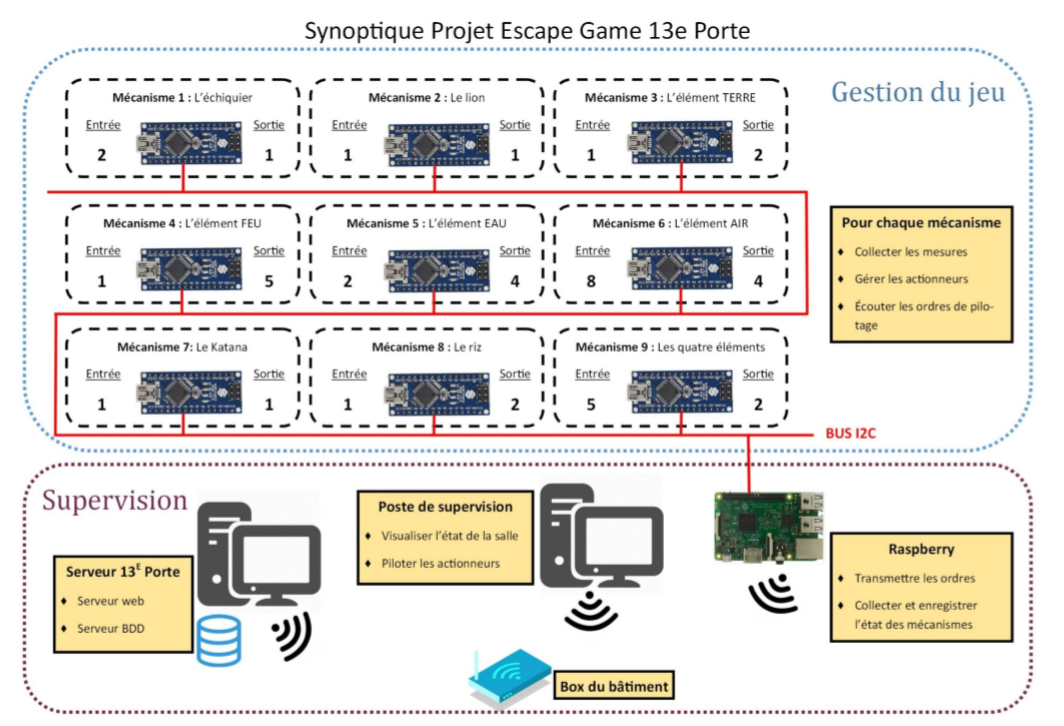
**Mécanisme 1.** L’échiquier   
**Mécanisme 2.** Le lion   
**Mécanisme 3.** L’élément TERRE   
**Mécanisme 4.** L’élément FEU   
**Mécanisme 5.** L’élément EAU   
**Mécanisme 6.** L’élément AIR   
**Mécanisme 7.** Le Katana   
**Mécanisme 8.** Le riz   
**Mécanisme 9.** Les quatre éléments

### Système de supervision

Le commanditaire souhaite disposer de deux fonctionnalités dédiées à la supervision et utiles à la maintenance :

• Le superviseur pourra visualiser depuis un ordinateur l’état du système. L’interface devra afficher à la fois l’état d’avancement dans le jeu, mais aussi la valeur de chaque capteur et actionneur.   
• Afin de faciliter la maintenance, une interface devra permettre de piloter depuis un ordinateur chaque actionneur.

Situation du projet dans son contexte  
  
image 1 – Domaine d’activité du projet

Synoptique Projet Escape Game 13ème Porte  
  
image 2 – Synoptique du projet

### Cahier des charges – Expression du besoin

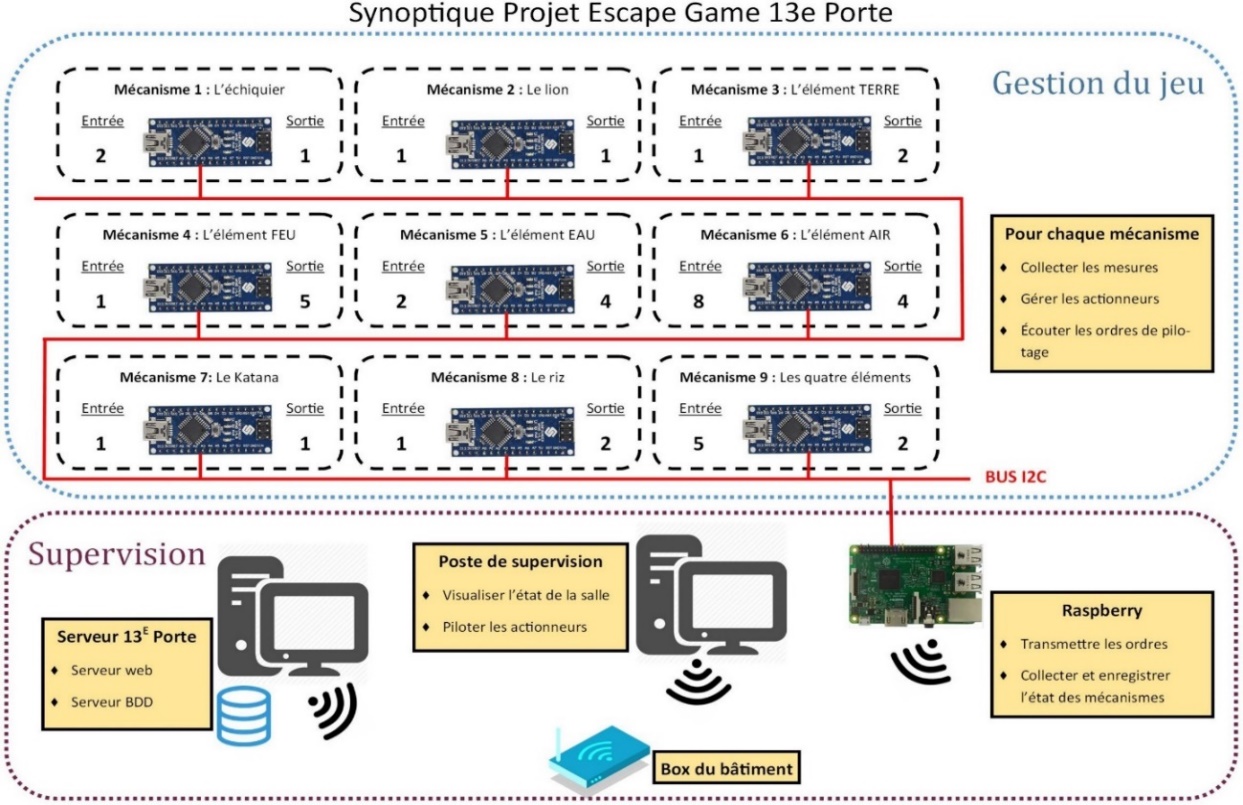
Le projet répondra aux besoins suivants :

✓ Gérer les neufs mécanismes du jeu   
✓ Collecter et enregistrer l’état de la salle (valeur de chaque capteur, état de chaque actionneur)   
✓ Visualiser l’état de la salle (valeur de chaque capteur, état de chaque actionneur)   
✓ Piloter les actionneurs

### Gérer les neufs mécanismes du jeu

Afin d’assurer la confidentialité du scénario, nous présentons dans ce document les différents mécanismes sans préciser la nature exacte des objets à manipuler dans la salle de jeu. Toutefois, les descriptions permettront d’évaluer le travail à réaliser.

## Rappel des tâches de l’étudiant



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

Joshua

* 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

image 3 – Synoptique du projet avec les tâches de l’étudiant

# 

# Mécanisme N°1 : L’Échiquier

## Synopsis du mécanisme

Le mécanisme repose sur deux **capteurs à effet Hall**.

Selon une condition portant sur les valeurs de ces deux capteurs, un électroaimant est activé ou désactivé via un relais et une LED s’allume ou non sur le tableau de contrôle.

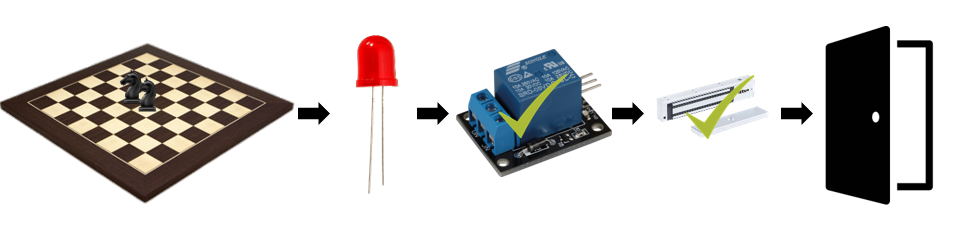
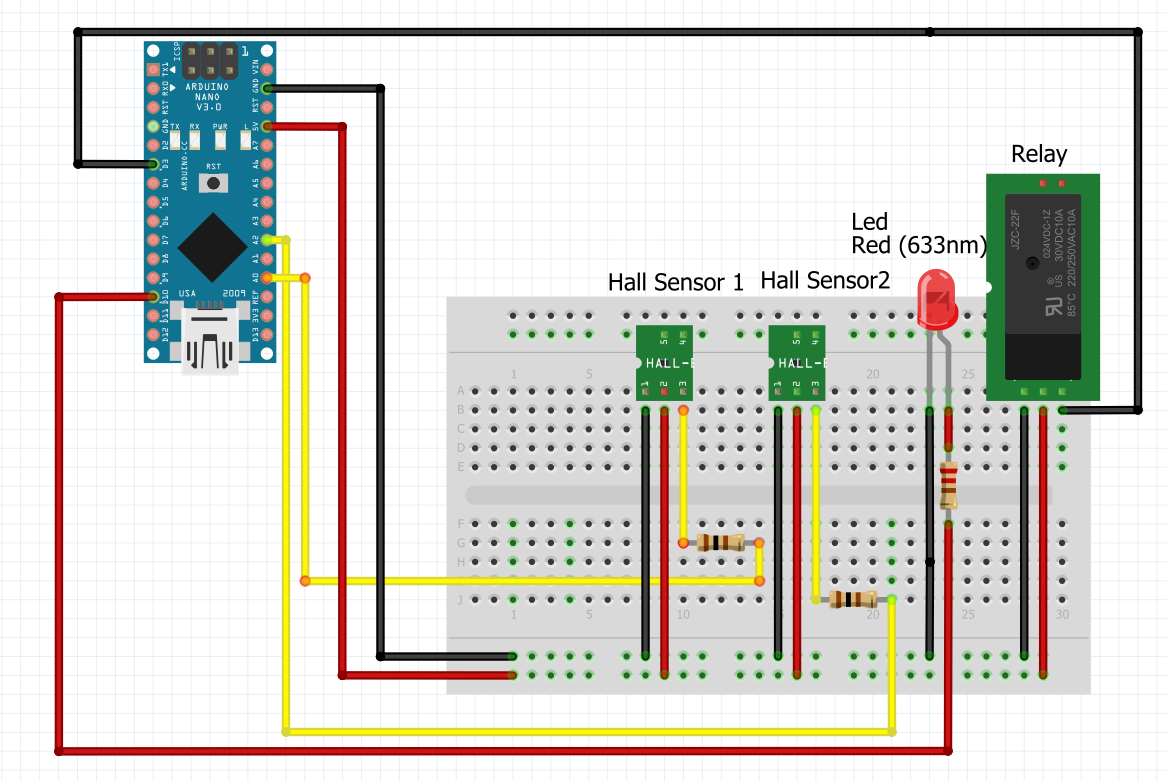
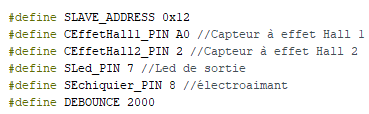
  
image 4 – Schéma explicatif du mécanisme

Schéma électrique du mécanisme

  
image 5 – Schéma électrique

## Programmation du mécanisme

  
image 6 – Présentation des différents composants

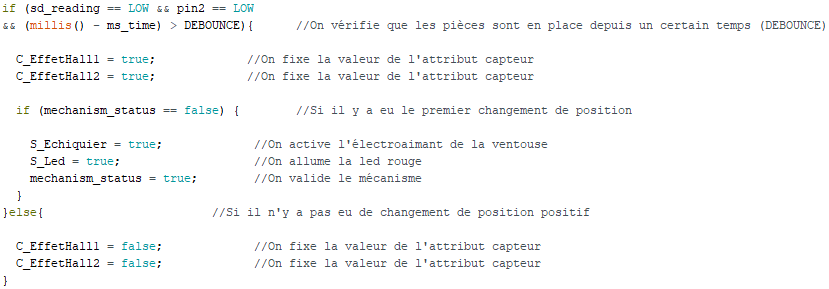
C’est ici que l’on va déclarer sur quel PIN de la carte Arduino Nano seront les différents éléments présents sur le schéma électrique ci-dessus.

Le premier « #define » va servir à l’étudiant n°1 pour le modèle « maître-esclave », il pourra, avec cette ligne, faire interagir les différentes cartes Arduino avec la Raspberry par le biais du bus I2C. Ensuite, nous avons nos deux capteurs à effet Hall, la LED qui sera présente sur le tableau de contrôle du gérant de la société et enfin l’électroaimant qui va libérer la porte.

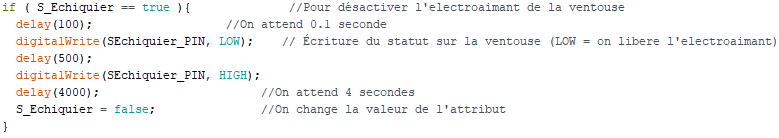
La programmation est effectuée sur le logiciel Arduino, on peut y coder nos mécanismes en langage C++. Par conséquent, nous allons coder notre mécanisme Echec avec deux fichier distincts « Echec.h » et « Echec.cpp » dans le même projet Arduino, comme suit :

  
image 7 – Fichier Echec.h

Ceci est la partie du fichier « Echec.h », on y définie les deux capteurs à effet Hall, l’actionneur « S\_Echiquier » et d’auters fonctions qui vont nous permettre de faire fonctionner le mécanisme.

  
image 8 – Condition pour vérifier que les pièces sont en place dans le fichier « Echec.cpp »

L’image ci-dessus est la condition qui vérifie que les deux cavaliers sont sur les bonnes cases, c’est le test : « sd\_reading == LOW && pin2 == LOW ». « sd\_reading » correspond au premier capteur à effet Hall et « pin2 » au deuxième. La notion de déphasage et de « DEBOUNCE » permet d’éviter que le client de l’Escape Game ne choisit pas des cases par hasard, « DEBOUNCE » est définie sur 2 secondes, donc le client doit laisser les pièces deux secondes sur les bonnes cases.

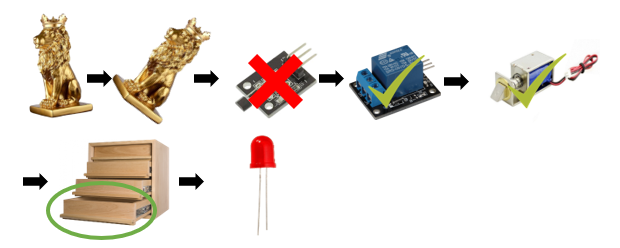
  
image 9 – Condition selon laquelle le mécanisme est validé

Cette condition nous montre comment l’actionneur fonctionne. Une fois le mécanisme validé, on désactive l’électroaimant, pour libérer la porte, c’est la ligne « digitalWrite(SEchiquier\_PIN, LOW) ; » qui nous le permet. Ensuite, on attend un demi second et on réactive l’électroaimant. Finalement, on attend 4 secondes et on change la valeur de l’attribut « S\_Echiquier » à false pour réinitialiser le mécanisme.

# Mécanisme N°2 : Le lion basculant

Le mécanisme repose sur un **capteur à effet Hall**.

Selon la mesure, un électroaimant est activé ou désactivé via un relais et une LED s’allume ou non sur le tableau de contrôle.

  
image 10 – schéma explicatif du mécanisme

# Mécanisme N°3 : L’élément Terre

Le mécanisme repose sur un **capteur à effet Hall**.

Selon la mesure, (i) un moteur et une LED sont activés ou désactivés via un relais pendant un laps de temps (ii) une sortie est paramétrée en destination du mécanisme des quatre éléments.

# Réalisation du projet

## Rappel de la tâche de l’étudiant

### Mécanisme 1 : l’échiquier

Le mécanisme repose sur deux capteurs à effet Hall.

Selon une condition portant sur les valeurs de ces deux capteurs, un électroaimant est activé ou désactivé via un relais et une LED s’allume ou non sur le tableau de contrôle.

### Mécanisme 2 : le lion basculant

Le mécanisme repose sur un capteur à effet Hall.

Selon la mesure, un électroaimant est activé ou désactivé via un relais et une LED s’allume ou non sur le tableau de contrôle.

### Mécanisme 3 : l’élément TERRE

Le mécanisme repose sur un capteur à effet Hall.

Selon la mesure, (i) un moteur et une LED sont activés ou désactivés via un relais pendant un laps de temps (ii) une sortie est paramétrée en destination du mécanisme des quatre éléments.

Visualiser l’état de la salle

Depuis une application WEB, le superviseur peut visualiser l’état de la salle. Pour chacun des neuf mécanismes, les valeurs des capteurs d’entrée sont affichées ainsi que l’état des différents actionneurs (état allumé ou éteint).

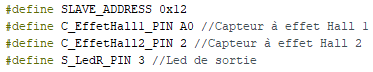
### Partie réseau

La partie réseau est constituée d’un schéma réseau de l’installation des différents éléments, un tutoriel pour l’utilisation de chacun des éléments a été transmis à l’entreprise.

# Mécanisme N°1 : L’Échiquier

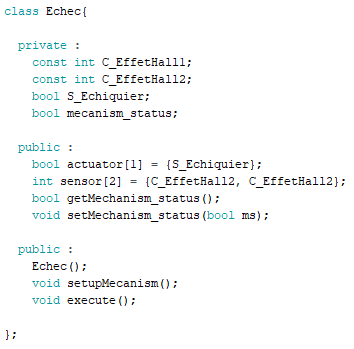
Programmation du mécanisme  
Un PIN est une entrée ou une sortie de la carte Arduino.

On commence par renseigner les PIN sur lesquelles sont installés nos composants.

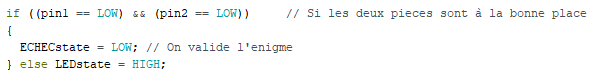
  
**Définition des PIN**

Le premier « #define » va être utile à l’étudiant n°1 pour expliquer le modèle « maître-esclave ».  
Le second et le troisième seront les PIN de nos deux capteurs à effet hall.  
Le dernier sera le PIN de la LED rouge qui sera représentée sur le tableau de contrôle.

La programmation est effectuée sur le logiciel Arduino. Celui-ci peut contenir du langage C ou C++. On va donc coder avec un fichier Echec.h et un fichier Echec.cpp.

  
**Echec.h**

Ceci est la partie du « fichier » Echec.h, on y définie les deux capteurs à effet hall, l’actionneur S\_Echiquier et d’autres fonctions qui vont nous permettre de faire fonctionner le mécanisme.

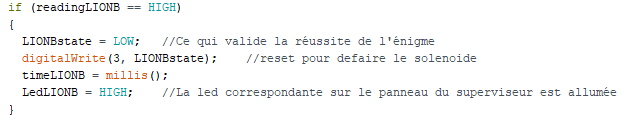
  
**La condition qui va vérifier que les pièces sont en place**

Cette condition va vérifier, à l’aide des deux capteurs à Effet Hall qui sont représentés par « pin1 » et « pin2 », que les deux pièces sont bien en place sur les bonnes cases afin de valider le mécanisme.

# Mécanisme N°2 : Le lion basculant

Programmation du mécanisme  
Comme pour le mécanisme précédent, on code avec un fichier LionB.h et un fichier LionB.cpp.  
Voici le fichier LionB.h :

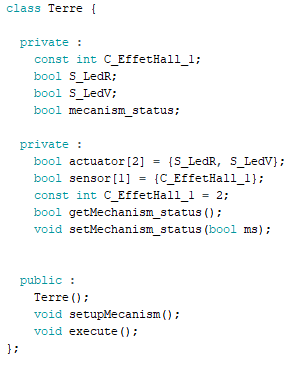
  
**Le fichier LionB.h**

Ceci est donc le fichier LionB.h, il est similaire à celui du premier mécanisme. Cependant, il n’y a, ici, qu’un seul capteur à effet hall qui fonctionne dans le sens inverse des deux capteurs du premier mécanisme.  
  
  
**La condition qui va valider l’énigme**

Cette condition va vérifier l’état du capteur à effet hall, s’il est désactivé, puisque dans ce mécanisme il faut déplacer la statue de Lion et donc enlever l’électroaimant du capteur à effet hall, alors le mécanisme est validé.

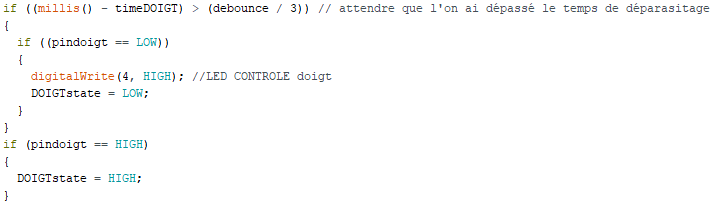
# Mécanisme N°3 : L’élément Terre

Programmation du mécanisme  
Comme pour les deux premiers mécanismes, on code avec un fichier Terre.h et un fichier Terre.cpp.  
Voici le fichier Terre.h :



**Terre.h**

Ceci est donc le fichier Terre.h, on retrouve un seul capteur à effet hall. Cette fois-ci, il y a une LED verte, représentée par « S\_LedV » qui va représenter la LED qui se trouve sur la tablette des quatre éléments. On y retrouve les différentes fonctions qui sont présentes dans le code des autres mécanismes.



**Partie de la fonction « execute() »**

Ceci est une partie de la fonction « execute() » du fichier Terre.cpp. On y voit la condition qui définie l’action de l’électroaimant du doigt de cinéma. Lorsque l’électroaimant est sur le capteur à effet hall pendant un temps donné par la première condition, le mécanisme se valide.

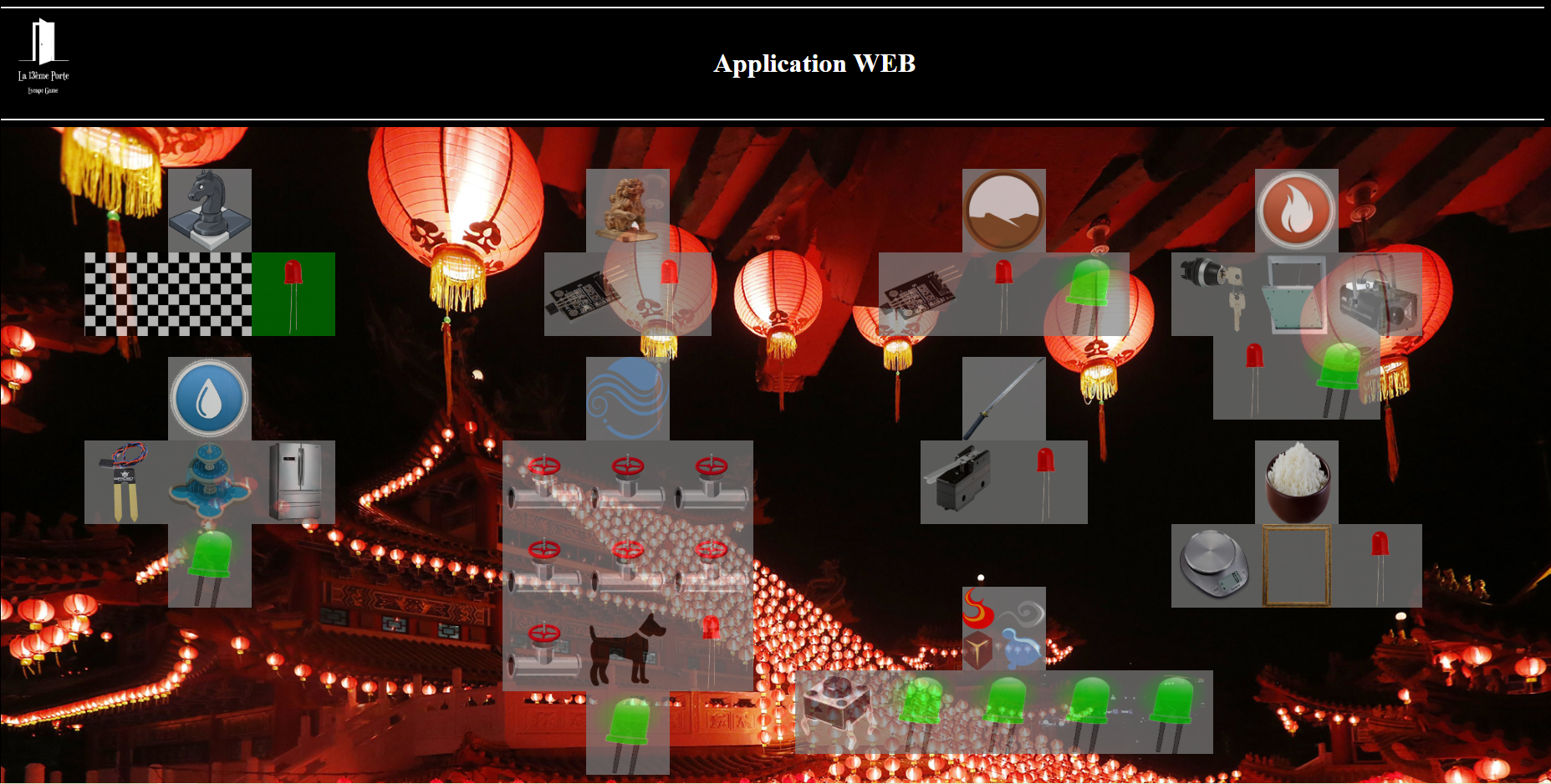
### Fonction void execute() :

Tout d’abord, il faut activer le capteur et lire sa première valeur (ValeurCapteur est initialisé à 0) :

# Application Web de supervision

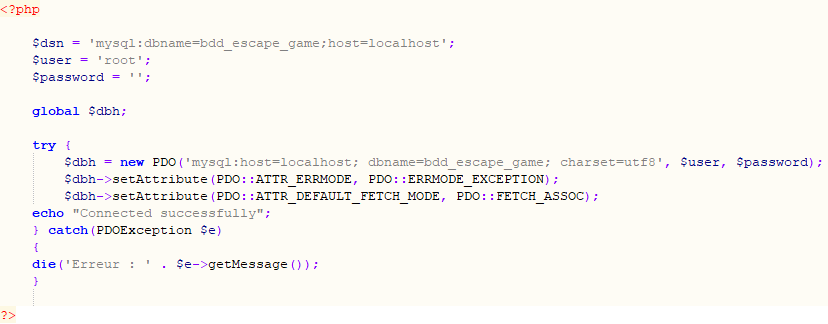
### Story Board de l’application Web



Aspect final  


Code  
  
Connexion à la base de données

Pour commencer, il nous faut relier notre application WEB à la base de données, pour cela on crée la page connexion.php comme suit :



Les premières lignes nous permettent de renseigner le nom et l’hôte de notre base de données ainsi que l’utilisateur et le mot de passe utilisé.

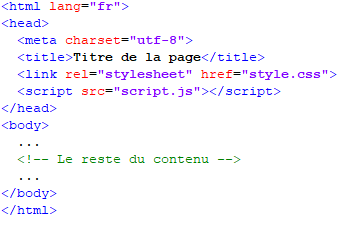
* global $dbh ;

Cette commande nous permet de pouvoir réutiliser la variable dans laquelle on va mettre les informations récupérées en base de données.

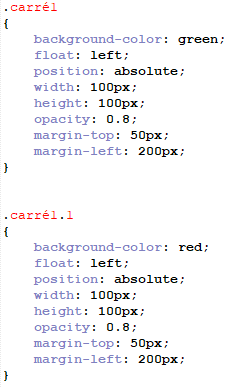
* Try
* {
* } catch
* {
* }

Grâce au « try/catch », on peut vérifier s’il y a une erreur dans la liaison avec la base de données, ce qui nous afficherait le message d’erreur contenu dans le « die() ».

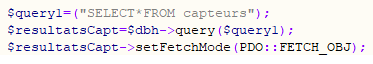
Code HTML/CSS  
Afin de commencer la programmation de l’application WEB de supervision, il faut, dans un premier temps, créer la structure de la page avec la structure traditionnelle HTML retrouvable sur internet :

  
  
Dans un second temps, il faut créer des formes nous permettant d’héberger nos images qui vont ensuite, à l’aide du code PHP, que nous verrons prochainement, nous permettre d’afficher l’état des différents capteurs ou actionneurs. Pour ceci, on procède de la manière suivante :

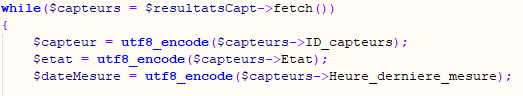
Cette portion de code va nous permettre d’afficher l’état « 1 » du mécanisme « Échec » par exemple :  
  
  
Il existe aussi une variante qui va nous permettre d’afficher l’état « 0 » du mécanisme en question :  


On utilise la classe « Carré1 » ou « Carré1.1 » juste pour modifier la couleur de la case qui va simuler l’état « 1 » ou « 0 » du mécanisme :   


#### Code PHP

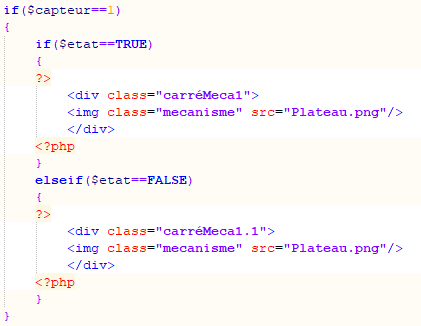


Cette portion de code nous permet de récupérer les informations de la table « capteurs » dans notre base de données. Le première ligne, à l’aide d’une requête SQL, récupère les informations de la table « capteurs ». Ensuite, il faut mettre toutes ces informations dans une variable, à l’aide de la fonction « query() » qui nous permet de récupérer un jeu de résultat provenant d’une requête SQL et de les récupérer en tant qu’objet PDOStatement.



Pour utiliser les données récupérées précédemment, il faut faire une boucle et utiliser la fonction « fetch() ». Cette fonction récupère une la ligne suivante, d’où l’utilisation d’une boucle pour récupérer toutes les lignes de la base de données.

Une fois toutes les données de la table « capteurs » récupérées, on doit utiliser des variables pour pour exploiter ces informations. On utilise « $capteur » pour les données de la colonne « ID\_capteurs », « $etat » pour les données de la colonne « Etat » puis enfin « $dateMesure » pour la colonne « Heure\_derniere\_mesure ».



Pour terminer, on doit faire plusieurs boucles comme celle-ci, où on récupère le numéro du capteur, s’il correspond au premier capteur, on entre dans la boucle et on observe l’état du capteur dans la base de données, s’il est à 1, on fait apparaître le carré vert, sinon ce sera le rouge comme suit :

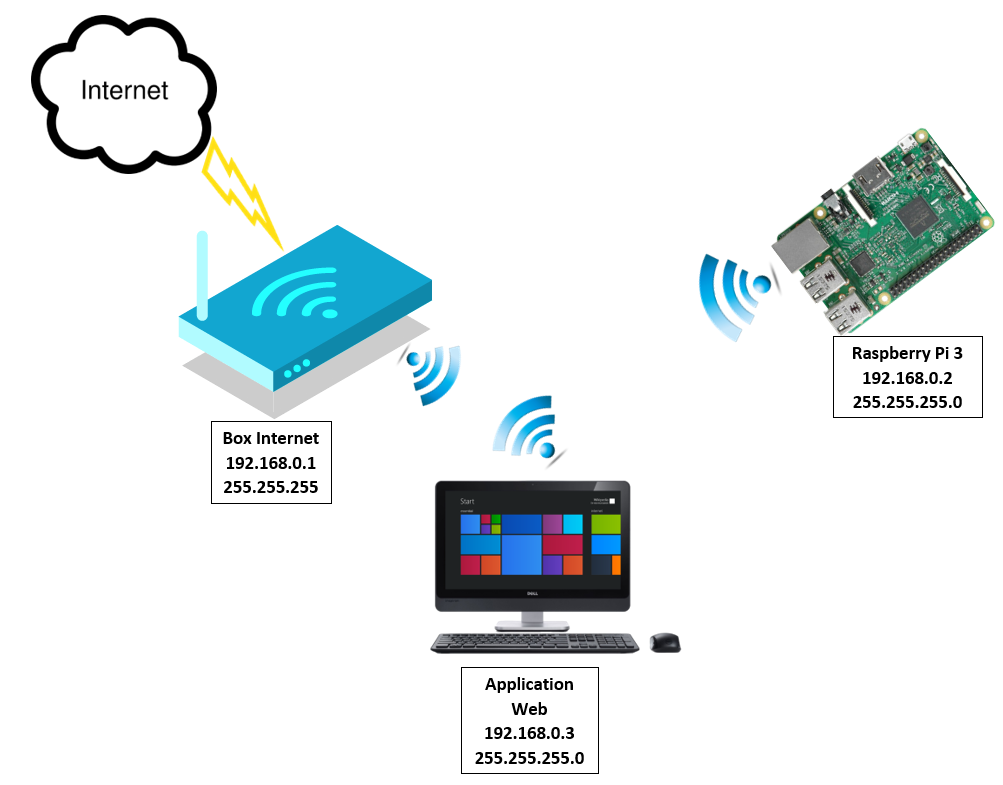


Si l’état du capteur est à 1, c’est sous cette forme que l’image va apparaître, si l’état est à 0, ce sera alors la même image mais sur un fond rouge. Cela nous permet aisément de différencier l’état des différents capteurs.

Partie réseau  
  
Matériel

Ordinateur  
**Un ordinateur** pouvant se connecter à la box internet de la société **13ème Porte**.  
  
Raspberry  
**Une carte Raspberry** qui va servir d’intermédiaire entre les différentes cartes Arduino Uno et le serveur de base de données.  
  
Box internet  
**Une box internet** qui est déjà présente sur le site puisqu’il s’agit du boîtier internet de la Société **13ème Porte.** Celle-ci va nous servir de routeur afin de faire communiquer les différents composants de ce réseau.

### Schéma réseau



Tutoriel de connexion  
Un tutoriel a été fourni à la société **13ème Porte** afin de permettre à cette entreprise de pouvoir déployer ce réseau en réservant, par exemple, une ou plusieurs adresses IP dans grâce au protocole DHCP.