Rapport Projet DEV

**Réalisé par :**

* BONMARIN Lucas
* COURET Hélène
* HILL Maximiliano
* OUALI Maher

**Nom du projet :** Escape Game

**Version du rapport :** 1

1. **Introduction :**

Depuis des années déjà, les nouvelles technologies ne cessent d’apparaître chaque jour. Si bien que les tablettes et les smartphones sont de nos jours des objets quotidiens. De plus, ce genre de technologie touche de plus en plus de jeunes. C’est à dire que la plupart d’entre eux ont accès à des smartphones ou des tablettes et savent s’en servir. C’est donc dans ce contexte que nous avons réalisé un jeu de type “Escape Game” sous la forme d’une application Android ayant un objectif pédagogique. En effet, ce jeu est destiné aux professeurs et aux élèves de 4ème et de 3ème du collège Saint Pol Roux de Brest pour mettre en pratique des compétences en programmation, mathématiques et technologie.

Grâce à l’application et à une maquette d’une maison pourvue de capteurs et d’actionneurs (LEDS, moteurs, alarmes, interrupteurs), l’élève pourra se déplacer de pièce en pièce en répondant à des questions que son professeur aura préalablement ajouté dans une base de données grâce à cette même application. Les élèves pourront évidemment développer leurs connaissances sur leurs cours mais aussi commencer à se familiariser avec la programmation. En effet, ils pourront se plonger dans des parties de code (Android ou Arduino) à leur portée.

Pour se faire, nous avons tout d’abord réfléchi à un scénario pour rendre cet apprentissage ludique, un objectif que l’élève devra attendre pour avancer dans le jeu et gagner. Ensuite, nous avons réalisé l’application Android sur App Inventor 2, qui est la pièce maîtresse du jeu. Pour finir, nous avons travaillé sur la partie maquette, c’est à dire relier l’application Android à la maquette de la maison grâce au Bluetooth et au code Arduino.

Rédacteur: BONMARIN Lucas

Relecteurs: OUALI Maher ,HILL Maximiliano

II. Résumé :

Depuis une cinquantaine d’années, l’éducation scolaire tend à faire passer tout apprentissage par le jeu. Ce document présente la réalisation d’une application utilisable par deux enseignants du collège Saint-Pol de Roux à Brest. Ces deux enseignants nous ont demandé de développer une application qui leur permettrait d’apprendre des notions mathématiques ou technologiques à leurs élèves de quatrième et de troisième. Cet apprentissage se ferait dans le cadre d’un jeu de type “escape room” qui demanderait aux élèves de répondre à des énigmes pour parvenir à sortir de la maison (une maquette domotique contenant des actionneurs afin de rendre le jeu interactif).

Afin de réaliser ce jeu, nous nous sommes servis d’une application Android qui permet aux élèves de répondre aux énigmes pour avancer dans le jeu, aux enseignants de gérer les énigmes et enfin qui exécute les liens avec les autres éléments du jeu. En effet, cette application est reliée par Bluetooth à la maquette domotique et également au serveur qui accueille la base de données formée des énigmes.

Rédactrice :  COURET Hélène

Relecteur : BONMARIN Lucas

III. Plan :

1. **Introduction**
2. **Résumé**
3. **Plan**
4. **Présentation du jeu**
5. **Application Android**
   * 1. **Interface du jeu**
     2. **Bases de données/PHP/Web serveur**
6. **Lien avec la maquette (Arduino)**
   * 1. **Actionneurs et détecteurs**
     2. **Protocole**
7. **Conclusion et mise en perspectives**
8. **Bibliographie**

IV. Présentation du jeu :

1. Déroulement du jeu :

Le jeu de type “Escape Room” que nous avons créé se base sur une maquette domotique connectée par un module Bluetooth à une application Android. Le but du jeu est de sortir de la maison en répondant correctement aux énigmes des différentes salles dans laquelle se trouvent fictivement l’élève au cours du jeu. Une fois que l’énigme a été correctement résolue sur l’application, un actionneur de la salle suivante s’active. Si la réponse donnée par l’élève est fausse, tous les actionneurs de la maquette s’actionnent pendant 5 secondes puis l’élève peut à nouveau répondre à la question.

1. Buts du jeu :

Le jeu que nous avons cherché à développer avait deux objectifs principaux définis par les professeurs du collège Saint-Pol de Roux à Brest.

Le premier objectif, qui serait utilisé par le professeur de mathématiques pour ses classes de quatrième, était de développer une application ludique, sous forme d’un jeu escape room, au cours duquel les élèves allaient devoir résoudre des énigmes mathématiques pour terminer le jeu. Cette application Android doit alors permettre aux professeurs de gérer facilement la base de données d’énigmes (en ajouter, en supprimer et modifier les propositions de réponses) et aux élèves de jouer à cet escape room en ayant accès aux énigmes et aux propositions de réponses. Bien répondre à l’énigme leur permet alors de progresser dans le jeu et en mathématiques.

Le second objectif de ce projet, à destination du professeur de technologie du collège pour ses classes de troisième, était de permettre à ce professeur d’initier ses élèves à la programmation informatique. Nous avons d’abord voulu réaliser les fonctions élémentaires pilotant les actionneurs en ArduBlock mais cela c’est révélé plus compliqué que prévu et nous avons pensé que ces élèves de troisième pourraient également répondre à des énigmes sous forme de code à trous dans le même cadre que les énigmes mathématiques.

Finalement, notre projet regroupe trois parties distinctes qui permettent de créer cette application d’un jeu de type escape room :

* une partie Android pour gérer l’apparition des énigmes, la connexion avec la maquette et le protocole général du déroulement du jeu
* une partie Arduino pour gérer les actionneurs de la maquette
* une partie Base de Données qui permet de stocker les énigmes

schéma des 3 parties

Rédactrice : COURET Hélène

Relecteur : HILL Maximiliano

V. Application Android: [5]

1. Interface du jeu:

L’application Android présente un élément très important du projet DEV puisqu’elle est l’interface à travers laquelle l’élève va être impliqué dans l’aspect pédagogique du jeu, ainsi, on doit garantir une application fonctionnelle et de haute qualité. De plus, puisque le client ciblé est un collège et en particulier des élèves de 3éme et 4éme années donc il faut présenter une bonne interface à l’échelle graphique et esthétique pour les encourager à utiliser ce nouveau concept d’éducation.

Parmi les exigences du cahier de charges du projet, on trouve la possibilité d’ajouter des énigmes par les professeurs, alors, il fallait que l’application soit non seulement un outil aux élèves pour jouer le jeu et répondre aux énigmes mais aussi aux enseignants pour accomplir cette tâche exigée. Par suite, on présente quatre écrans en total: un écran d’accueil pour différencier entre un élève et un professeur, deux écrans dédiés aux élèves, le premier pour choisir soit de l’aide, soit de commencer le jeu et le deuxième pour jouer le jeu et un dernier écran dédié aux professeurs pour ajouter de nouvelles énigmes à la base de données (voir la partie suivante).

1. **Écran d’accueil**

Cet écran contient deux boutons, l’un contenant l’image d’un prof et l’autre contenant l’image d’une élève (fig 1.a).

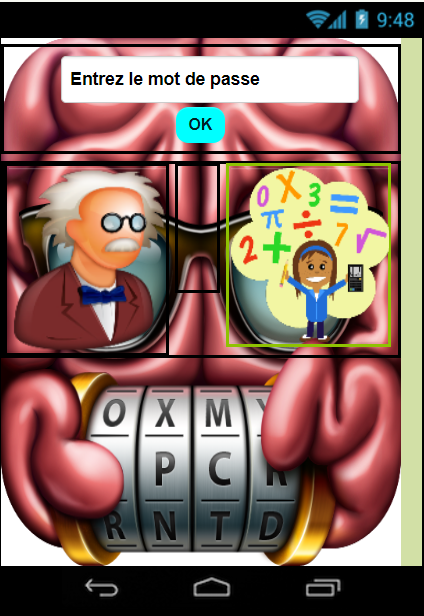


figure 1.a

L’accès à l’écran suivant se fait en cliquant sur l’image correspondante sachant que dans le cas d’un professeur, il faut entrer le bon mot de passe “123456789” qui est fixé par l’équipe et peut être changé à partir du code et non pas l’application (fig 1.b).

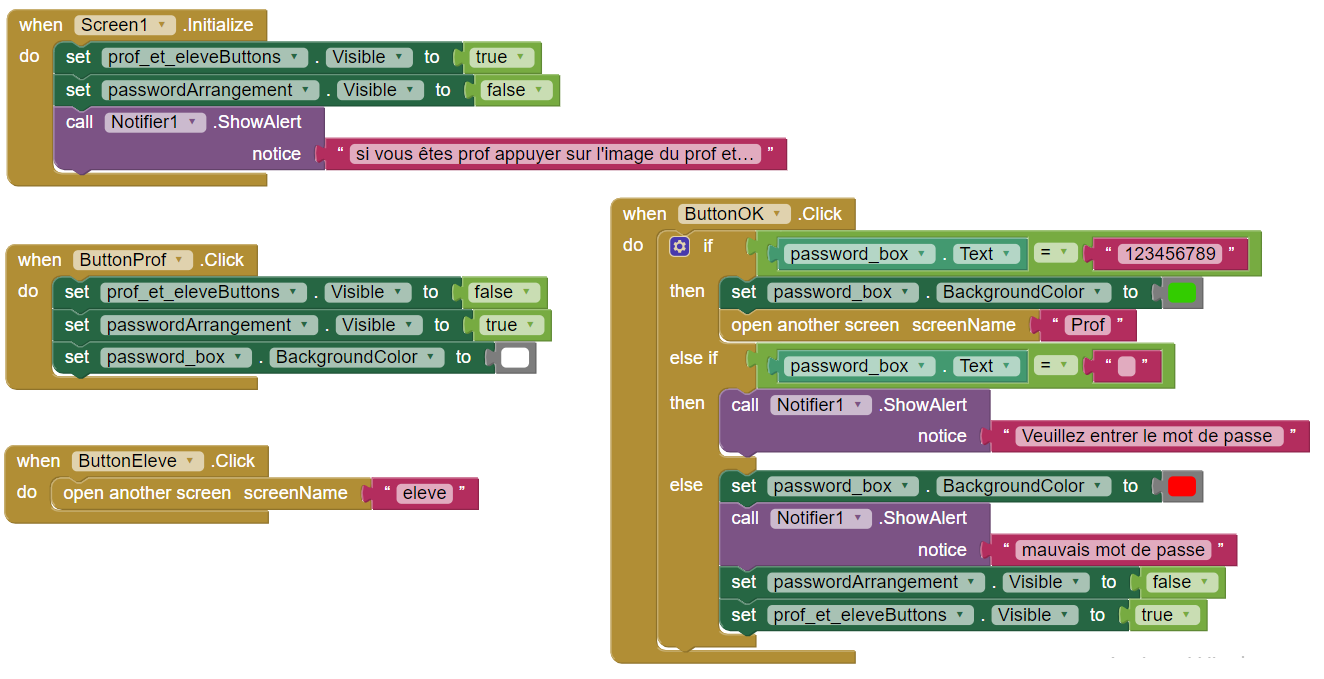


figure 1.b

1. **Écran Prof**

Cet écran (fig 2.b) contient deux boutons un de l’ajout et l’autre de l’envoie et quatre champs de texte pour faire entrer la question de l’énigme, une bonne réponse et deux mauvaises réponses.

Lorsque le professeur accède à cet écran, il ne voit que le bouton d’ajout (fig 2.a) et puis, lorsqu’il appuie dessus, ce bouton disparaît et les autres éléments s’affichent.

|  |  |
| --- | --- |
| https://lh4.googleusercontent.com/kszvgJa8OlL6r0C3uVWu1NueaVZFK9DeMiFoYfhbS-f0Y0qikFqOnpavlrYjSU266epBd8TZ0H9tAtHG6_SRlRw8f3KLBV5S7bHOCm2DN6x181g68U2L09HYkqjmboOqTb6Jj1UV | https://lh4.googleusercontent.com/zaAJukQmcb4_fdbt5rlXCwqKcvwzQB5qRsXIK35vBlQ5OcKhBl680oJ0TinZmlFtidZbaU6e2hhR6p0ohJf-BPwAy79XJY9Rew4KzuPcW15iNKBSUQceV9Jl1HBWAlcLUh1_T30z |
| figure 2.a | figure 2.b |

Après le remplissage des champs de texte et l’appuie sur le bouton de l’envoie, l’énigme sera stockée dans une base de données sur un serveur loué (voir partie suivante). (fig 2.c)

1. **Écran de l’élève**

Cet écran contient trois boutons:  le premier est sous la forme d’un cadenas pour lancer le jeu, le deuxième est sous la forme d’un point d’interrogation pour savoir quoi faire et le troisième est sous la forme d’une maison pour retourner vers la page d’accueil (fig 4.a).

Lorsque l’élève appuie sur le bouton de l’aide (celui en haut à droite), tous les boutons disparaissent et une image s’affiche pendant huit secondes (fig 4.c), c’est un papier sur lequel est écrite la mission demandée (fig 4.b)

|  |  |
| --- | --- |
| https://lh6.googleusercontent.com/dxgYCN4QhFCHRUf7B1SeKYj3xNwDBo3hw3sarMsh6aHH3P9Ko-zImt19QiGwmcYII3kZxqvv4N16MPZwUnHQpihGHBgV2MIr53zHDUMks75iXYB7u9YLHsb5iDw2DTxZb2dYa8sU | https://lh6.googleusercontent.com/zIEvzBdlmcsldjyWEKTC70XrY18x7b9vGZdpmbVKQRgXTxfA8iOm38vA6iLVU_73Trh2Ts8GYUsNDTvLON18XoxkP1pj6HYfFxDLAZb9dZ8V2Dua_rnasC0Ikg1oXNtR076NC6lQ |
| figure 4.a | figure 4.b |

Le code en bloc de cette partie:

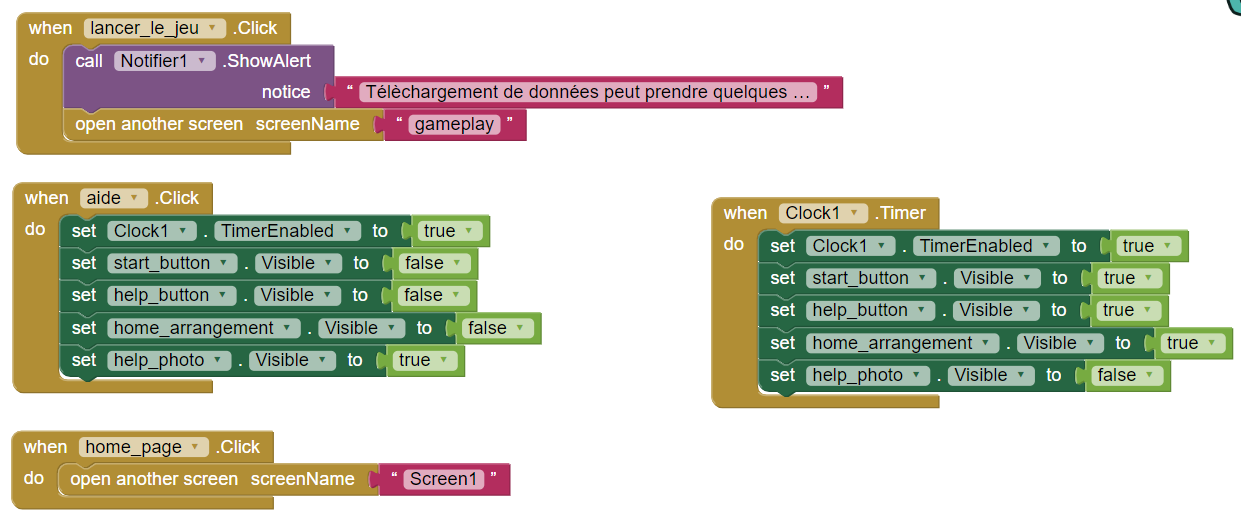


figure 4.c

4. **Écran du jeu**

Cet écran correspond à la partie la plus importante de l’application et la partie la plus compliquée au niveau du code Android puisque ça gère le déroulement du jeu, la connexion Bluetooth, le téléchargement des énigmes depuis la Base des données et l’envoie des tâches vers la carte Arduino.

Due à des problèmes de connectivité liés à AppInventor, il fallait qu’on utilise un même écran pour basculer entre deux sous-écrans: un premier (fig 5.a) pour poser l’énigme et vérifier la réponse de l’élève et un deuxième (fig 5.b) pour afficher à l’élève les tâches à faire pendant cette étape du jeu et envoyer les messages correspondants vers la maquette.

|  |  |
| --- | --- |
| https://lh3.googleusercontent.com/IE8oz-RR3jy2VAxKEKPl1tUt2php5EfOLsA9jXtB0oVByke3QQnQjlDjS8KuPR_k_U-eIM_T3u56dtPxce7UM-ZlxPWqkPCCUMAG4Qro0tpcQyc-EIc3jaGrSQaJbHde0gDe34HR | https://lh4.googleusercontent.com/ysBNWnSSO6czVj8PutiT9d4iK8bC_aV0ZOG07x2ieqUbijRG4jIeVMuOE369y_8OivtWjB8qyLIzHd3LzLu6SOvmQkibnNWlVolaUHXICXqT9aDjNpaV2q-TjsIIWPCcAwp2HsdR |
| figure 5.a | figure 5.b |

En fait, le code en bloc de cet écran contient plusieurs parties, ainsi, on va les détailler en partant de l’initialisation et en passant par la pose l’énigme jusqu’à atteindre l’affichage des tâches.

* *Initialisation:*

La phase d’initialisation consiste à télécharger un certain nombre d’énigmes et leurs réponses depuis le serveur et de les stocker dans 4 différentes listes, ce nombre dépend du scénario (déterminé par les professeurs en modifiants les listes des messages de contrôle à envoyer vers la carte Arduino) et d’assurer une connexion bluetooth entre l’application et la plateforme, c’est à dire que l’élève ne peut pas commencer à jouer sans appuyer sur le bouton “Bluetooth” pour se connecter.

* *Pose d’énigme*

L’énigme est ensuite posée à l’élève en choisissant un nombre aléatoire entre 1 et 3 qui va désigner l’indice de la bonne réponse et puis lorsque l’élève appuie sur un bouton parmi les trois existants, si ce dernier correspond au bon indice alors la réponse est comptée comme correcte et un smiley “heureux” apparaît sinon elle est comptée fausse et un smiley “triste” apparaît sur l’écran.

* *Affichage des tâches*

Lorsque la réponse est correcte, un autre écran s’affiche avec une petite animation d’une caisse de trésor qui s’ouvre et puis la mission à faire s’affiche sur l’écran et à ce moment là, l’application envoie le message correspondant à cette tâche vers la carte Arduino et attend un message contenant le chiffre “1” en retour pour soit passer à la tâche suivante ou l’énigme suivante.

Vers la fin du jeu, une note attribuée à l’élève s’affiche.

Cette note se calcule en additionnant un “1” à chaque bonne réponse et en soustrayant un “1” à chaque mauvaise réponse.

Rédacteur: OUALI Maher

Relecteur: HILL Maximiliano

1. Base de données/PHP/Web serveur :

(À modifier une fois qu’on aura le web serveur de l’école)

Une base de données informatique est un outil qui permet de stocker des informations de façon persistante même en dehors du système à implémenter. Plusieurs genres de base de données existent d’où il faut être capable de choisir quel type de base de données on utilisera. Cette décision se base ultimement sur la performance et les fonctionnalités qu’on attend de la BD (base de données) pour notre projet.

La BD doit être capable de stocker un grand nombre d’énigmes/questions avec leurs réponses (une correcte et deux incorrectes par définition du projet). Vu qu’il s’agit simplement de texte et d’un seul type de donnée (l’énigme) l’implémentation de la BD n’est pas un problème majeur. On utilisera une base de données fonctionnant avec le moteur InnoDB (InnoDB est un moteur de stockage de données pour les bases de données MySQL) sur une base de données MySQL (MySQL est simplement le système de gestion de base de données qui nous permet d’insérer, effacer, mettre à jour les données entre plusieurs autres fonctions). MySQL permet de gérer des BDs relationnelles où plusieurs tables avec des rangs (les données) sont relationnées les unes avec les autres. Cependant on utilisera une seule table puisque c’est une approche suffisante et beaucoup plus simple à implémenter.

En effet, on définit pour notre projet une table d’énigmes qui contient quatre colonnes; une identification (ID) qui va servir de marqueur unique pour chaque énigme, la question posée par l’énigme et finalement trois réponses, deux incorrectes et une correcte, puis une instance d’énigme est définie simplement comme une rangée de cette table.

Cependant, il y a plusieurs problèmes à adresser vu que la BD est retirée du système:

* Accès par internet de la BD.
* Connexion et interaction entre l’Application portable et la BD.
* Server/Web server pour la BD.

Pour ceci on utilisera un web hosting service gratuit qui supporte des BDs MySQL avec phpMyAdmin. Un web hosting service nous pourvoi avec un web serveur et un domaine (sous-domaine)  et une certaine capacité de stockage. Un serveur Web est un programme informatique qui traite une application côté serveur, établissant des connexions avec le client et générant ou donnant une réponse dans n'importe quelle langue ou application du côté client.

Le hosting service choisi est le paquet gratuit de hostinger[6] qui offre un web serveur avec protocole http et jusqu’à deux bases de données avec une capacité pour des centaines de tables. PhpMyAdmin est un outil logiciel gratuit écrit en PHP qui s’utilise à travers le navigateur web (chrome par exemple), destiné à gérer l'administration de MySQL sur le Web.

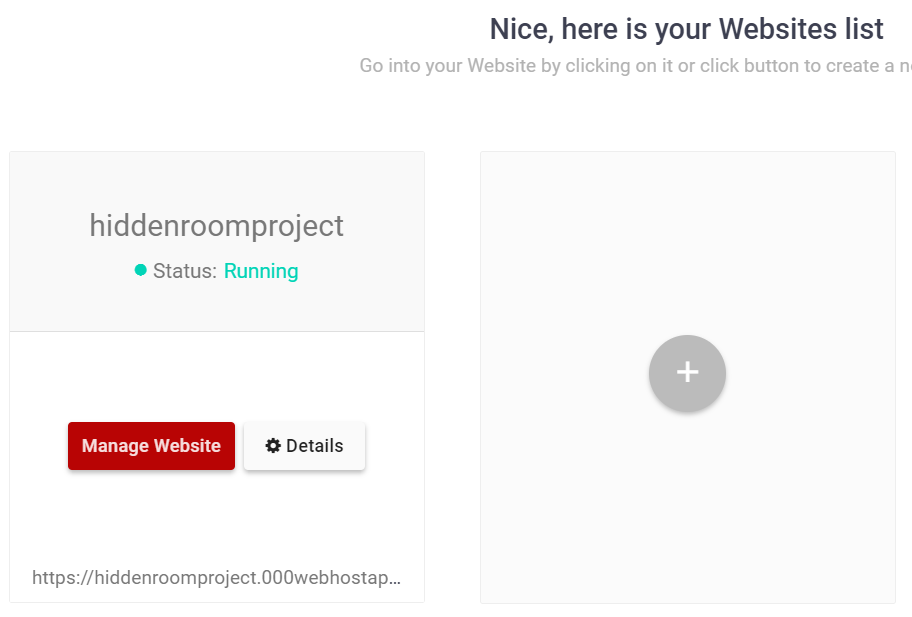


figure 1.a

Une fois que le web serveur est installé est en exécution, on crée une instance de BD et on lance PhpMyAdmin pour créer nos tables.

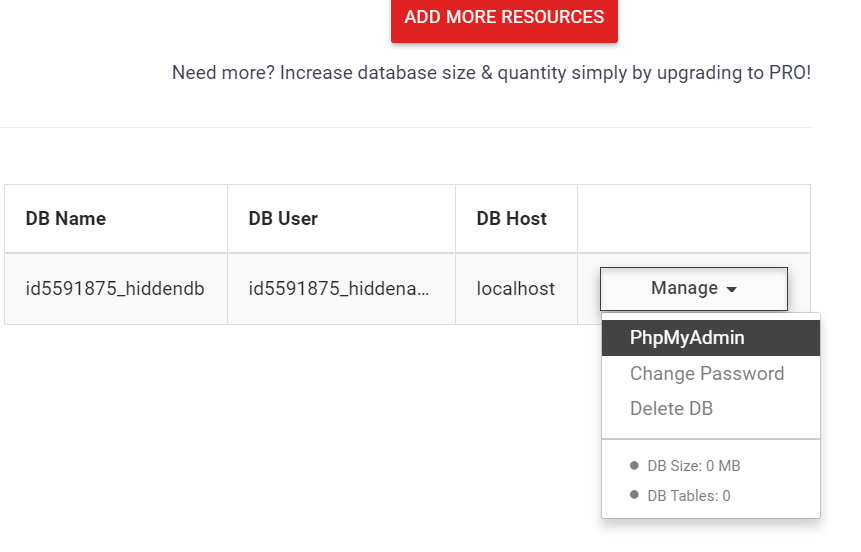


figure 2.a

Une fois dans PhpMyAdmin on doit créer notre table, ceci peut être fait directement par PhpMyAdmin ou à travers de “l’import” d’un fichier MySQL (.sql) qui s'exécute dans PhpMyAdmin. Le fichier .sql définit des tables et insère des valeurs, cette dernière étape est facultative dans ce projet, les énigmes pourront être rajoutées après à travers l’application.

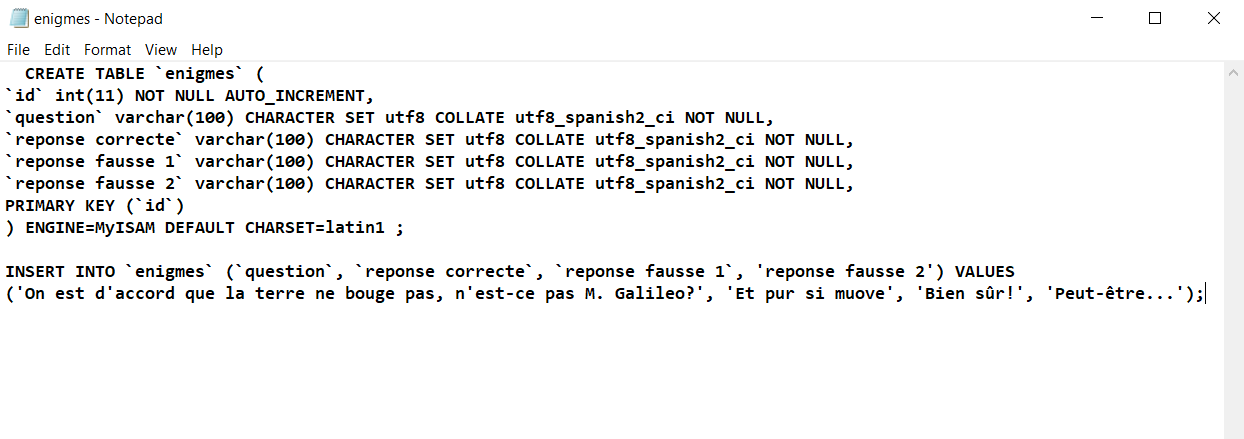


figure 3.a

Pour que le web server fonctionne comme voulu, il est nécessaire définir son fichier php et après le télécharger sur le serveur. Tel qu’il est dit en wikipedia, “Php est un langage de script côté serveur conçu pour le développement web” dont on a utilisé que quelques rudiments basiques ([7] et [8]). Le fichier PHP contrôle les demandes dirigées au serveur/BD. Puisque le serveur web est l'endroit où le code PHP est exécuté, l'accès à un fichier PHP à travers du browser ne vous donne pas accès au code mais vous fournit à la place le contenu HTML que le serveur génère. Dans le cas du projet, il faut établir une connection avec la BD et définir les fonctions qui traiteront les demandes de l’application.

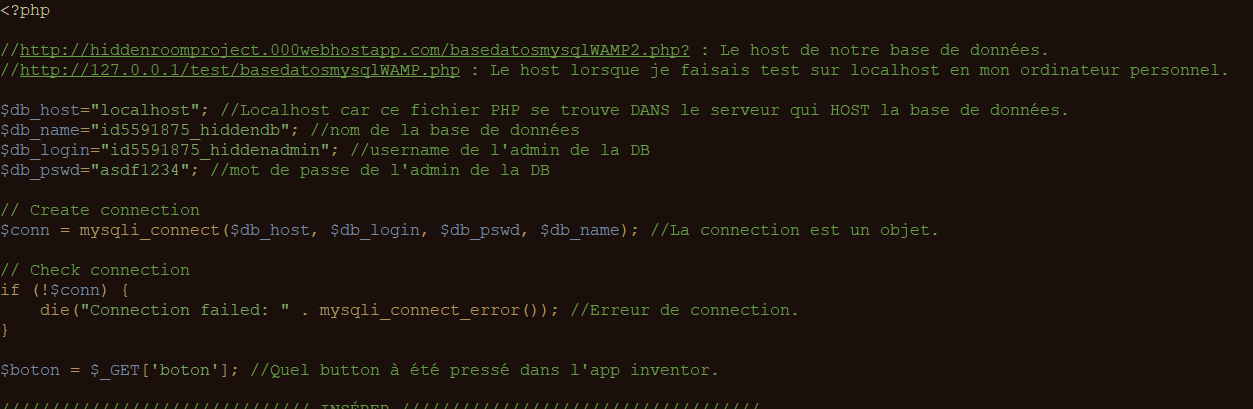


figure 4.a

Cependant, il faut être capable d’ajouter ce fichier dans le web server, pour ceci il faut utiliser un logiciel pour transfert de fichiers ou FTP.

En effet le service de 000webhost contient une interface avec services FTP pour ajouter le fichier .php dés le site web qu’on a créé à travers l’interface d’administrateur. Cependant une méthode plus sûre est l’utilisation d’un logiciel comme Filezilla qui nous donne un service FTP pour déposer des fichiers sur le serveur FTP de notre web server, ce serveur a un user, un mot de passe, une adresse et un port ouvert (21) à l’écoute en permanence pour faciliter notre dépôt de fichiers. Une fois connecté par Filezilla il s’agit d’une gestion de fichiers similaire à celle présentée par windows.

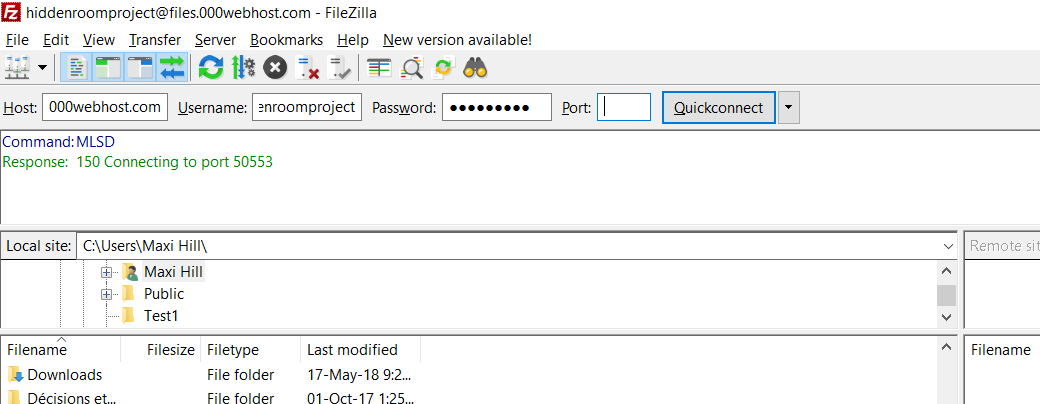


figure 5.a

Avec le fichier php déposé, notre serveur et BD sont prêts pour interagir avec l’application. Il faut maintenant préparer le code du côté de l’application Android pour la correcte intégration avec le fichier .php, contrôleur de nos demandes.

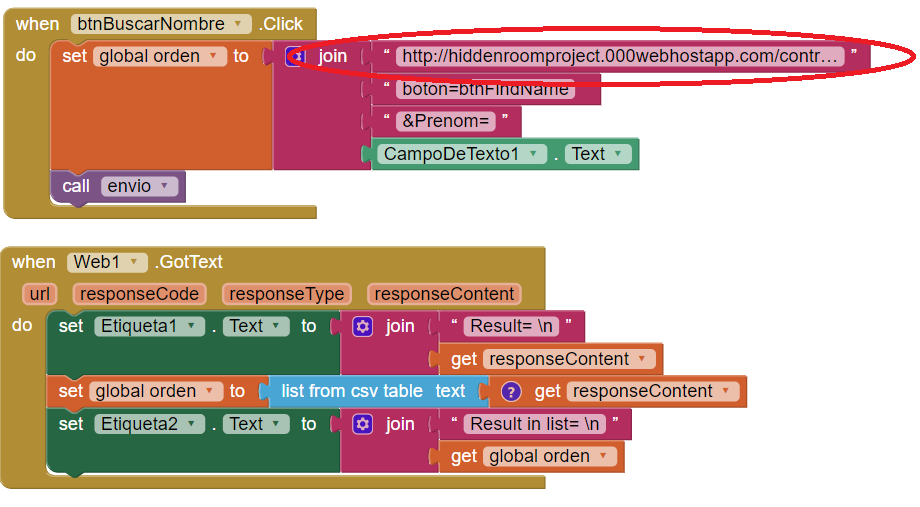


figure 6.a

Il s’agit de faire l’appel au fonctions correctes en donnant l’adresse de notre fichier .php sur le serveur, dans ce cas:

<http://hiddenroomproject.000webhostapp.com/controleurBaseDonnees.php>?

Ainsi, l’application et la BD sont intégrés.

Rédacteur: HILL Maximiliano

Relecteur: OUALI Maher

VI.   Lien avec la maquette :[4]

La maquette présentée est une plateforme composée de plusieurs actionneurs et détecteurs, ces derniers sont des composants électriques capables chacun de faire une tâche bien déterminée (générer de la lumière / son / mouvement rotationnel d’un objet …. détecter la lumière / un clic / un obstacle …).

Certes, ces éléments électroniques forment une partie très importante du projet puisque c’est grâce à eux qu’on va avoir des interactions physiques avec l’élève assurant la concrétisation du concept du jeu ( Escape Room ).

Vu que l’élève utilise l’application android fournie (voir II. ) pour recevoir les tâches à accomplir sur la maquette donc fallait garantir un moyen de communication entre l’application et la carte Arduino et pour cela on aurait recours à une liaison Bluetooth.

Comme on a déjà mentionné, plusieurs tâches existent et chaque tâche doit être transmise sous une forme unique, ainsi, cette connexion Bluetooth nécessite un protocole qui soit le mieux adapté aux exigences du projet.

Dans cette partie, on va détailler deux points majeurs de la partie Arduino qui sont :

* Les actionneurs: Leurs codes respectifs
* Le protocole de communication Bluetooth: Comment ça marche et comment il répond aux besoins du jeu.

1. Les actionneurs et les détecteurs :

Dans cette partie, on va présenter les codes écrits qui contrôlent les différents composants utilisés. Pour cela, on va exploiter l’interface de développement d’Arduino (fig 1.a).

|  |  |
| --- | --- |
| https://lh3.googleusercontent.com/xRvUxPm64fkxayUWh5yFTmanfHHcb0jkhfi5-eK_sDYyO7rvCZGkx823CAm5IsXL84PBWNU5ZsoQkr02rP_ogq7XeDyynbJU0xVKdRRTK8Okeyh6TxemDmB_Y0xfeW-HdNSrEPND | https://lh4.googleusercontent.com/xQGw3IPbkvXN5qUY20G4sAJdguJXap-zIXZYYO7K5YMtJybmJH0aHHhOsPo6nuQXB6zh1HQotgUkmn55EIT09eNevh8lsJXepv2E8UV0c7oZ7MuJERDFU2pzULe_CURDRBEnHZXg |
| figure 1.a | figure 1.b |

En effet, une carte Arduino (fig 1.b) est l’équivalent d’un mini-contrôleur programmable qui sert de faire des calculs et de donner des ordres aux composants électroniques connectés à travers les pins.

À l’aide de cet environnement de travail, on va pouvoir programmer notre carte et la connecter à la maquette et dans ce qui suit, on va présenter les codes écrits en langage d’Arduino (inspiré du C).

1. **LED: Light Emitting Diode (diode électro-luminescente)**

Une diode électro-luminescente est un dispositif capable d’émettre de la lumière lorsqu’on lui fournit du courant électrique grâce aux propriétés physiques d’un semi-conducteur [1]. Elle peut prendre plusieurs formes et elle est utilisée dans plusieurs domaine (téléviseurs /écrans …) mais celle qui va être utilisée dans notre projet est celle dans la figure 2.a.



figure 2.a

Dans notre projet, on présente deux manières pour utiliser les diodes :

* Allumer : La fonction utilisée pour réaliser cette tâche (voir fig 2.b) est “lightUpLED” dont les entrées sont “actuator” et “action” qui sont respectivement le pin correspondant à la diode en question et la durée en millisecondes pendant laquelle on va faire allumer la diode.

Le principe de la fonction est de faire passer le courant vers le pin correspondant [digitalWrite(LED[actuator-1],HIGH);], attendre pendant la durée demandée [delay(action);] et puis faire couper le courant [digitalWrite(LED[actuator-1],LOW);].

On remarque que si action = 0 alors, on laisse la diode allumée.

* Clignoter : La fonction utilisée pour réaliser cette tâche (voir fig 2.b) est “blinkLED” dont les entrées sont “actuator” et “action” qui sont respectivement le pin correspondant à la diode en question et le nombre de clignotement.

On remarque que la durée entre deux clignotements successifs est fixée par l’équipe du projet [delay(time\_btwFlashes);] mais peut être changée après par les professeurs puisqu’ils ont accès au code Arduino.

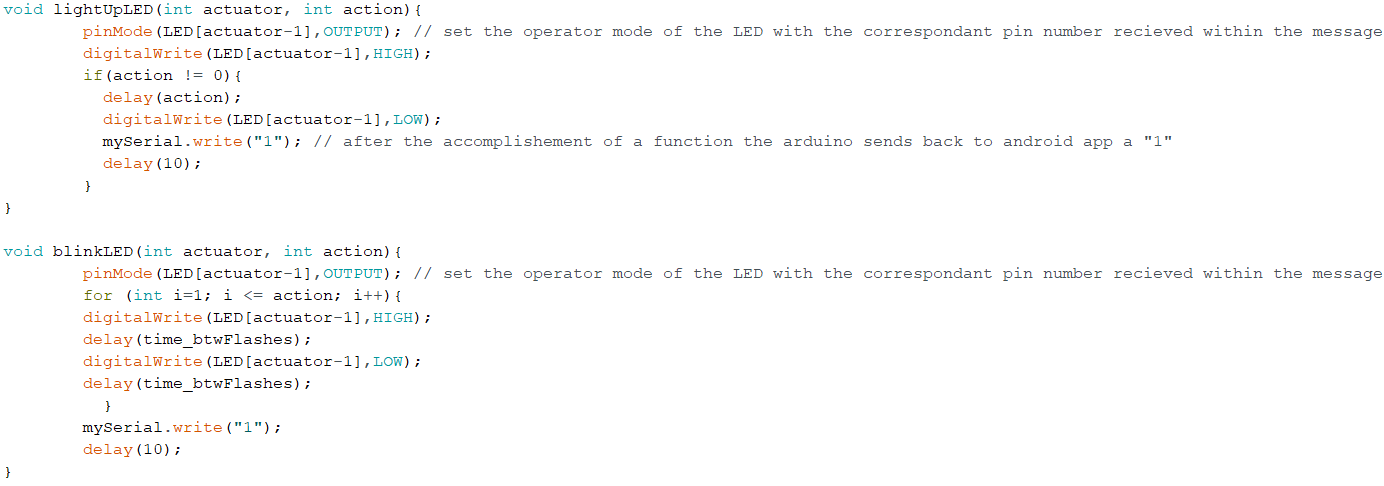


figure 2.b

1. **Servomoteur**

Un servomoteur (fig 3.a) est un dispositif capable de faire tourner un objet. La spécificité de ce composant c’est à dire la différence entre un servo-moteur et un simple moteur est que le premier est capable de s’opposer à un effort statique et maintenir sa position: c’est donc un système asservi [2].



figure 3.a

Dans ce projet, le servomoteur peut jouer le rôle d’un ascenseur pour passer d’un étage à un autre ou peut modéliser une porte qui s’ouvre et se ferme durant le déroulement du jeu.

Pour pouvoir utiliser un servomoteur, il faut télécharger une bibliothèque intitulée Servo et l’importer dans l’environnement de travail d’Arduino en utilisant la commande [#include <Servo.h>].

Dans notre projet, on présente une seule manière pour utiliser un servomoteur :

* Tourner : La fonction utilisée pour réaliser cette tâche (voir fig 3.b) est “turnServo” dont les entrées sont “actuator” et “action” qui sont respectivement le pin correspondant au servomoteur en question et l’angle de rotation en degrés.

L’idée derrière la fonction est de faire tourner le servo 1° à chaque fois jusqu’à atteindre la position finale en appelant la sous-fonction [myServo.write(position+lastPos[actuator - 1]);] et puis vers la fin on enregistre dans une liste la position finale [lastPos[actuator - 1] += action;]pour savoir par où commencer dans le prochain appel de fonction.

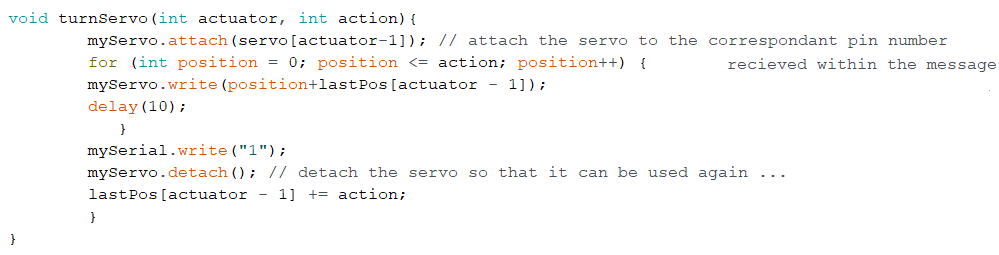


figure 3.b

1. **Buzzer (avertisseur sonore)**

Un avertisseur sonore (fig 4.a) est un élément électromécanique ou piézoélectrique qui produit un [son](https://fr.wikipedia.org/wiki/Son_(physique)) caractéristique quand on lui applique une [tension](https://fr.wikipedia.org/wiki/Tension_%C3%A9lectrique) : le [bip](https://fr.wikipedia.org/wiki/Bip_(son)) [3].



figure 4.a

Dans notre projet, on va utiliser le bipeur dans une seule manière:

* Sonner :  La fonction utilisée pour réaliser cette tâche (voir fig 4.b) est “toneBuzzer” dont les entrées sont “actuator” et “action” qui sont respectivement le pin correspondant au bipeur en question et la durée en milliSeconde pendant laquelle ce dernier va sonner.

Le fonctionnement de la fonction est le suivant: on commence par faire sonner le “buzzer” à une fréquence fixée par l’équipe du projet et indiquée à travers la variable “frequency”   [tone(buzzer[actuator-1], frequency);], puis on attend pendant la durée demandée et enfin on arrête le “buzzer” en appelant la commande [noTone(buzzer[actuator-1]);]

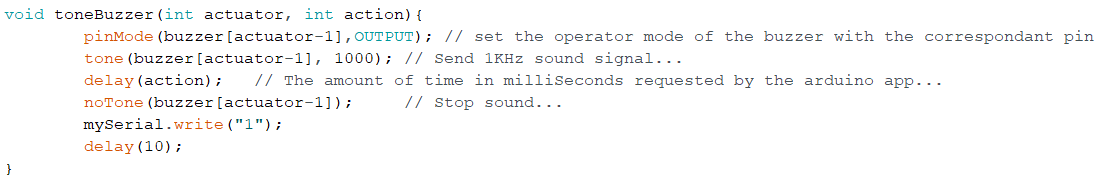


figure 4.b

Maintenant, on passe aux détecteurs dont la mission est de renseigner l’application sur l’état de la plateforme et des tâches physiques faites par l’élève sur la maquette (clic sur un bouton / taux de luminosité…)

1. **Bouton Poussoir**

Un bouton poussoir (fig 5.a) est un interrupteur simple qui est capable de passer à l’état logique “1” lorsqu’on appuie dessus et à l’état logique “0” lorsqu’on le relâche.



figure 5.a

Dans notre projet, l’utilité du bouton poussoir est la possibilité de faire des clics, ainsi, on peut par exemple demander à un élève de cliquer un certain nombre de fois sur un bouton précis pour avancer dans le jeu et passer à l’étape suivante (voir partie Interface dans la section de l’Android). Une seule fonctionnalité a été implémentée dans le code Arduino :

* Cliquer :  La fonction utilisée pour réaliser cette tâche (voir fig 5.b) est “clickButton” dont les entrées sont “actuator” et “action” qui sont respectivement le pin correspondant au bouton poussoir en question et le nombre de fois à cliquer dessus.

      Pour pouvoir compter le nombre de clics, il faut être capable de détecter les transitions d’état logique c’est à dire qu’il faut comparer à chaque fois l’état logique actuel au dernier état logique mesuré [if(digitalRead(button[actuator-1]) !=   lastButtonState)], si c’est différent, on incrémente une variable nommée “buttonPushCounter” et on met à jour le nécessaire jusqu’à atteindre le nombre demandée par l’application Android.

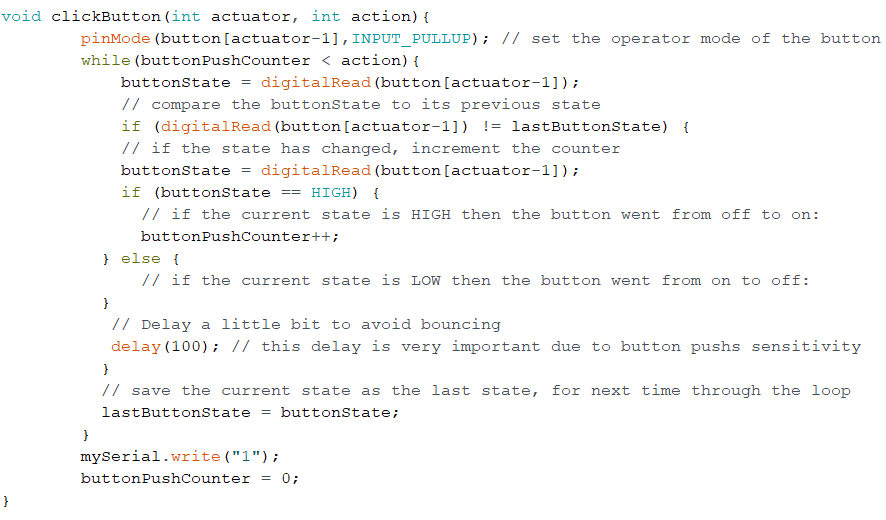


figure 5.b

1. **LDR: Light Dependent Resistor (résistance dépendante de la lumière)**

Un LDR (fig 6.a) est un composant contrôlé par la variation de la lumière absorbée. Plus il reçoit de photons, plus il génère du courant électrique et à partir de cette caractéristique, on va ajouter un autre moyen d’interaction entre l’élève et la maquette.

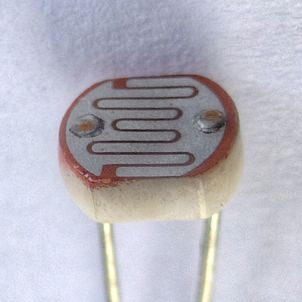


figure 6.a

Dans notre projet, on présente une seule fonction qui va modéliser deux états: - Lumière forte : lorsque la résistance dépasse un seuilFort fixé dans le code, on considère que la pièce en question est claire

- Lumière faible : lorsque la résistance dépasse un seuilFaible fixé dans le code, on considère que la zone en question est sombre

* Détecter :  La fonction utilisée pour réaliser cette tâche (voir fig 6.b) est “detectLuminosity” dont les entrées sont “actuator” et “action” qui sont respectivement le pin correspondant à la diode LDR en question et l’état désiré par l’application (1 pour Lumière forte et 0 pour Lumière faible).

On remarque que dans cette fonction, on utilise une commande d’entrée analogique et non pas numérique [analogRead (LDR[actuator-1])] qui retourne une valeur sur 10 bits entre 0 et 1023 proportionnelle à l’intensité du courant électrique dans le pin correspondant.

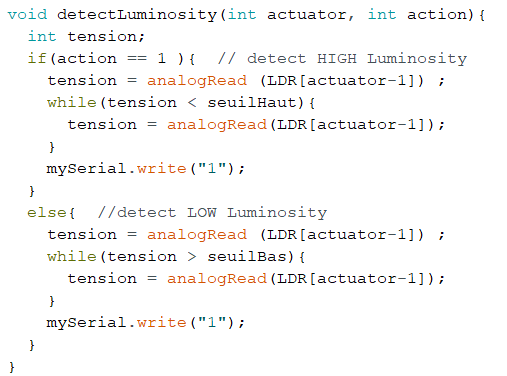


figure 6.b

1. **Capteur InfraRouge**

Un capteur InfraRouge (fig 7.a) est un détecteur qui réagit depuis un rayonnement InfraRouge (IR), il est composé de deux éléments qui sont respectivement  l’émetteur (diode blanche dans la figure): c’est une simple diode dont le domaine de rayonnement est dans le domaine IR invisible à l’oeil nu et le récepteur (diode noire dans la figure): c’est un photodétecteur qui fonctionne par l’absorption des photons réfléchies à partir d’un obstacle.

Ce composant est très précis comparé à d’autres dispositifs ayant des fonctionnalités équivalentes et dans notre cas, on va utiliser sa capacité à détecter un obstacle à proximité, en effet, l’élève reçoit le mission de faire passer sa main, qui va jouer le rôle de l’obstacle, devant le capteur.



figure 7.a

Dans notre projet, on présente une seule manière d’utiliser ce dispositif:

* Détecter : La fonction utilisée pour réaliser cette tâche (voir fig 6.b) est “detectInfraRed” dont l’entrée est “actuator” , c’est le pin correspondant au capteur InfraRouge en question. On utilise qu’un seul donné puisque l’action demandée est toujours la même:  faire passer sa main devant le capteur, ainsi, y’a pas besoin d’envoyer une commande à la carte Arduino qui va servir à rien.

Le principe du fonctionnement de la fonction est le suivant: On commence par envoyer un signal impulsionnel à partir de émetteur [digitalWrite(TRIGGER\_PIN[actuator-1], HIGH);], puis on active le récepteur [measure = pulseIn(ECHO\_PIN[actuator-1], HIGH, MEASURE\_TIMEOUT);]et on calcule le temps entre l’émission et la réception de l’écho et puis on multiplie par la vitesse de la lumière [distance\_mm = measure / 2.0 \* LIGHT\_SPEED;], ainsi, si la distance calculée est supérieure à certain seuil, on considère que la détection est bonne et on renvoie un message positif vers l’application (voir partie suivante sur le Protocole de communication).

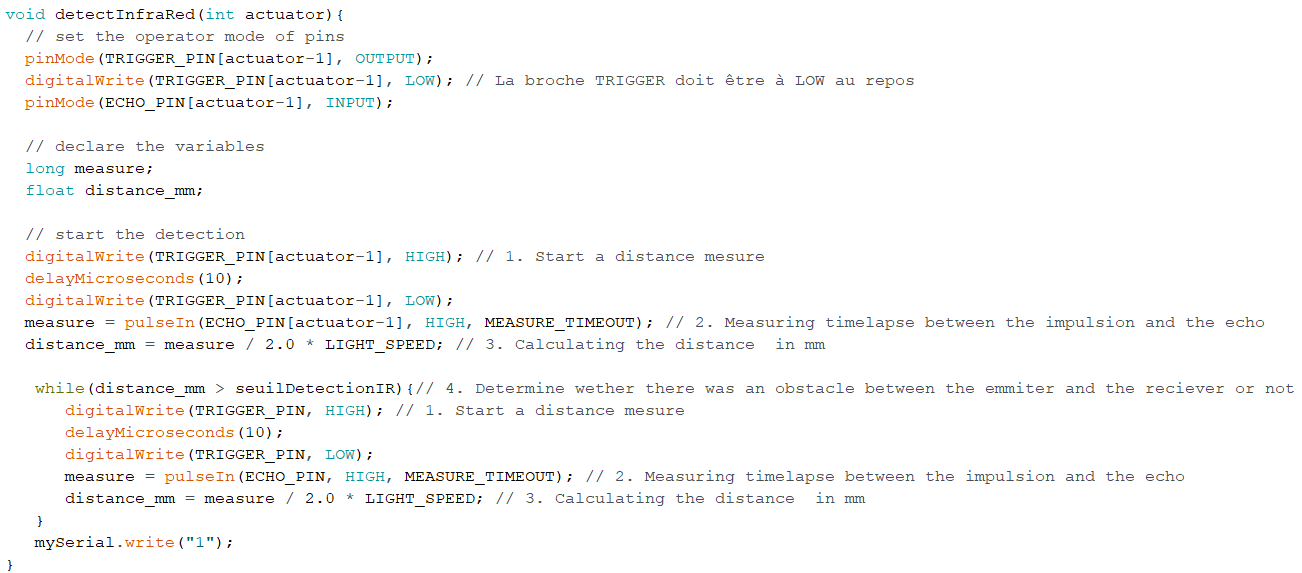


figure 7.b

Les composants présentés forment la liste des dispositifs qui vont être utilisés sur la maquette, mais si jamais on ajoute un nouveau composant, il suffit de lui donner son message unique du Protocole ( voir partie suivante ) et de coder sa tâche et puis d’appeler cette dernière dans la boucle principale du code Arduino.

Puisqu’on utilise différents dispositifs électroniques et qu’on peut trouver plusieurs instances de chaque élément, il fallait bien organiser la mémorisation des données et bien structurer les variables de stockage.

Ainsi, on a opté pour des listes (fig 8) où chacune définie l’ensemble des pins liés à un seul type de dispositif ordonnés dans le sens chronologique de leurs utilisations dans le jeu. Du coup, c’est cette partie qui faut changer après la connexions de tous les composants à la carte Arduino.

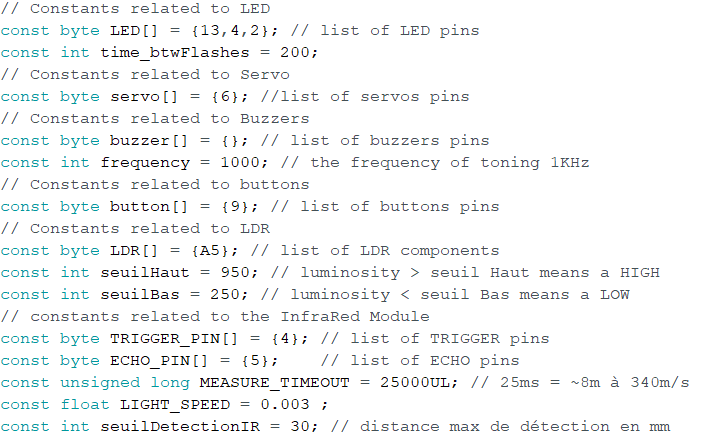


figure 8

Rédacteur: OUALI Maher

Relecteur: HILL Maximiliano

**B.**  Protocole :

Le concept du jeu se base sur l’utilisation de deux plateformes connectées durant le déroulement du jeu et pour cela, on a choisit une connexion Bluetooth à l’aide du module HC-05 d’Arduino (fig 9.a) d’un côté et la carte Bluetooth du smartphone de l’élève de l’autre côté.

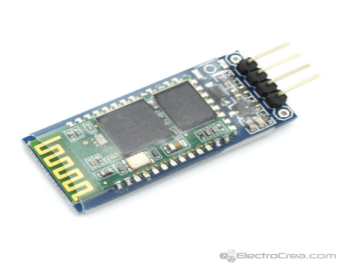


figure 9.a

Vu que les deux extrémités ne puissent pas participer au déroulement en même temps puisque ça sera très difficile de synchroniser les tâches à faire donc il fallait choisir un “maître” (master): celui qui va stocker les données du jeu et lancer des ordres qui est l’application dans notre cas et un “esclave” (slave): celui qui va décoder les messages reçus et répondre aux ordres qui est la carte Arduino dans ce projet.

Un protocole est un moyen de communication qui doit être compris par tous les acteurs en jeu ( Application Android et Carte Arduino dans notre projet ). Certes, il doit résister aux erreurs, être facile et rapide à construire et à décoder. De plus, il doit bien définir toutes les tâches présentes dans le jeu d’une manière explicite et unique sans oublier le fait qu’on doit reconnaître le début et la fin du message pour pouvoir où commencer le décodage, ainsi, on va utiliser un certain ensemble de caractères spéciaux pour chacun ayant sa propre signification.

Dans la partie précédente, on a évoqué deux termes à utiliser dans toutes les fonctions codées qui sont “actuator” et “action”, mais on n’a pas détaillé leurs origines.

Dans ce projet, on utilise plusieurs actionneurs et détecteurs faisant chacun un certain nombre de tâches (clignotement / détection / rotation …) ainsi le format type du message doit donner naissance à ces deux variables beaucoup utilisées dans notre code.

* actuator : Cette variable définit le composant en jeu, ainsi, elle nécessite deux éléments: le type du composant et son numéro qui correspond à son rang dans l’ordre chronologique du jeu (par exemple le 1er LED appelé aura le numéro 1 et le 2éme aura le numéro 2..). Du coup, le message doit contenir en premier lieu un champs pour indiquer le type et un autre pour définir son numéro tout en séparant entre ces deux champs.
* action : Cette variable correspond à la tâche demandée par l’application, ainsi, elle nécessite deux éléments: un caractère définissant le type de la tâche à faire et une grandeur liée à cette tâche (par exemple un entier en mS pour Allumer / un nombre de fois pour Clignoter / un booléen pour Détecter …). Ainsi, le message doit contenir en second lieu un champs pour indiquer le type et un autre pour définir le taux tout en séparant entre ces deux champs.

À partir de ces explications, on met en place le format suivant:

**#X@Y>W%Z**

L’utilité des caractères spéciaux est pour définir le début du message et séparer entre les quatre champs comme exigé.

* X est une lettre parmi cet ensemble { L, S, B, P, D, I } pour représenter respectivement LED, Servo moteur, Bipeur, bouton Poussoir, Diode photorésistante, capteur Infrarouge.
* Y est un entier lié au composant
* W est une lettre parmi cet ensemble { A, C, T, S, C, L, P } pour représenter respectivement Allumer, Clignoter (LED) ; Tourner (Servo moteur) ; Sonner (Bipeur) ; Cliquer (bouton Poussoir) ; Luminosité (Diode photorésistante) ; Passer (Infrarouge).

On remarque qu’on peut pas mettre dans le même message, par exemple, L pour LED et T pour Tourner car c’est incompatible.

* Z est un entier lié à la tâche.

1. ***Connection:***

La connection bluetooth du côté Arduino nécessite l’importation d’une bibliothèque intitulée softwareSerial à l’aide de la commande suivante [#include <SoftwareSerial.h>] et puis l’établissement de la connection se fait en créant une image à l’aide de la commande [SoftwareSerial mySerial(8,7);] avec 8 et 7 sont respectivement les pins de la réception et la transmission.

1. ***Décodage du message:***

La particularité d’une connection Bluetooth est quelle est une liaison série, ainsi, la transmission se fait caractère par caractère et pour pouvoir mettre en place les deux variables évoqués au-dessus, il fallait utiliser des outils booléens pour suivre le début et la fin de chaque champs et celles de la transmission en général (fig 9.b) et du coup, on construit les deux variables “actuator” et “action” petit à petit pour savoir quelle fonction à appeler (fig 9.c).



figure 9.b



figure 9.c

Rédacteur: OUALI Maher

Relecteur: HILL Maximiliano

VII. Conclusion et mise en perspectives :

L’application réalisée permet, grâce aux différentes technologies utilisées, de dynamiser l’apprentissage des mathématiques et de la technologie au collège.

Durant ce projet, nous avons d’abord réalisé les fonctions permettant d’activer les actionneurs de la maquette pour rendre le jeu interactif. Les actionneurs sont maintenant activés selon un protocole propre à la maquette connectée à l’application.

Ensuite, nous avons créé la base de données qui permet de stocker les énigmes et leurs réponses sur un serveur internet externe à l’application.

Enfin, l’application Arduino permet de faire le lien entre les élèves ou les enseignants et les composants de l’application.

Finalement, l’application réalisée répond aux besoins spécifiés par les enseignants. Cependant, il serait possible d’étendre le public intéressé notamment en diversifiant la base d’énigmes. C’est-à-dire en ajoutant une ou deux colonnes permettant de spécifier les thèmes des énigmes (matière scolaire et niveau des énigmes).

Rédacteur :  COURET Hélène

Relecteur : OUALI Maher

VIII. Bibliographie :

[1] *Light-emitting diode* :

<https://en.wikipedia.org/wiki/Light-emitting_diode>(consulté le 15/05/2018)

[2] *Servomoteur* :

<https://fr.wikipedia.org/wiki/Servomoteur> (consulté le 15/05/2018)

[3] *Bipeur* :

<https://fr.wikipedia.org/wiki/Bipeur> (consulté le 15/05/2018)

[4] Site Web Arduino

<https://www.arduino.cc/> (consulté depuis le 10/03/2018)

[5] Site Web MIT AppInventor 2

<http://appinventor.mit.edu/explore/>  (consulté depuis le 21/04/2018)

[6] Sit Web 000webHost Hostinger

<https://000webhost.com>

[7] Tutos php

www.w3schools.com/pHP/default.asp

[8] Php wikipedia

https://fr.wikipedia.org/wiki/PHP

Rédacteurs: BONMARIN Lucas, COURET Hélène, HILL Maximiliano, OUALI Maher

Relecteurs: BONMARIN Lucas, COURET Hélène, HILL Maximiliano, OUALI Maher