

# 2018

## Rapport de Projet

LP SESAM - MECSE



# ClimbGame

Réalisé par : Samuel Flipo / Miguel Barrocas / Rémi Jahnich

# Sommaire

Introduction	3
Présentation du projet	3
Groupe de projet	3
Partie : site web	4
I.    Matériels et prérequis	4
II.   Présentation du site web	4
Partie : interface de jeu	6
I.    Matériels et prérequis	6
II.   Présentation de l'interface de jeu	6
III.  Exécution et explication du code sur Windows	7
Partie : réalité augmentée	8
I.    Matériels et prérequis	8
II.   Récupération du squelette	8
Poursuite du projet	10

## Introduction

Ce rapport a pour but de présenter le projet sur lequel nous avons travaillé : « Escalade en réalité augmentée ».

En tant qu'étudiants/alternants à l'IUT de Cachan, préparant une licence professionnelle MECSE (Métiers de l'Electronique : Communication et Systèmes Embarqués) de parcours SESAM (Systèmes Embarqués et Supervision par Application Mobile), nous avons un module d'activités de professionnalisation (U5) comprenant la réalisation d'un projet.

Ce projet devra être réalisé en l'espace de 72h (18 séances) et a pour but, dans notre cas, de développer une application d'escalade en réalité augmentée.

## Présentation du projet

Le projet « Escalade en réalité augmentée », doit remplir le cahier des charges suivant :

- Développer un site web permettant à un utilisateur d'accéder (localement) à l'interface de jeu ; l'utilisateur devra être recensé dans une base de données ;
- Développer une interface de jeu, dans laquelle l'utilisateur pourra accéder à différents niveaux de jeu, ainsi qu'à un mode de personnalisation de niveau ;
- Intégrer une Kinect Pro au programme : celle-ci devra renvoyer une image (mur d'escalade) à l'interface de jeu et également détecter les mains du squelette de l'utilisateur ;
- Projeter le jeu le un mur d'escalade.

Ce projet est en partenariat avec : Roc & Résine (Thiais Village).

## Groupe de projet

Les différentes tâches à accomplir ont réparties entre chaque membre du groupe de projet et selon les préférences de chacun.



Rémi Jahnich  
SESAM 1 – MECSE  
Développeur Python

Développer une interface de jeu, dans laquelle l'utilisateur pourra accéder à différents niveaux de jeu, ainsi qu'à un mode de personnalisation de niveau.



Samuel Flipo  
SESAM 1 – MECSE  
Webdesigner

Développer un site web permettant à un utilisateur d'accéder (localement) à l'interface de jeu ; l'utilisateur devra être recensé dans une base de données.



Miguel Barrocas  
SESAM 1 – MECSE  
Développeur

Intégrer une Kinect Pro au programme : celle-ci devra renvoyer une image (mur d'escalade) à l'interface de jeu et également détecter les mains du squelette de l'utilisateur.  
Projeter le jeu le un mur d'escalade.

## Partie : site web

Le site web représente la première étape vers l'accès au jeu d'escalade en réalité augmentée ; il s'agit d'un site web local, pour cela nous avons utilisé WampServer (Windows).

### I. Matériels et prérequis

La réalisation du site web ainsi que des bases de données a nécessité :

- Un ordinateur.
- Un éditeur de texte : Notepad++ ou Sublime Text.
- Une plateforme de développement web permettant de faire fonctionner localement des scripts php : WampServer ; celui-ci comprend des serveurs (Apache, MySQL), un interpréteur de script (php), ainsi que phpMyAdmin pour l'administration web des bases MySQL.
- Langage de programmation web : html, css, javascript et php.

### II. Présentation du site web

Pour construire un site web on a recours à des langages client, ceux-ci décrivent comment le site web doit s'afficher :

- Html : son rôle est de gérer et organiser le contenu (texte, images, des liens...) ;
- Css : son rôle est de gérer l'apparence de la page web (agencement, positionnement, décoration, couleurs, taille du texte...) ; ce langage complète le html.
- Javascript : son rôle est de dynamiser la page web (animation, mobilité d'un élément sur la page...) ; il s'agit beaucoup plus évolué que les deux précédents.

L'utilisation de ces trois langages a permis de réaliser notre site web (Figure n°1).

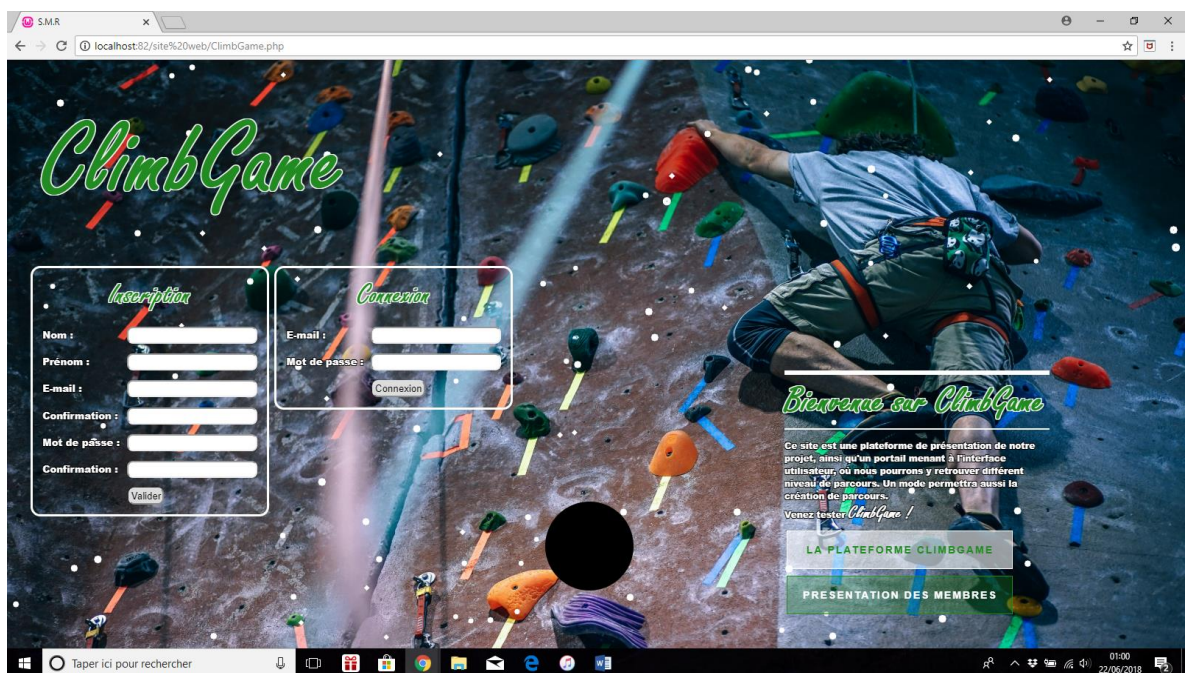


Figure n°1 : site web

Notre site comporte des formulaires (Inscription/Connexion) ; en effet il s'agit d'un site dynamique puisque son contenu peut changer sans intervention du webmaster. Ce type de site utilise d'autres langages et outils en plus de ceux vus précédemment :

- Php : il s'agit d'un langage que seuls les serveurs comprennent et qui permet de rendre son site dynamique ; c'est php qui génère la page web et est combiné à l'outil MySQL.
- MySQL : il s'agit d'un SGBD (Système de Gestion de Base de Données) ; son rôle est d'enregistrer des données. C'est grâce à MySQL que les futurs utilisateurs pourront s'enregistrer sur le site.
- Apache : le serveur web

WAMP (sous Windows) regroupe tous ces outils, ainsi que phpMyAdmin (Figure n°2).

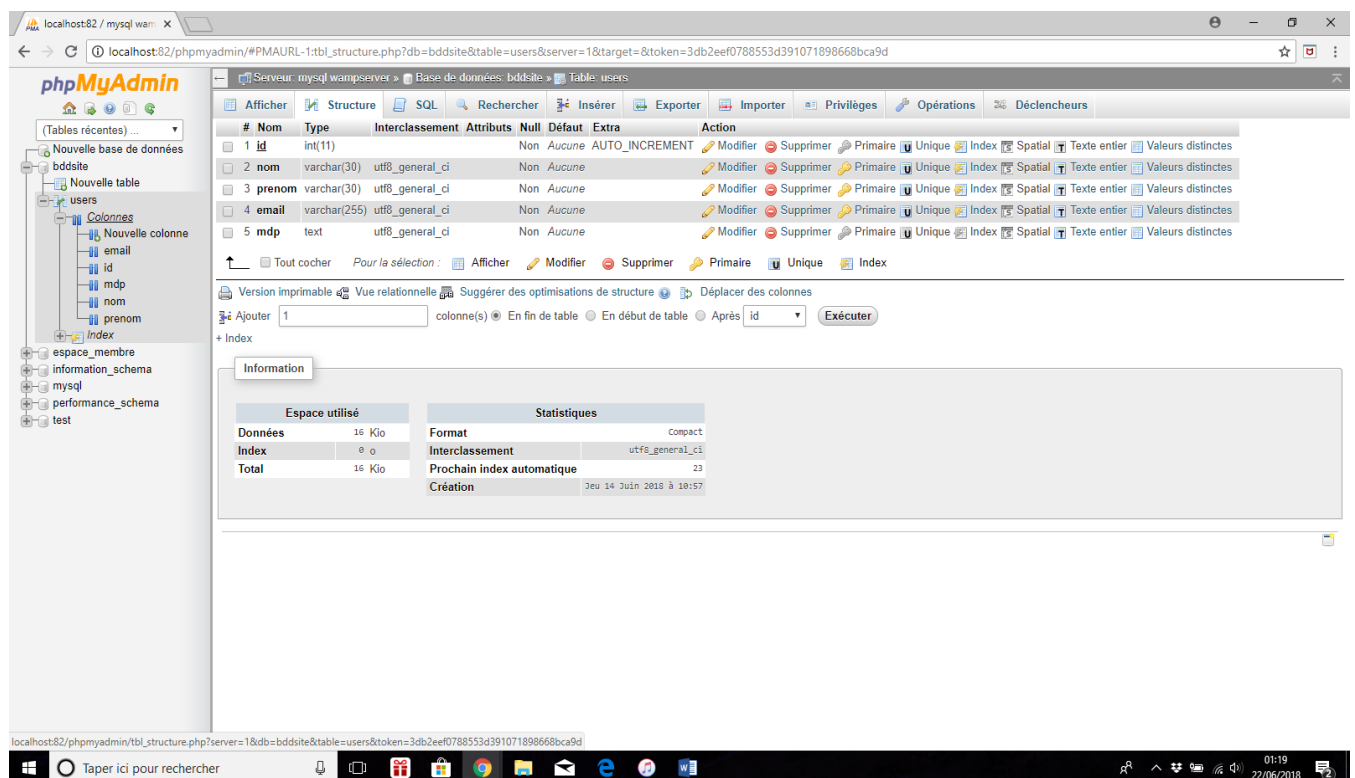


Figure n°2 : table users phpMyAdmin

PhpMyAdmin est un outil qui permet de visualiser rapidement l'état de notre base de données ainsi que de la modifier. Pour ce site web, nous avons créé une table relative aux informations demandées dans le formulaire d'inscription (nom, prénom, mail, mot de passe) ; lorsqu'un utilisateur remplit le formulaire, il est alors référencé dans la base de données.

Plusieurs liens mènent ensuite à l'interface de jeu :

- Validation du formulaire d'inscription ;
- Connexion à son compte si l'utilisateur est déjà inscrit ;
- Le bouton « Démo » (Figure n°1, rond noir).



## Partie : interface de jeu

Il s'agit de l'interface où l'utilisateur, après s'être inscrit ou connecté depuis le site web, pourra soit choisir un niveau prédéfini d'escalade (facile, moyen, difficile), soit en personnalisé un.

### I. Matériels et prérequis

Dans cette partie, nous avons utilisé le langage de programmation Python 3.6.4 (32bit) sur un PC Windows 10.

Pour l'interface, nous avons utilisé les librairies Pygame et os.

Pygame permet de gérer facilement l'affichage vidéo 2D, le son et les périphériques de commandes (clavier, souris...).

Os permet la gestion de fichiers et de dossiers. Celle-ci va nous servir à faire des sauvegardes de donnée dans des fichiers texte.

Pygame <https://www.pygame.org/>

Os <https://docs.python.org/fr/3.5/library/os.html>

### II. Présentation de l'interface de jeu

Dans le premier écran (Figure n°3) nous avons un choix entre quatre niveaux de difficulté. Après le choix fait (en cliquant sur un niveau) on peut cliquer sur le bouton lancer qui nous envoie au deuxième écran.

Dans le second écran (Figure n°4) nous avons un autre choix entre 3 parcours. Les noms des différents parcours sont modifiables (avec le bouton éditer de l'interface) et enregistrés à chaque modification dans un fichier texte. Après le choix d'un parcours, le programme lit les positions des cibles dans deux fichiers texte (un pour la position X, et l'autre pour la position Y). Un bouton "modification cibles" présent sur l'interface permet de déplacer les cibles et de les réenregistrer par la suite.

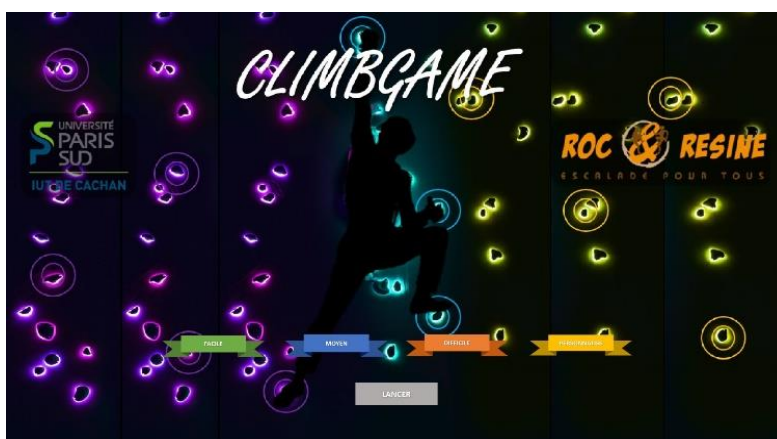


Figure n°3 : écran d'accueil de l'interface de jeu

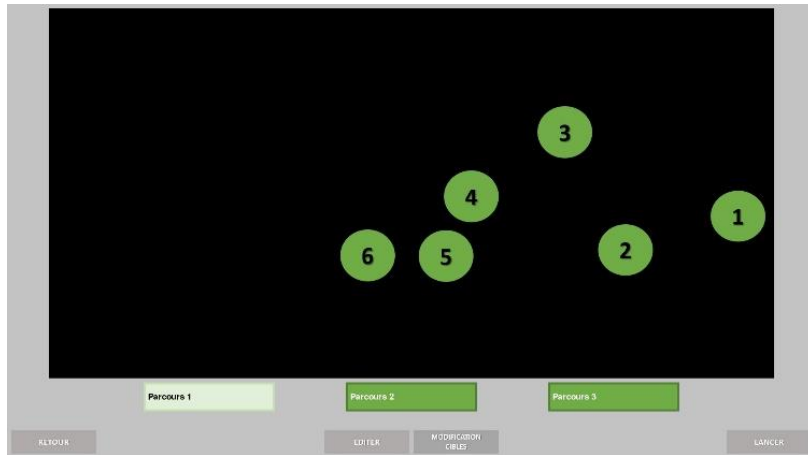


Figure n°4 : écran d'édition de l'interface de jeu

### III. Exécution et explication du code sur Windows

- Exécuter le programme avec IDLE

Faites un clic droit sur le fichier `.py`. Dans le menu contextuel, cliquez sur `edit with IDLE`.

Vous pouvez voir votre code, ainsi que plusieurs boutons. Cliquez sur `module` puis sur `run module` (ou appuyer sur `F5` directement)

- Lancer le programme depuis la console Windows

Il faut ajouter le chemin vers l'exécutable `python.exe` (par défaut se trouve sur `C:\`, je pense) dans la variable d'environnement `PATH`. (Voir sur google)

Puis lancez une console Windows (`cmd.exe`), déplacez-vous vers le dossier contenant le programme et saisissez le nom du programme avec `.py` à la fin.

Ces deux solutions permettent de garder la fenêtre d'exécution du programme à la fin de celui-ci. Ce qui permet de voir les erreurs s'il y en a (**peut être reformulé**). Si vous n'avez pas besoin de cette fenêtre, vous pouvez juste double cliquer sur le fichier `.py`.

#### Explication brève du code :

- Affichage des images selon des conditions ;
- Gestion des périphériques de commandes (clavier, souris) ;
- Enregistrement des données dans des fichiers texte (position X et Y des cibles, nom des parcours).

## Partie : réalité augmentée

Le dispositif : Kinect Pro et projecteur, permet au jeu d'être en réalité augmentée.

### I. Matériels et prérequis

Dans cette partie, nous avons utilisé une caméra KinectV2 qui était à notre disposition. La caméra dispose d'une caméra couleur, de trois capteurs et caméra infrarouge en plus d'une rangée de micro (Figure n°5).

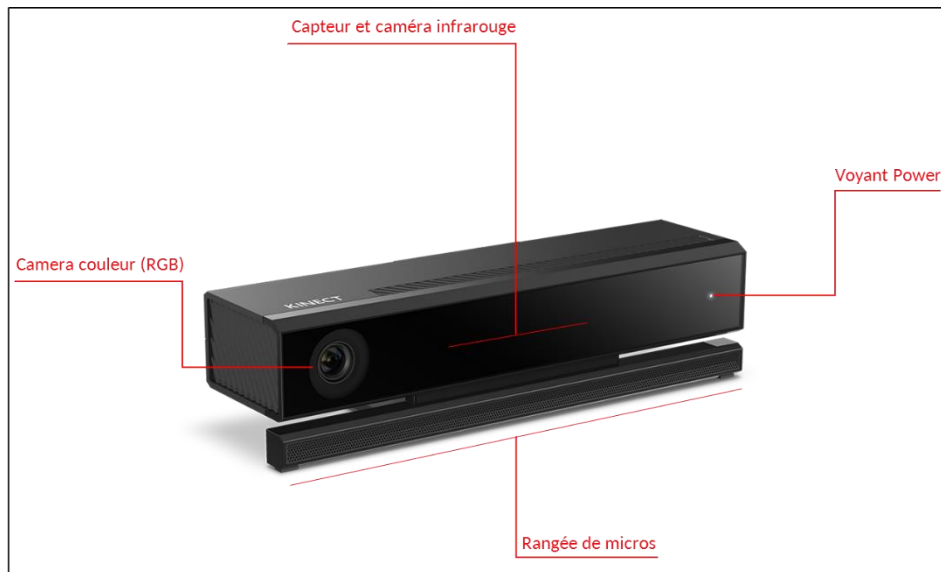


Figure n°5 : Kinect Pro

Nous avons également utilisé un ordinateur avec Windows 10 et le langage de programmation Python 3.6.4 (32bit).

Sur python nous avons utilisé particulièrement les librairies Pygame, PyKinect2

Pygame permet de gérer facilement l'affichage vidéo 2D, le son et les périphériques de commandes (clavier, souris...).

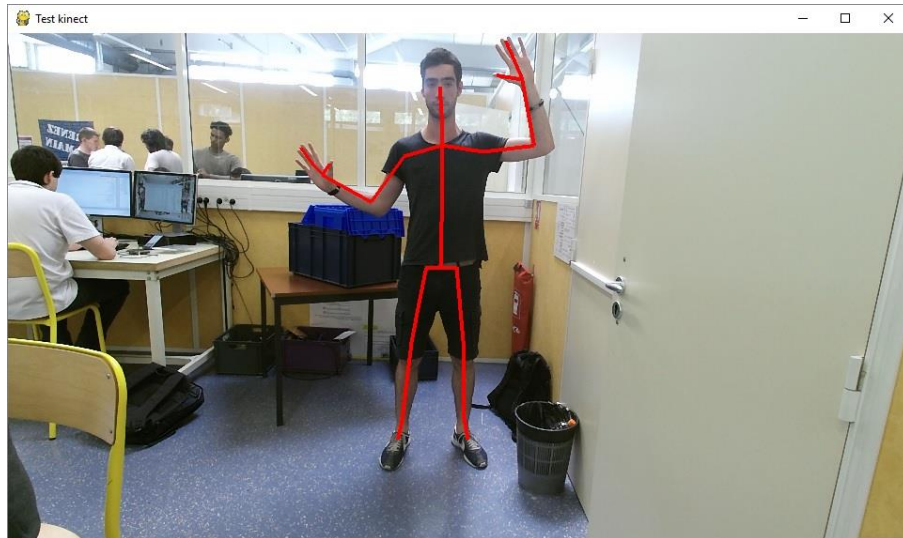
Os permet la gestion de fichiers et de dossiers. Celle-ci va nous servir à faire des sauvegardes de donnée dans des fichiers texte.

### II. Récupération du squelette

Dans un premier temps nous avons utilisé des prototypes de jeu déjà crée afin de se familiariser avec la caméra et la façon dont elle détectait les mouvements.

La caméra détecte les différentes parties du corps humain comme les bras, avants bras, pieds ou encore la tête (Figure n°6).





*Figure n°6 : squelette entier*

Ensuite nous allons assembler tous ces morceaux de corps grâce au langage python en partant de la tête jusqu'au tronc.

Puis à partir du tronc nous allons finir par les mains et pied. (Tronc – bras- avant-bras – mains)

Cette fenêtre de détection doit apparaître une fois dans l'édition des parcours sous forme d'image fixe afin de positionner les cibles sur le mur. Et une seconde fois lors du lancement du jeu afin de détecter les actions du joueur.

Une fois la détection du squelette faite nous avons pu isoler la détection. C'est-à-dire que nous allons réduire le squelette pour qu'il ne détecte que les mains.

Ainsi la fenêtre de jeux n'affiche à présent que les deux parties de la main (Figure n°7).



*Figure n°7 : squelette des mains*

Il nous a ensuite fallu choisir une méthode d'affichage plus propres et plus en adéquation avec le thème du jeu choisis. Ainsi à la place de simple bâton pour la main nous avons choisis des ronds (Figure n°8).

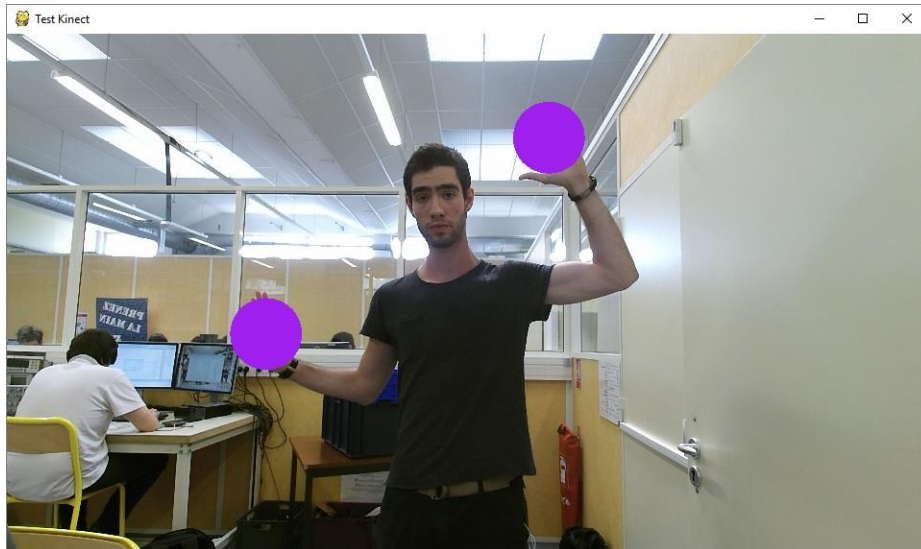


Figure n°8 : squelette rond des mains

Ainsi les ronds nt parfaitement pour la validation des cibles positionné dans l'interface.

## Poursuite du projet

Le projet n'étant pas fini, plusieurs ajouts seront à faire pour le finaliser :



Lien Github : <https://github.com/Rjahnich/Projet-RA>