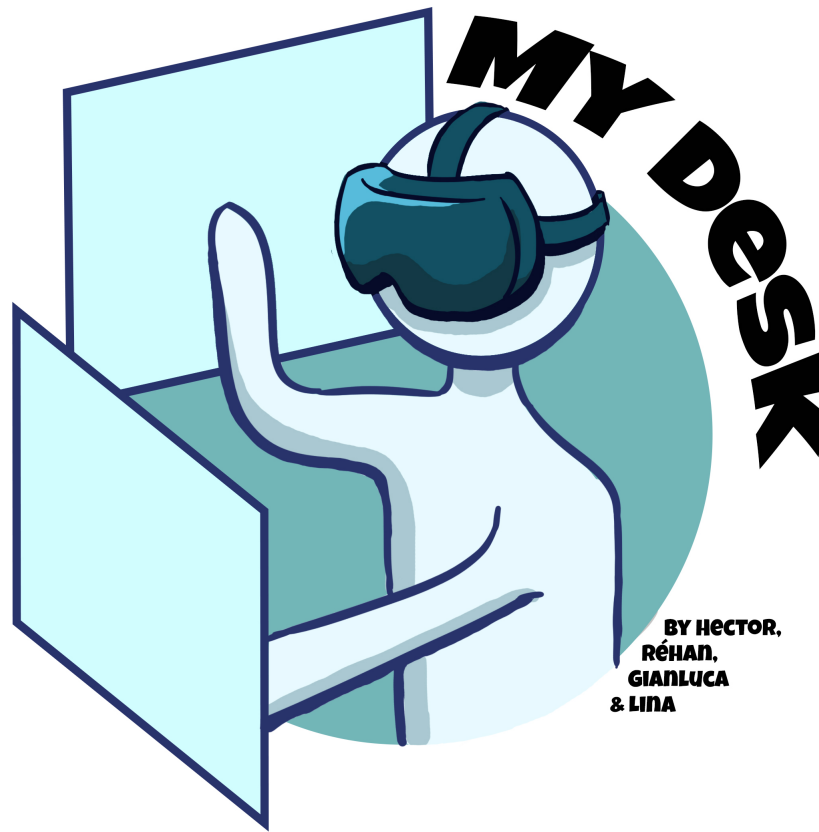


# Rapport de soutenance intermédiaire



EITM

Gianluca Minoprio, Rehan Khan,  
Lina Diringer, Hector De Turckheim

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Rappel du cahier des charges</b>	<b>4</b>
2.1	Projet . . . . .	4
2.2	Tâches à réaliser pour la soutenance intermédiaire . . . . .	4
<b>3</b>	<b>Avancement du projet</b>	<b>5</b>
3.1	Souris et clavier virtuel . . . . .	5
3.1.1	Souris et pointeur	5
3.1.2	Clavier virtuel . . . . .	6
3.2	Création du site web . . . . .	10
3.2.1	Forme du site web . . . . .	10
3.2.2	Hébergement du site web . . . . .	14
3.3	A faire pour la prochaine soutenance . . . . .	14
<b>4</b>	<b>Conclusion</b>	<b>15</b>
4.1	Individuelle . . . . .	15
4.2	Générale . . . . .	16
<b>5</b>	<b>Sources</b>	<b>16</b>
5.1	Tutoriels vidéos . . . . .	16
5.2	Sites web . . . . .	16

## 1 Introduction

Vous vous apprêtez à lire notre rapport de soutenance intermédiaire. Celui-ci aborde les nouveautés apportées à notre projet depuis sa dernière présentation du 12 mars, en particulier la création d'un site web présentant *My Desk*. Nous expliquerons la répartition des tâches par les membres du groupe, ainsi que celles attendues pour la soutenance finale. Bonne lecture !

## 2 Rappel du cahier des charges

### 2.1 Projet

On rappelle que notre groupe souhaite réaliser un bureau virtuel sur Unity. Lors de la dernière soutenance, nous avons réussi à créer un environnement de travail avec un décor changeant. La visualisation des différentes fenêtres du bureau a aussi été implémentée.

### 2.2 Tâches à réaliser pour la soutenance intermédiaire

Voici le tableau présenté dans le cahier des charges, rappelant ce qui doit être effectué pour la soutenance intermédiaire :

Parties	Groupe
Souris et clavier virtuel : -Affichage d'un curseur de souris sur les écrans virtuels à partir du pointage de la manette -Affichage d'un clavier virtuel utilisable pour écrire	Gianluca, Lina
Site internet -Création du site internet présentant le projet	Hector, Rehan

## 3 Avancement du projet

### 3.1 Souris et clavier virtuel

#### 3.1.1 Souris et pointeur

##### EXPLIQUER IMPLEMENTATION DU POINTEUR

#### Modification de la taille des fenêtres à l'aide du pointeur

Une fois notre pointeur de souris implémenté, nous avons voulu offrir à l'utilisateur une nouvelle possibilité : celle de modifier la taille de ses fenêtres ouvertes.

L'idée était la suivante : lorsque le pointeur de la manette est dirigé vers une des fenêtres, la taille de celle-ci pourrait être modifiée à l'aide du joystick de la manette gauche. Lorsque ce dernier est pressé vers l'avant, la taille de la fenêtre augmenterait de façon fluide. A l'inverse, l'enfoncement du joystick en arrière diminue progressivement la taille de la fenêtre pointée.

Pour cela, nous avons modifié la fonction qui donne aux fenêtres leur taille initiale à l'ouverture. Chaque objet *windowTexture* représentant nos fenêtres ont pour attribut une dimension (ou échelle pour 1000 pixels) : *scalePer1000Pixel*. C'est donc cet attribut qu'il a fallu modifier pour redimensionner la fenêtre. Comme on peut le voir ci-dessous, la fonction *ScaleWindow* est entre autres appelée à chaque frame pour chacune des fenêtres ouvertes :

```
void Update()
{
    foreach (var kv in manager_.windows) {
        var windowTexture = kv.Value;
        CheckWindow(windowTexture);
        MoveWindow(windowTexture, usePositionFilter);
        ScaleWindow(windowTexture, useScaleFilter);
    }
}
```

Fonction *Update* appelée à chaque frame

Nous avons donc rajouté deux conditions à notre fonctions : si le joystick est enfoncé vers l'avant, ou bien s'il est enfoncé vers l'arrière. Cela se traduit par l'utilisation des booléens *OVRInput.Get(OVRInput.Button.PrimaryThumbstickUp)* et *OVRInput.Get(OVRInput.Button.PrimaryThumbstickDown)* qui respectivement renvoient vrai si le joystick est appuyé vers l'avant ou l'arrière.

Dans un de ces cas, nous augmentons ou diminuons la dimension actuelle de la fenêtre, en multipliant sa taille par 1.05 ou 0.95. Cette modification de 5% permet un rendu fluide et agréable.

### 3.1.2 Clavier virtuel

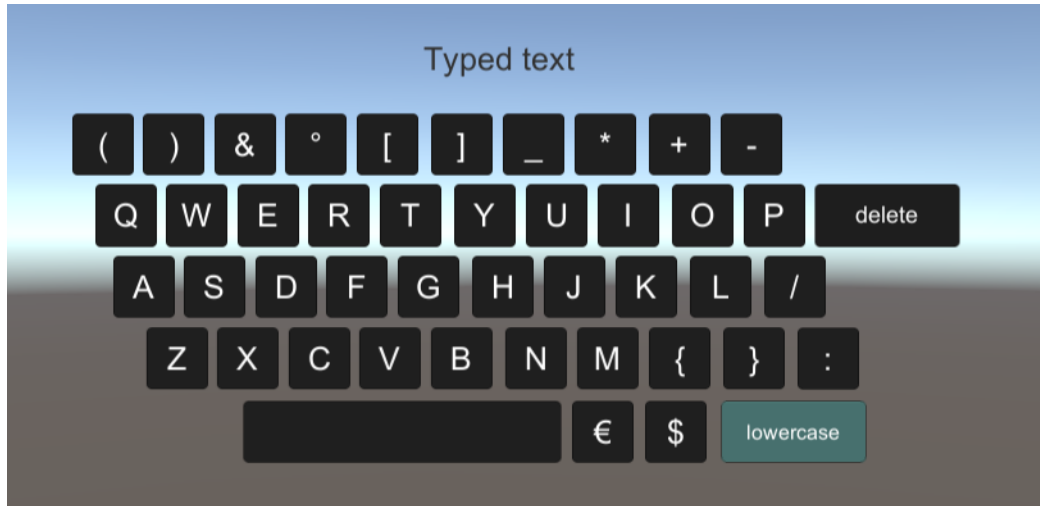
Qui dit souris... dit clavier ! En effet, l'utilisation d'un clavier physique dans la réalité virtuelle n'est pas vraiment pratique, puisqu'on ne verrait pas celui-ci ! Il était donc évident d'intégrer un clavier virtuel dans notre *Virtual Desktop*.

Nous avons fait le choix d'implémenter un clavier de type QUERTY, puisque c'est le plus utilisé au niveau international. Ensuite, afin de rendre notre clavier le plus fonctionnel et le plus similaire à un clavier classique, nous avons créer un bouton permettant d'alterner entre un clavier de lettres minuscules et de lettres majuscules.

Avant d'expliquer son implémentation, en voici un aperçu :



## Clavier de lettres minuscules



## Clavier de lettres majuscules

Notre clavier prend la forme d'un ensemble de boutons noirs en 2D. La plupart de forme carrée, ils permettent chacun d'écrire soit une lettre, un symbole ou un chiffre. Notre texte constitue en fait une chaîne de caractères à laquelle on ajoute au fur et à mesure le caractère correspondant au bouton actionné.

**Ajout de caractère :**

Chaque bouton applique la fonction *alphabetfunction*, qui prend en paramètre une chaîne de caractères *alphabet* :

```
string word = null;
string alpha;
public Text myName = null; ◀ ...
public GameObject minuscule, capitale; ◀ capitale

◀ Event handler ◀ 90 asset usages
public void alphabetFunction (string alphabet)
{
    word = word + alphabet;
    myName.text = word;
}
```

Code de la fonction *alphabetfunction*

Ainsi, notre objet de type Text nommé MyName qui affiche le texte tapé prend pour attribut text notre chaîne *word*, qui permettra son affichage au-dessus des boutons. Pour chaque bouton, c'est donc le paramètre alphabet qui change.

Prenons pour exemple le bouton dont l'étiquette est « B ». Lorsqu'on clique sur le bouton, l'appel à alphabetfunction se fait avec la chaîne « B » passée en paramètre. Le texte affiché sera alors celui originellement présent auquel on ajoute le caractère « B ».

### Suppression de caractère :

Sachant que personne n'est parfait, il fallait pouvoir pallier les fautes de frappes. C'est là qu'apparaît le bouton « delete », qui lui a droit à une fonction particulière. Son principe est simple : la chaîne *word* devient sa sous-chaîne de longueur  $\text{word.Length} - 1$ . Ainsi on affichera la chaîne *word* initiale, mais sans son dernier caractère.



```
public void delete ()  
{  
    word = word.Substring( startIndex: 0, length: word.Length - 1);  
    myName.text = word;  
}
```

Code de la fonction *delete*

### **Alternance du clavier de minuscules et de majuscules :**

A l'image d'un clavier de téléphone mobile, nous avons choisi de ne pas mettre les lettres minuscules et les majuscules côte-à-côte, mais de pouvoir alterner entre deux claviers. Cela rend son utilisation plus ergonomique et plus intuitive.

Pour ce faire, nous avons créé deux GameObject : minuscule et capitale, qui regroupent respectivement l'ensemble des boutons correspondant aux lettres minuscules et aux lettres majuscules. Seul un des boutons reste commun aux deux objets : celui qui permet leur alternance.

Lorsqu'on clique dessus, il appelle la fonction ShowLayout que voici :

```
public void CloseAllLayouts()
{
    minuscule.SetActive(false);
    capitale.SetActive(false);
}

Event handler 2 asset usages
public void ShowLayout(GameObject SetLayout)
{
    CloseAllLayouts();
    SetLayout.SetActive(true);
}
```

Code de la fonction *ShowLayout*

Comme son nom l'indique, elle affiche le `GameObject` passé en argument, grâce à la méthode `SetActive` dont le booléen est `true`. Avant cela, elle fait appel à `CloseLayout`, qui elle masque nos deux `GameObject` `minuscule` et `capitale`. Elle assure qu'un des `GameObject` ne masque pas l'autre, et que seul celui passé en argument de `ShowLayout` soit visible.

Ainsi, le bouton « lowercase » dans `capitale` affichera le `GameObject` `minuscule`, et le bouton « capitals » de `minuscule` affichera `capitale`.

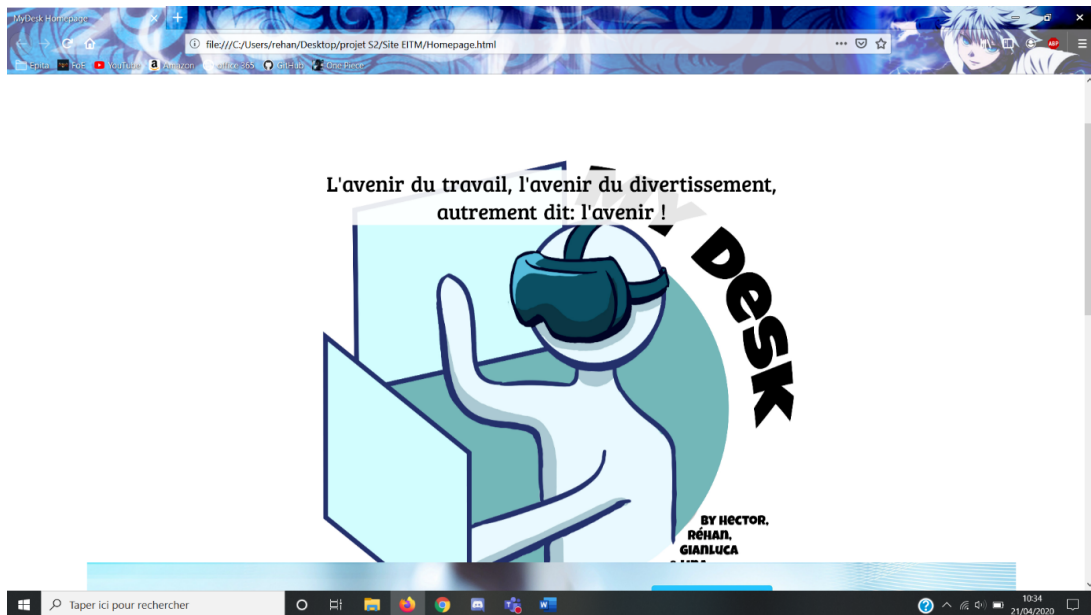
## 3.2 Création du site web

### 3.2.1 Forme du site web

Pour le site web, il a tout d'abord fallu découvrir les différents langages web communs (`html`, `css`, `php`). En apprenant ces langages (dont on a réalisé la différence de syntaxe avec le `C#`, `OCaml` ou `python` qu'on a connu), on a voulu expérimenter l'effet de « parallaxe » sur la page d'accueil, qui a été fait en `css`.

## L'effet de parallaxe

L'effet de parallaxe consiste à donner l'impression qu'il y a différentes couches qui composent le site web et que chaque couche se déplace à une vitesse différente. On voit sur les captures d'écrans ci-dessous que l'image du bas recouvre l'image du haut quand on descend dans la page. Bien que cet effet soit répandu sur les sites web, l'appliquer à notre site a été bien plus difficile que ce à quoi on s'attendait.

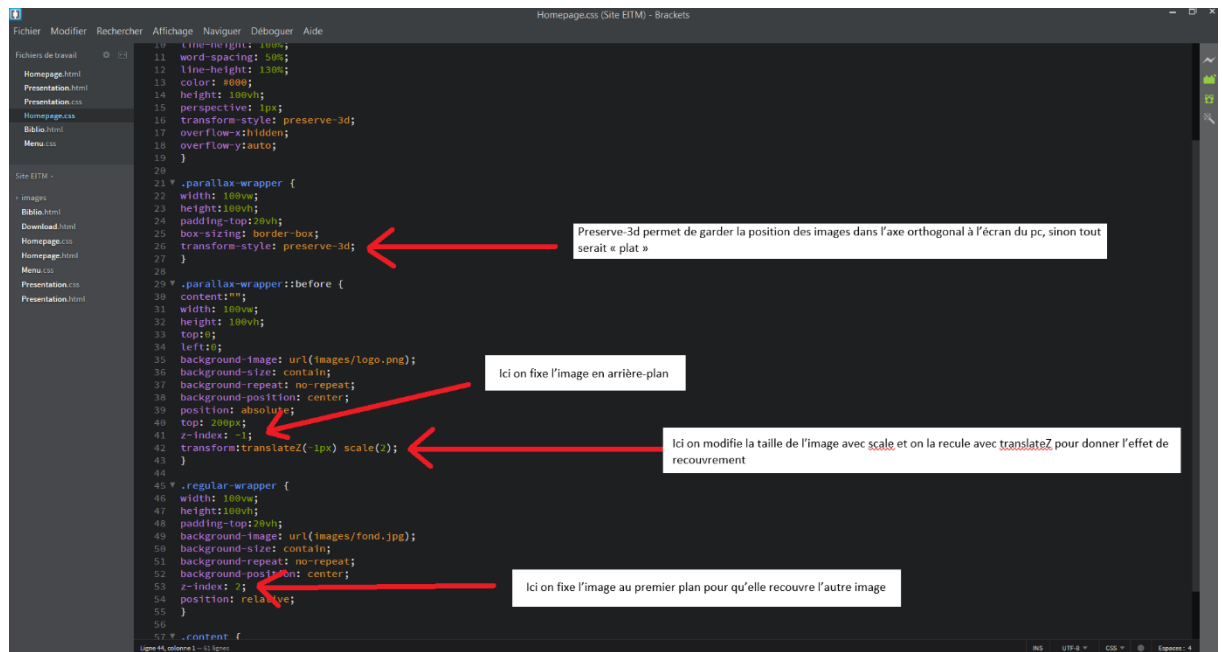


Aperçu de la page d'accueil du site web



Aperçu de la page d'accueil du site web : l'image en bas de la page d'accueil recouvre celle du haut après défilement vers le bas

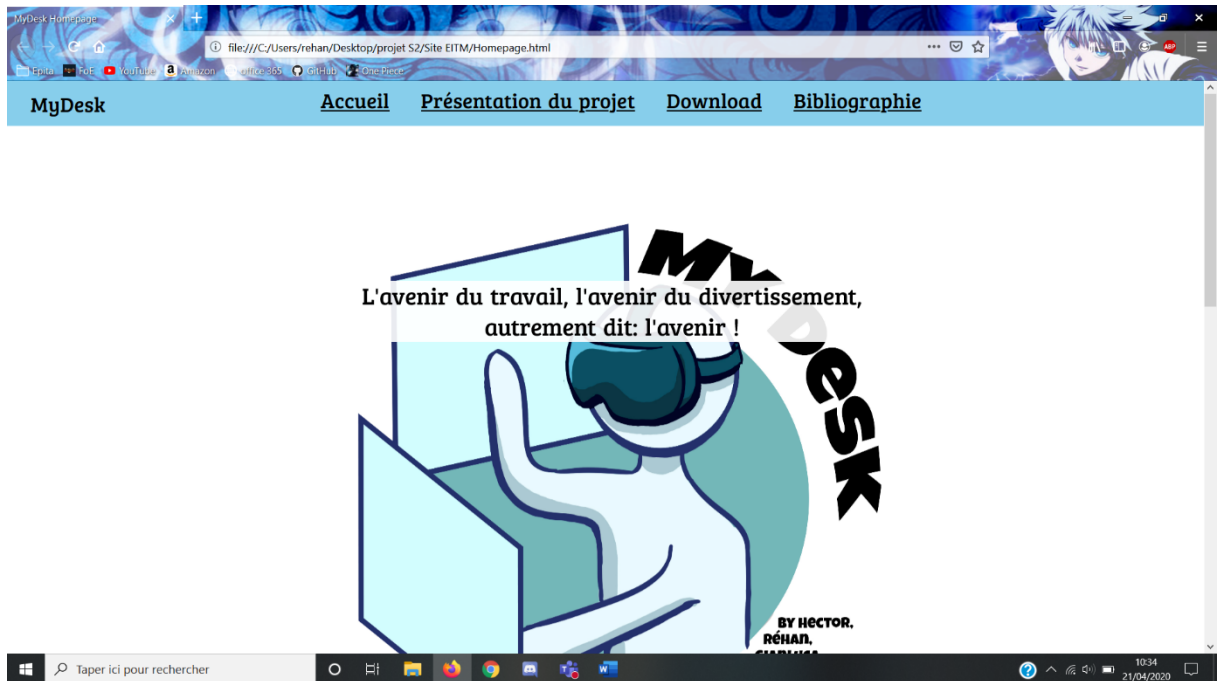
Pour mieux comprendre, voici notre code annoté :



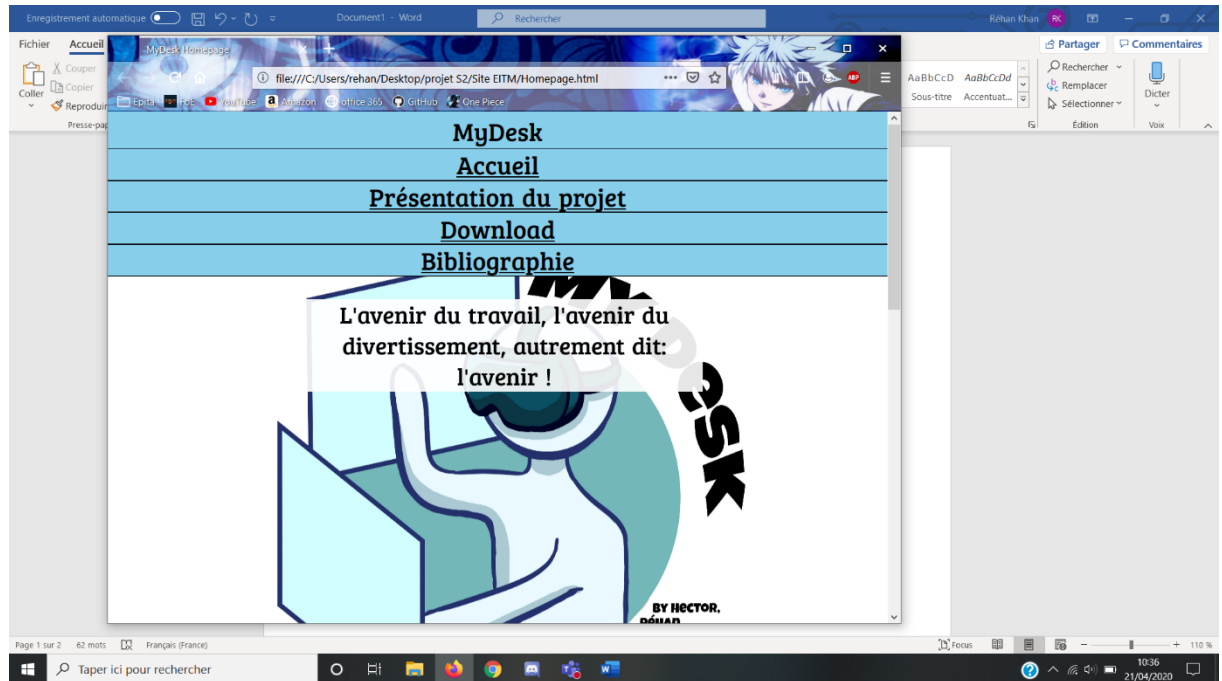
Code de l'effet de parallaxe

## Le menu réactif

On a ensuite créer un menu réactif, qui s'adapte à la taille de la fenêtre, et change de forme si la fenêtre devient trop petite. Voici ci-dessous les deux dispositions possibles :



Disposition du menu sur grande fenêtre



Disposition du menu sur une fenêtre réduite

Cette disposition dépend donc de la taille de la fenêtre. Si celle-ci passe en dessous de 1080 px (pixels css), les liens vers les autres pages disparaissent et le titre « MyDesk » est centré. Si on passe la souris sur le logo, un menu déroulant apparaît avec les liens vers les autres pages du site, ce qui est plus optimisé pour une petite fenêtre. Pour ce menu, on a également dû en apprendre plus sur le css.

### 3.2.2 Hébergement du site web

#### DESCRIPTION HEBERGEMENT

## 3.3 A faire pour la prochaine soutenance

Lors de la prochaine soutenance, nous allons améliorer l'environnement de notre Virtual Desktop et nous implémenterons une nouvelle fonctionnalité à notre projet : une barre de tâches fonctionnelle.

On rappelle l'attribution de ces tâches ci-dessous :

Parties	Groupe
Environnement : -Création d'environnements à thèmes (la mer, la forêt, etc...) basé sur les points de référence déjà mesurés -Création d'un tutoriel d'utilisation de l'application qui apparaîtra à la première utilisation	Gianluca, Hector
Implémentation de la barre de tâche : -Création d'une barre de tâche ergonomique en 3D -Récupération des informations de la barre de tâche du bureau Windows -Affichage des icônes correspondantes sur la barre de tâche en 3D	Rehan, Lina

## 4 Conclusion

### 4.1 Individuelle

#### **Hector de Turckheim**

##### CONCLUSION INDIVIDUELLE

#### **Gianluca Minoprio**

##### CONCLUSION INDIVIDUELLE

#### **Réhan Khan**

Pour cette deuxième soutenance, je devais faire le site web avec Hector. Je pensais avoir des bases en programmation web mais je me suis vite rendu compte que je n'y connaissais presque rien.

Avec le confinement, le travail en groupe était compliqué, par exemple pour vérifier que le fichier .css de l'un fonctionnait avec le .html de l'autre. Finalement on a quand même réussi à bien s'organiser, on s'y est pris suffisamment à l'avance et on a pu faire un site web qui nous plait. On a eu le temps d'apprendre comment le faire et on s'est partagé les tâches avec précision, ce qui nous a permis de finir le site en avance et de régler les petits détails pour qu'il nous convienne vraiment.



Malgré le confinement, on a su s'organiser pour fournir un travail efficace en tirant profit du temps libre dont nous disposions. Je suis satisfait de notre site et fier de notre bonne entente ainsi que de notre organisation.

### **Lina Diringer**

J'ai trouvé ces dernières semaines plus difficiles, notamment dues aux conditions de travail assez inédites et originales, en particulier pour un travail de groupe. En effet, la distance ne joue pas en la faveur de la communication et de l'entraide : voilà un nouveau défi pour ce travail de groupe !

Cependant, malgré ce paramètre, les outils actuels de communication n'ont jamais été aussi utiles : un moyen aussi de mieux s'y familiariser. Grâce à eux, nous avons réussi à faire évoluer notre projet comme nous l'avions souhaité, et je suis encore une fois satisfaite du travail de notre équipe. J'ai encore pu en apprendre plus sur la réalité virtuelle et le fonctionnement d'Unity, auquel je n'avais jamais eu affaire auparavant.

## **4.2 Générale**

# **5 Sources**

## **5.1 Tutoriels vidéos**

- Ep 16. UI keyboard par *Troublesome Girl* sur Youtube
- How to create keypad system in Unity3D par *WatchFindDo Media* sur Youtube

## **5.2 Sites web**

- <https://www.alsacreations.com/tuto/>
- <https://www.openclassrooms.com/>
- <http://www.css-faciles.com/>
- <https://developer.mozilla.org/fr/>