

---

# RAPPORT SOUTENANCE 2023

## **ATIPE**

**Robin R. / Fabien R. / Mehdi T. / Yassin H.**

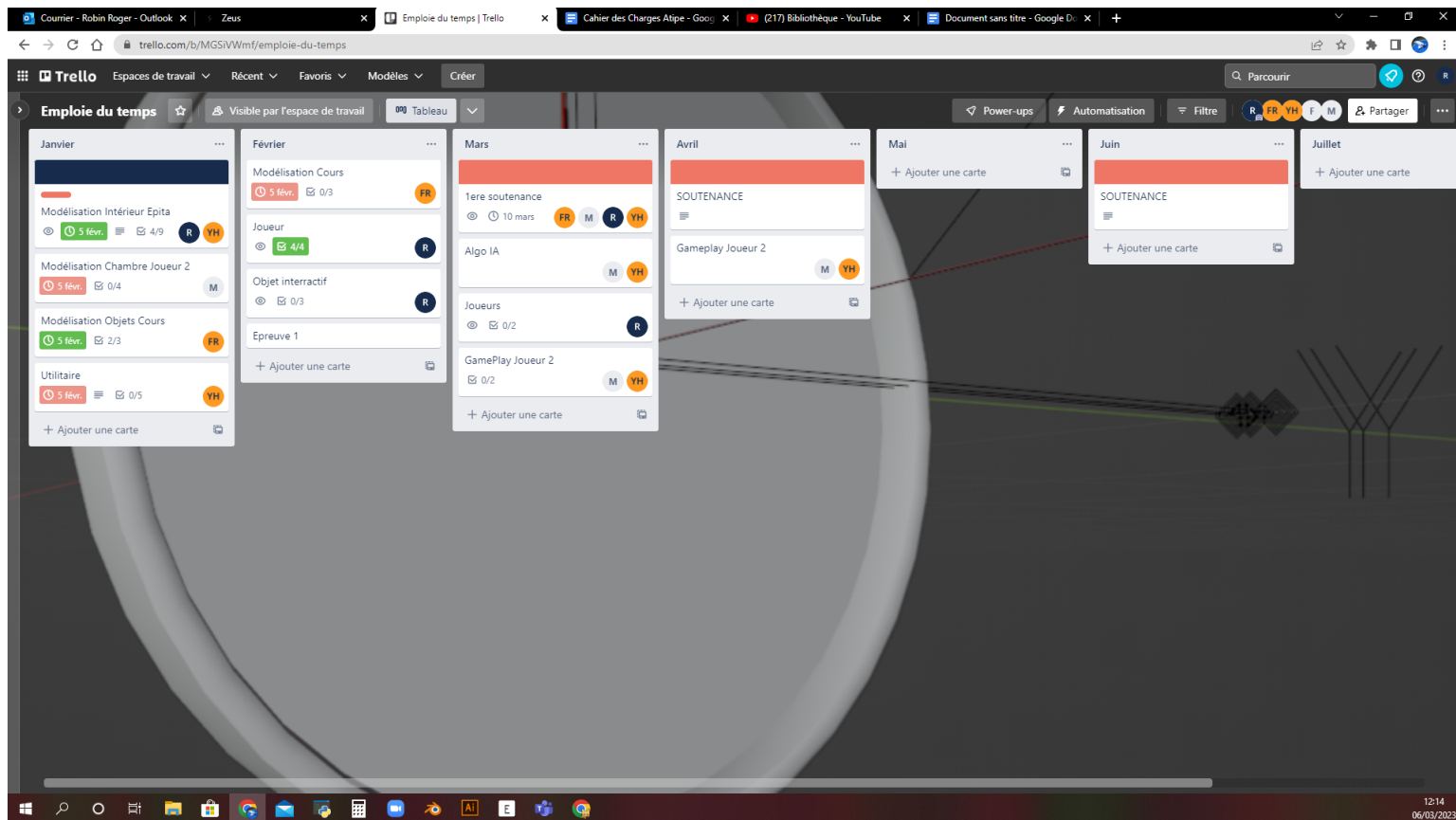
---

## **Tables des matières**

1. Introduction	1
2. Découpage du projet	2
a. Assignment des tâches (Planning)	
3. Avancé	
a. Script	
b. Modélisation	7
c. Programmation	
d. Algorithmes trouvés	

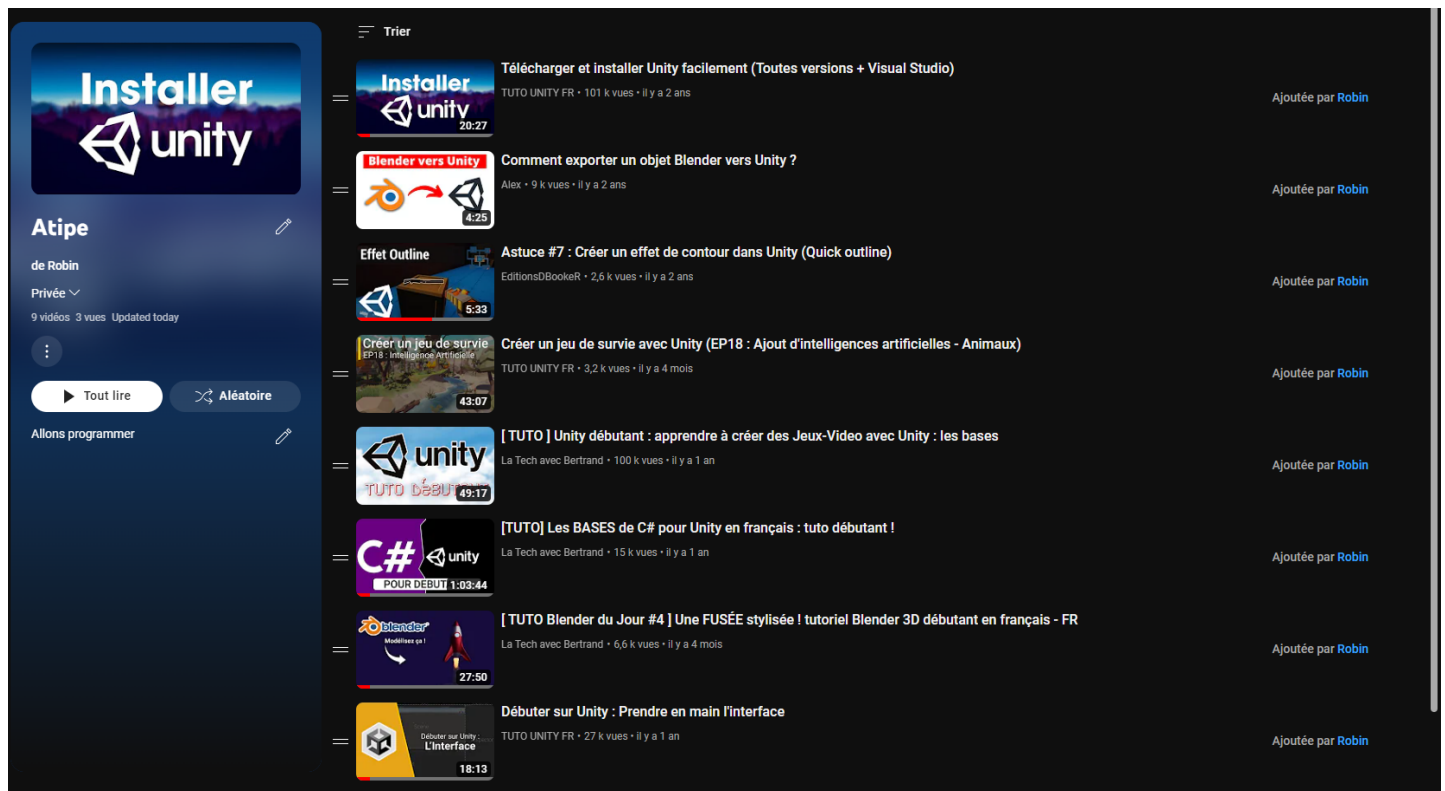
## Introduction :

Pour débuter notre projet nous avons mis en place plusieurs outils afin d'être au maximum organiser. En effet, notre plus grande crainte fut le désordre. Pour pallier cela nous avons ouvert un emploi du temps sur Trello, ou nous pouvons suivre l'avancée de chacun. Le point fort de Trello est que nous pouvons assigner plusieurs personnes sur plusieurs tâche diviser en checkList.



<https://trello.com/b/MGSiVWmf/emploi-du-temps>

De plus, nous partageons une PlayList Youtube, ou nous regroupons toutes les vidéos utiles à la réalisation du projet. Les youtubeurs Unity fr, La tech avec Bertrand ou encore Alex nous ont fortement accompagné dans la découverte de Unity.



En ajoutant à cela, nous avons un serveur discord afin de communiquer efficacement.

**ÉVÉNEMENT  
ANNULÉ**

Nous avons créé un compte google commun afin de se transmettre les modèles blender, mais nous avons arrêtés de l'utiliser car trop de plate forme freinait l'efficacité de notre travail.

## Découpage du projet :

Notre cahier des charges fut légèrement modifié :

### *Ancien découpage du projet*

SOUTENANCE		R.R	F.R	M.T	Y.H	Total
1ère	Programmation du Jeu	5%	5%	5%	5%	20%
	Game design					
	Squelette du Jeu	5%	10%	10%	10%	35%
	Création du terrain	50%	20%	10%	5%	85%
	Déroulé des énigmes	25%	25%	25%	25%	100%
	Animations 3D	5%	5%	5%	5%	20%
2ème	Recherche sur I.A.			50%	50%	100%
	Recherche Multijoueur	50%	50%			
	Programmation du Jeu	10%	10%		10%	50%
	Game Design					
	Squelette du Jeu	40%		30%		85%
	Création du terrain				15%	100%
3ème	Implémentation enigmes	15%	35%			50%
	Conception de L'I.A.			25%	25%	50%
	Multijoueur	50%	50%			100%
	Audio Communication	10%	15%	15%	10%	50%
	Animations 3D	30%		30%		100%
3ème	Programmation du Jeu	15%	10%	10%	15%	100%
	Game design					
	Squelette du Jeu	10%		5%		100%
	Implémentation enigmes		50%			100%
	Création du terrain					100%
	Conception de L'I.A.	50%				100%
	Multijoueur		15%		25%	100%
	Audio Communication				50%	100%
	Cinématique		20%	80%		100%
3ème	Déroulé enigmes					100%
	Animations 3D					

■ **Eléments ajoutés**

■ **Eléments supprimé**

■ **Eléments modifiés**

Nous avons pu affiner notre découpage du projet avec une meilleure répartition des tâches. Deux personnes travaillent sur une tâche plutôt que tout le monde travaille sur la même. Cela est beaucoup plus efficace.

Voici l'ancien cahier des charges :

SOUTENANCE		R.R	F.R	M.T	Y.H	Total
1ère	Programmation du Jeu	5%	5%	5%	5%	20%
	<u>Game design</u>	15%	5%		5%	25%
	Squelette du Jeu	5%	10%	10%	10%	35%
	Création du terrain		10%	15%	10%	35%
	Déroulé des énigmes	25%	25%	25%	25%	100%
2ème	Animations 3D	5%	5%	5%	5%	20%
	Programmation du Jeu	10%	10%		10%	50%
	Game Design		10%	15%	10%	60%
	Squelette du Jeu	20%	10%	10%	10%	85%
	Création du terrain	20%	15%	15%	15%	100%
	Implémentation enigmes	15%	35%			50%
	Conception de L'I.A.	10%	10%	20%	10%	50%
	Multijoueur	10%	15%	10%	15%	50%
3ème	Audio Communication	10%	15%	15%	10%	50%
	Animations 3D	15%		25%		60%
	Programmation du Jeu	15%	10%	10%	15%	100%
	Game design	10%	10%	10%	10%	100%
	Squelette du Jeu	10%		5%		100%
	Implémentation enigmes		50%			100%
	Création du terrain					100%
	Conception de L'I.A.	50%				100%
	Multijoueur		15%		25%	100%
	Audio Communication				50%	100%
	Cinématique		20%	80%		100%
	Déroulé enigmes					100%
	Animations 3D	25%		15%		100%

## ■ Modification

## Avancé :

### a. Script

#### Sommaire (chronologique)

Extérieur : Récupération des clés

RDC : Lumières

Etage 1 : Caméra

Etage 1 :

Etage 2 :

Etage 3 : PAS D'ÉTAGE 3

Etage 4 : Amphi

Etage 5 :

Sous-sol :

Etage 6 : Récupération des Partiels (fin)

#### Informations utiles

Personnage 1 : personnage qui veut récupérer les infos sur les partiels.

Personnage 2 : personnage qui est chargé de guider son coéquipier.

First Person Controller, zqsd pour se déplacer, "e" pour interagir avec un objet, "i" pour avoir accès à son inventaire.

Les deux personnages sont en communication à partir du moment où le vocal s'active et ils le resteront toute la partie. (Vocal)

Lorsque le joueur à Epita se fera toucher par le gardien, il reviendra à l'entrée mais NE PERDRA PAS SA PROGRESSION. (ceci nous évite les conflits de sauvegarde trop complexes.)

#### Environnement :

Scène 1 : Dans la rue devant Epita

Scène 2 : La chambre du perso 1.

Scène 3 : Intérieur d'Epita

Scène 4 : Cour (à la sortie du réfectoire)

## Extérieur :

### Description :

*Perso dans la rue. Ambiance angoissante, avec du brouillard, la nuit des lampadaires avec peu de lumière.*

- POV Perso 1 :

### Enigme :

*Personnage explore, il va vers l'entrée d'Epita.*

*Il trouve un papier qui dit : "Le gardien est ici toute la nuit, il paraît qu'il détient un double des clés du Campus dans sa voiture".*

*Il doit trouver la voiture du gardien.*

*Un message en bas de l'écran s'affiche "Mmmh... la voiture est fermée il me semble avoir vu une caisse à outil traîné devant, je pourrai jeter un coup d'oeil "*

*Le personnage récupère un pied de biche.*

*Il s'en sert pour ouvrir la voiture et récupérer les clés*

*Il peut ainsi s'introduire dans Epita.*

- POV Perso 2

### Enigme :

*Personnage se trouve dans sa chambre.*

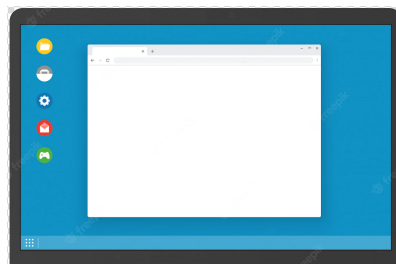
*Il reçoit un message de son coéquipier.*

*Il explore sa chambre et trouve son ordinateur.*

*Il doit le déverrouiller.*

*Lorsque le perso accède à son ordinateur, il est envoyé sur une nouvelle scène un écran d'ordinateur*

*Exemple :*



*Le joueur devra avant tout déverrouiller l'ordinateur.*

*Il aura ainsi accès à des caméras de surveillance d'Epita.*



*Le joueur a trop fait la fête la veille il doit se souvenir de son mdtps.  
Pour cela, il doit trouver des indices dans sa chambre.  
rébus pour trouver*

**RDC :**

Description :

*Le joueur arrive dans Epita. Le gardien dort l'IA n'est donc pas activé.  
Porte entrouverte avec gardien qui dort.*

- **POV Perso 1 :**

Enigme :

*Le personnage est dans l'obscurité. Il trouve une lampe torche pour  
l'aider à voir plus clair.*

*Son but est d'aller dans le réfectoire.*

*Il doit trouver un disjoncteur.*

*Lorsqu'il rentre un message sera affiché : "Il fait bien sombre ici, il me  
semble avoir déjà vu un disjoncteur près d'une machine à café"*

*Il trouve le disjoncteur.*

*Le disjoncteur aura un modèle particulier. Le joueur devra renseigner le  
modèle à son coéquipier.*

*Son ami aura accès au mode d'emploi du disjoncteur. Ce mode d'emploi  
contiendra un code couleur à taper. Ce qui permettra d'actionner les  
lumières.*

- **POV Perso 2 :**

Enigme :

*Le perso a accès au PC.*

*Il devra fouiller dans les fichiers afin de trouver le mode d'emploi du bon  
modèle de disjoncteur.*

*Il renseignera le code à son coéquipier.*

## 1er étage :

### Description :

*Le joueur arrive dans Epita. Le gardien dort l'IA n'est donc pas activé.*

- POV Perso 1 :

### Enigme :

*Le personnage devra suivre les indications de son coéquipier.*

*Il trouvera au 1er étage une affiche avec des informations sur des changements d'Epita.*

*“GRAND CHANGEMENT À EPITA*

*Rénovation des toilettes*

*Atelier Création de jeu de Palet*

*Ouverture d'un nouveau club de Polos*

*Nouveaux Casiers situé au 1er étage*

*Soirée au Perroquet Bourrée le 31 février*

*“*

*Cette affiche lui donnera l'indication de trouver les casiers. Il aura le code grâce à son coéquipier.*

*Le joueur détient la clé usb.*

*Avec l'aide de son coéquipier, le joueur devra trouver les serveurs d'Epita.*

*Les serveurs se trouveront dans une pièce secrète.*

*Cette pièce est située dans les escaliers reliant le rez-de chaussé et le premier étage.*

*Le logo IONIS est une porte.*

*Pour ouvrir cette pièce le joueur passera devant la machine à café et verra affiché “Tiens, la machine à café à l'air fonctionnelle, pour une fois...”*

*Un bouton sur cette même machine permettra d'ouvrir la porte.*

*Il introduira la clé USB à l'intérieur.*

*Ceci débloque l'accès au caméra de l'étage 1, au joueur n2. Mais active aussi l'arrivée du gardien.*

- POV Perso 2 :

Enigme :

*Le joueur aura accès à une application Caméra.*

*L'application sera bloquée au départ.*

*Il devra fouiller dans ses fichiers et trouver un dossier Projet Hacking Caméra.*

*Ce dossier contiendra un README avec dedans "Code casier : 5944" (style emacs CamL)*

*Dans ce même dossier il y aura un fichier qui renseignera que la clé usb devra être introduite dans les serveurs de Epita.*

## 2eme Etage :

Description :

*Le joueur arrive au second étage, les caméra sont activé a l'étage n1*

- POV Perso 1 :

Enigme :

*Afin d'accéder à l'étage supérieur, le joueur devra se rendre ensuite dans les bureaux des professeurs afin de trouver une clé qui permettra d'accéder au prochain étage.*

*Les portes seront déverrouillées lorsque les caméras seront activées.*

*Le joueur en arrivant devant les bureaux verra affiché : "Il me semble que je me trouve dans les bureaux des professeurs, j'ai entendu dire que M.Saber travaillait très tard, il faut que je sois bien vigilant."*

*Le joueur aura donc deux choix d'ouverture de porte.*

*Il pourra ouvrir le bureau intitulé Professeur, ou il y aura un professeur en train de travailler. Si le joueur rentre dans la pièce, il perd la partie.*

*Le bon choix sera la salle de la secrétaire.*

*A l'intérieur il sera guidé par son coéquipier afin de fouiller son bureau.*

- POV Perso 2 :

Enigme :

*Le joueur aura accès à un dossier d'archive avec des anciennes vidéos des caméras  
L'une d'elle indique l'emplacement de la clé. (vidéo de la secrétaire qui pose sa clé dans un certain endroit.)*

### 3eme Etage

#### Description :

Pas d'étage 3 à Epita, Ce sera précisé dans l'interface des caméras du joueur 2.

### 4eme Etage

#### Description :

*Découverte de l'amphithéâtre, les deux joueurs seront coupés en arrivant à ce niveau.*

- **POV Perso 1 :**

#### Enigme :

*Le joueur devra essayer d'allumer le projecteur, il devra aller chercher un ordinateur situé dans les rangs de l'amphithéâtre, dans un sac oublié par un élève.*

- **POV Perso 2 :**

#### Enigme :

*Le PC affichera un problème réseau, le joueur sera contraint à quitter son PC et devra résoudre le problème. Il sortira de sa chambre*

## 5eme Etage (final)

### Description :

*Etage final, le joueur trouvera une salle secrète où repose les partiels.*

- POV Perso 1 :

### Enigme :

*Le joueur devra trouver une salle avec un tableau ou il y aura marqué ACDC. Les pc seront allumées et sur chaque pc il y aura marqué la commande linux "man ascii"*

- POV Perso 2 :

### Enigme :

*Le joueur aura accès à un fichier contenant la table ascii et devra guider son ami pour trouver le code.*

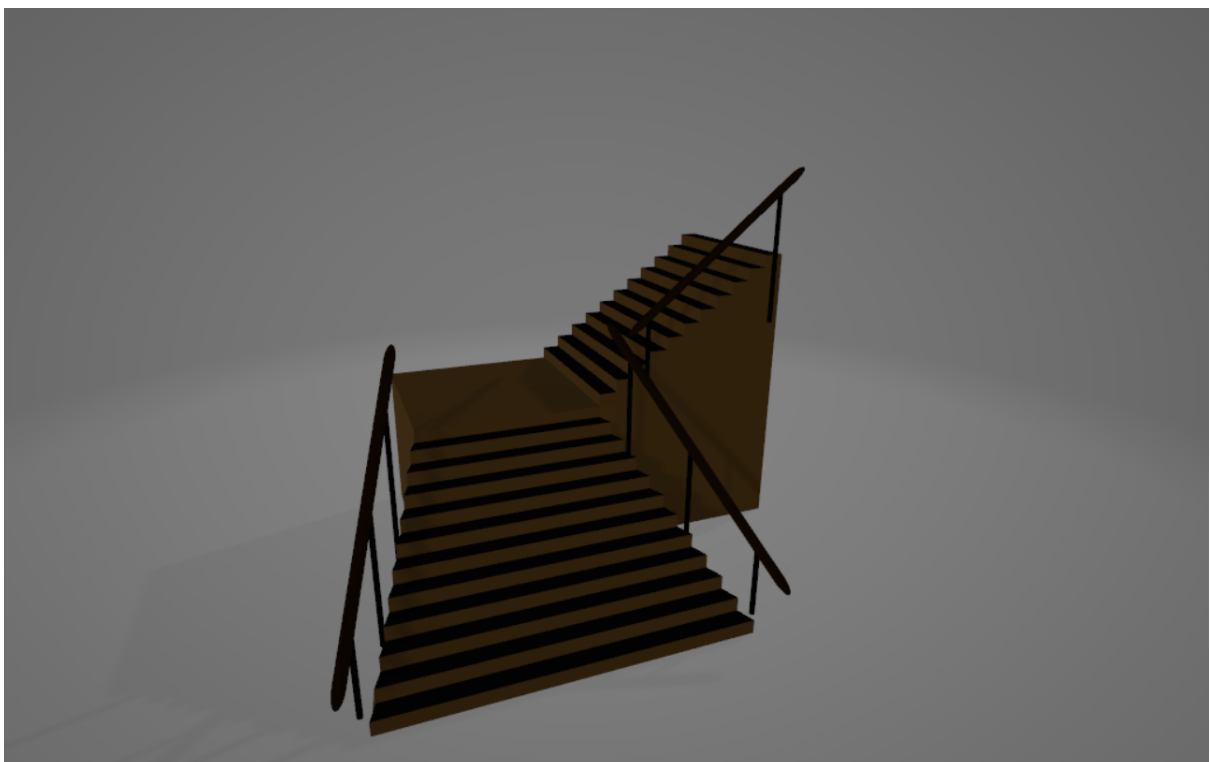
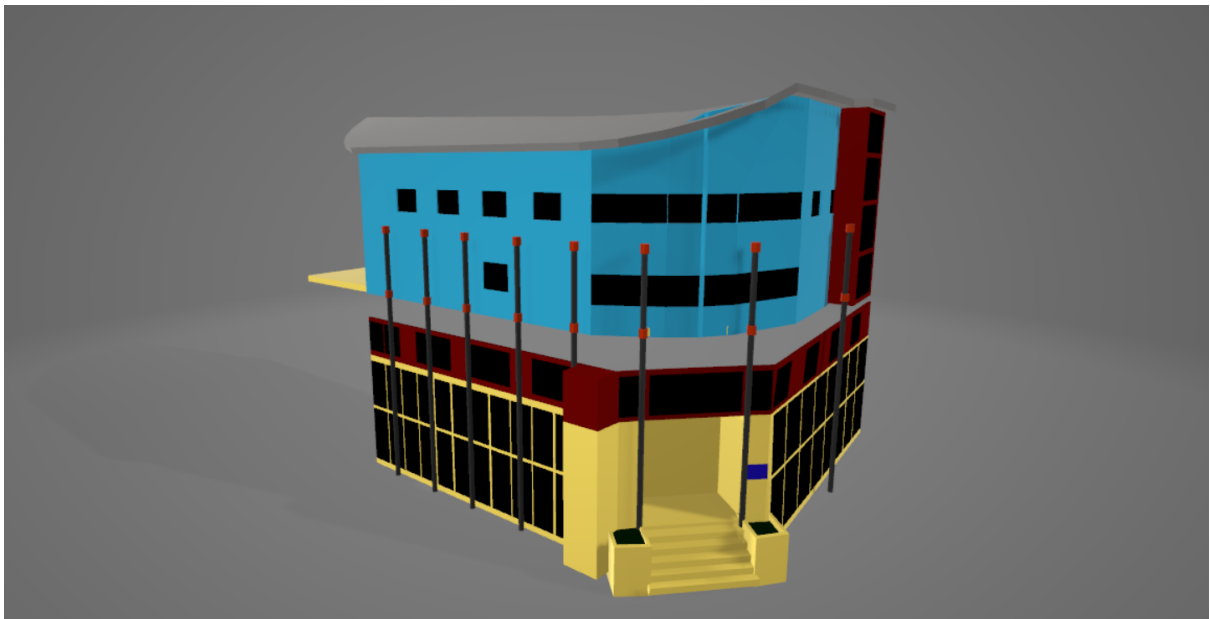
### Fin :

*Le joueur récupérant les partiels débloque la sortie et peut s'échapper tout en fuyant le vigile.*

## b. Modélisation

Pour modéliser le jeu nous avons tout appris avec le logiciel blender. Nous avons créé la grande partie des objets, néanmoins nous avons fait appel à des sites gratuits qui proposent des modèles 3D. Comme par exemple, TurboSquid, Open 3D Model, Free 3D.

Voici deux exemples de modèles que nous avons réalisés :



## c. Programmation

Nous avons débuté la programmation avec la création du squelette du joueur.

Voici comment nous avons divisé la programmation du personnage : (Robin)

- Player Controller

Ici on gère les touches pressées par le joueur afin de bouger avec l'aide des commandes `GetAxisRaw`. De plus, nous avons fait appel à la structure `Vector3` déjà implémentée dans Unity. Cette structure est utilisée dans Unity pour transmettre des positions et des directions 3D. Il contient également des fonctions pour effectuer des opérations vectorielles courantes. Dans cette classe on gère aussi la rotation du joueur, en gérant les mouvement de la souris

- Player Motor

Le script gère le mouvement du joueur. Il est relié à Player Controller qui lui s'occupe comme son nom l'indique des contrôles du joueur. Il utilise le `Rigidbody` de notre personnage et lui appliquera des changements de positions. La méthode `PerformMovement` est chargée de faire le mouvement.

- Player interact

Ce script est chargé de l'interaction du joueur avec les objets. Nous utilisons un Raycast qui est une ligne invisible constamment présente dans la caméra du joueur. Ainsi lorsque le joueur aura au centre de son point de vue un objet il sera en capacité d'interagir avec. Le système d'inventaire étant en développement, nous pouvons pour le moment choisir quel objet peut interagir avec le joueur.

## Intelligence artificielle : (Yassin)

L'intelligence artificielle (le gardien) devra s'activer à l'étage 2. Elle se déplacera dans tout le bâtiment à une vitesse assez lente. Elle aura un rayon de détection, si le joueur se trouve dans ce rayon (et qu'il n'est pas caché), elle le poursuivra, à une vitesse légèrement inférieure à celle du joueur pour permettre à ce dernier de lui échapper.

Pour coder l'IA, nous allons utiliser le module UnityEngine.AI. Nous allons coder 3 fonctions principales, Patrolling, Chase et Update. La fonction patrolling permettra de définir l'état de base de l'IA, c'est à dire la faire se déplacer dans tout le bâtiment sans poursuivre le joueur. La fonction Chase sera l'état de poursuite du joueur. Enfin, la fonction update permettra la détection du joueur dans le rayon de l'IA et le changement d'état.



Après avoir codé très rapidement ces trois fonctions, on obtient bien le résultat souhaité. Evidemment, il reste encore la modélisation et l'intégration dans le jeu, mais les bases de l'IA sont là, elle effectue bien tout ce que l'on lui a demandé et qui est décrit ci-dessus.

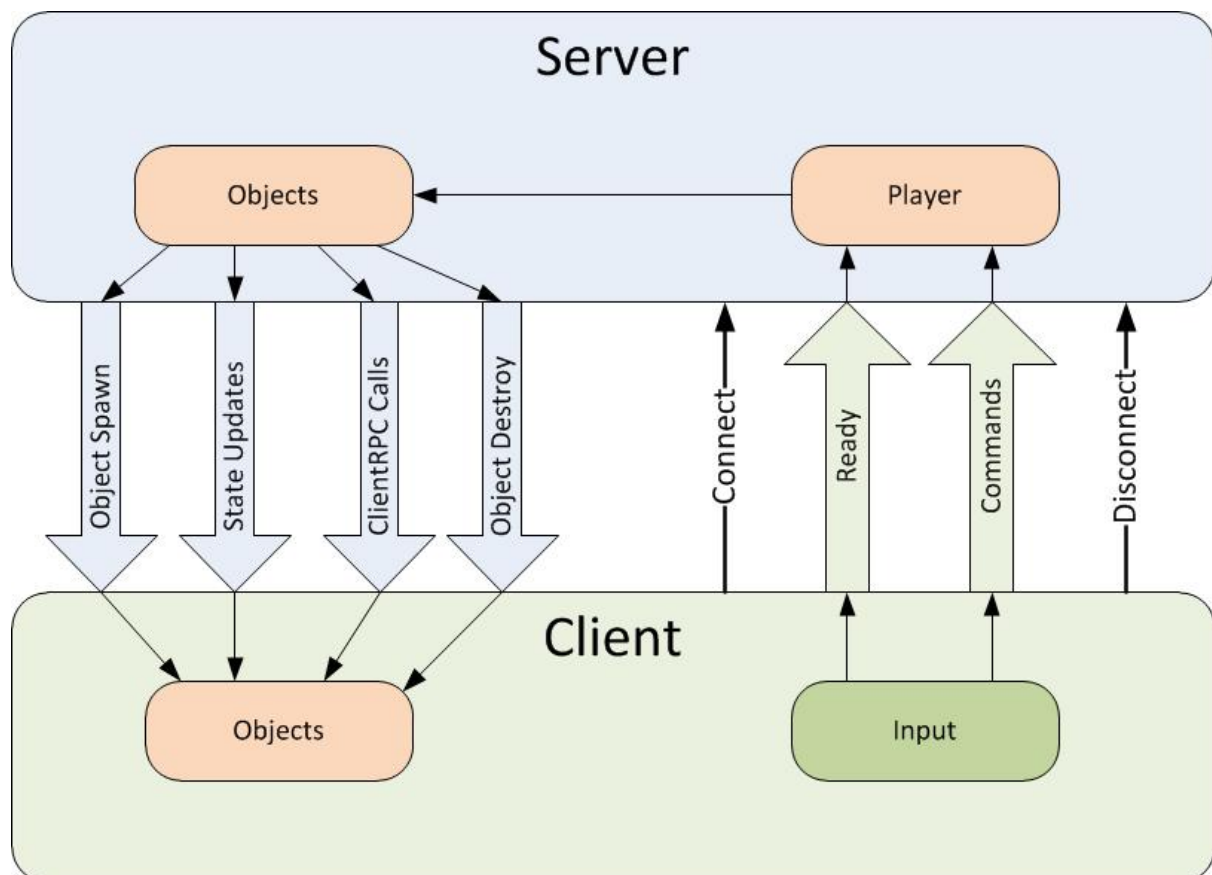


## Multijoueur : (Robin)

Pour synchroniser nos joueurs nous allons utiliser l'asset gratuite proposé par Unity, Mirror. Mirror est un outil efficace afin de générer la mise en réseau du projet. Deux composants de Mirror le NetworkManager (permet de gérer le réseau) et le NetworkHUD (permet de gérer les personnage accueillie sur le réseau).

Mirror utilise un système de communication entre chaque objet qui interagit dans le jeu ainsi nous devons ajouter un NetworkIdentity a chaque objet utile de notre jeu. (ex : joueur / item / etc ...)

### Transmission des informations avec Mirror



Un problème que nous avons rencontré lors de la réalisation du multijoueur est que les joueurs sont connectés les uns avec les autres et dès lors qu'un personnage bouge l'autre aussi. Ainsi nous avons réalisé un Script nommé PlayerSetup, qui sera implémenté sur chacun des joueurs. Le script précisera que les actions réalisées seront faites uniquement sur le joueur sélectionné. En d'autres termes nous préciserons les composants à désactiver pour le joueur.

Du côté de la synchronisation, nous avons choisis d'optimiser les packages envoyer au serveur en les limitant à 100 millisecondes entre chaque envoie. En effet, si nous baissions au maximum ce temps, le serveurs seraient surchargés et cela ferait l'effet inverse les packages seraient triés en file d'attente et le jeu serait saccadé.

## Changement de scène : (Mehdi)

Lors du passage sur l'interface représentant un bureau d'ordinateur, un changement de scène sera promulgué projetant le joueur présent dans la chambre d'une scène en 3 Dimension à une scène en 2 dimension. L'utilisation des scènes permet une scénarisation plus efficace et moins complexe en plus d'assurer une bonne gestion de la mémoire de l'ordinateur. En effet, lorsqu'on quitte une scène pour se diriger dans une nouvelle, la première scène est purgée de la mémoire. On note cependant que les variables statiques sont conservées en mémoire lors des changements de scène.

Tout d'abord il faut créer un script C#, qu'on va appeler par exemple ChangeScene.cs.

Première chose à faire c'est d'importer une classe que l'on va appeler par exemple SceneManager:  
`using UnityEngine.SceneManagement;`

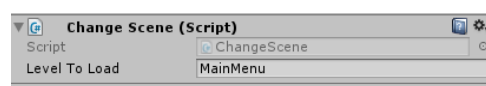
Après cela on commence à construire notre script en créant une variable :

```
public string LevelToLoad;
```

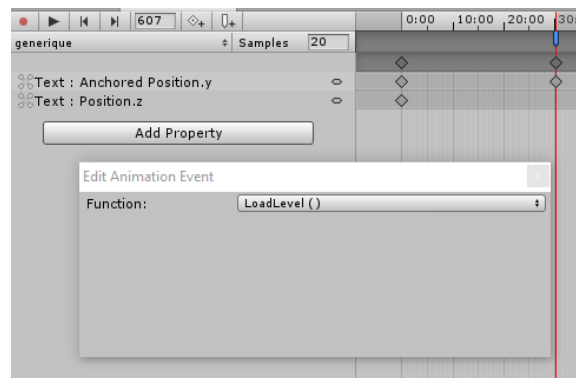
Ce qu'il reste à faire c'est d'écrire une fonction qui permet de charger le contenu de notre variable qu'on va insérer dans le GameObject. Pour cela on va utiliser la fonction LoadScene de la librairie SceneManager.

```
void LoadLevel()  
{  
    SceneManager.LoadScene(LevelToLoad);  
}
```

Une fois fait cela, il faudra écrire le nom de la scène à modifier dans le champ vide du script qu'on va attacher au GameObject, dans notre cas « MainMenu » :



Ce qu'il reste à faire c'est de déclencher l'événement dans l'animation en allant dans « Animation » par exemple :



Conclusion:

Les objectifs remplis pour la première soutenance :

Nou avons terminée la modélisation d'Epita.

Nos objectifs pour la prochaine soutenance:

Nous voulons avoir intégralement fini le développement des joueurs (en particulier le joueur 1, et si possible le joueur 2), avoir intégré l'IA dans le jeu (afin qu'elle puisse se déplacer sans problème, avec la vitesse adéquate), avancer voire finir les énigmes du script et enfin terminer le multijoueur.