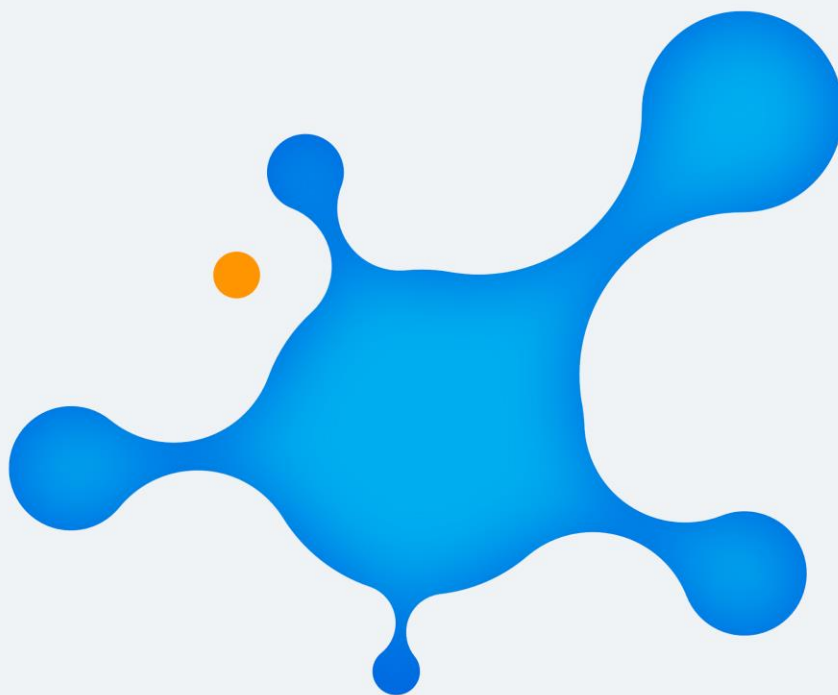


## Rapport d'activité

# RUDY MICHAU

Stage du 25 Mai au 25 Juin 2021



## RUDY MICHAU

créations web et graphiques

# Sommaire

<b>Rapport d'activité</b>	1
<b>Sommaire</b>	2
<b>Présentation de l'entreprise</b>	4
<b>Présentation du service</b>	5
<b>Présentation de l'environnement matériel et logiciel</b>	6
<b>Présentation du contexte de votre situation professionnelle et de la mission confié</b>	7
<b>Semaine 1</b>	8
Les tâches réalisés	8
La participation à des entretiens ou réunions	8
Les problèmes rencontrés	8
Les actions menées pour leur résolutions	9
<b>Semaine 2</b>	10
Les tâches réalisés	10
Les problèmes rencontrés	11
Mouvement du personnage - Retour visuel	11
Saut	12
Les actions menées pour leur résolutions	13
<b>Semaine 3</b>	14
Le saut	14
Optimisation de la gestion des collisions	16
Avancement du projet	19
<b>Semaine 4</b>	20
Correction des bugs	20
Pourquoi le personnage ne voit plus le sol à une certaine distance de la scène ?	20
Pourquoi le personnage fait une fois sur deux un saut statique puis un saut dynamique ?	20
Options sans caméra, caméra en 1er et 3eme personnes	21
Simulation du saut en fonction de la gravité	23
Ajout d'une nouvelle classe et bouton pour sélectionner le sol	24
Avancement du projet	25
<b>Semaine 5</b>	26

Site web pour la vente de l'add-on	26
Portfolio	27
Sassy CSS	29
Recréer facilement la structure de Bootstrap	31
One page, oui mais...	33
Formulaire de contact	34
<b>Ce que m'a apporté ce stage</b>	36
<b>Conclusion</b>	37
<b>Remerciement</b>	38

# Présentation de l'entreprise

Rudy MICHAU est une entreprise qui a été créée le premier février 2011 à Joué-Lès-Tours et qui aujourd'hui se situe à Sorigny. A ce jour, aucun salarié n'est dans l'entreprise car monsieur MICHAU travaille en freelance. Il est spécialisé dans la création web (sites internet et applications web) et l'infographie.

## Présentation du service

Je n'ai pas été accueilli dans un service particulier car M. MICHAU est un auto-entrepreneur.

Je suis en permanence avec mon tuteur de stage pour la mission qu'il m'a confiée.

# Présentation de l'environnement matériel et logiciel

M. MICHAU a un PC qui est adapté au développement de rendu 3D, un SLI de GTX 1070 pour une très bonne puissance de calcul.

Il a un NAS Synology avec 2 disques durs de 8 To en mirroring pour les sauvegardes.

Il a un switch 8 ports pour pouvoir connecter en filaire son routeur avec son NAS et son PC.

Son PC a 2 systèmes d'exploitation : Windows 10 et Ubuntu.

Il utilise Blender (un logiciel de 3D) pour son activité principale.

Atom comme éditeur de texte pour le développement.

Notion comme outil de gestion de projet

Discord et Twitter comme espace de communication des projets.

Pour ma part, au départ, j'ai travaillé sur mon PC portable personnel mais au final, je suis passé sur mon PC fixe par manque de puissance de calcul.

J'utilise le logiciel blender (2.93) pour faire la mission qui m'est demandée.

J'utilise Visual Studio Code comme IDE pour le développement en Python.

Mon tuteur m'autorise à utiliser Visual Studio Code plutôt qu'Atom car j'y suis déjà habitué et connais les raccourcis, qui ne sont pas forcément les mêmes sur Atom.

Enfin, j'utilise le logiciel Notion pour la gestion de projet.

# Présentation du contexte de votre situation professionnelle et de la mission confiée

La mission est de créer un addons pour le logiciel de rendu 3D Blender en utilisant son API.

L'API (Application Programming Interface) est une nouvelle notion encore pas vue en cours qui est un ensemble de classes et de fonctions qui permet de piloter le logiciel en question.

Cet addon à créer est un moteur de jeu dans un environnement en 3 dimensions non prédictible. Le but est de permettre d'automatiser un personnage qui marche devant, derrière, à gauche et à droite de lui tout en gérant les collisions, les pentes, ...

A chaque étape "importante" du projet, c'est mon tuteur qui teste le programme et le regarde pour voir s'il n'y a pas des optimisations à faire. De plus, pour les notions qui ne sont pas vues en cours et dont j'ai besoin, Mr MICHAU crée une partie de code et m'explique son fonctionnement.

Une fois le projet terminé, l'addon en question sera proposé à la vente pour le logiciel blender.

# Semaine 1

## Les tâches réalisées

Durant cette première semaine, la tâche essentielle était de comprendre comment utiliser l'API Blender.

La seconde tâche était de créer le cœur du moteur de jeu soit la gestion des collisions dans tous les sens (devant, derrière, gauche, droite et en bas) ainsi que la gestion des pentes.

## La participation à des entretiens ou réunions

Je ne participe pas aux entretiens car c'est confidentiel.

## Les problèmes rencontrés

La plus grande difficulté est de comprendre les fonctions de l'API Blender car cela demande énormément d'investissement en temps pour comprendre les fonctions pilotes du logiciel.

La seconde difficulté est de comprendre les vecteurs en 3 dimensions qui demande bien plus de compréhension d'un environnement local (Vecteur théorique) que d'un environnement global (Vecteur réel en utilisant les vecteurs théoriques).

La troisième difficulté est de comprendre comment on peut vérifier s'il y a une collision en utilisant les formules de trigonométrie (cosinus, sinus et tangente) et les adaptées en code.

Enfin des notions de mathématiques qui se voient en BAC+5 scientifique qui sont les matrices mondes, les matrices inversées ainsi que leurs multiplications.



## Les actions menées pour leur résolutions

Dans le premier cas, il faut encore que je m'adapte à l'utilisation de l'API Blender pour continuer dans ma lancer.

Dans le second cas, le fait de s'imaginer dans une pièce m'a suffit pour comprendre les vecteurs en 3 dimensions.

Dans le troisième, j'ai dû revoir mes cours de troisième pour palier aux problèmes.

Dans le dernier cas, c'est mon tuteur qui m'a donné la solution pour que je puisse continuer.

## Semaine 2

### Les tâches réalisées

Durant cette seconde semaine, j'ai commencé à développer plus de fonctionnalités pour que le futur utilisateur puisse utiliser l'addon sans difficulté de compréhension

Dans un premier temps, il a fallu définir automatiquement l'origine du personnage pour éviter une sélection manuelle.

Dans un second temps, il a fallu adapter la distance des collisions qui au départ était en valeur dure en valeurs variables en fonction de la hauteur, largeur et longueur du personnage.

Dans un troisième temps, ajouter une caméra automatiquement qui sera reliée sur le personnage.

Puis, je crée un personnage s'il n'y en a pas encore un sur la scène.

Je vérifie quand je sélectionne un objet, s'il n'existe pas de caméra parent, s'il en existe déjà une, je la supprime et j'en crée une nouvelle sinon je la crée.

Quand je crée mon personnage alors qu'il n'y en a aucun sur la scène, ses déplacements sont différents.

En appuyant sur la touche shift et z en même temps, cela fait courir votre personnage.

Enfin, je donne la possibilité de sauter à notre personnage.

## Les problèmes rencontrés

Cette semaine, l'API m'est légèrement plus familière et je commence à mieux comprendre comment il faut chercher les fonctions à utiliser, mais l'API Blender est tellement vaste qu'il me faudra bien plus de temps que le temps que je passe en stage pour comprendre l'ensemble de ses fonctions.

Pour la fonction qui permet de définir automatiquement l'origine d'un personnage, je n'ai pas eu de difficulté particulière à part pour placer les arguments à utiliser.

Pour adapter la distance de collision des éléments avec le personnage, la seule difficulté était d'adapter parfaitement le rayon qui permet de définir la collision.

Pour le placement de la caméra et sa potentielle suppression, le problème était lors de la création d'une caméra alors que l'autre a été supprimé. La position de la caméra dérivait sur l'axe X. Grâce à mon tuteur qui connaît bien l'utilisation de cette API, il m'a montré qu'il fallait forcer la scène à rafraîchir les coordonnées.

## Mouvement du personnage - Retour visuel

Nous avons souhaité augmenter le retour visuel lors des déplacements du personnage. Pour des questions de simplicité le personnage ne sera pas un humain mais le robot suivant



Il n'y aura donc pas de marche mais un mouvement de balancier dans la direction du déplacement. Exemple : pencher en avant lorsque l'on avance.

### Problème

La rotation du personnage impacte tous les angles et directions de détections préalablement établis. En effet, les directions suivent des vecteurs dont les coordonnées sont définies dans le repère du personnage. Si celui-ci tourne, les directions suivies pour les collisions aussi.

### Solutions envisagées

1. Modifier l'algorithme pour compenser les mouvements de balancier
  - a. Algorithmiquement compliqué.
  - b. Mathématiquement abordable après quelque rappel de trigonométrie, mais pas évident.

2. Modifier le contenu de l'objet "personnage" pour créer l'illusion du mouvement

C'est la solution 2 qui a été retenue. L'astuce a été de faire tourner le contenu au lieu du contenant. Ainsi, le personnage tourne visuellement bien que l'orientation de l'objet n'est pas changée.

## Saut

Le saut est toujours à l'étude, ce sera peut-être le plus dur du projet.

Le problème majeur est dû à l'environnement de Blender. Dans son fonctionnement, le rafraîchissement de l'écran ne se fait que si un événement est détecté (mouvement de souris, touche clavier...) Cela impose donc de maintenir la touche de saut tout le long du mouvement. Si on lâche en cours de route, le personnage reste dans les airs. Il est visuellement en l'air bien que ses coordonnées continuent d'évoluer. Il faut trouver un moyen de contourner cette limitation.

## Les actions menées pour leur résolutions

Dans le premier cas, il faut que je m'efforce à chercher dans l'API Blender si une commande ou une fonction existe déjà pour piloter le logiciel.

Dans le second cas, j'ai avisé en continuant le code et en ajoutant les arguments petit à petit en fonction des besoins pour l'utilisation de la fonction.

Dans le troisième cas, ajouter un pourcentage de distance supplémentaire au résultat pour que sa soit plus "réel"

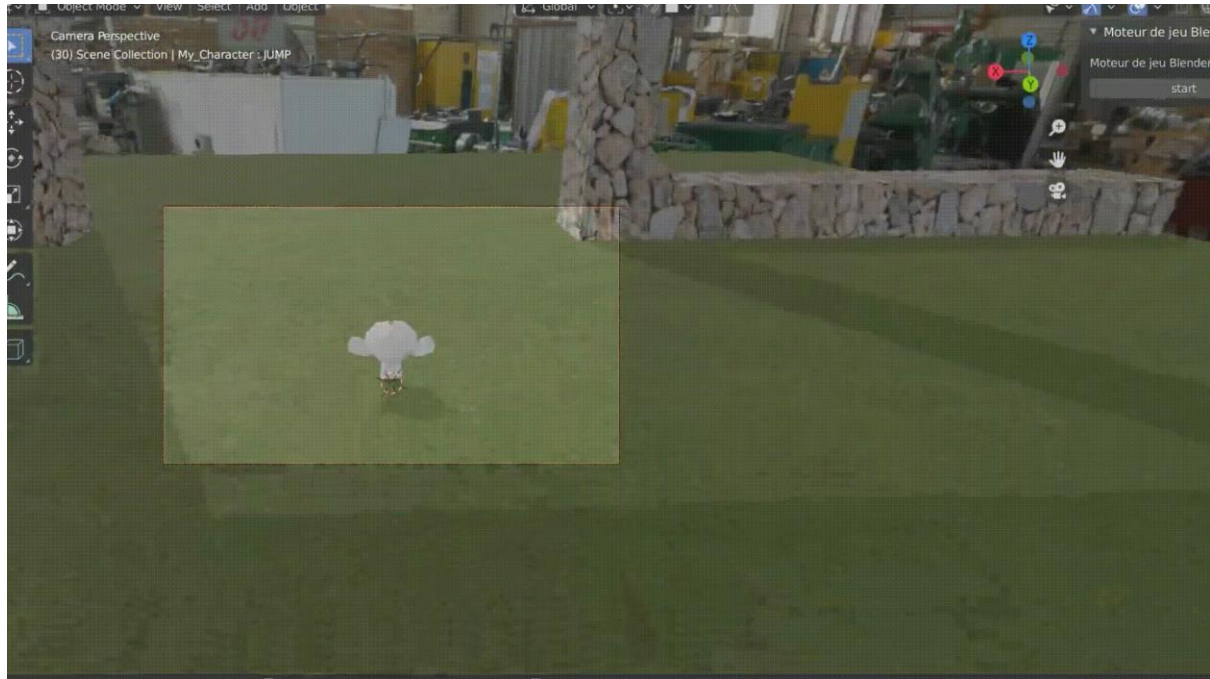
Dans le 4ème cas, c'est mon tuteur qui m'a aider a créer la fonction qui me permet de forcer le rafraichissement de la scène.

Dans l'avant-dernier cas, mon tuteur m'a donné un indice sur ce qui existait lors de l'utilisation d'un objet, et j'ai adapté mon code au cas qui se présentait.

Dans le dernier cas, je cherche encore la solution.

## Semaine 3

### Le saut



Durant la fin de semaine dernière, je me suis confronté au problème du saut du personnage .

En effet, Blender fonctionne quand un événement est détecté (mouvement de souris, touche du clavier, ...), ce qui imposait de maintenir la touche espace durant toute la durée du saut.

Durant le début de cette semaine, mon tuteur et moi avons trouvé la solution.

Dans l'API Blender, on a remarqué qu'il y avait la possibilité d'utiliser un événement TIMER qui tourne à la vitesse du processeur. J'ai donc pu brider cet événement pour qu'il tourne 60 fois par seconde (soit 60 FPS).

```
if (datetime.datetime.now() -  
self.previous_time).total_seconds() >= 0.02:  
    self.previous_time + datetime.timedelta(seconds=0.02)  
    self.deplace(context, event)
```

J'ai réutilisé la même technologie que la semaine dernière qui consiste à déplacer le contenant du personnage mais pas sa position. (Il est visuellement en l'air bien que ces coordonnées continuent d'évoluer). C'est le shape keys.

La seconde partie a consisté à pouvoir sauter et avancer en même temps. J'avais déjà l'essentiel.

Il fallait donc que je puisse connaître la dernière touche qui a été pressée et la touche pressée. Blender ne montre que les événements actifs. Il fallait donc trouver un moyen de sauvegarder l'ancienne valeur. J'ai donc fait un dictionnaire a deux valeur:

now qui est la touche qui est pressée

before qui est la touche qui a été pressé

Pour que cela fonctionne, à chaque tour de boucle, je remplace la valeur before par now puis now devient une nouvelle valeur.

```
if not event.type == 'TIMER':  
    if event.type != self.tab_event_type['BEFORE'] and not  
self.jump_state:  
        self.tab_event_type['BEFORE'] =  
self.tab_event_type['NOW']  
        self.tab_event_type['NOW'] = event.type
```

Pour pouvoir faire sauter le personnage, je dis que si la touche pressée est 'SPACE' et que la touche qui à été pressée est 'Z': je fais avancer mon personnage et utilise le shape keys.

Deux nouveaux problèmes sont apparus :

- Le saut crée un lag important.
- le personnage saute une fois sur deux en statique puis en avançant.  
(Problème de détection des touches), Ce problème n'est toujours pas résolu.

Dans ce cas, j'ai dû utiliser l'API Blender pour pouvoir avancer. Mon tuteur m'a aiguillé lors du développement de ce cas pour que je puisse avancer de façon plus rapide.

## Optimisation de la gestion des collisions



La gestion des collisions de base était plus bonne pour une scène qui s'avérait petite. En revanche, plus la scène était grosse et contenait beaucoup d'objets plus il y avait de calcul. De plus, lorsque le personnage sautait, d'importants ralentissements se produisaient. De fil en aiguille, mon tuteur et moi avons conclu que cela venait de la gestion des collisions.

Le but est de tester moins d'objets pour la collision en testant les objets les plus proches.

2 Solutions s'offraient à moi:

- rafraichir le tableau de vérification des collisions toutes les X secondes (Simple à mettre en oeuvre mais encore beaucoup de calcul)
- rafraichir le tableau à chaque fois que le personnage parcourt  $\frac{2}{3}$  d'un rayon donné (plus dur mais bien plus optimisé)

J'ai donc naturellement choisi la solution n°2.

J'ai commencé par créer une fonction sphère qui retourne un tableau contenant l'ensemble des objets dont la distance maximum entre le personnage, est l'objet le plus gros de la scène ÷ 2.

J'ai donc développé une fonction `i_m_th_biggest` qui permet de renvoyer l'objet le plus gros de la scène pour l'utiliser dans ma fonction sphere.



On imagine le tableau de la fonction sphere comme une sphère qui à un rayon dynamique. Tout ce qui est dans le périmètre de cette sphère est ajouté comme objet à tester.

Malheureusement, le fait de changer de rayon dynamiquement fait que lorsqu'on s'approche d'un objet trop gros, le rayon s'élargit trop et teste donc encore bien trop d'objets à la fois.

Mon tuteur s'est dit qu'il fallait faire un mixe de ces 2 fonctions mais inversé. Au lieu que la sphère parte de l'origine du personnage, il vaudrait mieux que la sphère parte des objets et qu'une fois qu'on s'en approche trop, on l'ajoute dans le tableau des objets à tester.

Repartie de 0, J'ai créé la fonction `search_near_objects` qui me permet de faire un calcul de la distance entre l'origine du personnage et l'origine des objets. Finalement, j'ai dû revoir des notions vues au collège pour que mon cerveau puisse repartir sur de bonnes bases avec Mr Pythagore (pas toujours facile).



En effet, le fait de calculer une distance en hauteur (donc coordonnées Z) puis faire sa différence pour avoir la dernière distance est le théorème de Pythagore.

J'ai fait une fonction distance entre le personnage et l'objet qui permet de calculer la distance de celui-ci

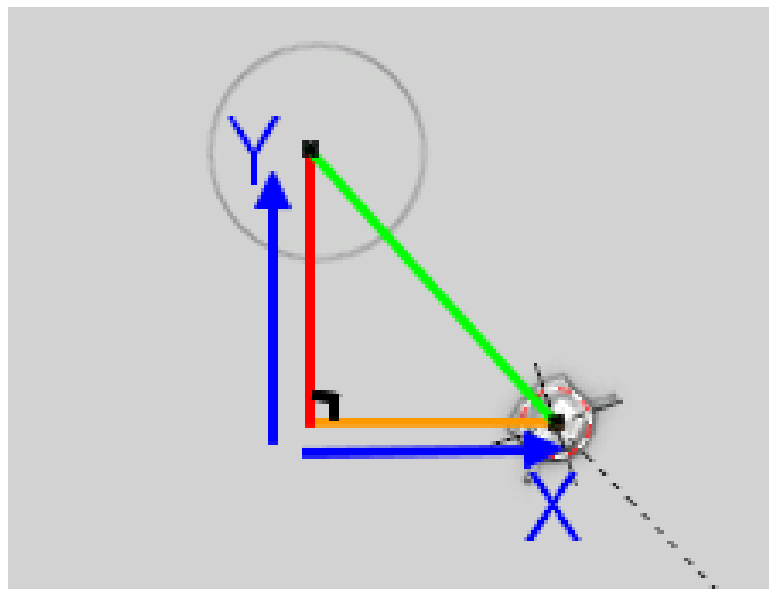
(trait vert sur le triangle rectangle)

```
def distance(perso, objects=None,
location = Vector((0, 0, 0))):
d = math.sqrt(((perso.location[0] -
objects.location[0]) ** 2) +
((perso.location[1] -
objects.location[1]) ** 2) +
((perso.location[2] -
objects.location[2]) ** 2))
return d
```

Dans notre fonction `search_near_object`, on calcule donc la distance de AB grâce à notre fonction `distance`. Puis on calcul BC grâce au théorème de Pythagore pour calculer la distance avant la collision d'un objet.

```
def search_near_objects(perso):  
    for obj in bpy.context.visible_objects:  
        ab = perso.location[2] - obj.location[2]  
        ca = distance(perso, obj)  
        bc = math.sqrt(ba**2 - bc**2)
```

L'optimisation de la gestion des collisions fonctionne enfin. Mais durant ce Vendredi après-midi, mon tuteur a vu que l'on pouvait supprimer la fonction distance en ne calculant pas les coordonnées Z mais par les coordonnées X et Y. Ce qui permet d'éliminer un calcul; donc encore plus d'optimisation.



```
def search_near_objects(perso):  
    for obj in bpy.context.visible_objects:  
        bc = perso.location[0] - obj.location[0]  
        ba = perso.location[1] - obj.location[1]  
        ca = math.sqrt(ba**2 + bc**2)
```

Aucun problème ne s'est pour l'instant révélé avec cette nouvelle gestion des collisions

## Avancement du projet

Le projet à été lancé sous la demande d'un client Américain réalisant des environnement virtuelle pour les jeux vidéos. Le programme lui servira alors pour le bon dimensionnement des scènes.

Il n'y avait pas d'objectif précis durant chaque semaine du stage. En revanche, le projet doit être terminé ou sur le point d'être fini lors de la fin du stage.

Etant donné que le cœur du moteur de jeu est déjà fait et que les fonctionnalités sont faites. Il ne reste plus qu'à corriger les bugs, optimiser ce qui est optimisable et voir s'il y a besoin de rajouter d'autres fonctionnalités.

Un fois l'add-on terminé, elle passera en bêta test puis sera en vente libre pour les utilisateurs de Blender

## Semaine 4

### Correction des bugs

#### Pourquoi le personnage ne voit plus le sol à une certaine distance de la scène ?

En fin de semaine dernière, lors de l'optimisation de la gestion des collisions j'ai relevé un bug que je n'avais pas réussi à résoudre.

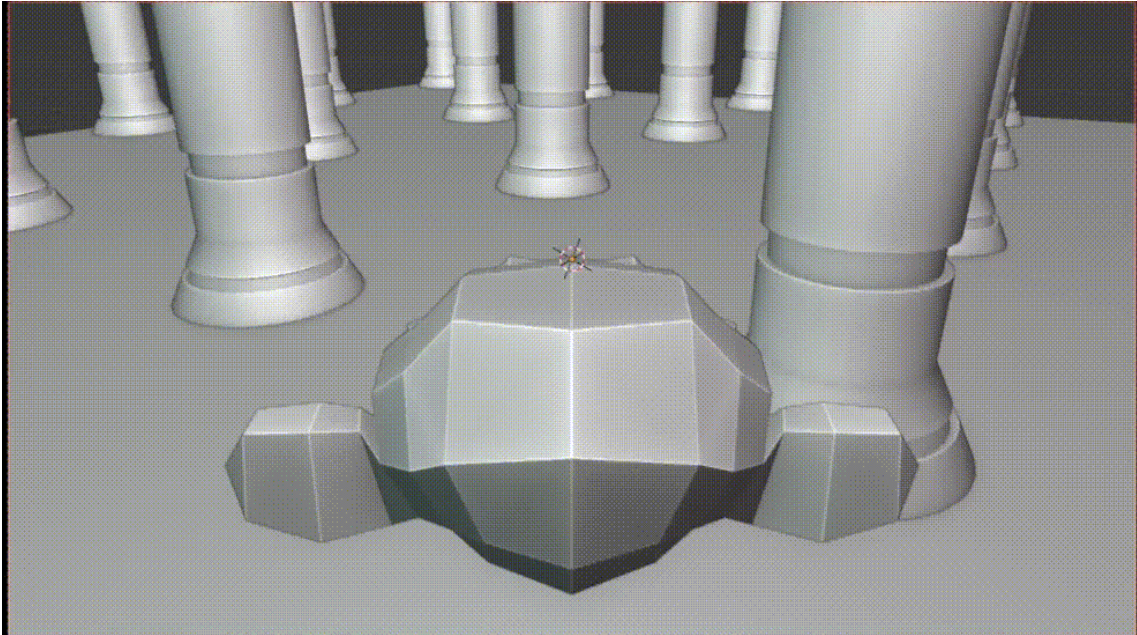
A force de recherche, j'ai remarqué que j'avais oublié d'ajouter obligatoirement le sol dans le tableau de tests des collisions.

#### Pourquoi le personnage fait une fois sur deux un saut statique puis un saut dynamique ?

En fin de semaine dernière, j'ai remarqué lorsque j'ai fait le saut pour le personnage importé qu'il faisait une fois sur deux un saut correct alors que les touches pressées étaient les bonnes. Après moult recherche dans le code source, j'ai finalement compris que cela venait de la condition d'entrée dans le dictionnaire pour la touche pressée et la touche qui a été pressée.

Tous les bugs ont finalement été résolus !

## Options sans caméra, caméra en 1er et 3eme personnes



Pour le confort de l'utilisateur, je lui propose de choisir la vue qu'il souhaite pour correspondre à l'ensemble des situations auquel on peut se confronter.

Pour l'option caméra en 1er et 3ème personnes, j'ai juste utilisé la fonction camera créée la semaine dernière en ajoutant un argument vecteur qui est un vecteur qui permet de placer la caméra selon celui-ci .

J'ai relevé le nom des événements molettes avant et arrière de la souris pour faire cette option.

Pour ce faire, j'avais juste à modifier le vecteur de fixation de la caméra. De plus, j'ai créé une variable view\_style qui me permet de ne pas recréer une caméra inutilement (si on est en première personne et que je veux aller en première personne).

```
if not self.event_shift:
    if event.type == 'WHEELUPMOUSE' and self.view_style == "TROISIEME":
        self.camera = camera(self.perso, Vector((0, -
self.perso.dimensions[1] / 4, 0)), 90)
        self.view_style = "PREMIERE"
    elif event.type == 'WHEELDOWNMOUSE' and self.view_style ==
"PREMIERE":
        self.camera = camera(self.perso)
        self.view_style = "TROISIEME"
```

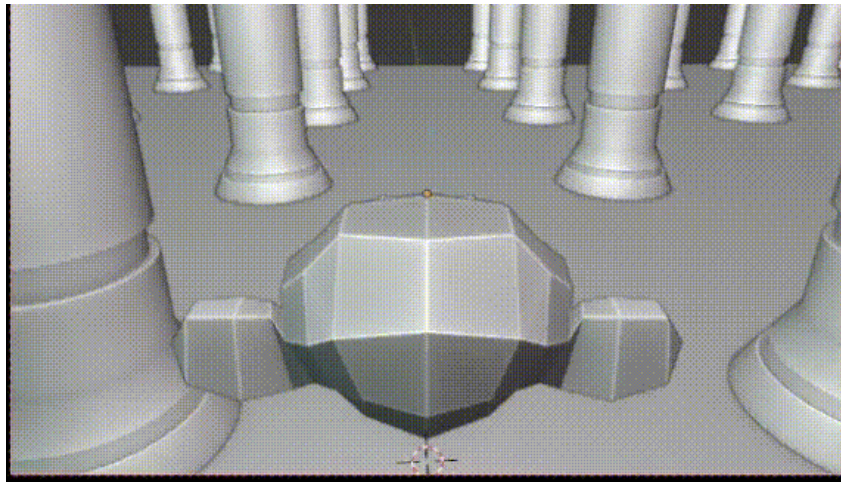
Dans le cas où je ne veux pas de caméra, je précise à l'utilisateur d'appuyer sur la touche shift avant d'appuyer sur le bouton qui exécute la classe.

```
class Engine(bpy.types.Operator):  
    """EN : To start without a camera \n  hold SHIFT then click start  
\n\nFR : Pour démarrer sans la caméra \n  maintenez SHIFT puis cliquez  
sur start"""
```

```
if not event.shift:  
    self.camera = camera(self.perso)  
    self.event_shift = None  
elif event.shift:  
    self.event_shift = event.shift
```



## Simulation du saut en fonction de la gravité



La semaine dernière, j'ai fait sauter mon personnage importé en fonction de ma boucle de 60 FPS pour faire évoluer ses `shape_keys` selon une valeur fixe, ce qui donnait un air pas naturel au saut de mon personnage. C'est pourquoi je me suis attelé à faire évoluer mon saut en fonction de la formule de la gravité qui est en fait une parabole en mathématique.

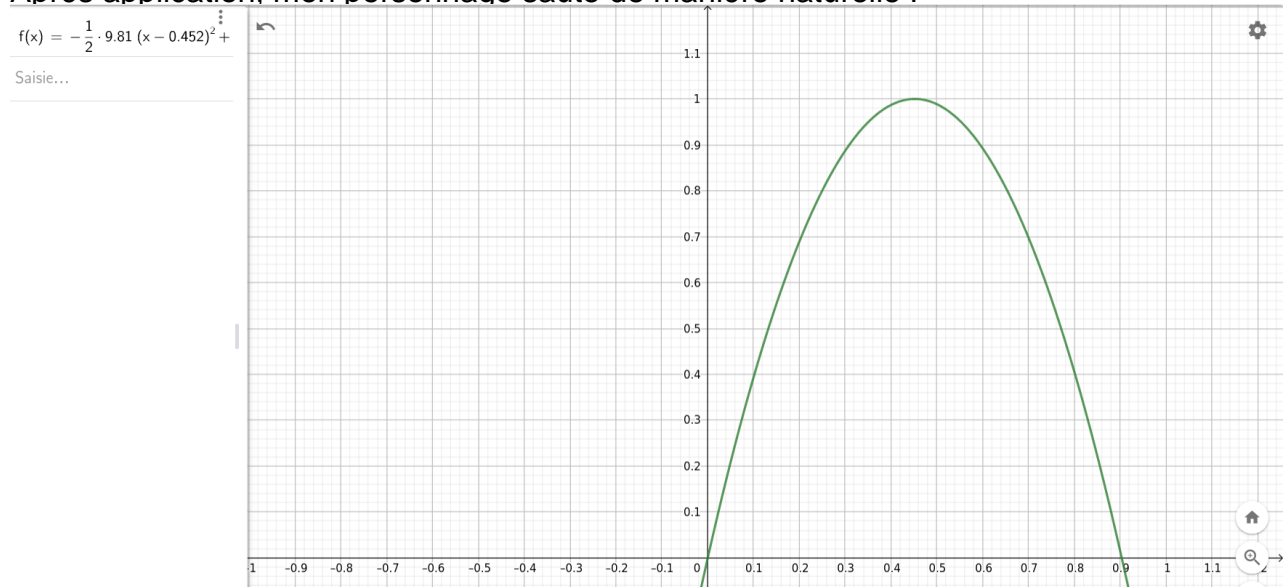
$$f(x) = -\frac{1}{2} * 9.81 * t^2 + 1$$

J'ai donc dû trouver la formule de la gravité que j'ai appliquée mais le saut de mon personnage bugger en s'exécutant de manière aléatoire. Mon tuteur a remarqué qu'il fallait déplacer la courbe de façon à ce qu'elle parte du point 0. Il a donc calculé la différence et m'a donné la nouvelle formule.

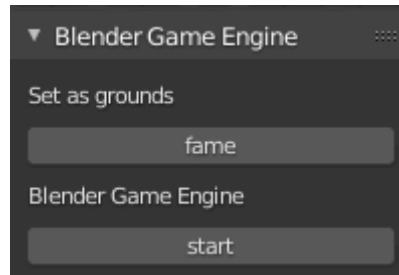
$$f(x) = -\frac{1}{2} * 9.81 * (t - 0.452)^2 + 1$$

```
def gravity(temps):  
    parabole = (-1/2) * 9.81 * (temps - 0.452)**2 + 1  
    return parabole
```

Après application, mon personnage saute de manière naturelle !



## Ajout d'une nouvelle classe et bouton pour sélectionner le sol



Une dernière étape est nécessaire avant de terminer ce projet. Faire un bouton qui permet d'ajouter le nom 'ground' pour ensuite l'ajouter au tableau des objets à tester pour les collisions. En effet, l'utilisateur ne va pas forcément regarder le code source pour comprendre son fonctionnement et comprendre comment s'utilise l'add-on. Il faut donc (même les plus petits détails), que ce soit pensé.

J'ai donc créé une classe Ground qui ajoute le nom "\_ground" à la fin de son nom pour la bonne exécution de la seconde classe.

```
class Ground(bpy.types.Operator):
    """EN : Set as grounds \n Press this button \n\n FR : Définir
comme sols \n Appuyez sur ce bouton"""
    bl_idname = "wm.ground"
    bl_label = "Simple View Operator"

    offset: FloatVectorProperty(
        name="Offset",
        size=3,
    )

    def invoke(self, context, event):
        """EN : Renames selected objects as ground \n FR : Renomme les
objets sélectionnés en tant que sol"""
        for obj in bpy.context.selected_objects:
            obj.name += '_ground'
        return {'FINISHED'}
```



## Avancement du projet

Le projet est fini !

L'add-on sera proposé dans peu de temps à des bêta testeurs qui rapportent les éventuels bugs avant de proposer celui-ci à la vente.

Le nouveau projet de la semaine prochaine sera de créer une page web qui mettra en valeur l'add-on.

## Semaine 5

### Site web pour la vente de l'add-on

Lors de la fin de la semaine dernière, le projet était de créer un site vitrine pour l'add-on qui a été créé.

Malheureusement, mon tuteur n'a pas trouvé l'inspiration pour créer ce site et donc il m'a demandé de faire mon portfolio.

## Portfolio

Dans un premier temps, j'ai montré le site que j'ai commencé à faire chez moi.

Mon tuteur m'a fait part de quelques remarques :

- Je n'ai pas de maquette pour construire le style du site
- J'utilise un framework CSS qui est castrateur au niveau du style et de la mise en place

Pour commencer, il faut absolument construire une maquette pour réfléchir à comment notre site sera designé. Une fois la maquette faite, on peut imaginer la structure html du document et développer le site sans avoir à réfléchir au design.

Dans un second temps, Bootstrap s'est montré lourd pour le peu qu'il y avait à faire. C'était l'occasion de faire progresser mes connaissances en HTML / CSS, de me rendre plus indépendant et plus libre dans mes créations.



J'ai donc commencé par faire une maquette pour mon portfolio à l'aide du logiciel Photoshop.

Une fois la maquette terminée, on passe au développement

## Sassy CSS

Mon tuteur m'a montrer une autre façon de faire du CSS :

Le SCSS est simplement du CSS imbriqué qui permet d'éviter les répétitions.

L'avantage de celui-ci est que l'on peut déclarer des variables et les utiliser dans le code. De plus, ce code est plus facile à maintenir en cas de changements.

Son désavantage est que l'on doit utiliser un compilateur pour le passer en CSS car le html ne supporte pas le SCSS.

J'ai donc installé et utilisé le logiciel Scout-app qui est un compilateur gratuit de SCSS en CSS.

<https://scout-app.io/>



exemple de code CSS:

```
.reverse {  
  background-color: #FFC107;  
  border-radius: 50px;  
  box-shadow: 15px 10px 10px #26212D;  
}  
  
.reverse h2, .reverse span {  
  color: #343A40;  
  padding: 5px;  
}  
  
.reverse div p {  
  color: #343A40;  
}
```

Exemple du même code en SCSS:

```
$jaune : #FFC107;  
$gris : #343A40;  
.reverse {  
  background-color: $jaune;  
  border-radius: 50px;  
  box-shadow: 15px 10px 10px #26212D;  
  h2, span {  
    color: $gris;  
    padding: 5px;  
  }  
  div {  
    p {  
      color: $gris;  
    }  
  }  
}
```

## Recréer facilement la structure de Bootstrap

Mon tuteur a plus d'un tour dans son sac quand il s'agit de créer des sites web. Il crée des classes qui permettent de créer une structure parfaite pour développer un site web (comme Bootstrap) mais dont le fonctionnement est compréhensible par tout le monde !

Chaque ligne est une classe row qui contient des div w-x (x allant de 1 à 12). La somme de tous les x doit être égale à 12 (comme les col dans Bootstrap).

HTML :

```
<div class="row">
  <div class="w-4">
    
  </div>
  <div class="w-8">
    <h3>Présentation</h3>
    <p>Lorem Ipsum</p>
  </div>
</div>
```

CSS :

```
.row {
  width: 75%;
  margin: 128px auto;
}

.row div {
  display: inline-block;
  vertical-align: top;
}
```

SCSS :

```
.row {
  width: 75%;
  margin: 128px auto;
  div {
    display: inline-block;
    vertical-align: top;
  }
}
```

### Commun CSS SCSS :

```
.w-1 { width : 7.3%; }  
.w-2 { width : 15.6%; }  
.w-3 { width : 24%; }  
.w-4 { width : 32.3%; }  
.w-5 { width : 40.6%; }  
.w-6 { width : 49%; }  
.w-7 { width : 57.3%; }  
.w-8 { width : 65.6%; }  
.w-9 { width : 74%; }  
.w-10 { width : 82.3%; }  
.w-11 { width : 90.6%; }  
.w-12 { width : 99%; }
```



## One page, oui mais...

J'ai décidé de faire mon portfolio en One page / une page. Cela me permet d'être plus rapide pour le développement du site, mais cela donne plusieurs inconvénients:

- Un site trop long est très vite fatigant. Il faut donc une page qui soit assez contenu
- A cause d'une page contenu, les informations doivent être succinctes.

C'est pour cela que mon tuteur m'a proposé de faire des overlays et des accordéons pour rendre plus de données accessibles tout en gardant une page contenu en longueur.

J'ai donc créé un script JavaScript qui me permet de changer des propriétés CSS en cliquant sur certains endroits de la page.

```
$("#open_overlay1").click(function() {  
    $("#overlay1").css({  
        "visibility" : "visible"  
    });  
});  
$("#close_overlay1").click(function() {  
    $("#overlay1").css({  
        "visibility" : "hidden"  
    });  
});
```

## Formulaire de contact

Je compte garder mon portfolio un moment pour que cela fasse office de CV numérique pour trouver du travail à la sortie du BTS.

C'est pour cela que j'ai fait un formulaire de contact. Au cas où une entreprise veuille me contacter.

J'ai d'abord pensé à utiliser une méthode "barbare" qui est de transformer ma page html en php pour directement récupérer les données puis m'envoyer un mail.

Après réflexion, j'ai voulu séparer les données en utilisant de l'Ajax qui a été vu en cours. Mon tuteur m'a répondu que c'est une manière plus propre et ordonnée de traiter les données.

Je suis donc partie sur une transmission des données par l'Ajax pour ensuite les transmettre au PHP pour envoyer un mail.

Ajax :

```
$(document).ready(function() {
    $("#submit").click(function(e) {
        e.preventDefault();
        $(".message").html('<div class="message">Envoi en
cours...</div>');
        $("#spin").html('<div class="loader"></div>');
        $.post(
            'contact.php',
            {
                civilite : $("#civilite").val(),
                nom : $("#nom").val(),
                prenom : $("#prenom").val(),
                tel : $("#tel").val(),
                mail : $("#mail").val(),
                sujet : $("#sujet").val(),
                message : $("#message").val()
            },
            function(data) {
                if(data == 'success') {
                    $("#resultat").html("<div>Mail envoyé</div>");
                }
                else {
```

```
        $("#resultat").html("<div>Echec de l'envoi du  
mail</div>");  
    }  
}  
);  
});  
});
```

PHP :

```
<?php  
    $civilite = $_POST['civilite'];  
    if($civilite == 'on'){  
        $civilite = "Mr";  
    }  
    else {  
        $civilite = "Mme";  
    }  
    $nom = $_POST['nom'];  
    $prenom = $_POST['prenom'];  
    $mail = $_POST['mail'];  
    $tel = $_POST['tel'];  
    $mail = $_POST['mail'];  
    $sujet = $_POST['sujet'];  
    $message = $_POST['message'];  
  
    $msg = "De $civilite $nom $prenom" . "\r\n" . "$message" . "\r\n" .  
"Contact : $tel";  
  
    if(mail($mail, $sujet, $msg)) {  
        echo "success";  
    }  
    else {  
        echo "failed";  
    }  
?>
```

## Les recherches effectuées

J'ai dû utiliser les documentations de l'API blender et des exemples de scripts sur internet pour pouvoir comprendre comment je pouvais utiliser les fonctions de l'API.

## Ce que m'a apporté ce stage

Ce stage m'a beaucoup appris dans plusieurs domaines de programmation.

D'une part, j'ai appris à correctement déboguer un site / logiciel en partant de l'erreur relevée par la console, qui ensuite peut révéler que cela vient d'une fonction que l'on a mal fait juste avant. De plus, j'ai même dû faire exprès d'ajouter une erreur dans le programme pour trouver d'où venait le bug.

D'autre part, développer en utilisant une API si grande, cela demande énormément d'investissement pour comprendre et savoir l'utiliser.

Pour la partie web, j'ai beaucoup appris pour réfléchir au site en faisant une maquette, et surtout, la partie style qui n'a pas été vu en cours.

Il ne faut pas qu'un site soit uniquement fonctionnel car aujourd'hui, c'est normal. Ce qui fait la différence entre un site et un autre pareil techniquement aujourd'hui, c'est le style. Aujourd'hui, un site web se doit d'être BEAU et FONCTIONNEL.

## Conclusion

Pour conclure, je dois dire que ce stage m'a énormément plus, en termes de connaissance acquise. Au niveau relationnel, tout allait bien étant donné que je connais Rudy depuis plusieurs années.

J'ai remarqué durant ce stage qu'en cours, l'on approfondit pas certaines notions qui sont pourtant essentielles comme le débogage, les maquettes, le style, ...

Je reste aujourd'hui dubitatif quant au choix de mon orientation future. En effet, dans le développement, j'aime me torturer l'esprit, mais j'ai une petite préférence pour le web.

Ce stage sème aujourd'hui un doute :

Je ne sais pas aujourd'hui si je veux me diriger dans les applications web ou dans les logiciels.

# Remerciement

Je remercie sincèrement Mr MICHAU Rudy de m'avoir accueilli en stage pour ces 5 semaines et ainsi pouvoir m'apporter son savoir et avoir de nouvelles connaissances et compétences en développement.