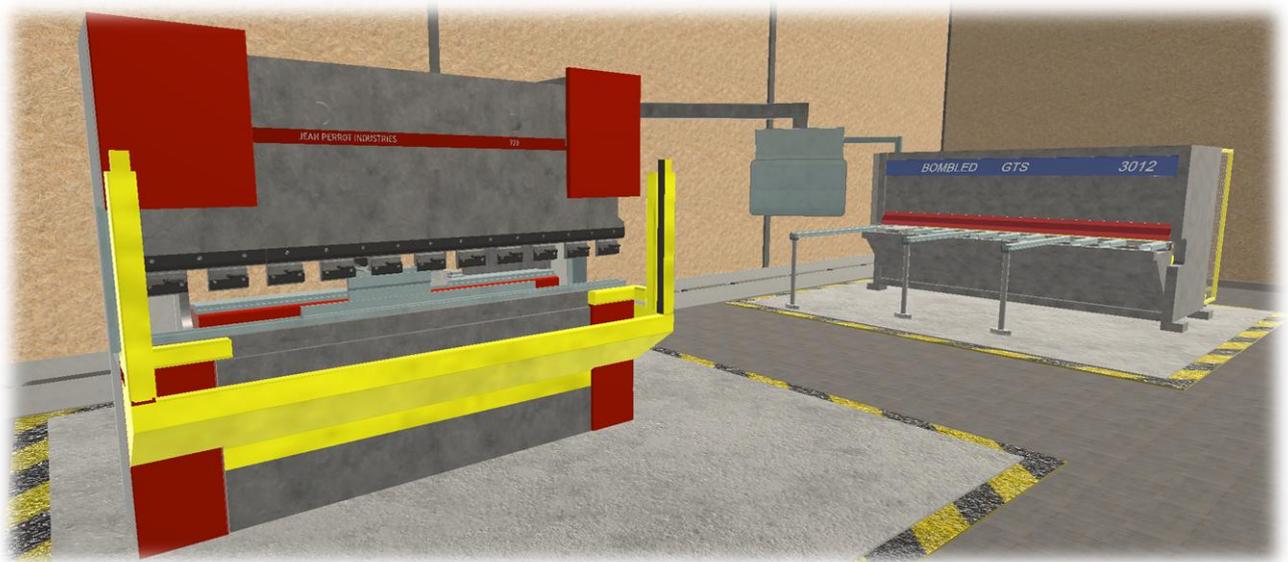


Création d'un jeu de rôle avec



Exploitation et modification de l'asset Inventory master



pierre.chauvin@ac-orleans-tours.fr

Lycée des métiers Jean Lurçat

41600 Fleury les Aubrais

Préambule :

Les vidéos disponibles sur le site d'Eduscol ont été réalisées sans montage. Cela signifie qu'elles ont été tournées en une seule prise, ce qui réduit le risque de surprises et d'éléments qui apparaissent et disparaissent entre des rushes différents.

Sommaire

1- Présentation.....	3
2- Les assets.....	4
3- Prise en main.....	5
4- Création d'items.....	5
5- EPI.....	6
6- Blueprint.....	7
7- Création de poste de travail.....	8
8- Gagner/perdre de l'expérience.....	10
9- Insérer des personnages non jouables.....	11
10- Création des vendeurs.....	12
11- Les lieux de stockage.....	13
12- Création d'une mission.....	13
13- Conclusion.....	15

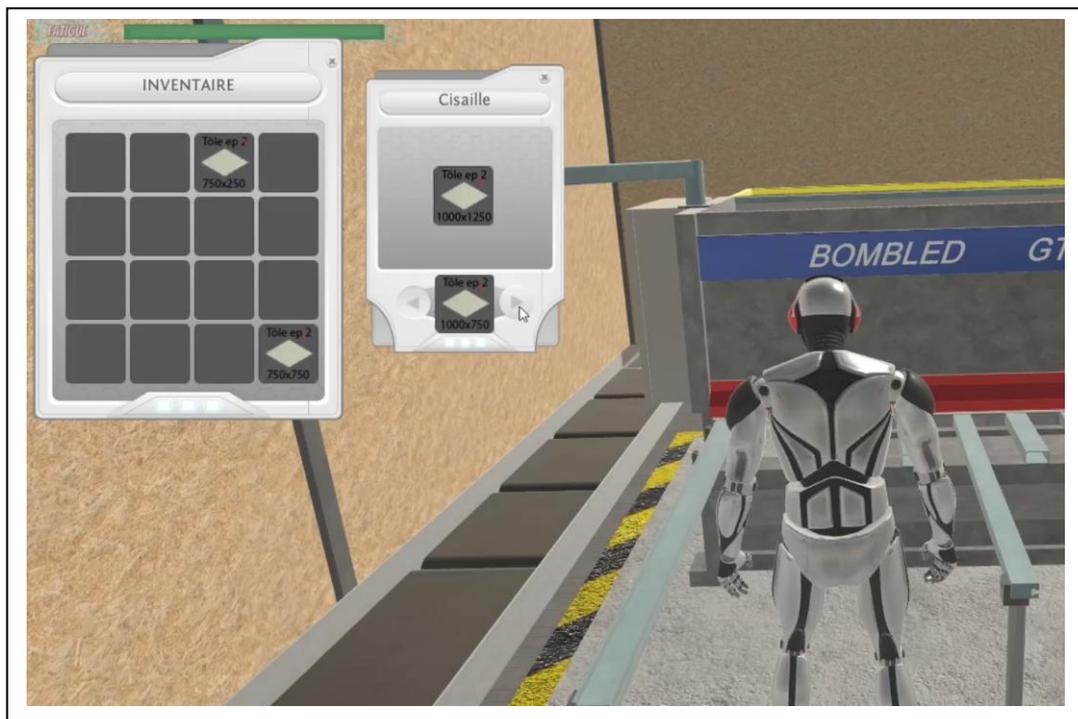
1- Avant-propos et présentation.

L'évaluation des compétences au travers du jeu de rôle est un projet développé cette année à destination d'élèves scolarisés en filière Chaudronnerie. Je ne peux malheureusement le partager dans son intégralité, certains « assets » n'étant pas libres de droits. Cependant, les éléments contenus dans la ressource - Vidéo et fichiers de travail - permettent de se rendre compte du résultat que l'on peut atteindre sans être spécialiste en développement informatique.

Ce format d'évaluation, que j'ai pratiqué avec mes élèves de Chaudronnerie, m'a permis un retour très positif de leur part. En plus de mettre en situation l'élève, elle permet de valider les compétences visées et de développer des facultés d'abstraction nécessaires à la bonne pratique de leur métier. Exemple : positionner mentalement sa tôle dans la cisaille afin d'optimiser la découpe (voir photo ci-dessous).

En développant différentes missions, je me suis rendu compte de la nécessité de découper de manière logique la séance et de scénariser les activités de l'élève. Chaque mission doit être remise dans un contexte, ce qui m'a imposé de faire un travail que je ne faisais pas forcément dans mes cours plus classiques. La présence de Michel et cette personification du client ont eu un impact très positif sur la motivation de mes élèves à réussir à rendre leur client satisfait.

Dans les années à venir, je pense impliquer mes élèves dans la réalisation des scénarios, il me semble intéressant que les élèves imaginent et produisent eux-mêmes les différentes situations qu'ils seront amenés à vivre dans leur future vie professionnelle.



2- Les assets

Les assets gratuits :

2.1- Inventory master :

<https://assetstore.unity.com/packages/tools/gui/inventory-master-ugui-26310>

Asset permettant d'avoir un système d'inventaire et de craft.

2.2- Invector :

<https://assetstore.unity.com/packages/tools/game-toolkits/third-person-controller-basic-locomotion-free-82048>

Asset permettant d'avoir un personnage à la 3^{ème} personne avec animation et déplacement.

2.3- Quick Outline :

<https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488>

Asset permettant d'avoir une ligne colorée sur la silhouette des modèles numériques.

2.4- Standard Asset :

<https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-for-unity-2018-4-32351>

Asset donnant accès à beaucoup d'éléments de base (Textures, scripts, modèles numériques, etc....)

Les assets payants :

2.5- Snap and plug :

<https://assetstore.unity.com/packages/tools/modeling/snap-and-plug-21930>

Asset permettant l'aimantation des GameObjects.

2.6- Dialogue System

<https://assetstore.unity.com/packages/tools/ai/dialogue-system-for-unity-11672>

Asset permettant des phases de dialogue entre l'élève et les personnages non jouables.

2.7- RT Voice :

<https://assetstore.unity.com/packages/tools/audio/rt-voice-pro-41068>

Asset permettant d'exploiter le narrateur « Hortense » de Windows. (Fonctionne très facilement avec Dialogue system)

2-8- HUD Navigation System

<https://assetstore.unity.com/packages/tools/gui/hud-navigation-system-103056>

Asset permettant d'avoir accès à une boussole, minicarte, repères spatiaux, etc. (Très facile d'exploitation)

3- Prise en main

Commencer par installer [Unity 2019.4.13](https://www.unity.com/2019/4/13/unity-2019-4-13) et configurer Visual Studio comment dans les tutoriels disponibles ci-dessous :

Installer Unity :

<https://www.youtube.com/watch?v=sKUST4YQkIk>

Installer et configurer Visual Studio :

<https://www.youtube.com/watch?v=hWiL3QdKDiw&t=118s>

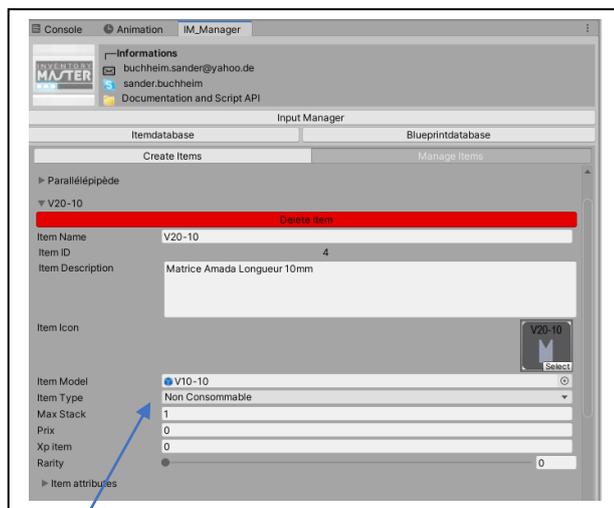
Puis télécharger les deux projets disponibles sur le site d'Eduscol. Le premier projet appelé « Rpg Eduscol » contient le projet à l'état Initial. Le second projet « Rpg Eduscol Final » est le projet dans l'état de la dernière vidéo du tutoriel. Vous pouvez vous en servir pour vérifier l'évolution de votre travail. Je vous conseille de partir du dossier initial et de suivre les différents tutoriels dans l'ordre.

4- Création d'items

Pour créer de nouveaux items, deux éléments sont indispensables :

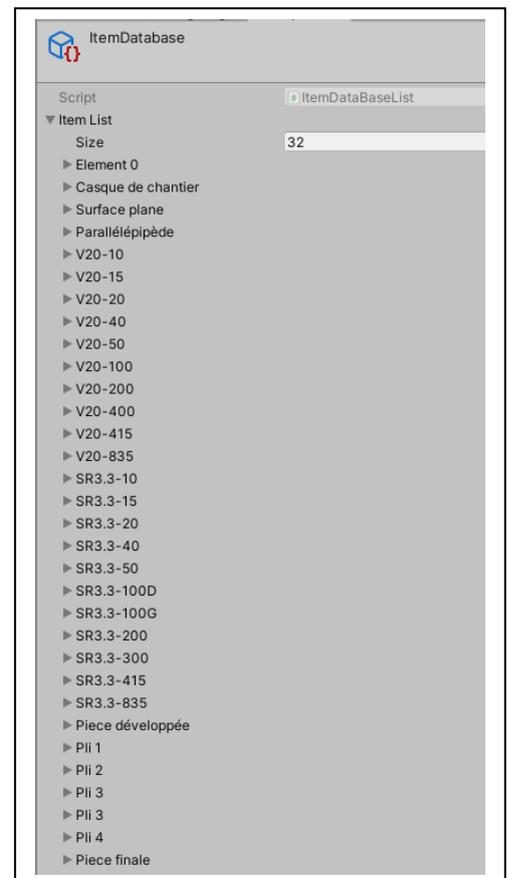
- Le modèle numérique en .fbx ou .Obj
- Une image au format d'un icône. Partir de l'icône d'origine nommé « Slot », disponible dans GriusInventory/Textures/ExampleInventory/Slot. Vous pouvez le retravailler avec un logiciel de retouche de photo (PhotoShop ou GIMP)

R
e
m
a
r
q
u
e
:
l
e
t
y
p
e

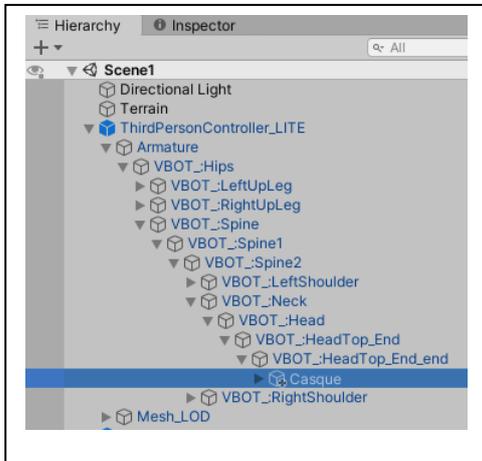


'item « non consommable » permet de ne pas faire disparaître l'item lors des phases de « craft »

La liste complète, ci-contre, de vos items sont disponibles dans la base de données située dans GriusInventory/Ressources/Itemdatabase.



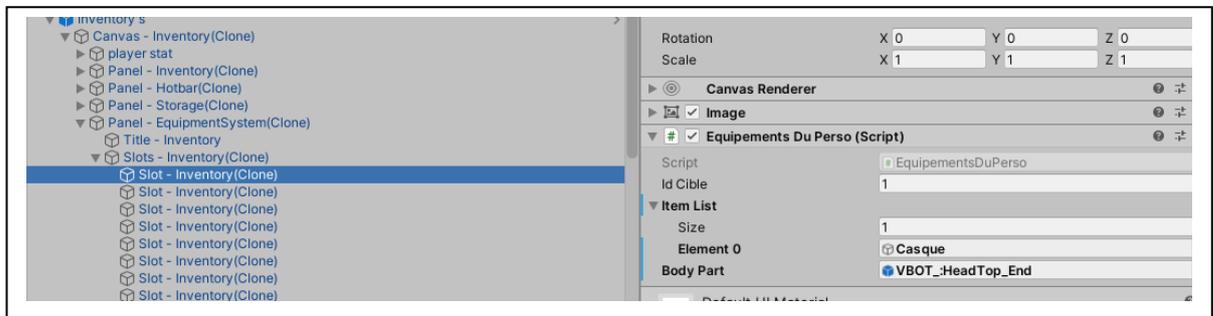
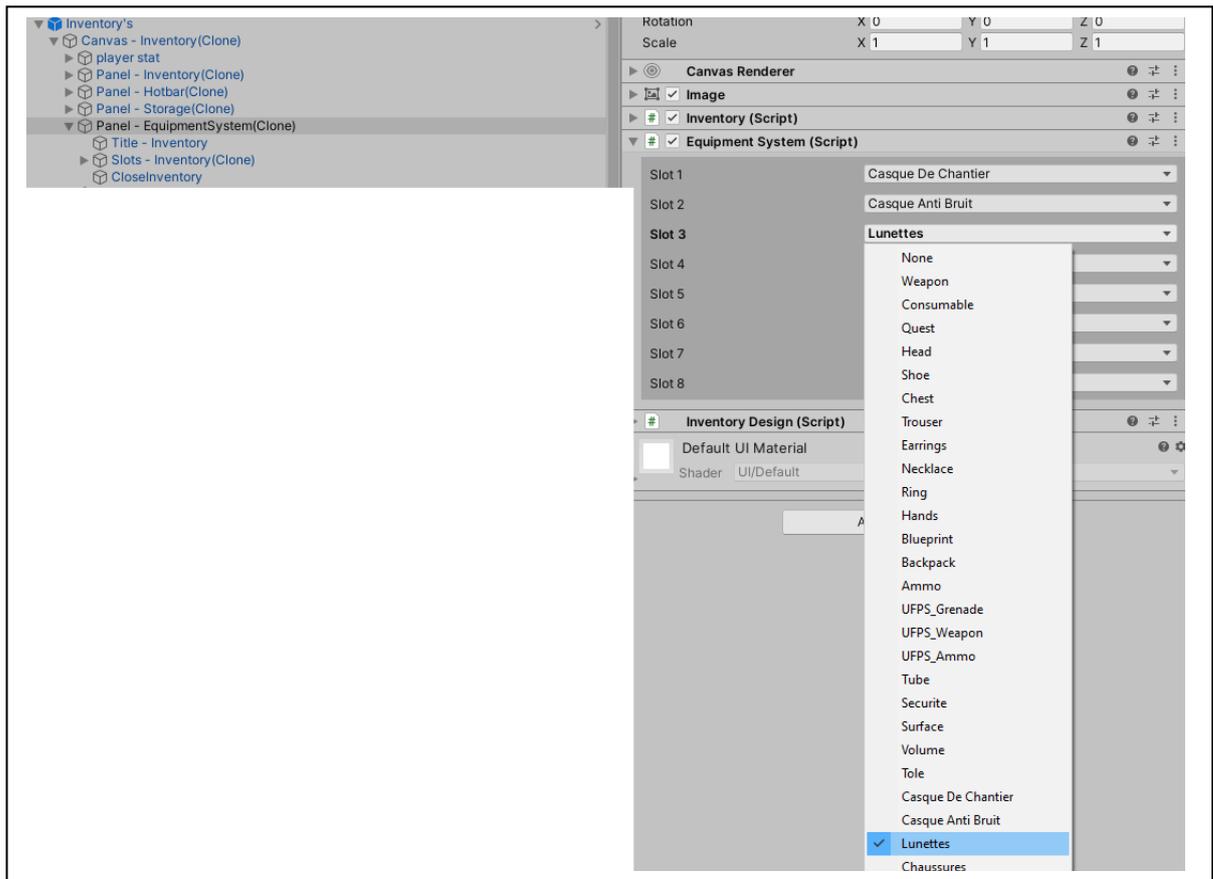
5- EPI



- Les Epi doivent être placés dans l'armature du personnage (Photo ci-contre) et doivent être désactivés.

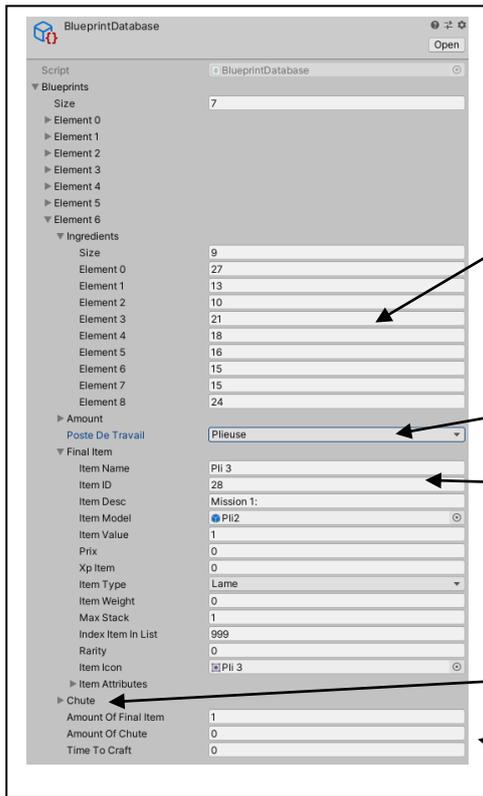
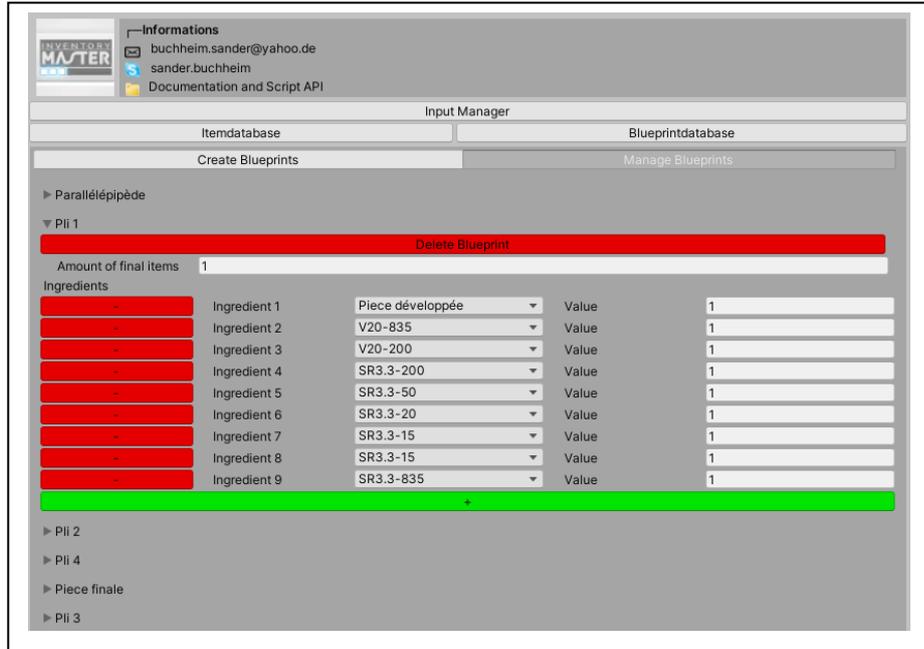
- Dans panel-EquipementSystem vous devez affecter le bon Epi au slot désiré comme dans la capture d'écran ci-dessous.

- Dans le slot correspondant, placer le script « Equipement du perso » et configurer le (Voir dernière capture d'écran ci-dessous)).



6- Blueprint

Les Blueprints sont les recettes permettant l'obtention de nouveaux items, ils sont définis dans une base de données nommée « Blueprintdatabase » et disponible dans :
6riusInventory/Ressources/Blueprintdatabase



Liste des Id des items de la recette

Poste de travail où sera autorisé le « craft »

Item final

Chute éventuelle, si l'élément est 0 il n'y aura pas de chute.

Quantité d'items/chute finale

7- Création de poste de travail

Ouvrir le script « WorkBenchType » puis rajouter une virgule au dernier élément puis rajouter votre nouveau poste de travail et ne pas terminer par une virgule, comme dans l'exemple ci-contre.

Dans le script « PlayerInventory », rajouter les variables correspondant à votre nouveau poste de travail

```

WorkBenchType.cs  Consumeltem.cs  PlayerInventory.cs
Assembly-CSharp
1  using UnityEngine;
2  using System.Collections;
3
4  public enum WorkBenchType
5  {
6      CraftSystem,
7      Cisaille,
8      Poinconneuse,
9      Plieuse,
10     Michel,
11     PerceuseAColonne,
12     Etabli
13 }
  
```

```

WorkBenchType.cs  Consumeltem.cs  PlayerInventory.cs
Assembly-CSharp
1  using UnityEngine;
2  using System.Collections;
3  using UnityEngine.UI;
4  using UnityEngine.EventSystems;
5  using TMPPro;
6  using System.Collections.Generic;
7  using UnityEngine.Events;
8  using Invector.vCharacterController;
9
10 public class PlayerInventory : MonoBehaviour
11 {
12
13     public GameObject inventory;
14     public GameObject characterSystem;
15     public GameObject craftSystem, plieuse, cisaille, etabli;
16     public GameObject slotCasqueDeChantier, slotCasqueAntibruit, slotBackPack;
17     int secuCasqueDC, secuCasqueAB;
18     private Inventory craftSystemInventory, plieuseInventory, cisailleInventory, etabliInventory;
19     private CraftSystem cS, cSPlieuse, cSCisaille, cSEtabli;
  
```

```

Consumeltem.cs  PlayerInventory.cs
PlayerInventory
}

void Start()
{
    menu.SetActive(false);
    menuOk = false;
    experienceBarre = GameObject.Find("XpCourant").GetComponent<Image>();
    securiteBarre = GameObject.Find("SecuriteBarre").GetComponent<Image>();
    niveauDuJoueurTxt = GameObject.Find("NiveauDuJoueur").GetComponent<TMP_Text>();
    niveauXp = GameObject.Find("NiveauXp").GetComponent<TMP_Text>();
    brouzoufText = GameObject.Find("brouzouf").GetComponent<TMP_Text>();
    SecuText = GameObject.Find("SecuText").GetComponent<TMP_Text>();

    if (inputManagerDatabase == null)
        inputManagerDatabase = (InputManager)Resources.Load("InputManager");

    if (craftSystem != null) {...}

    if (plieuse != null) {...}

    if (cisaille != null) {...}

    if (etabli != null)
    {
        cSEtabli = etabli.GetComponent<CraftSystem>();
        etabliInventory = etabli.GetComponent<Inventory>();
    }

    if (GameObject.FindGameObjectWithTag("Tooltip") != null)
        tooltip = GameObject.FindGameObjectWithTag("Tooltip").GetComponent<Tooltip>();
    if (inventory != null)
        mainInventory = inventory.GetComponent<Inventory>();
    if (characterSystem != null)
        characterSystemInventory = characterSystem.GetComponent<Inventory>();
}
  
```

Toujours dans le script « PlayerInventory », dans la méthode « Start », copier et coller la condition non nul (« != » signifie non nul) et adapter la avec les variables de votre nouveau poste de travail, comme dans l'exemple de l'établi.

Dans le script « Consumeltem » rechercher la condition de détection du clic droit, Et faire un « copier/coller » de la condition « else if » et l'adapter à votre nouveau poste de travail.

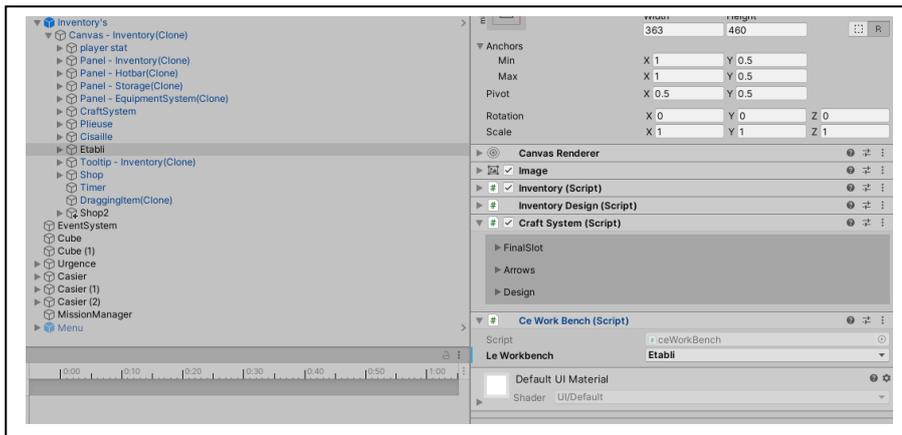
```

if (data.button == PointerEventData.InputButton.Right)
{
    //item from craft system to inventory
    if (transform.parent.GetComponent<CraftResultSlot>() != null)
    {
        bool check = GameObject.FindGameObjectWithTag("Player").GetComponent<PlayerInventory>().inventory.GetComponent<Inventory>().checkIfItemAlreadyExist(item.itemID, item.itemValue);
        chute = this.transform.parent.GetComponent<CraftResultSlot>().chuteItem;

        if (!check)
        {
            GameObject.FindGameObjectWithTag("Player").GetComponent<PlayerInventory>().inventory.GetComponent<Inventory>().addItemToInventory(item.itemID, item.itemValue);
            GameObject.FindGameObjectWithTag("Player").GetComponent<PlayerInventory>().inventory.GetComponent<Inventory>().stackableSettings();
        }

        if (transform.parent.parent.name == "CraftSystem")...
        else if (transform.parent.parent.name == "Plieuse")...
        else if (transform.parent.parent.name == "Cisaille")...

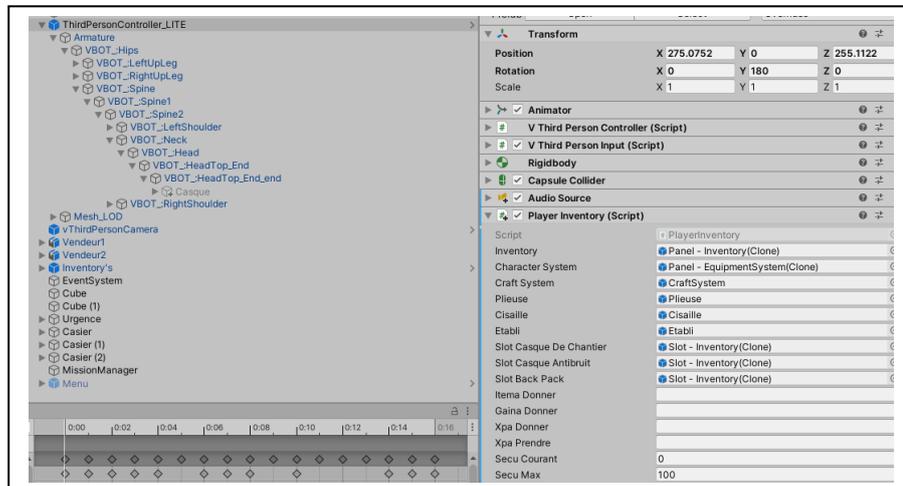
        else if (transform.parent.parent.name == "Etabli")
        {
            CraftSystem cEtabli = GameObject.FindGameObjectWithTag("Player").GetComponent<PlayerInventory>().etabli.GetComponent<CraftSystem>();
            cEtabli.deleteItems(item);
            CraftResultSlot resultEtabli = GameObject.FindGameObjectWithTag("Player").GetComponent<PlayerInventory>().etabli.transform.GetChild(3).GetComponent<CraftResultSlot>();
            resultEtabli.temp = 0;
            if (chute.itemID != 0)
            {
                GameObject.FindGameObjectWithTag("Player").GetComponent<PlayerInventory>().inventory.GetComponent<Inventory>().addItemToInventory(chute.itemID, item.itemValue);
            }
            else
            {
                chute = null;
            }
        }
    }
}
    
```



Dans la Hiérarchie, dans Inventory's, faire un copier-coller d'une fenêtre de craft et l'adapter à vos besoins (Titre, nombre de cases, etc)

Configurer le WorkBenchtype

Dans PlayerInventory, faire un glisser/déposer de la fenêtre précédemment mise en place.

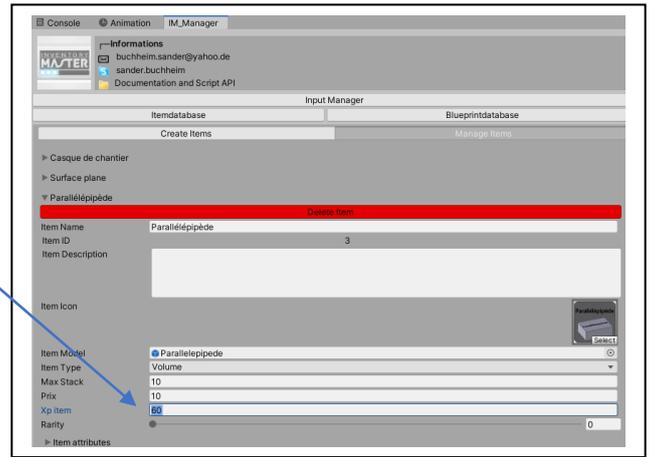


8- Gagner/perdre de l'expérience

- Gagner de l'expérience

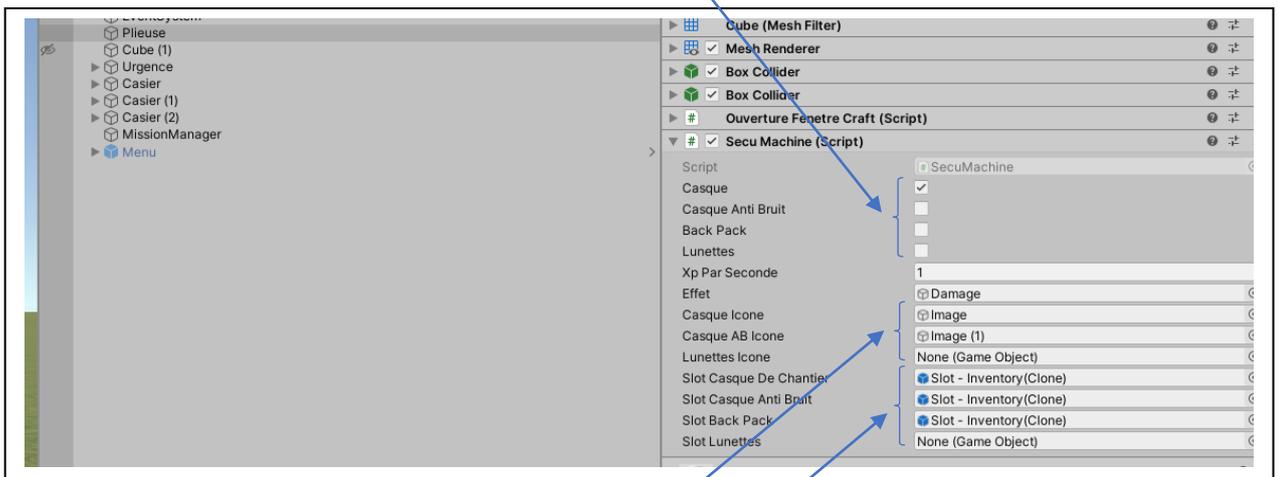
Chaque fois que l'élève fabriquera de nouveaux Items des points expériences lui seront affectés. Ces points sont définis lors de la création de l'item comme dans l'exemple ci-dessous.

Attention, à chaque modification de la valeur « Xp Item », les blueprints correspondant doivent être supprimés et créés de nouveau.



- Perdre de l'expérience :

Chaque fois que l'élève se présentera sans ces EPI devant le poste de travail, des points d'expérience lui seront retirés. Pour ce faire, placer le script « SecuMachine » sur le poste de travail et définir les EPI nécessaires en cochant les éléments correspondants.



Icones d'information sur les EPI nécessaires

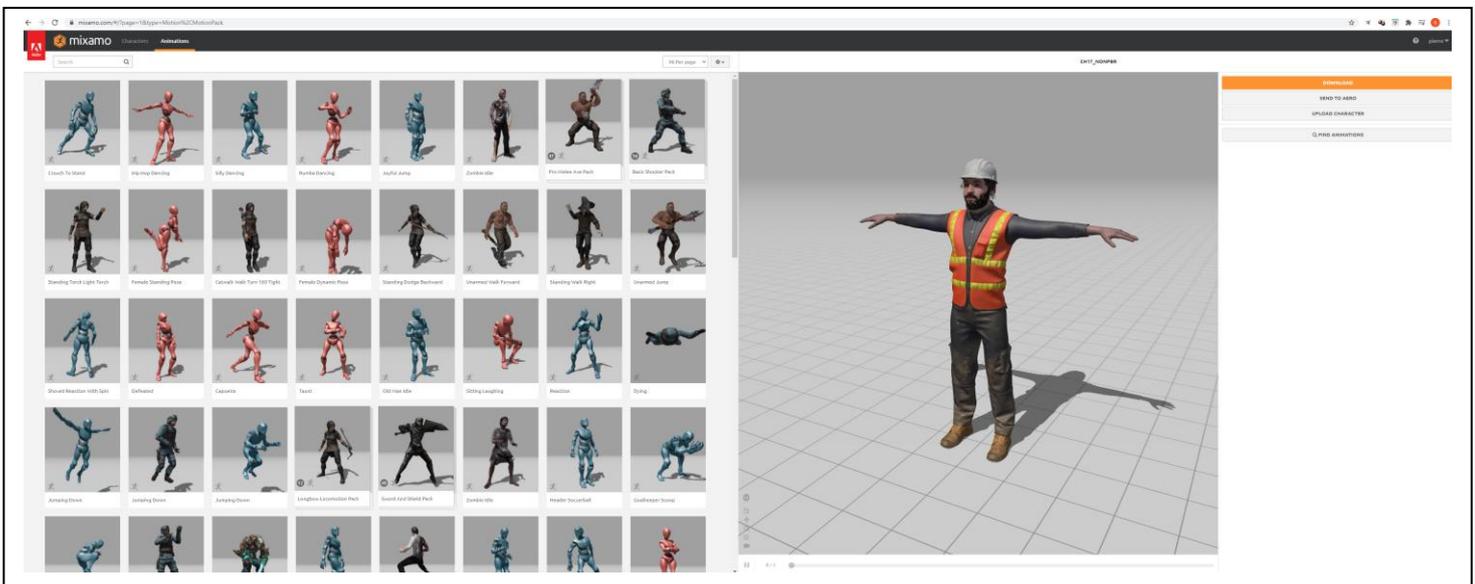
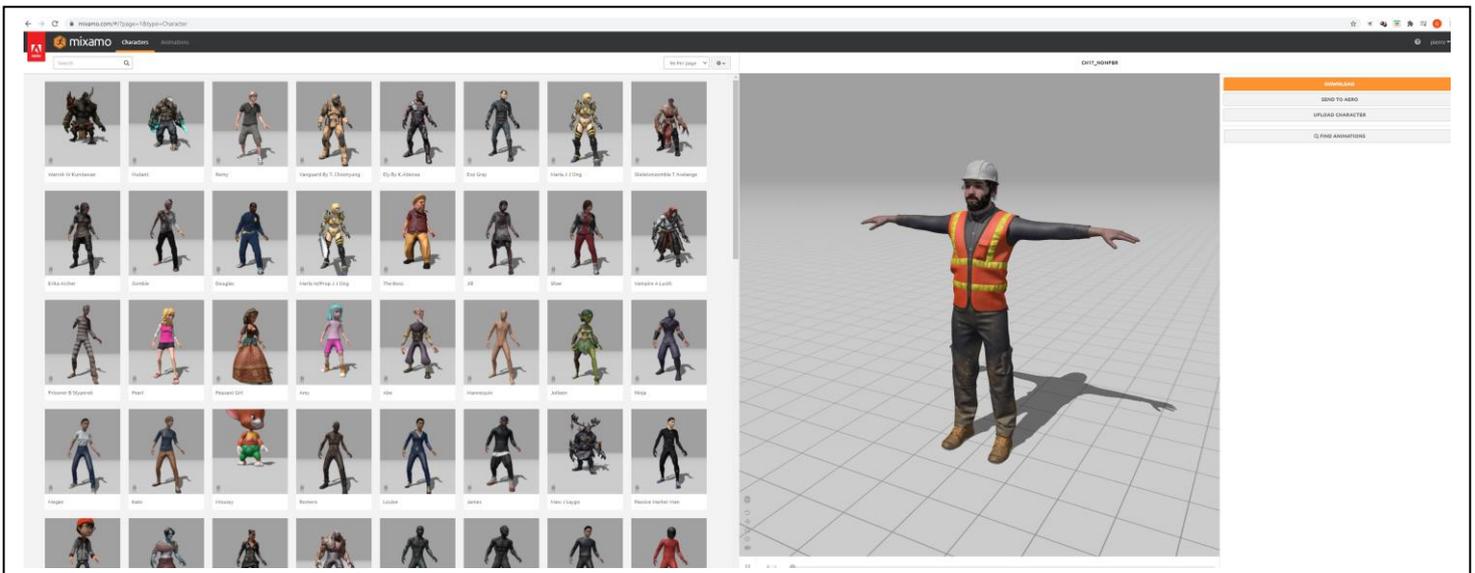
Slots correspondants disponibles dans Panel-EquipementSystem

9- Insérer des personnages non jouables (PNJ)

Le site Mixamo, créer par Adobe, met à disposition gratuitement des personnages 3d et des animations. Le site est disponible à l'adresse suivante :

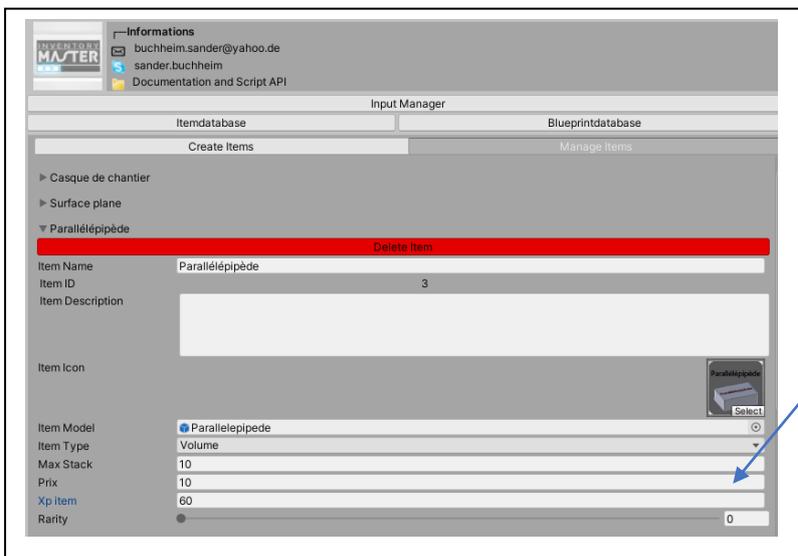
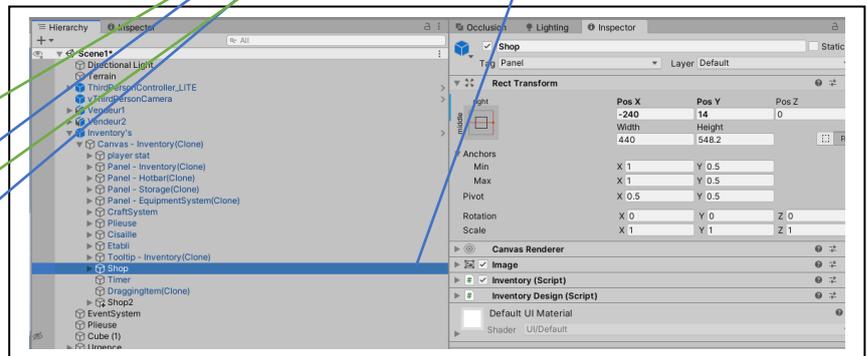
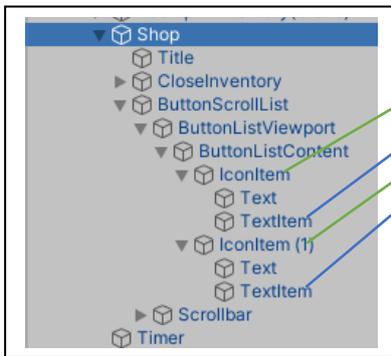
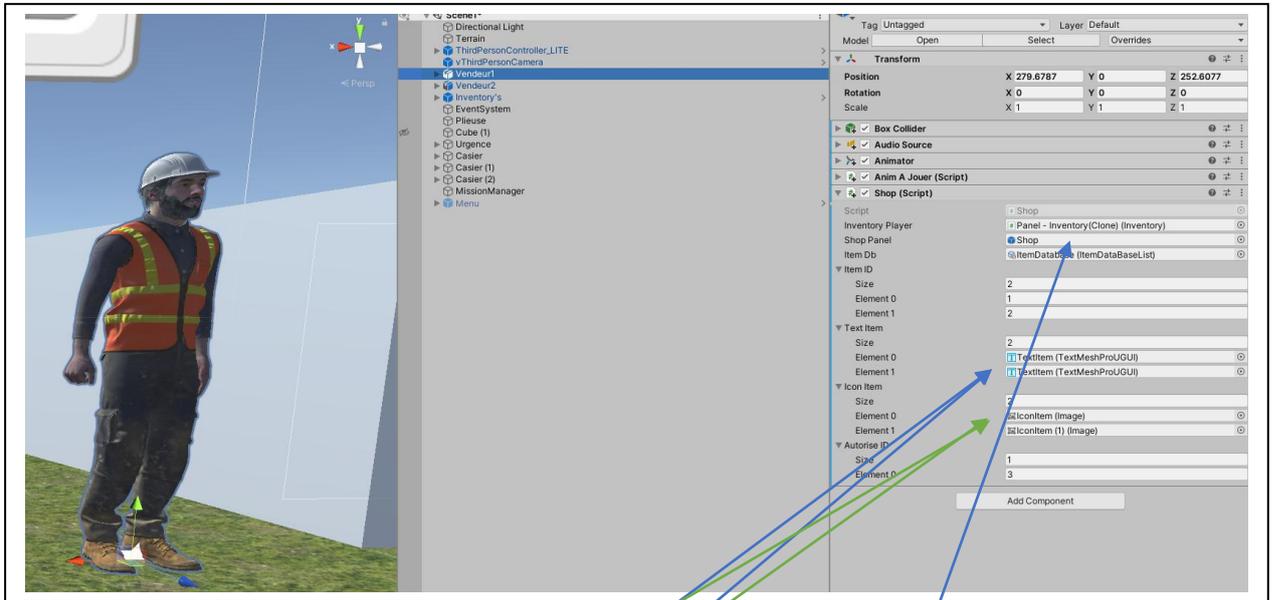
<https://www.mixamo.com/>

Ce site nécessite de créer un compte.



10- Créer des vendeurs

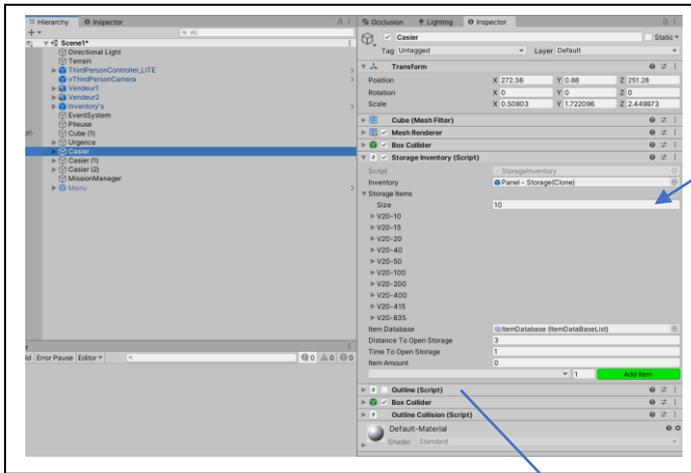
Pour créer de nouveaux vendeurs, il suffit de rajouter le script « Shop » à votre PNJ. Ce script permet de définir la liste des items que le vendeur fournira et la liste des Items qu'il acceptera d'acheter.



Le prix des items affiché par les vendeurs est défini au moment de la création de l'item comme indiqué ci-contre

11- Les lieux de stockage

N'importe quel « GameObject » peut être utilisé comme un lieu de stockage. Il suffit de lui rajouter le script « Storage Inventory », comme dans l'exemple ci-dessous :



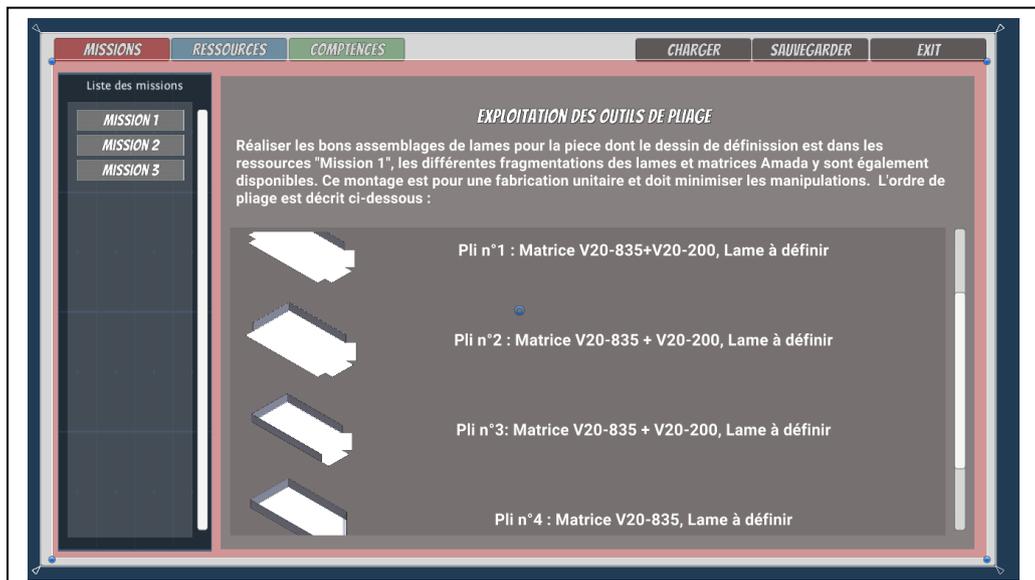
Ci-contre le script « Storage Inventory » appartenant au gameObject « casier ». Il possède une liste de dix items.



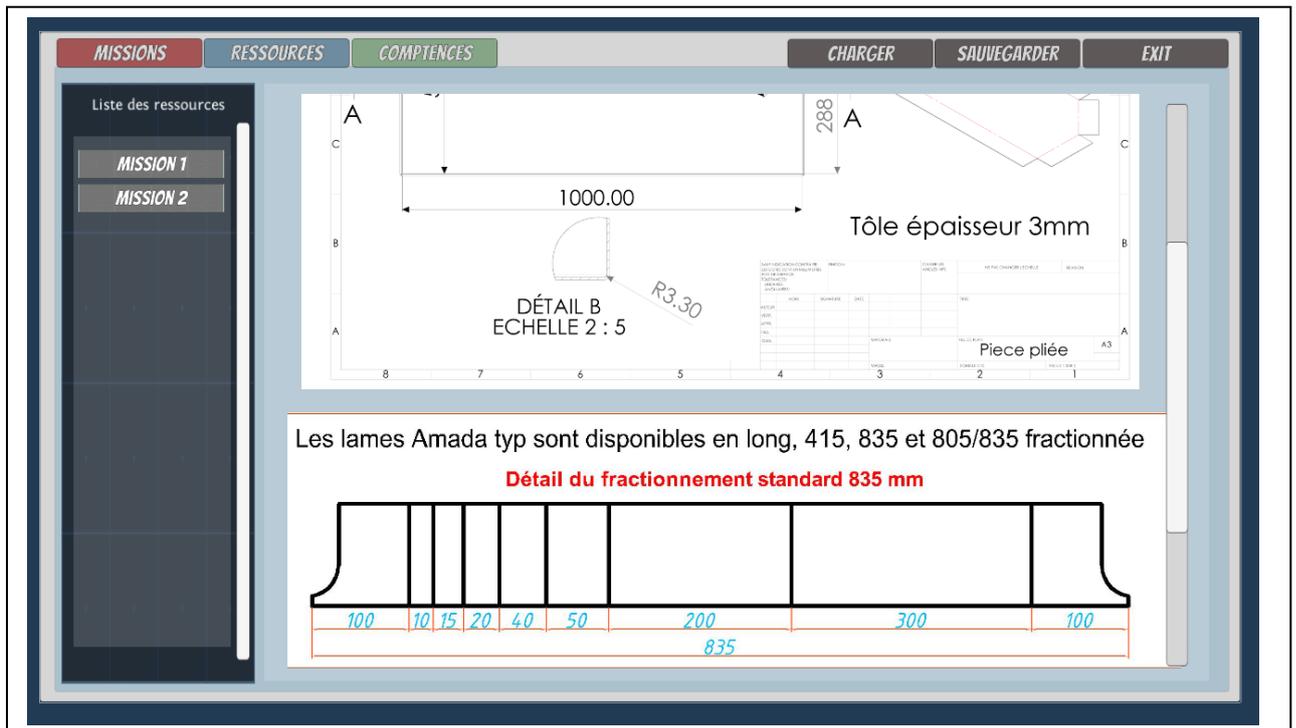
Le script « Outline Collision » permet d'activer la ligne délimitant le contour du casier. Elle permet à l'élève d'identifier une interaction possible avec le casier, et donne des informations supplémentaires sous forme de texte visible dans l'interface utilisateur.

12- Création d'une mission

La touche « M » permet d'accéder au menu, qui est constitué de trois onglets (Mission, ressources, et compétence). Ci-dessous l'onglet « Mission » permet de décrire la consigne et de donner la trame de la mission.

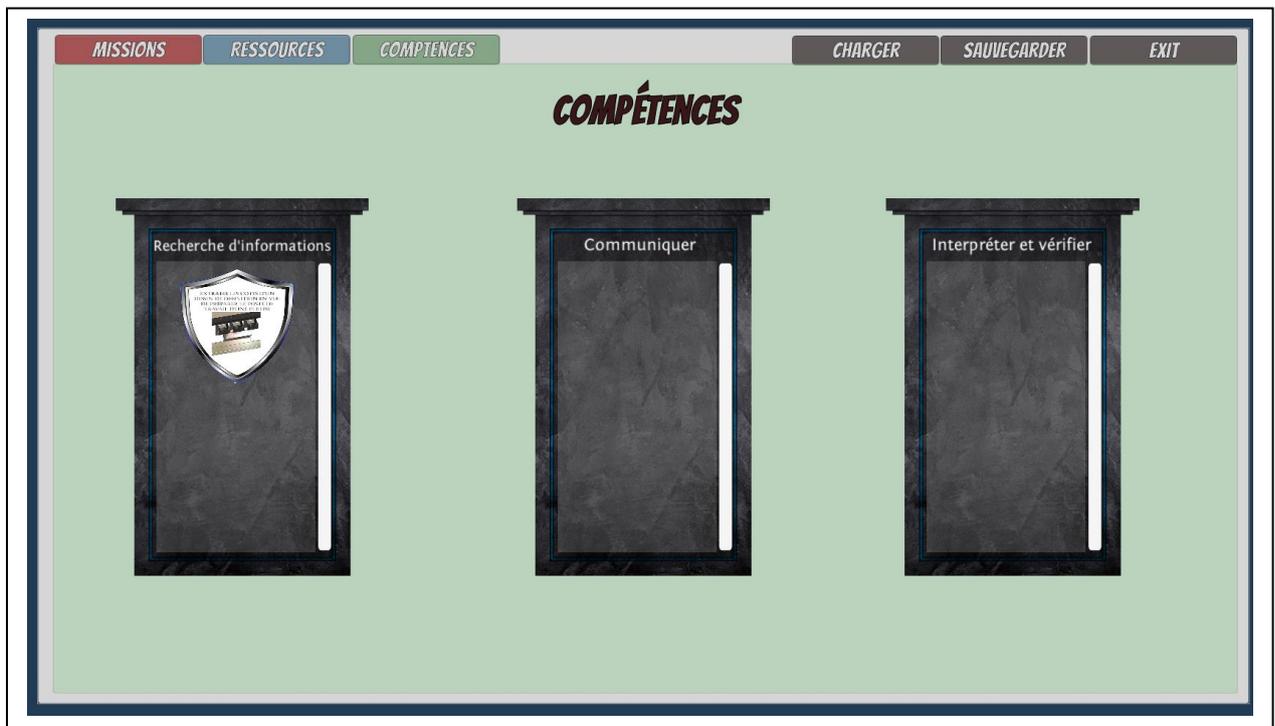


Ci-dessous l'onglet « Ressources » permet d'accéder au dossier technique et aux documentations du



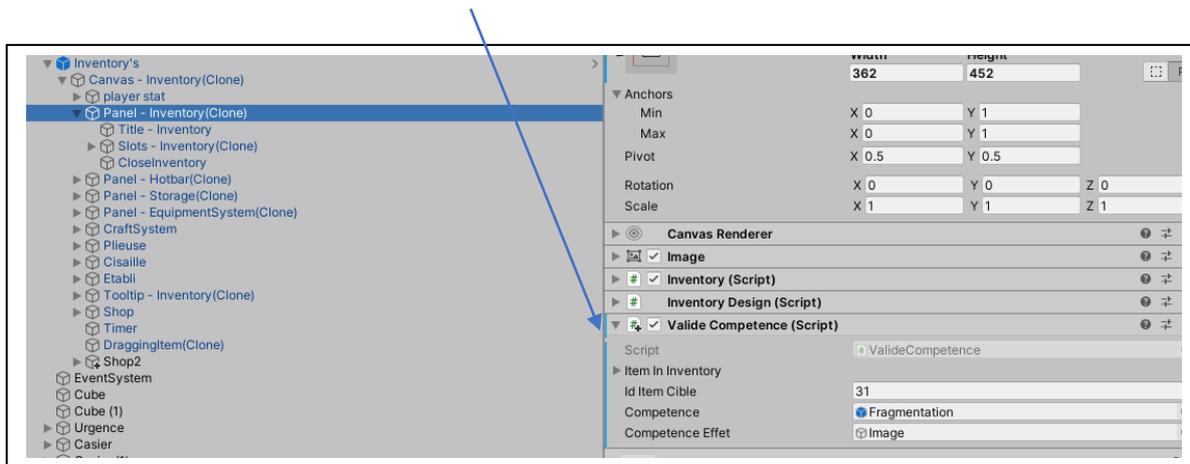
constructeur.

Ci-dessous, l'onglet « compétences » permet de mettre en avant la progression de l'élève en lui



octroyant des badges à chaque validation de mission.

Le script « Valide Compétence » permet de détecter la présence d'un item spécifique dans la fenêtre d'inventaire du personnage. Dans l'exemple proposé, si l'élève a l'item 31 (Pièce finale après pliage) alors il déclenchera la validation de la compétence correspondante.



13- Conclusion

L'avènement de l'outil informatique et d'internet ont démocratisé l'accès au savoir, imposant aux enseignants de modifier leur pratique. L'usage de contenus interactifs permet désormais de susciter l'envie auprès des élèves en créant des situations d'apprentissage originales.

Unity est un logiciel puissant de création d'animations, certes complexe, mais les éléments contenus dans la ressource doivent permettre une prise en main rapide des fonctions principales afin de proposer des contenus adaptés aux besoins.

J'ai personnellement pris énormément de plaisir à développer ces outils et à les partager. Ils sont sans doute imparfaits et évolueront en fonction des progrès accomplis dans le futur. D'autres applications de ce type verront également le jour au profit des élèves qui, je l'espère, seront les grands gagnants de ce travail.

Vous pouvez me contacter si vous rencontrez des difficultés ou si vous souhaitez aller plus loin dans l'utilisation du logiciel (pierre.chauvin15@ac-orleans-tours.fr).