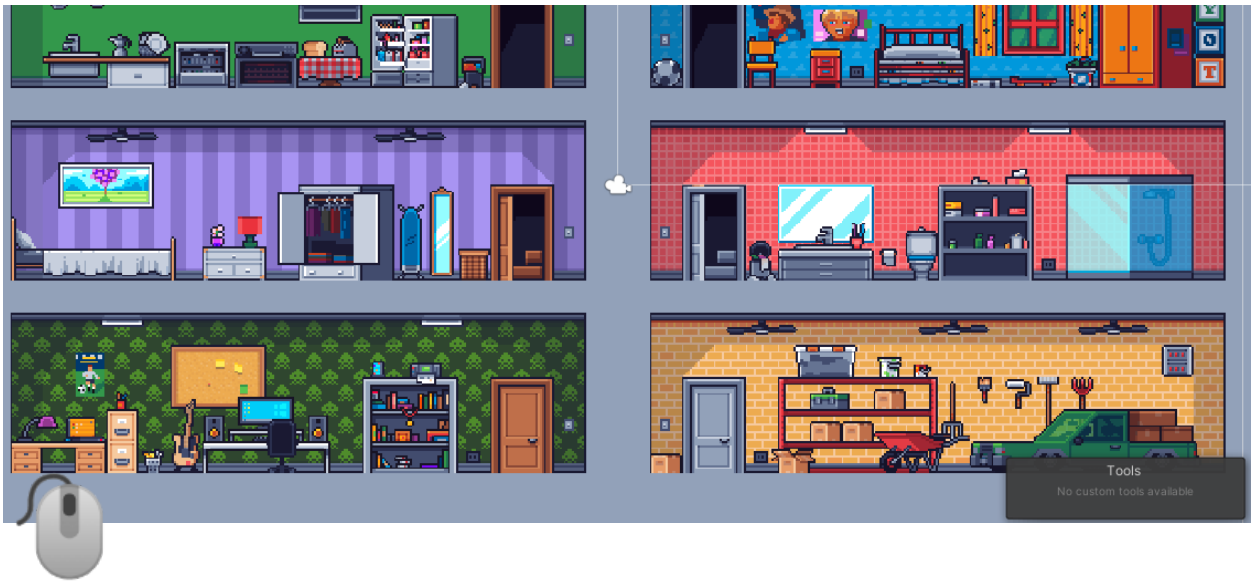



# Workshop Point & Click 2D

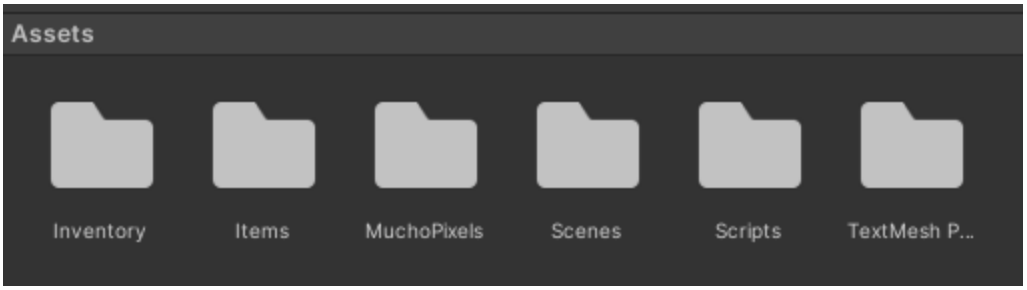
## Introduction

Durant ce workshop, on va découvrir Unity et voir comment créer un jeux point and click en 2D

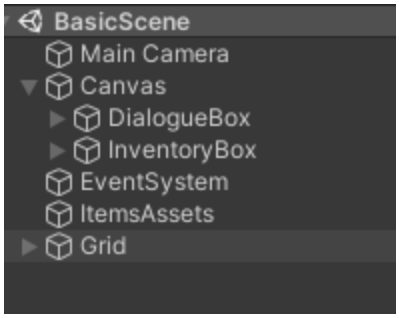
---

## Setup

- Pour commencer, il faut avoir installer Unity version **2019.4.20.**
- Maintenant vous pouvez créer un projet 2D, en ajoutant le chemin et un titre par exemple *workshopPointAndClick.*
- Une fois que vous avez créer un projet, on vas télécharger quelques assets
    - En haut à gauche, Assets → Import Package → Custom Package et importez le **Workshop-point-and-click-setup.unitypakage** veillez à avoir toutes les cases cochez lors de l'import.
- Bravo, vous avec réussi à setup le projet. Passons en revus ce que vous venez de télécharger




- Inventory: les textures pour l'Ui de l'inventaire
- Items: les png des items à intégrer dans l'inventaire
- MuchoPixels: Les élements de la map
- Scène: les Scènes du jeux, vous pouvez double clic sur BasicScene
- Scripts:
  - Inventoty: le systeme d'inventaire avec des méthode pour ajouter, supprimer des éléments et les déclarer
  - Ui: le système d'UI, avec les dialogues et l'inventaire
  - Player: le joueur, un component qui contient une UI et un inventaire




- La scène contient :
  - Main Caméra: une caméra fixe sur notre map

- Canvas, si vous avez fais le workshop sur les menus vous savez déja comment ça fonctionne. On à des component un pour les dialogues et un autre pour l'inventaire
- ItemAssets, un script qui lie un png à un enum
- Grid: très pratique pour la 2d, elle représente notre map

---

## Step 1

Maintenant qu'on à passer en revue les components on peut commencer à créer un jeux et les premiers game event

Pour commencer,

- Créez un **gameObject** vide en l'appelant *GameEvent* par exemple, il contiendras toutes nos futurs intéractions
- Pour plus de clarté, on vas découper les events en fonctions des pièces ou ils prennent place, ajoutez dans *GameEvent* un **GameObject** *Garage*
- On peut maintenant ajouter des components dans notre garage,

- Créez un **GameObject** dans le garage en l'appelant B*ox* ou tous autres objets qui vous inspires. ajouter lui un component **Box Collider2D** et positionné le autour de votre objet. Vous pouvez également **Edit Collider**

afin de le redimensionner. Il s’agit de la zone dans laquelle on vas cliquer pour lancer un event

- Ajoutez un script à votre component, deux fonctions sont normalement générés
  - Start appélé une fois, lors de l’initialisation de notre component.
  - Update appelé à chaque frame, vous pouvez la supprimée

Vous pouvez déclarer deux variables

```
/*
dialogueUI représente notre dialogueBox dans le Canvas,
il contient une méthode ShowDialogue(string dialogue)
qui fait défiler ... un dialogue, pas simple à devinez
*/

[SerializeField] private DialogueUI dialogueUI;

/*
player permet d'acceder à l'inventaire et son UI
*/

[SerializeField] private Player player;
```

Puis, ajoutez une méthode qui est appelé lors du clique sur un box collider,

```
void MaMéthode() { /* A remplacer par La méthode que vous avez trouvé */
    player.inventory.AddItem(
      // vous pouvez ajouter un Fusible ou un autre type
      // présent dans l'enum ItemType de Script/Inventory/Item.cs
      // Mais, seul les fusibles sont stackable, voir IsStackble()
      new Item { itemType = Item.ItemType.Fusible, amount = 1 }
    );
    dialogueUI.ShowDialogue("Votre Dialogue");
}
```

Vous pouvez maintenant appuyer sur **play** et cliquer sur votre objet, vous devriez voir apparaitre votre dialogue et l’objet que vous avez choisi êtres

ajouter dans votre inventaire.

Par défault c’est un fusible, pour fermer le dialogue appuyer sur **espace**

---

## Step 2

Une boite qui contitent des fusibles à l’infinie n’est pas très réaliste, créez deux comportement si la boite est vide ou pleine, avec deux dialogues distinct.

N’hésiter pas essayer de changer le type d’objet ou à rajouter les votres

La prochaine étape c’est de ce servir de l’objet que vous venez de trouver

Avec ces deux fonctions vous pouvez détecter les objets dans l’inventaire

```
public bool IsTypePresent(Item.ItemType itemType)
public bool IsEnoughTypePresent(Item.ItemType itemType, int amount)
```

Et vous pouvez en supprimez avec

```
public void RemoveItem(Item item)
```

Maintenant, crée un **GameEvent** qui vous convient

Par vous réparé l’ordinateur avec un fusible, ouvrez l’armoire avec une clée, créez différents dialogues si vous avec l’objet, si il vous le manque, etc...

---

## Step 3

Vous savez tous se dont vous avez besoins pour créer un jeux, essayer de multiplier les **GameEvent** dans différentes pièce de la maison.

Avec au moins 4 GameEvent différents, 3 items et pourquoi pas une fin en trouvant les clées de la voiture, le mot de passe de l’ordinateur ou autre.