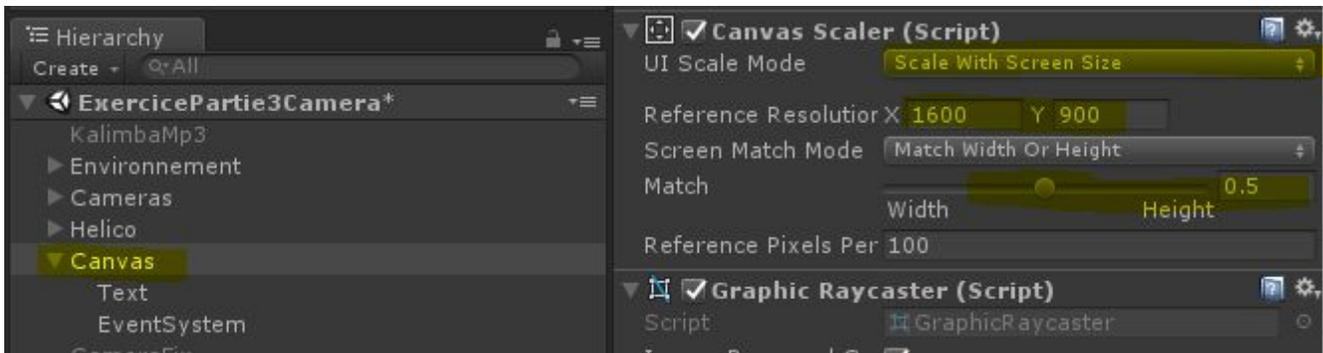


## Éléments d'interface (UI) texte, bouton et image

### Redimensionnement automatique des éléments d'un Canvas

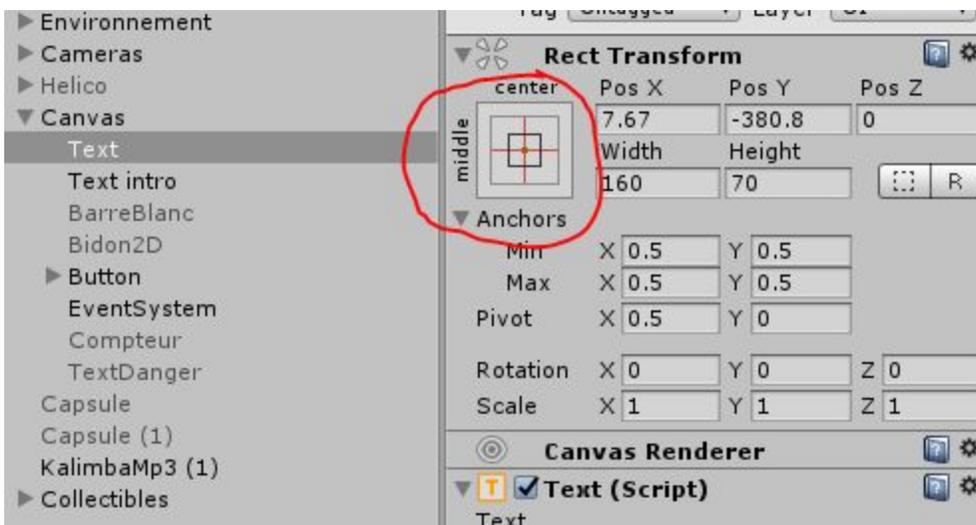
Le composant Canvas Scaler est utilisé pour contrôler l'échelle globale et la densité de pixels des éléments d'interface utilisateur dans le canevas. Cette mise à l'échelle affecte tout ce qui se trouve sous le canevas, y compris les tailles de police et les bordures d'image.



### Fixer la position relative des éléments UI

Il faut fixer l'emplacement des ancrs des éléments UI dans l'inspecteur (valeur par défaut est le centre) pour leur permettre de rester dans l'écran lorsque les dimensions de l'écran changent.

Les éléments qui se trouvent dans les coins de l'écran doivent être ancrés aux coins correspondants.



## L'élément Button (Create/UI/Button)

### Propriétés:

**Interactable:** Ce composant acceptera-t-il la saisie? Voir Interactable.

**Transition:** Propriétés qui déterminent la façon dont le contrôle répond visuellement aux actions de l'utilisateur.

**Navigation:** Propriétés qui déterminent la séquence des contrôles à l'aide de TAB. (Choisissez *None* pour gestion avec la souris)

### Événements

**On Click:** Un événement qui est invoqué quand un utilisateur clique sur le bouton et le **relâche**.

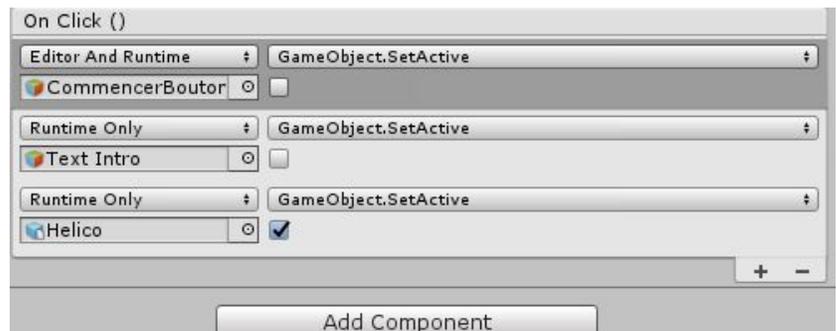
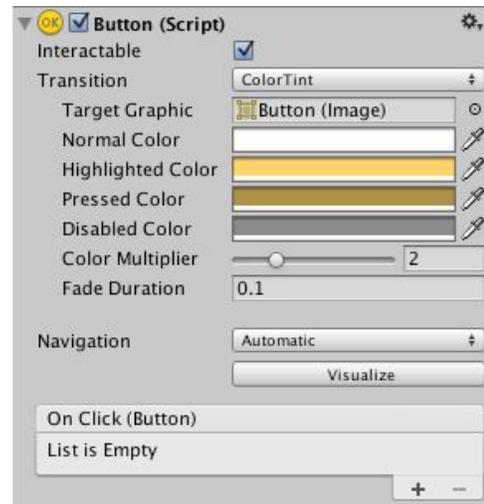
Permet d'exécuter une ou plusieurs commandes sur les objets (ou appeler une des fonctions d'un objet) lorsque le bouton est cliqué et **relâché**.

La fonction doit être public.

Si la souris est déplacée à l'extérieur du bouton avant que le bouton soit relâché, l'action ne se produira pas.

Les cas d'utilisation typiques incluent:

- Confirmer une décision (par exemple, lancer un jeu ou enregistrer un jeu)
- Déplacement vers un sous-menu dans une interface graphique
- Annulation d'une action en cours (par exemple, téléchargement d'une nouvelle scène)



## Consignes pour l'exercice

### Texte d'introduction

- Désactivez l'hélicoptère.
- Créez un texte présentant les instructions. (**Create/UI/Text**)
  - Ex. *Vous devez atteindre la base en 120 secondes...Attention de ne pas manquer d'essence*
- Fixez les ancrs du texte (Anchors) au centre de l'écran. (center,middle)  
Ceci permet de fixer l'emplacement de l'élément texte par rapport à la fenêtre de canvas.

### Compteur

- Créez un texte permettant d'afficher un compteur.
- Positionnez le texte et fixez les ancrs du texte (Anchors) en haut à droite de l'écran. (right,top)
- Créez un script pour cet objet qui permet de gérer un compteur descendant (de 120 à 0).
  - Ajoutez **using UnityEngine.UI** au début du script, pour pouvoir utiliser les méthodes UI.
  - Déclarez une variable de type **int**.
  - Indice: vous pouvez utiliser InvokeRepeating() ou autre méthode.

### Exemple

```
// Gestion d'un compteur qui augmente de 10 à chaque seconde et affichage dans un UI Text.
....
using UnityEngine.UI; //ajoute de bibliothèque UI

public class CompteurScript : MonoBehaviour {

    public int valCompteur=0;

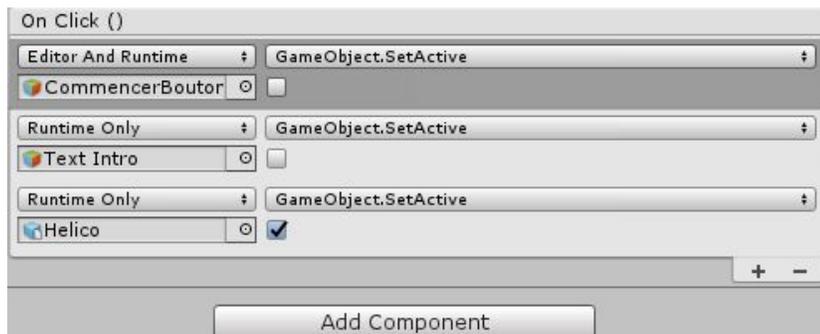
    // Déclenche InvokeRepeating une fois
    void Start ()
    {
        InvokeRepeating("Compteur", 1, 1); //à chaque seconde appelle Compteur
    }
    void Compteur()
    {
        valCompteur += 10;
        GetComponent<Text>().text = valCompteur.ToString(); //convertit la valeur en texte et
                                                             // l'affiche
    }
}
```

- Le compteur doit diminuer de 1 à chaque seconde.
- Il ne doit pas descendre plus bas que 0.
- Lorsque le compteur est rendu à 0, le moteur doit s'arrêter et l'hélico commencer à tomber. Le InvokeRepeating doit aussi être annulé (CancelInvoke("Compteur");

**Remarque:** Pour faciliter l'explosion, placez le code d'explosion du script de l'hélico dans une fonction (ex: ExploderHelico()), qui pourra être appelée à partir du script du compteur lorsque le compte arrive à 0. La fonction doit être public.

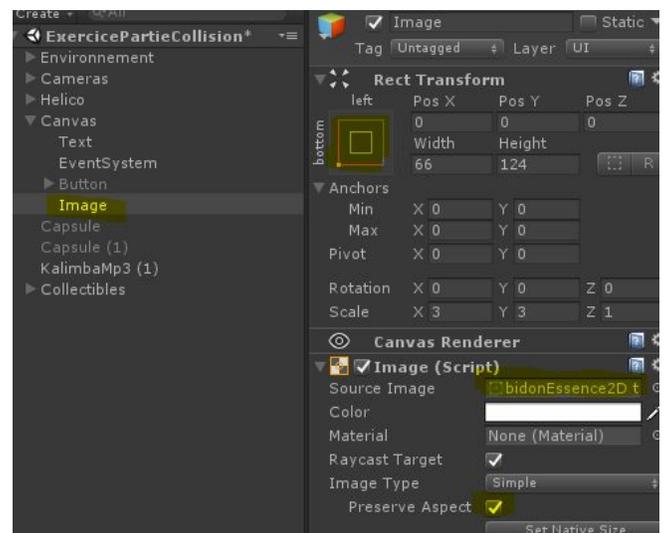
### Bouton Commencer

- Créez un bouton permettant de commencer la partie (**Create/UI/Button**). Lorsqu'on clique sur ce bouton :
  - Le texte d'introduction et le bouton disparaissent;
  - L'hélicoptère apparaît;
  - Le compteur et la barre d'essence apparaissent également. (pas présents dans le graphique)

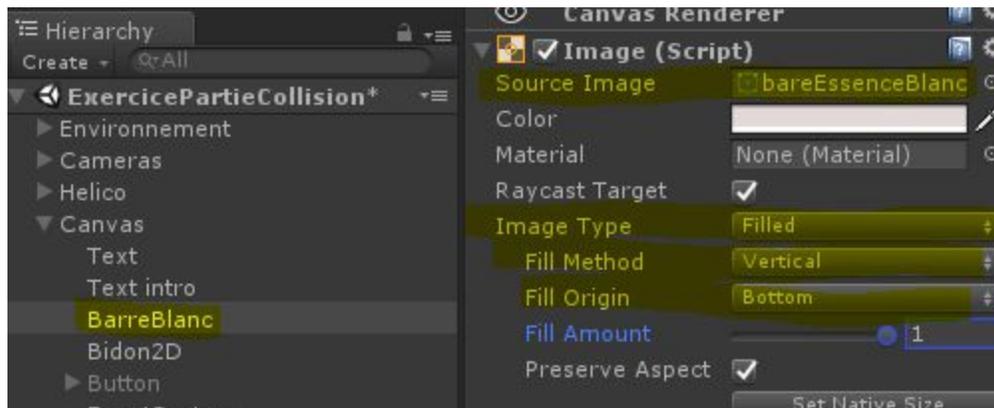


### Image Barre d'essence

- Importez les textures du bidon d'essence et de la barre blanche. Modifiez le type des textures à **Sprites** dans l'inspecteur.
- Créez une image UI (Create/UI/Image) et placez la dans la partie inférieure gauche de la scène.
- Fixez les ancres de l'image (Anchors) en bas à gauche. (left,bottom)  
Ceci permet de fixer l'emplacement de l'image par rapport à la fenêtre de canvas.
- Assignez l'image du "bidonEssence2D" à la propriété "Source Image".
- Cochez la case *Preserve Aspect*. Ceci permet de garder le même ratio largeur vs hauteur lors du redimensionnement..
- Créez une autre image UI (Create/UI/Image) pour la barre blanche "barreEssenceBlanche".
- Ajustez les dimensions et placez-la dans la partie inférieure gauche en arrière du bidon. Dans la fenêtre *Hierarchy*, elle doit se trouver en haut de l'image du bidon (dans le canvas, les éléments plus haut sont en arrière des autres).



- Modifiez les propriétés de cette image (comme dans l'exemple) pour permettre de créer une jauge de remplissage.



- Pour modifier les propriétés de l'image, vous aurez besoin d'une variable de type *Image*. Dans l'inspecteur, vous pourrez glisser dans cette variable le *gameObject* qui contient l'image de la barre blanche.
- Pour redimensionner l'image, assurez-vous que le type de votre image est à "Filled", ce qui vous donnera accès à la propriété "fillAmount".

#### Exemple

```

public float quantiteEssence
public Image monImage; //l'image initialisée dans l'inspecteur
void GestionEssence()
{
    ....
    quantiteEssence -= ...;
    monImage.fillAmount = quantiteEssence; //changement de la hauteur de l'image
    ....
}
}

```

- **Dans le script de l'hélico**, créez une fonction pour la gestion de la quantité d'essence (*GestionEssence()*) Cette fonction est appelée dans le *FixedUpdate()* de l'hélico. Pour gérer la quantité d'essence disponible, vous aurez besoin d'une variable *quantiteEssence*.
  - n'oubliez pas d'ajouter **using UnityEngine.UI** au début du script
  - L'essence diminue à chaque frame si l'hélico est démarré;
  - L'essence augmente lorsqu'un bidon d'essence est ramassé;
  - La hauteur de la barre d'essence diminue ou augmente en fonction de la valeur de la variable "*quantiteEssence*";
  - Si la quantité arrive à 0, les hélices de l'hélico doivent arrêter et il doit tomber et exploser en touchant le terrain.

## Défis optionnels

#### Avertissement du niveau d'essence

- Si le niveau d'essence chute en bas de 30%, afficher un message clignotant "**Danger ! Écrasement imminent !!!**"
- Si le niveau d'essence remonte à plus de 30%, le message doit cesser d'apparaître.