

# 10

## Interactions par le regard

---

Vous le savez maintenant, avec la réalité virtuelle presque tout passe par le regard. Les expériences de réalité virtuelle se vivent avec un casque sur les yeux, et l'utilisateur peut ne disposer que du casque, c'est pourquoi tout doit pouvoir se faire via le regard et en tournant la tête.

Par exemple, si dans votre expérience il y a un dialogue avec un personnage virtuel et que vous voulez que l'utilisateur lui réponde, vous pouvez imaginer un système qui détecte si ce dernier fait oui ou non de la tête. Un interrupteur peut être activé si on le regarde fixement pendant une seconde. Ce sont des moyens simples que vous pourriez être amenés à implémenter selon vos besoins.

Pour notre visite d'appartement, nous devons gérer les lumières, la porte et la télévision. Ces trois exemples seront suffisants pour pratiquer les interactions avec l'environnement. Étant donné que le Google Cardboard dispose d'un bouton de clic sur le casque, je vais aussi vous montrer comment utiliser ce système. Vous verrez donc dans ce chapitre plusieurs façons de procéder afin de vous laisser le choix pour vos futurs projets.

### 10.1. Allumage de la télévision : regard + clic

L'objectif est de pouvoir allumer la télévision en regardant la télécommande une petite seconde. Le principe sera sensiblement le même que pour le bouton de marche sauf que l'objet qui en subira la conséquence sera la télévision.

Commencez par ajouter un [EventTrigger](#) à votre télécommande afin de pouvoir détecter les interactions avec celle-ci. Vous pouvez mettre en place le même système de timer qu'au chapitre précédent mais pour changer nous utiliserons ici l'événement `POINTER CLICK`. Comme son nom l'indique, celui-ci est déclenché lors d'un clic. L'utilisateur devra donc actionner le bouton du Google Cardboard pour le déclencher.

*Note > Beaucoup de casques ne disposent pas du clic, c'est pourquoi je vous invite à plutôt mettre en place le déclenchement par timer et réutiliser le code que nous avons vu au chapitre précédent pour vos projets si vous souhaitez toucher un plus large public.*

Nous associerons une fonction à cet événement une fois notre script créé. Ajoutez donc un nouveau script à la télécommande. Appelez-le `Te1ecommande` et ouvrez-le. Il nous

faudra deux variables, la première pour stocker l'image de la télévision allumée et la seconde qui sera un booléen permettant de savoir si la télévision est éteinte ou allumée afin de pouvoir changer son état.

Créez une fonction `UtiliserTelecommande` dans laquelle vous utiliserez un `if/else` permettant d'activer ou de désactiver le `GameObject` de l'image. Le code est assez simple :

```
using UnityEngine;
using System.Collections;

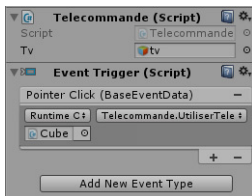
public class Telecommande : MonoBehaviour {

    public GameObject tv;
    private bool isOn = false;

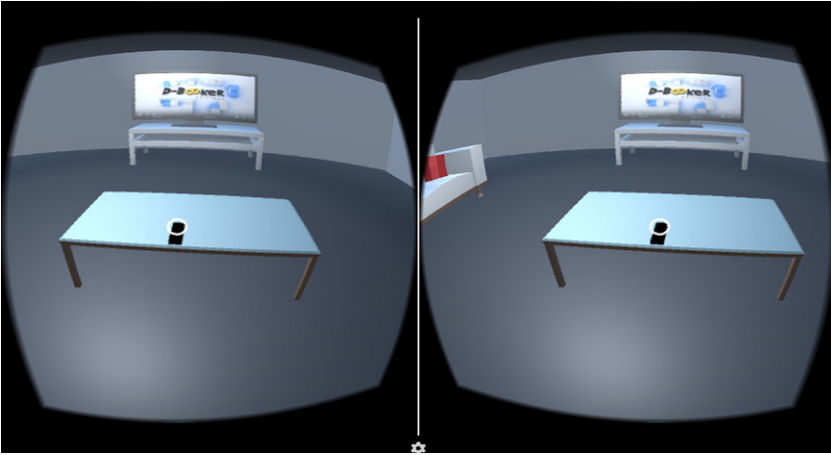
    public void UtiliserTelecommande()
    {
        if(!isOn)
        {
            tv.SetActive(true);
            isOn = true;
        }
        else
        {
            tv.SetActive(false);
            isOn = false;
        }
    }
}
```

Vous pouvez maintenant attribuer les variables du script et associer la fonction `UtiliserTelecommande` à l'événement de clic :

**Figure 10.1** : Configuration du script



Notre script est terminé ! Ça n'est pas plus compliqué que ça. Vous pouvez tester votre programme afin de constater le bon fonctionnement de votre système fraîchement programmé :

**Figure 10.2 :** Activation de la télé en regardant la télécommande

Je me dirige vers la télécommande, je clique et la télé s'allume. Nous pouvons l'éteindre de la même façon. Ce système est très basique mais il fonctionne et tient sur quelques lignes de code. Si vous décidez de mettre en place un timer, vous pouvez ajouter une bulle avec la barre jaune qui permettra au joueur de bien se situer par rapport au timer.

*Note > Dans la première partie de ce livre, nous avons vu qu'il était important, pour une bonne immersion, d'ajouter des effets sonores. Vous pouvez facilement intégrer des effets sonores en ajoutant un AudioSource à vos objets (comme par exemple à la télévision) et en appelant la fonction `tv.GetComponent<AudioSource>().PlayOneShot(tvSfx)`; Ici, `tvSfx` est un `AudioClip`. Ce bout de code peut être appelé dans votre fonction `UtiliserTelecommande()` lors de l'allumage de la télévision.*

*Nous avons également parlé de la lumière. Vous pouvez ajouter une Spotlight au-dessus de la télécommande pour attirer le regard de l'utilisateur sur elle afin qu'il comprenne qu'il peut interagir avec.*

## 10.2. Allumage des lumières à l'entrée dans une pièce

Nous allons maintenant passer à la programmation des lumières de la scène. Cette fois-ci, nous allons développer un système très différent. Nous utiliserons les  **cubes invisibles géants qui remplissent les pièces de la maison**  pour détecter la position du joueur et ainsi allumer uniquement la lumière de la pièce dans laquelle il se trouve.

La fonction `OnTriggerEnter` va permettre de détecter lorsque le joueur pénètre dans une nouvelle pièce et entre donc en collision avec un cube. Pour l'utiliser, vous devez ajouter un Collider au personnage ainsi qu'un RigidBody (menu `COMPONENT/PHYSICS`). Vous pouvez utiliser le Collider de votre choix. Pour le RigidBody, vous pouvez contraindre