

Saisies usager (Input)

CSTJ – TIM
Jeu Vidéo 3



On peut saisir...

- Touches de clavier
- Boutons de la souris
- Contrôleur de type console (gamepad)
- Manette de jeu (joystick)
- Touchés (tablettes, écrans tactiles)

La majorité des projets à ce jour utilisent l'objet **Input** et ses méthodes pour cela.

Un nouveau système de saisie est en cours d'intégration dans Unity (pas abordé dans ce cours).

Saisir une touche du clavier...


- Input.**GetKey()** – Une touche est pressée (détectée avec répétition)
- Input.**GetKeyDown()** – Une touche est pressée (une fois)
- Input.**GetKeyUp()** – Une touche est relâchée

Pour indiquer la touche spécifique à détecter, on utilise la classe **KeyCode** et ses propriétés (KeyCode.A, KeyCode.B, etc):

```
If( Input.GetKeyDown( KeyCode.Space ) == true )  
{  
    // faire quelque chose ici...  
}
```



Détecter un clic de souris

- 
- Input.**GetMouseButton()** - Un bouton de souris est pressé en continu (répétition)
 - Input.**GetMouseButtonDown()** - Un bouton de souris est pressé (une fois)
 - Input.**GetMouseButtonUp()** - Un bouton de souris est relâché

Pour déterminer le bouton à détecter, on utilise un chiffre qui représente le bouton (0 = bouton de gauche, 1 = bouton de droite, 2 = bouton central)



Fonctions en lien avec les paramètres des préférences *(Edit > Project Settings > Input Manager)*

- Input.**GetAxis()** - de -1f à 1f (Spécifier "Horizontal" ou "Vertical")
- Input.**GetAxisRaw()** - de -1 à 1 (Spécifier "Horizontal" ou "Vertical")

GetMouseButton(), GetMouseButtonDown(), GetMouseButtonUp()

Pour ces méthodes, des noms sous forme de chaînes permettent de relier des actions de l'utilisateur avec des groupes d'entrées (touches, clics de souris, joystick, etc.).

Par exemple, Input.**GetMouseButtonDown("Fire1")** détecte si la touche espace ou le bouton de gauche de la souris sont pressés.

Quelques propriétés souvent utiles

- `Input.anyKey` - true, dès qu'une touche est pressée (répétition)
- `Input.anyKeyDown` - true, dès qu'une touche est pressée (une fois)





Quelques propriétés souvent utiles (suite)

- **Input.mousePosition** - Correspond à la position de la souris

Attention!

Cette coordonnée doit être convertie selon la vue de la Caméra quand on est en 2D pour correspondre à une coordonnée valide à l'écran:

```
Vector3 pt = Camera.main.ScreenToWorldPoint(Input.mousePosition);
```

