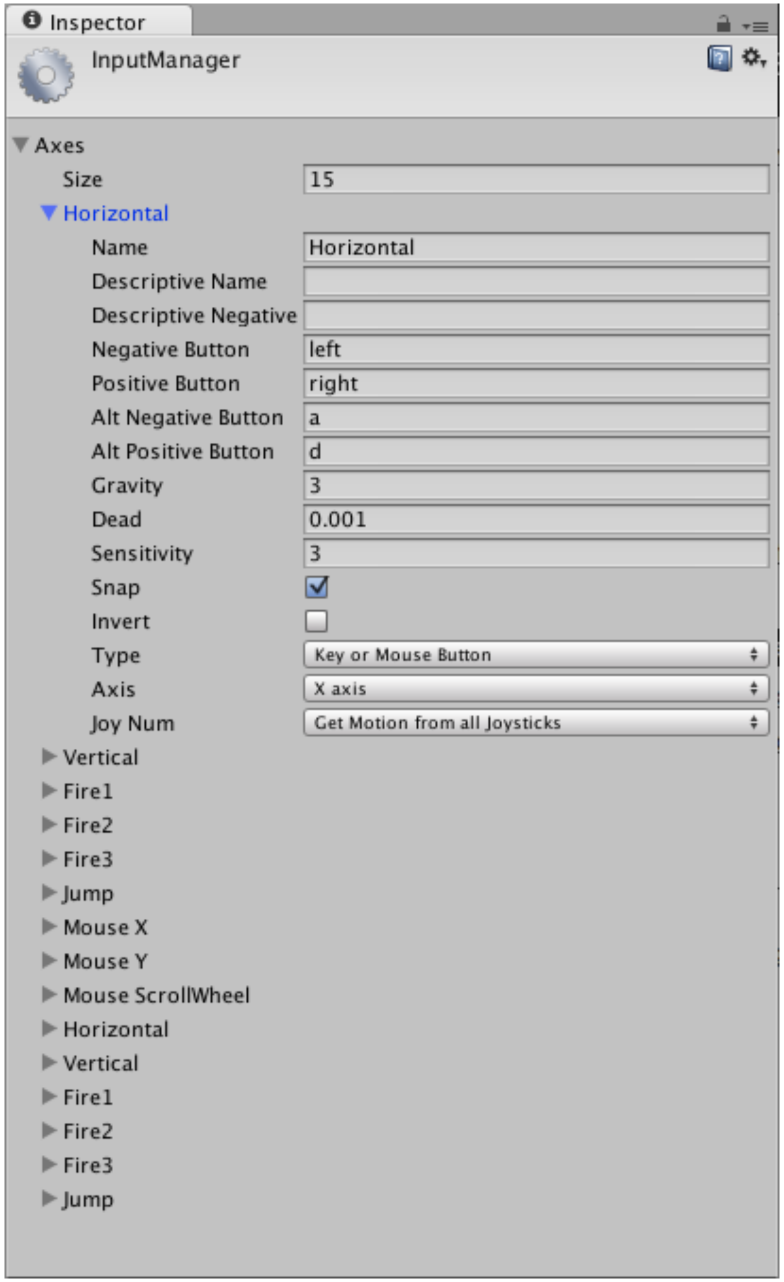
# CT3536 (Unity3D)

## Keyboard & Mouse Input

# Input Manager

Edit > Project Settings > Input

- Used to define named input control schemes mapped to keyboard, joystick, mouse, etc.
- Standard predefined axes:
  - Input.GetAxis("Vertical")
  - Input.GetAxis("Horizontal")
  - The value will be in the range -1...1 for keyboard and joystick input.
  - Default mapping to arrow keys and to WASD
- Also standard predefined:
  - Input.GetAxis("Mouse X")
  - Input.GetAxis("Mouse Y")



Inspector

InputManager

▼ Axes

| | |
|---|---|
| Size | 15 |
| ▼ Horizontal | |
| Name | Horizontal |
| Descriptive Name | |
| Descriptive Negative | |
| Negative Button | left |
| Positive Button | right |
| Alt Negative Button | a |
| Alt Positive Button | d |
| Gravity | 3 |
| Dead | 0.001 |
| Sensitivity | 3 |
| Snap | ✓ |
| Invert | ☐ |
| Type | Key or Mouse Button |
| Axis | X axis |
| Joy Num | Get Motion from all Joysticks |

▶ Vertical
▶ Fire1
▶ Fire2
▶ Fire3
▶ Jump
▶ Mouse X
▶ Mouse Y
▶ Mouse ScrollWheel
▶ Horizontal
▶ Vertical
▶ Fire1
▶ Fire2
▶ Fire3
▶ Jump

## Static Methods of the Input class

- Input.GetKey(KeyCode key)
  - Returns true as long as the identified key is held down
- Input.GetKeyDown(KeyCode key)
  - Returns true only on the frame that the key was pressed down
  - So this should *only* be used in Update( ), not in FixedUpdate( )
- Input.GetMouseButton(int button)
  - Returns true as long as the identified mouse button is held down (0=left, 1=right, 2=middle)
- Variations….
  - Input.GetKeyUp(KeyCode key)
  - Input.GetMouseButtonDown(int button)
  - Input.GetMouseButtonUp(int button)

# Static Member Properites of the Input class

- Input.compass
  - Returns data from the device's compass (if any)
- Input.gyro
  - Returns data from the device's gyroscope (if any)
- Input.location
  - Returns location data from device's GPS receiver (if any)
- Input.mousePosition
  - Returns as a Vector3 indicating pixel coordinates
  - The bottom-left of the screen or window is at (0, 0). The top-right of the screen or window is at (Screen.width, Screen.height).

# Input.mousePosition Example

- Iterate thru a list of objects and indicate which (if any) are under the mouse according to their **collider bounds**, by using a Coroutine to pulse their size (see also next slide)

```
public class CubeObject : MonoBehaviour {
    private bool isPulsing = false;
    public void StartPulsing() {
        if (!isPulsing)
            StartCoroutine(Pulse());
    }
    private IEnumerator Pulse() {
        isPulsing = true;
        float size = 2f;

        float angle = 0f;
        while (angle<Mathf.PI*2f) {
            angle += 10f*Time.deltaTime;
            size = 2f+Mathf.Sin(angle);
            transform.localScale = new Vector3(size,size,size);
            yield return null;
        }
        isPulsing = false;
    }
}
```

This script makes an object pulse its size when StartPulsing() is called

```
public class GameManager : MonoBehaviour {

    // inspector settings
    public GameObject templateGameObject; // prefab we want to makes instances of
    //

    private List<GameObject> gameObjects = new List<GameObject>(); // list of game objects we've spawned

    void Start () {
        // instantiate and position a bunch of game objects
        for (int i=0; i<20; i++) {
            GameObject go = Instantiate(templateGameObject);
            go.transform.position = new Vector3(Random.Range(-40f,40f), Random.Range(-20f,20f), 0f);
            gameObjects.Add(go);
        }

        // position the camera (to which this GameManager script is attached)
        this.transform.position = new Vector3(0f, 0f, 50f);
        this.transform.LookAt(Vector3.zero);
    }

    void Update () {
        // is any object under the mouse?.. First we need to turn mouse's screen position to a world position
        Vector3 mousePosOnScreen = Input.mousePosition; // 2d position on screen (pixels)
        // 50 units in front of camera, i.e. z=0 in the world based on camera position:0,0,50/lookat:0,0,0
        mousePosOnScreen.z = 50f;
        Vector3 mousePosInWorld = Camera.main.ScreenToWorldPoint(mousePosOnScreen);

        for (int i=0; i<gameObjects.Count; i++) {
            GameObject go = gameObjects[i];
            Collider c = go.GetComponent<Collider>();
            if (c.bounds.Contains(mousePosInWorld)) {
                go.GetComponent<CubeObject>().StartPulsing();
            }
        }
    }
}
```