UNIVERSITAT POLITÈCNICA DE VALÈNCIA
CAMPUS D'ALCOI

Collision detection

Jordi Linares

unity

OnCollisionEnter(), OnCollisionStay(), OnCollisionExit()

- When a **GameObject** has the **Rigidbody** component then it will be under the physics engine control (**Rigidbody2d** for 2D)
- In this way, this GameObject will properly react in front of collisions with other objects (if they have Colliders)
- Quite often it is interesting to receive a notification, an event, when this collision takes place in order to carry out a specific action (play a sound, for example)
- A frequent case consists of reacting differently depending on what are we colliding with (it is not the same a wall or an enemy)
- The method **OnCollisionEnter()** is invoked when a collision occurs, **OnCollisionStay()** while the collision is still taking place, and **OnCollisionExit()** when the collision finishes (in 2D we have the same methods but with the '2d' as suffix)
- A very common action will be to figure out with which object has collided with, being frequent using for this **Tags**. We can create tags and assign them to objects in order to group them in families depending on their type (enemies, walls etc.). In this way, it is common to check out if the Tag of the object we are colliding with is the one that triggers a specific action.

OnCollisionEnter(), OnCollisionStay(), OnCollisionExit()

- **Some rules regarding collisions:**

- If we want the object to be under the physic forces it must have a **Rigidbody**

-Detecting collisions means to have a **Collider** (it is also required if you want to detect the object using a raycast or methods such as OnMouseDown)

- Colliders generally has a different, simpler, geometry of the object (Mesh Filter) so the required calculation becomes also simpler.

- We can create Colliders as the sum of different Colliders, just by adding children with colliders to the GameObject

UNIVERSITAT POLITÈCNICA DE VALÈNCIA
CAMPUS D'ALCOI

Collision detection

Jordi Linares

unity