GameObjects and components

Jordi Linares

Gameobjects

- There are many **GameObjects** we can put in our scene from the menu of the **Unity** editor:

- **Empty**. It is a GameObject with just the compulsory Transform component and it is useful, for example, as a parent of a collection of GameObjects, for grouping objects and work with all of them together or for making some transformations easier. We can also use it just to add components and create a specific GameObject.

- Cube, sphere, capsule, cylinder, plane and quad. Simple objects we can create inside **Unity** (perfect for our first examples without having to import anything or for creating more complex elements transforming and combining them)

- **Lights** (directional, point, spotlight and area)

- **Cameras**. Besides the main camera, added by default in each scene, we can add several additional cameras, that we can enable/disable or make them render into different areas of the screen, for example.

- **Others**: interface elements (GUI), cloth, ragdolls, trees etc.

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CAMPUS D'ALCOI

Most important components

- We can add many different components to a GameObject, the main ones are:

- **Transform**. Always present by default, even in an empty GameObject, and holds position, rotation and scale of the object
- **Mesh Filter**. Stores the polygonal mesh of the GameObject, so its geometry
- **Mesh Renderer**. Its function is to define how the mesh is going to be rendered (painted), what materials and properties we will use.
- **Collider**. It's another geometry that our GameObject is going to use when colliding with other objects. For performance reasons this collider is usually simpler than the original geometry of the object.
- **RigidBody**. Required when we want our object to be under the control of the physics engine (gravity, reaction against collisions etc.)
- **Sound**. By using *AudioSource* (whenever the object can produce sounds) and *AudioListener* (where the 'mic' of the scene is)
- **Script**. It will control the behavior of the GameObject using code. We can add more than one to a GameObect.

- **Many more**: particle systems, animations, etc.

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CAMPUS D'ALCOI

More about components

- We can add any component to any object, enable/disable it, remove it, edit any of its properties in the inspector view, etc.

- From inside our scripts (code) we can also get access to GameObject's components, add them, modify them, consult or change their parameters etc.

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA
CAMPUS D'ALCOI

GameObjects and components

Jordi Linares