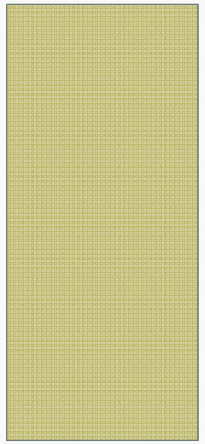
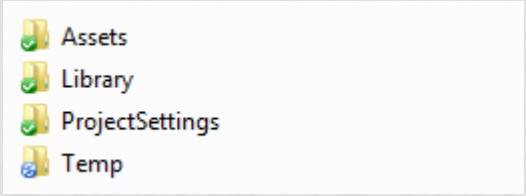
ANIMACIONS 3D, JOCS I ENTORNS INTERACTIUS

M7: DESENVOLUPAMENT D'ENTORNS INTERACTIUS MULTIDISPOSITIU I VIDEOJOCS

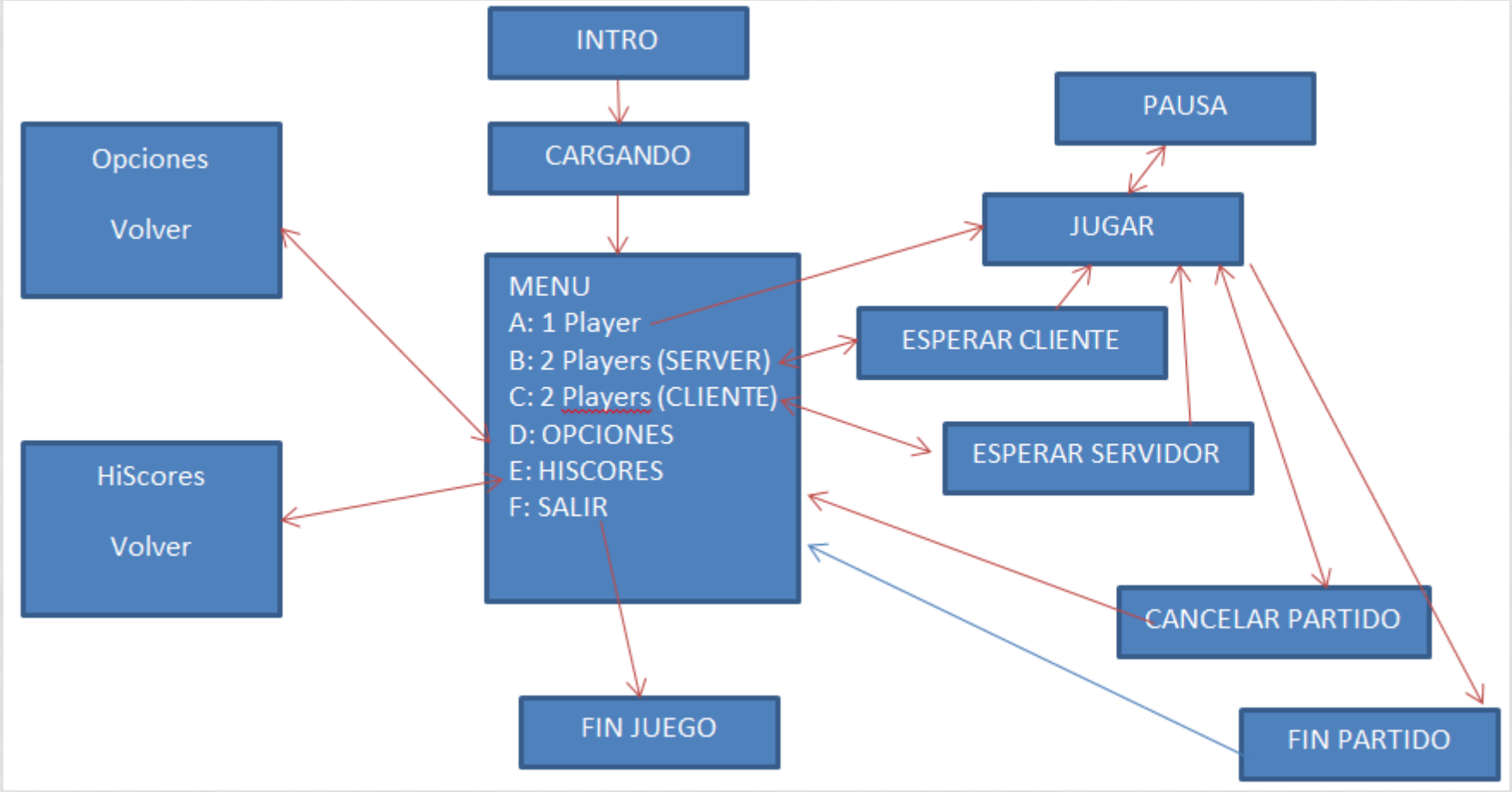


1. UNITY 3D: PRIMER CONTACTE

- Descàrrega i instal·lació Unity 3D
- Versió lliure vs versió pro
- Obrint Unity
- Els projectes
- Estructura a disc
- Les escenes (màquina finita d'estats)

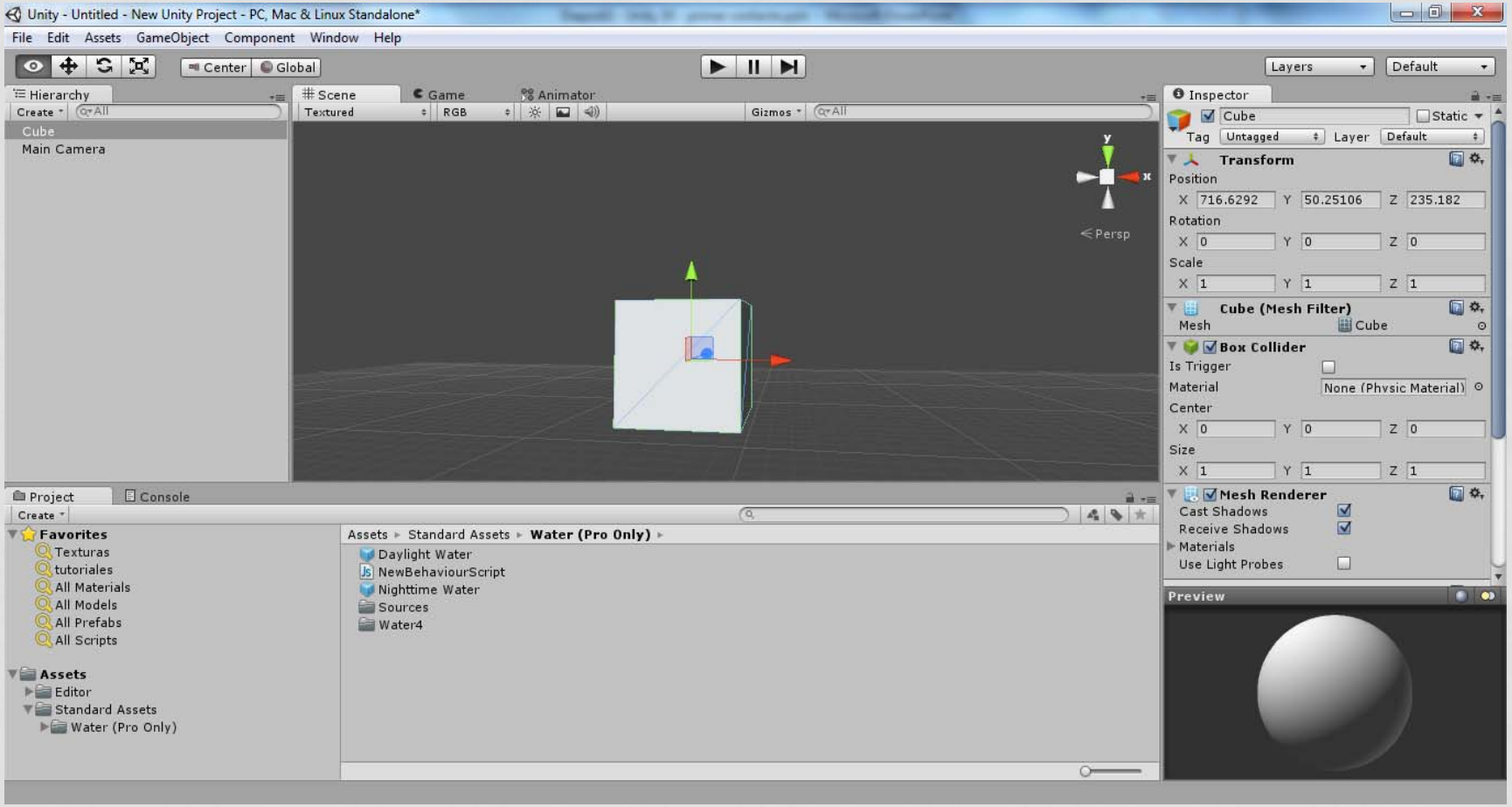


1. UNITY 3D: PRIMER CONTACTE



1. UNITY 3D: PRIMER CONTACTE

- Interfície



1. UNITY 3D: PRIMER CONTACTE

- Algunes hotkeys
  - Ctrl-1: vista Scene
  - Ctrl-2: vista Game
  - Ctrl 9: asset store
  - ESPACIO: maximitzar finestra
  - Ctrl-D: duplicar objecte
  - Ctrl-Sghift-F: encuadrar càmera (objecte) amb la vista
  - W, E, R: moure, escalar, rotar
- Roda mouse: zoom
- botó central: pan
- RMB: orbitar
- Shift-LMB: orbitar al voltant de l'objecte seleccionat
- 2x click (a hierarcchy): encuadrar objecte

1. UNITY 3D: PRIMER CONTACTE

- Àrees de pantalla
- Eixos de coordinades
- Vista perspectiva vs ortogonal
- Coordinades locals vs Globals
- Conceptes principals
  - Game objects: qualsevol objecte actiu a l'escena. Contenidor de components
  - Components: les peces que juntes conformen un game object. Poden ser audios, regles físiques, scripts...
  - Assets
  - Prefabs
  - Terrain
  - Scripting: mono developer

1. UNITY 3D: PRIMER CONTACTE

- Afegir game objects. Opcions bàsiques
  - Moviment
  - Rotació
  - Escala
  - Duplicat
- **Component** transform
- Previsualitzar projecte: RUN
- La càmera principal: viewport

1. PRIMER PROJECTE

- Afegirem, pas a pas:
  - Un cub. Què passa al executar? Encuadrant la imatge
  - Aixecar
  - Aplicar comportament rigid body
  - Posar-li un pla
  - Afegir una llum
  - Permetre moviment de càmera (component LookAt)
  - Afegir un sky-box
  - Ara traurem rigid body y el fem rotar. **Primer script**
    - Funcions Start() i Update()
    - Gir a velocitat constant
    - Fent servir variable *velocitat*
    - *Fent servir Time.deltaTime*

2. PRIMER PROJECTE

- Hem fet servir el component **Transform**.
- Ajuda de **Transform** al *Reference manual*

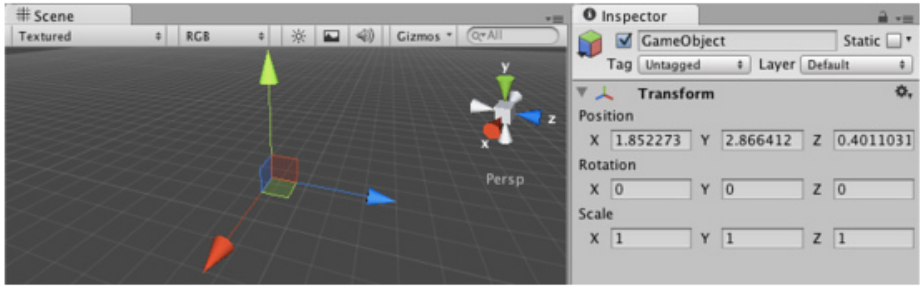
Transforms

Transforms are a key **Component** in every **GameObject**. They dictate where the GameObject is positioned, how it is rotated, and its scale. It is impossible to have a GameObject without a Transform. You can adjust the Transform of any GameObject from the **Scene View**, the **Inspector**, or through Scripting.

The remainder of this page's text is from the Transform Component Reference page.

Transform

The **Transform** component determines the **Position**, **Rotation**, and **Scale** of each object in the scene. Every object has a Transform.



The Transform Component is editable in the Scene View and in the Inspector

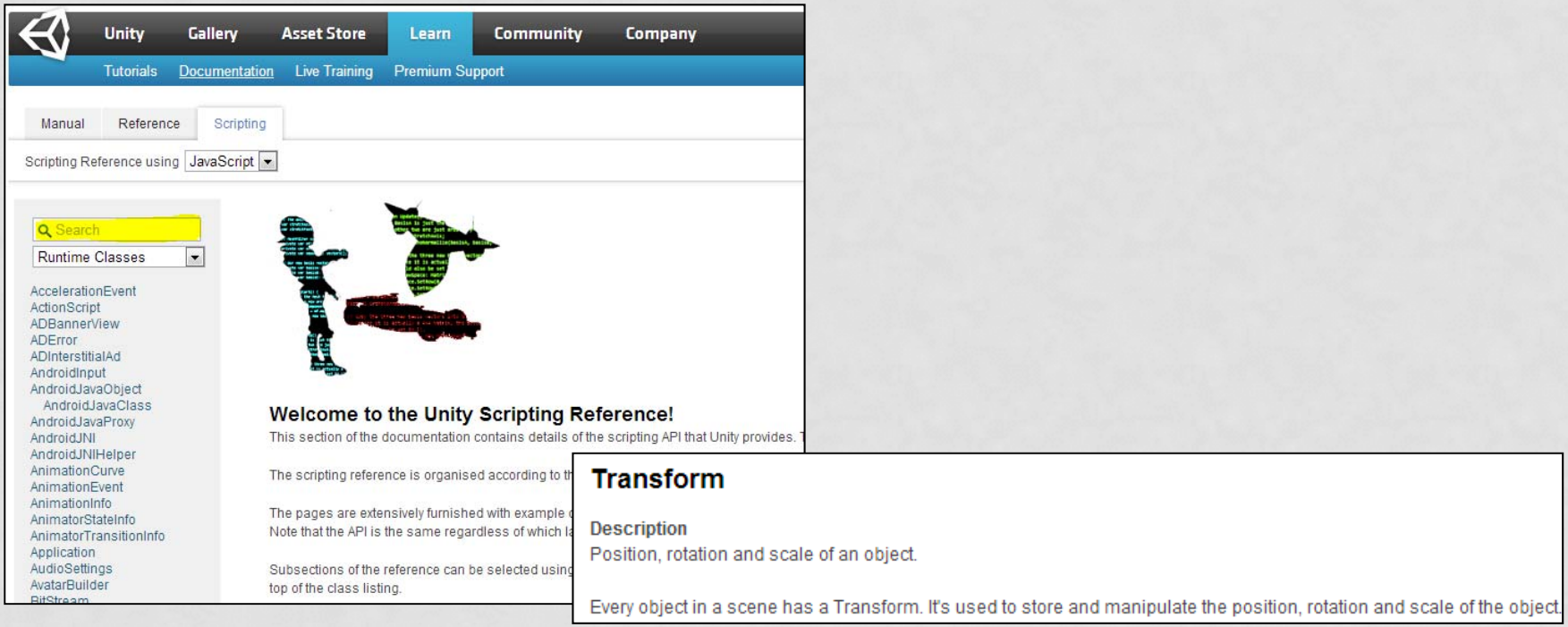
Properties

Position	Position of the Transform in X, Y, and Z coordinates.
Rotation	Rotation of the Transform around the X, Y, and Z axes, measured in degrees.
Scale	Scale of the Transform along X, Y, and Z axes. Value "1" is the original size (size at which the object was imported).

2. PRIMER PROJECTE

- Per buscar informació de les diferents classes existents a Unity, els seus atributs, mètodes, etc... accedim a:

  MENU : Help → Scripting Reference

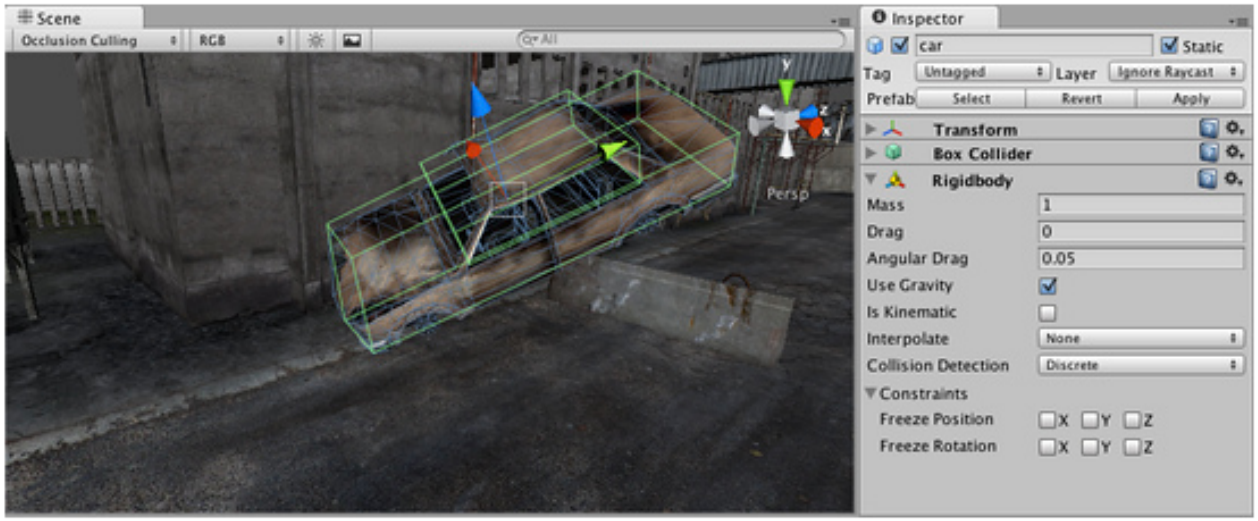


2. PRIMER PROJECTE

- També hem fet servir el comportament Rigid Body

Rigidbody

Rigidbodies enable your **GameObjects** to act under the control of physics. The Rigidbody can receive forces and torque to make your objects move in a realistic way. Any GameObject must contain a Rigidbody to be influenced by gravity, act under added forces via scripting, or interact with other objects through the NVIDIA PhysX physics engine.

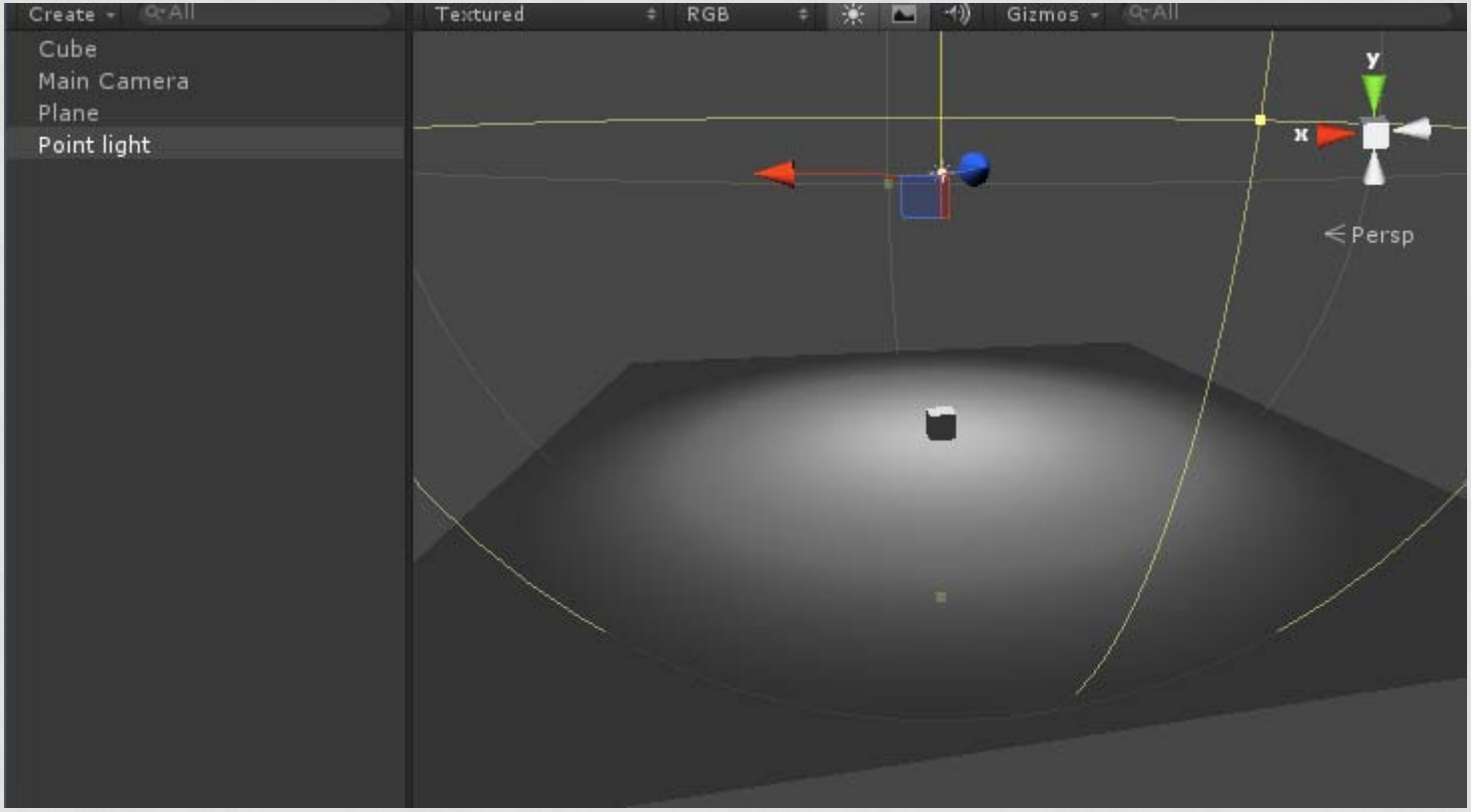


Rigidbodies allow GameObjects to act under physical influence

3. PRIMER JOC

1. Creació escenari amb un cub
2. Moviment del cub
3. Fent saltar al cub
4. Afegir pedestal. Importació des de 3ds max
5. Modificant les mecàniques de desplaçament i salt
6. Mostrar informació per pantalla
7. Finalitzar partida

3.1 CREACIÓ ESCENARI AMB UN CUB



3.2 MOVIMENT DEL CUB

Utilitzarem la classe **Input...**

Input

Description
Interface into the Input system.

Use this class to read the axes set up in the Input Manager, and to access multi-touch/accelerometer data on mobile devices.

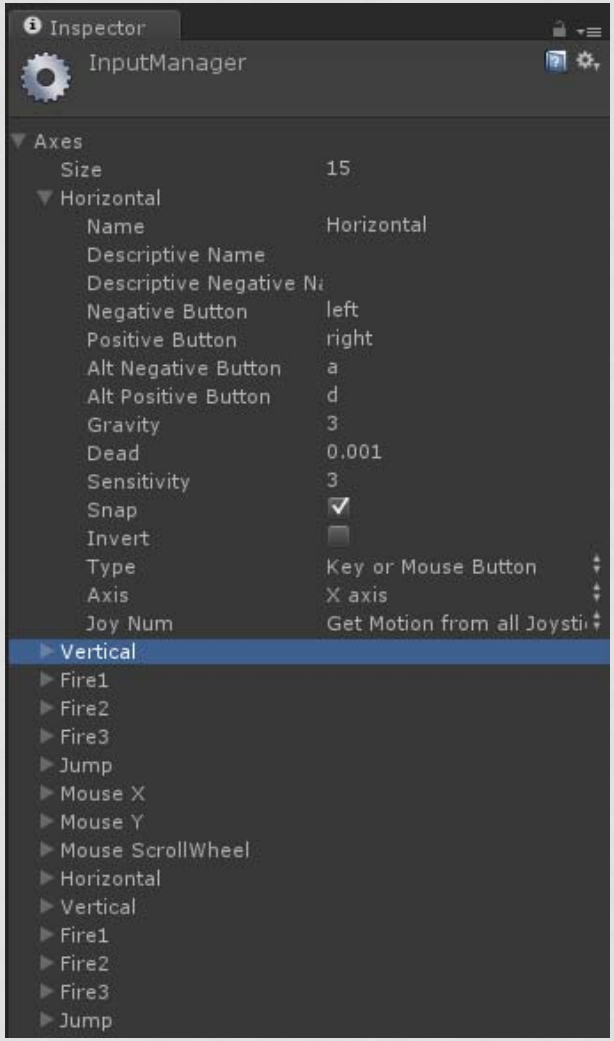
To read an axis use Input.GetAxis with one of the following default axes: "Horizontal" and "Vertical" are mapped to joystick, `A`, `W`, `S`, `D` and the arrow keys. "Mouse X" and "Mouse Y" are mapped to the mouse delta. "Fire1", "Fire2" "Fire3" are mapped to `Ctrl`, `Alt`, `Cmd` keys and three mouse or joystick buttons. New input axes can be added in the Input Manager.

If you are using input for any kind of movement behaviour use Input.GetAxis. It gives you smoothed and configurable input that can be mapped to keyboard, joystick or mouse. Use Input.GetButton for action like events only. Don't use it for movement, Input.GetAxis will make the script code smaller and simpler.

Note also that the Input flags are not reset until "Update()", so its suggested you make all the Input Calls in the Update Loop.

3.2 MOVIMENT DEL CUB

- Per treballa amb els control de joc, farem servir **Input.GetAxis**
- Per veure/modificar totes les possibilitats referents als controls disponibles, accedim a MENU : Edit → Project Settings → Input



3.2 MOVIMENT DEL CUB

També farem servir la funció **Translate** de la classe **Transform**

Transform.Translate

function **Translate**(**translation**: Vector3): void;
function **Translate**(**translation**: Vector3, **relativeTo**: Space): void;

Description
Moves the transform in the direction and distance of `translation`.

If `relativeTo` is left out or set to Space.Self the movement is applied relative to the tr
world coordinate system.

```
function Update() {
    // Move the object forward along its z axis 1 unit/second.
    transform.Translate(Vector3.forward * Time.deltaTime);

    // Move the object upward in world space 1 unit/second.
    transform.Translate(Vector3.up * Time.deltaTime, Space.World);
}
```

function **Translate**(**x**: float, **y**: float, **z**: float): void;
function **Translate**(**x**: float, **y**: float, **z**: float, **relativeTo**: Space): void;

3.2 MOVIMENT DEL CUB

- Versió 1: es mou amb velocitat que dependrá dels fps:

```
function Update () {
    transform.Translate(Input.GetAxis("Horizontal"),0,Input.GetAxis("Vertical"));
}
```

- Versió 2: independent dels fps (**Time.deltaTime**):

```
function Update () {
   var despX:float;
   var despZ:float;
   despX=Input.GetAxis("Horizontal")*Time.deltaTime*100;
   despZ=Input.GetAxis("Vertical")  *Time.deltaTime*100;
   transform.Translate(despX,0,despZ);
}
```

3.2 MOVIMENT DEL CUB

Time

Description

The interface to get time information from Unity.

Static Variables

captureFramerate	Slows game playback time to allow screenshots to be saved between frames.
deltaTime	The time in seconds it took to complete the last frame (Read Only).
fixedDeltaTime	The interval in seconds at which physics and other fixed frame rate updates (like MonoBehaviour's FixedUpdate) are performed.
fixedTime	The time the latest FixedUpdate has started (Read Only). This is the time in seconds since the start of the game.
frameCount	The total number of frames that have passed (Read Only).
maximumDeltaTime	The maximum time a frame can take. Physics and other fixed frame rate updates (like MonoBehaviour's FixedUpdate).
realtimeSinceStartup	The real time in seconds since the game started (Read Only).
smoothDeltaTime	A smoothed out Time.deltaTime (Read Only).
time	The time this frame has started (Read Only). This is the time in seconds since the start of the game.
timeScale	The scale at which the time is passing. This can be used for slow motion effects.
timeSinceLevelLoad	The time this frame has started (Read Only). This is the time in seconds since the last level has been loaded.

3.2 MOVIMENT DEL CUB

1. Versió 3: afegint la variable velocitat

```
var velocitat:int=50;
function Update () {
  var despX:float;
  var despZ:float;

  despX=Input.GetAxis("Horizontal")*Time.deltaTime*velocitat;
  despZ=Input.GetAxis("Vertical")  *Time.deltaTime*velocitat;
  transform.Translate(despX,0,despZ);
}
```