



UNIVERSIDAD NACIONAL DEL LITORAL
FACULTAD DE INGENIERÍA Y CIENCIAS HÍDRICAS



**Tecnicatura en diseño
y programación de videojuegos**

UNL VIRTUAL



Manipulación de objetos en 3D

Unidad 4: Carga de Niveles

Docentes
Jaime Fili
Nicolás Kreiff

Contenidos

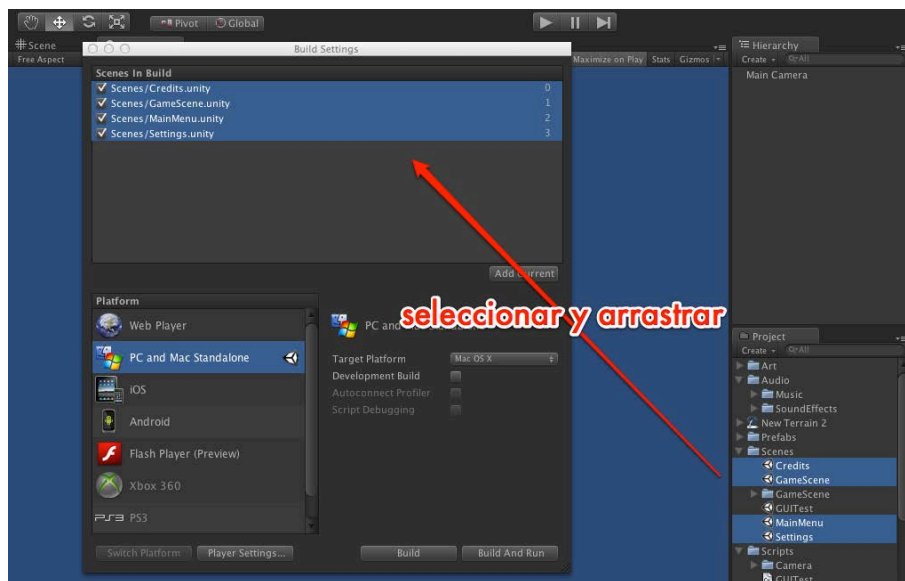
Carga de Niveles	3
Carga de Niveles con Application.LoadLevel	3
Carga de Escenas Aditivas con Application.LoadLevelAdditive	4
Carga Asincrónica de Niveles	5
Carga Asincrónica de Niveles Aditivos	5

Carga de Niveles

Carga de Niveles con Application.LoadLevel

La función Application.LoadLevel carga los niveles por su nombre o índice en el proyecto. En el contexto de Unity los niveles están representados por las escenas (scenes).

Antes, dichos niveles se deben agregar a los utilizados en el juego; para ello se usa la opción *File >> Build Settings* en Unity, que permite agregar las escenas que se necesiten a la lista de niveles.

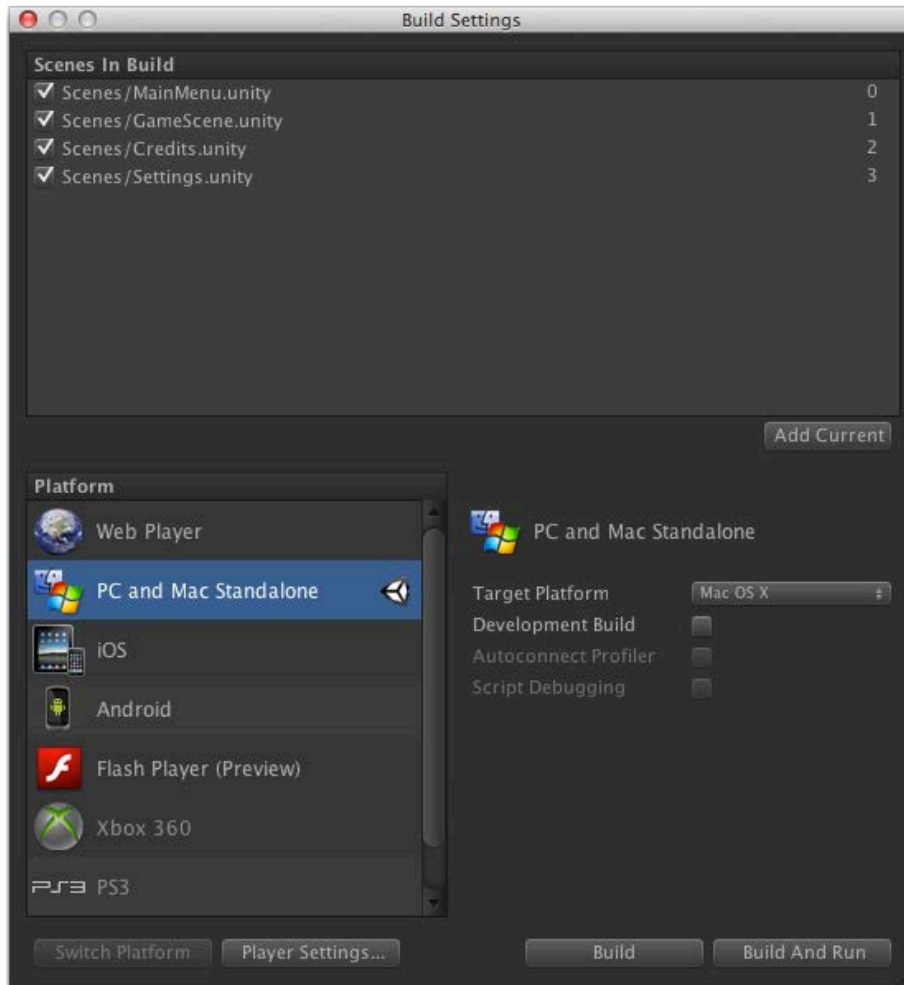


El siguiente ejemplo muestra la forma de utilizar Application.LoadLevel()

```
01 using UnityEngine;
02 using System.Collections;
03
04 public class example : MonoBehaviour {
05     void Example() {
06         Application.LoadLevel("HighScore");
07     }
08 }
```

Cuando se carga un nuevo nivel todos los GameObjects que han sido cargados antes se destruyen. Si se desea que un objeto sobreviva cuando se carga un nuevo nivel se debe usar Object.DontDestroyOnLoad.

También hay que tener en cuenta que cuando la aplicación se inicia, se carga la primera escena que aparece en la lista de niveles. No importa el orden de las escenas, siempre y cuando la escena que se desea cargar como inicial esté primera en la lista.



Carga de Escenas Aditivas con Application.LoadLevelAdditive

Esta función carga una escena aditivamente. Su diferencia con respecto a LoadLevel es que LoadLevelAdditive, no destruye los objetos en el nivel que está siendo utilizado; los objetos del nuevo nivel son agregados a la escena actual. Esto es muy útil, por ejemplo, para la creación de mundos virtuales continuos donde el contenido se va cargando a medida que se avanza por el escenario. Lo que sigue es un ejemplo de carga de escenas aditivas:

```
01 using UnityEngine;
02 using System.Collections;
03
04 public class example : MonoBehaviour {
05     void Example() {
06         Application.LoadLevelAdditive(1);
07         Application.LoadLevel("MoreStuff");
08     }
09 }
```

Carga Asincrónica de Niveles

Esta función de Unity PRO, llamada `LoadLevelAsync`, carga el nivel de fondo para que actúe asincrónicamente¹.

Unity carga completamente todos los assets y objetos de la escena en un hilo de fondo. Esto permite cargar nuevos niveles mientras se juega el nivel actual, se muestra una barra de progreso o se crea un mundo completo ya que brinda la posibilidad de realizar un *streaming* donde constantemente se cargan y descargan diferentes partes basadas en la posición del jugador, sin alterar el flujo del juego.

Para saber si un nivel se ha terminado de cargar, se puede utilizar la variable `isDone` resultante de la opción `AsyncOperation`. El resultado de un `LoadLevelAsync` puede ser también usado por una corrutina (este término puede consultarse en <http://en.wikipedia.org/wiki/Coroutine>).

Cuando se construye el *player*, Unity optimiza en forma automática los assets de tal manera que `LoadLevelAsync` los carga linealmente del disco para evitar tiempos de búsqueda.

Debe tenerse en cuenta que la performance de la carga de fondo en el editor de Unity es mucho más lenta que en el *web player* o en un paquete *standalone*, y que también se pueden tener más interrupciones de carga que en el *player*.

Esto puede verse con un ejemplo:

```
01 using UnityEngine;
02 using System.Collections;
03
04 public class example : MonoBehaviour {
05     IEnumerator Start() {
06         AsyncOperation async = Application.LoadLevelAsync("MyBigLevel");
07         yield return async;
08         Debug.Log("Loading complete");
09     }
10 }
```

Carga Asincrónica de Niveles Aditivos

Esta función de Unity PRO carga las escenas aditivamente y asincrónicamente en el fondo.

A diferencia del anterior, `LoadLevelAdditiveAsync` no destruye los objetos en el nivel actual; los objetos del nuevo nivel son agregados a la escena actual. El resto de su comportamiento es igual al de `LoadLevelAsync`. Su esquema es también similar:

```
01 using UnityEngine;
02 using System.Collections;
03
04 public class example : MonoBehaviour {
05     IEnumerator Start() {
06         AsyncOperation async =
07         Application.LoadLevelAdditiveAsync("MyAddLevel");
08         yield return async;
09         Debug.Log("Loading complete");
10     }
11 }
```

¹ Asincrónico es todo aquel suceso que tiene lugar sin total correspondencia temporal con otro suceso.