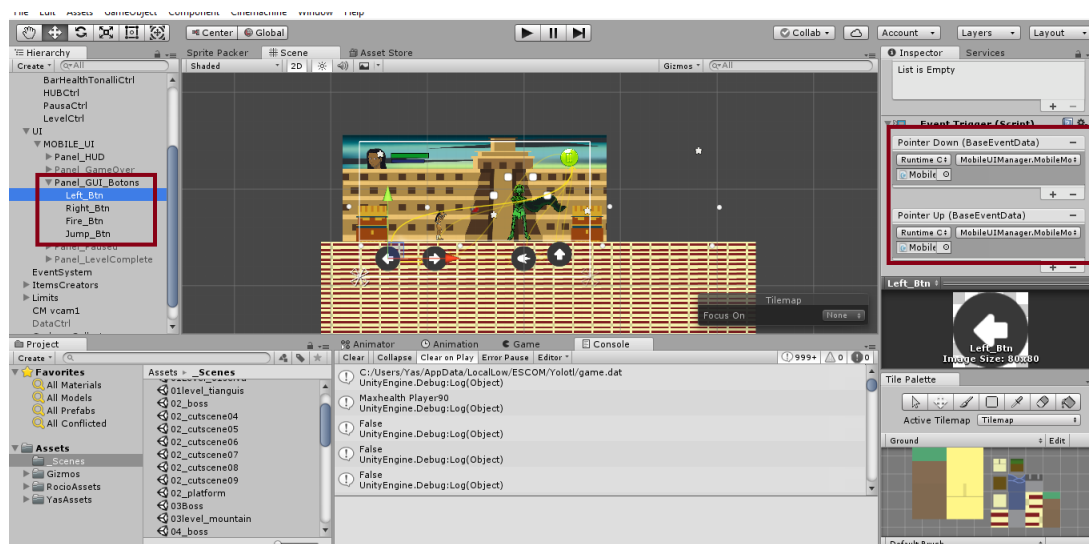
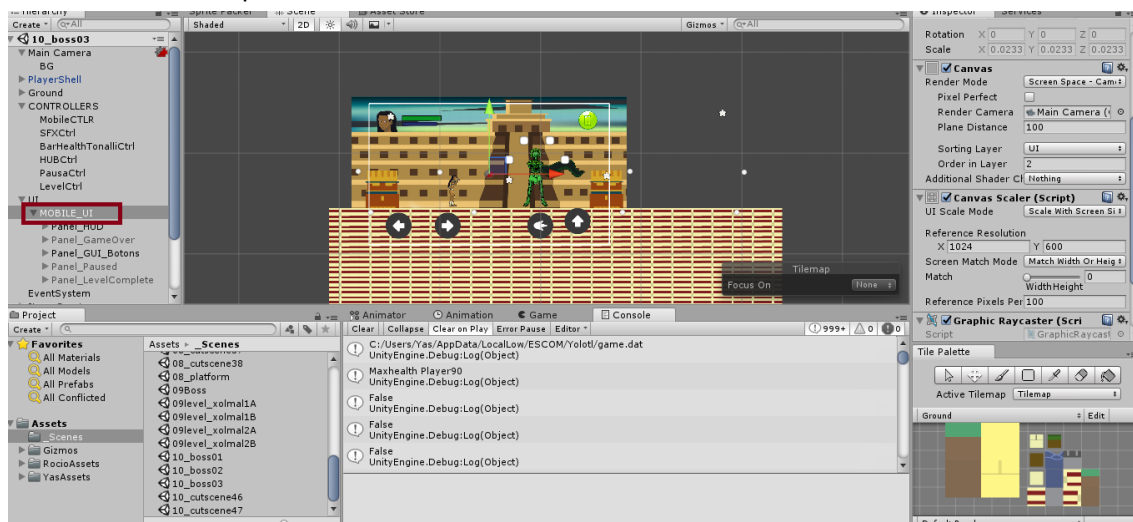


## Reconfiguración después de la integración.

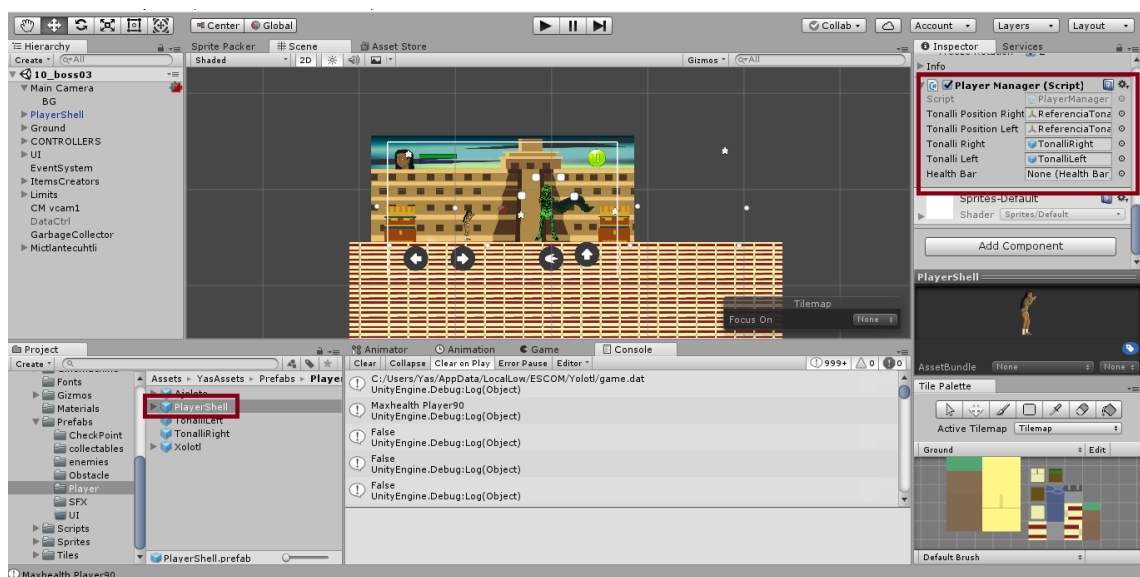
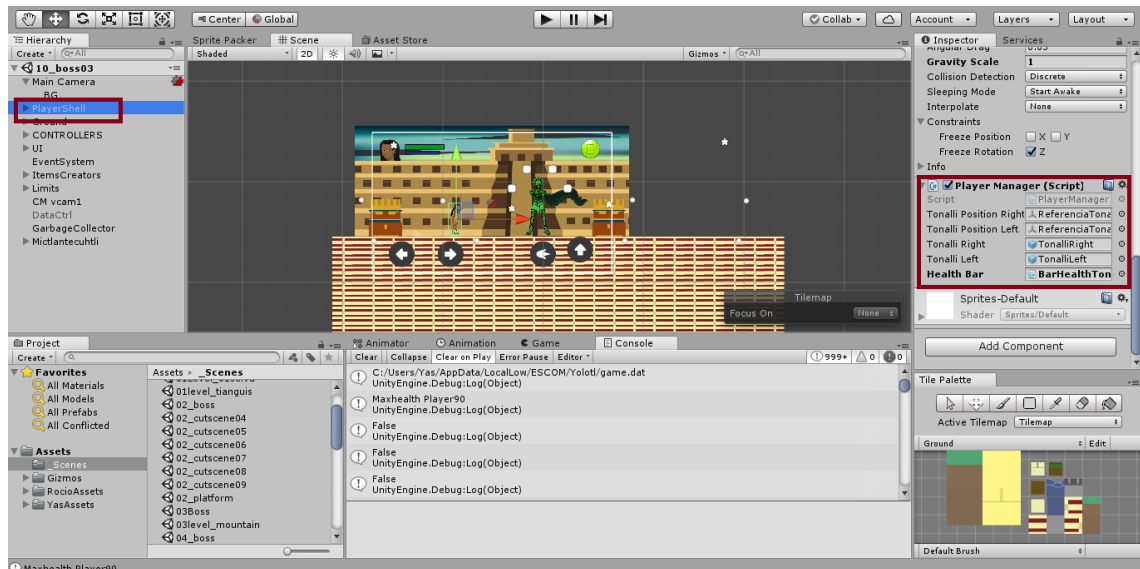
Cuando se integran proyectos de Unity con uso de paquetes en algunas ocasiones se pueden llegar a perder las referencias de sus objetos. En esta guía se ofrece una ayuda para verificar la funcionalidad de sus objetos.

Pasos para verificar la funcionalidad de sus escenas una vez se ha completado la integración:

1. Compruebe las referencias de la GUI, este es una de las referencias que más comúnmente se pierden:



2. Compruebe las referencias a atributos públicos dentro de sus GameObjects, si ese objeto es instanciado desde un prefab, corrija directamente la referencia desde el prefab y Unity actualizará todos los objetos que se deriven del prefab en todas sus escenas.

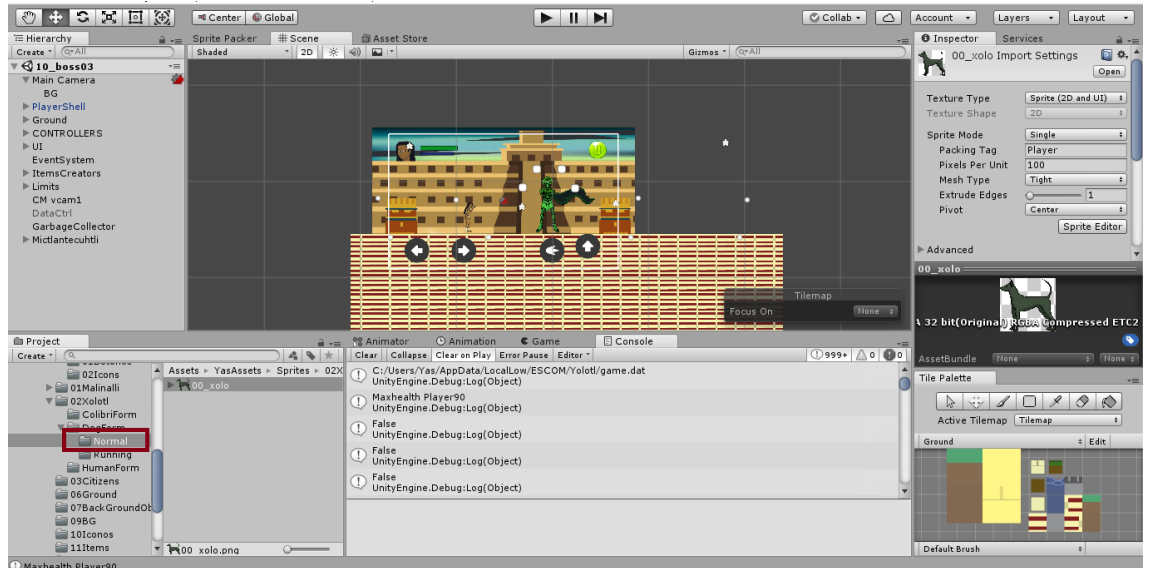


3. En caso de que no recuerde la configuración inicial. Puede abrir de manera paralela el proyecto original y comparar la configuración de su escena.

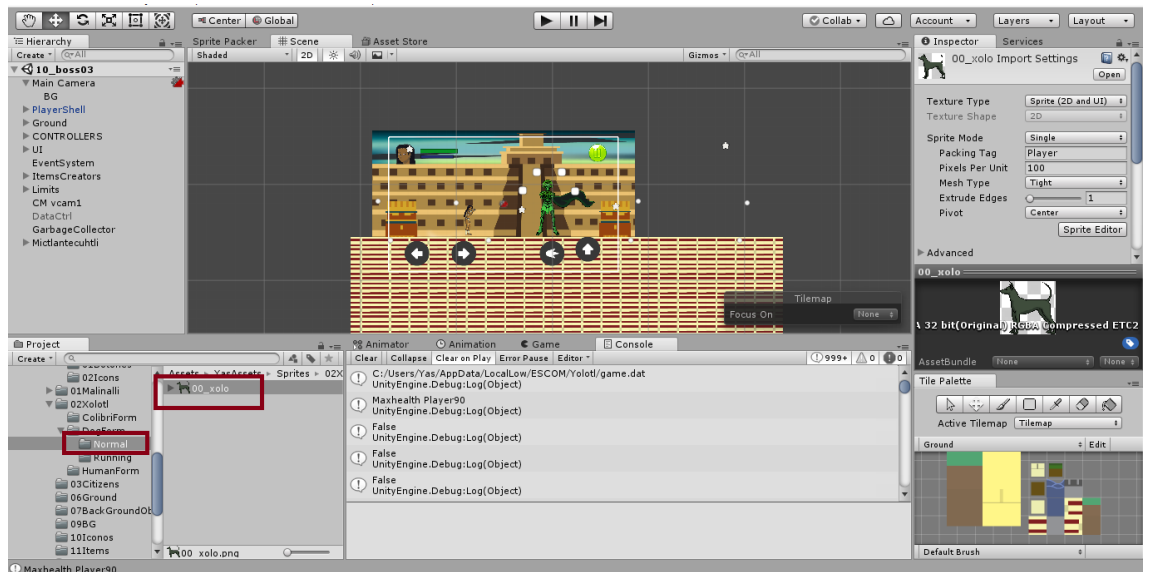
## Optimizaciones importantes

Para mejorar el rendimiento de la aplicación en cuanto a tiempos de render es necesario empaquetar los Sprites que utilice. A continuación, se presentan los pasos de empaquetado de sprites:

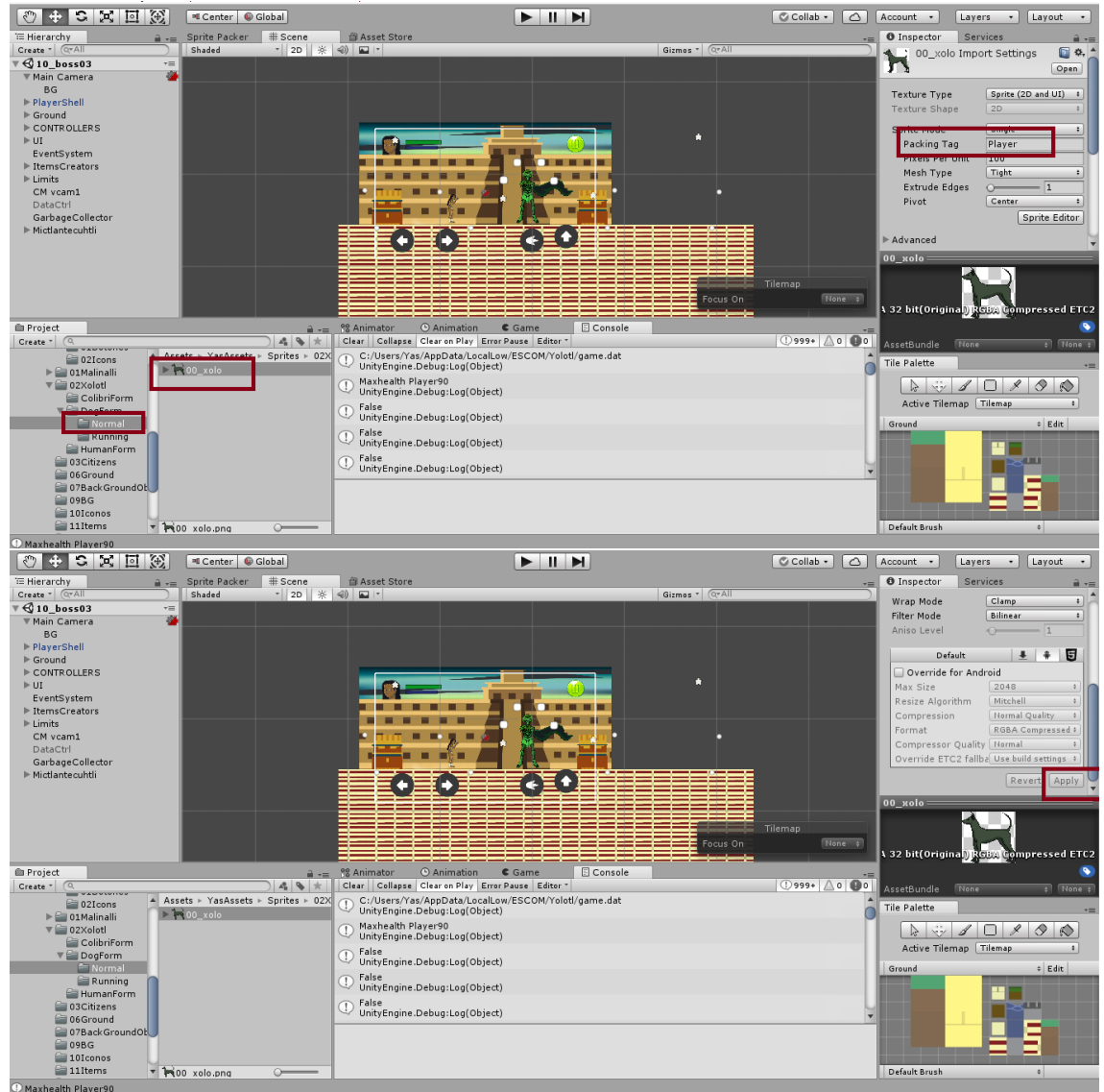
1. Diríjase a la carpeta donde tiene sus sprites.



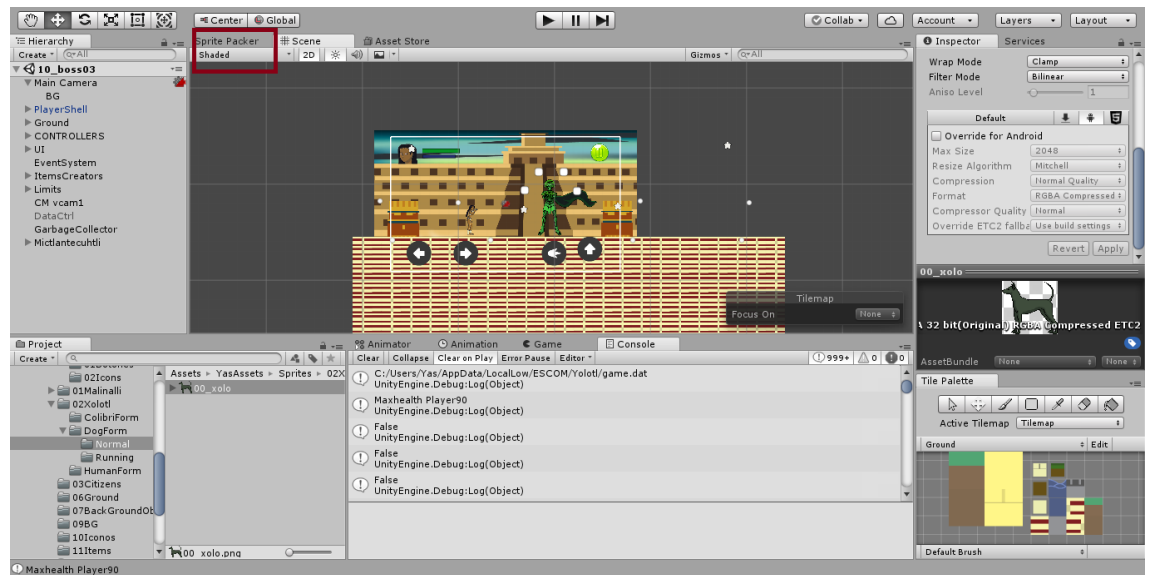
2. Seleccione su Sprite (también puede seleccionar de manera multiple)



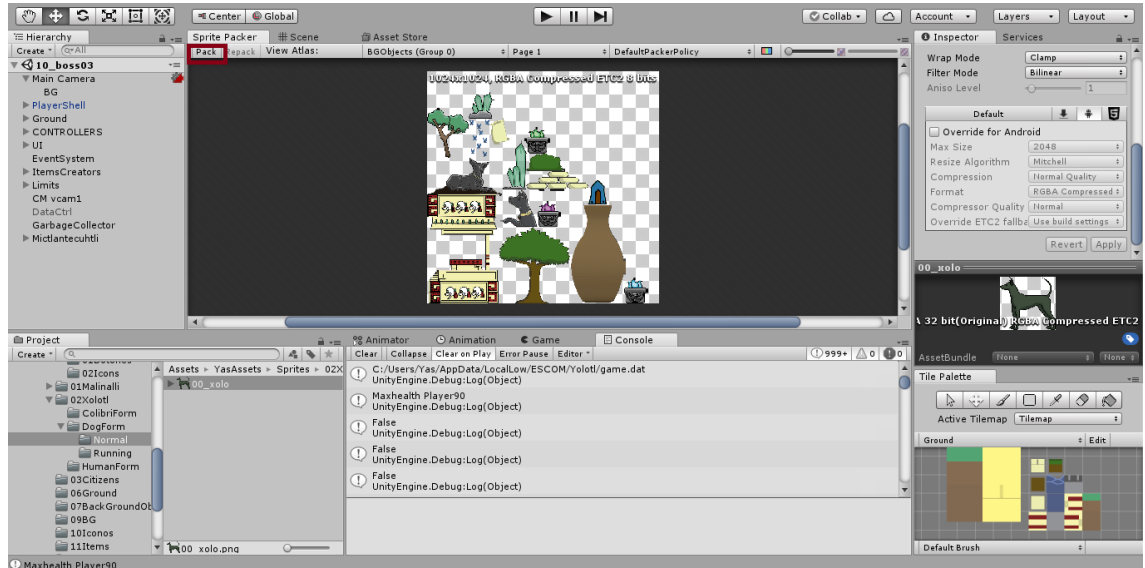
3. Asigne un nombre a su paquete y de la opción de aplicar.



4. Esto no es suficiente para empaquetar los Sprites. Diríjase a la pestaña de Sprite Packer



5. De click en pack, esto generara sus paquetes de sprites



6. El criterio para generar paquetes varía de persona a persona pero se recomienda que se generen los paquetes con base a la utilidad de los sprites ejemplo un paquete para los sprites de enemigos, otro para objetos de fondo, otros para la GUI, etc. Para que diferentes sprites entren en un mismo paquete deben de tener el mismo valor en la opción de Packing Tag del punto 1.

## Salvar datos de la partida.

El manejo de datos del juego se hace a partir de la clase `GameDataCtrl`, esta clase funciona con el paradigma de diseño de un **singleton**. Por lo que para el correcto funcionamiento del juego durante pruebas desde computadora es necesario siempre iniciar la prueba desde la escena **00\_MenuPrincipal**. Es importante no agregar una instancia de esta clase a ninguna de las escenas salvo la escena **00\_MenuPrincipal**.

1. Agregue la siguiente línea de código en donde realice las transiciones de sus escenas:
  - a. Escena 01 tianguis  
`GameDataCtrl.instance.SaveData (50f, 30f, 5f, 0, 1);`
  - b. Escena 01 selva  
`GameDataCtrl.instance.SaveData (50f, 30f, 5f, 0, 2);`
  - c. Escena 03 plataforma  
`GameDataCtrl.instance.SaveData (60f, 30f, 10f, 0, 5);`
  - d. Escena 03 boss  
`GameDataCtrl.instance.SaveData (60f, 40f, 10f, 0, 6);`
  - e. Escena 05 plataforma  
`GameDataCtrl.instance.SaveData (70f, 40f, 15f, 0, 9);`
  - f. Escena 05 boss  
`GameDataCtrl.instance.SaveData (70f, 50f, 15f, 0, 10);`
  - g. Escena 07 plataforma  
`GameDataCtrl.instance.SaveData (80f, 50f, 20f, 0, 13);`
  - h. Escena 07 boss  
`GameDataCtrl.instance.SaveData (80f, 60f, 20f, 0, 14);`
  - i. Escena 09 plataforma A  
`GameDataCtrl.instance.SaveData (80f, 60f, 25f, 0, 17);`

- j. Escena 09 plataforma B  
GameDataCtrl.instance.SaveData (80f, 60f, 25f, 0, 18);
  - k. Escena 09 boss  
GameDataCtrl.instance.SaveData (90f, 60f, 25f, 0, 19);
2. Se recomienda que se sigan esos valores de lo contrario el juego no podrá funcionar correctamente.