

Un Sistema de Guardado

÷Unity 2D÷

Por: Nadia Gonzalez

÷Programación para Videojuegos÷

Universidad Tecnológica Americana:
UTECA



ÍNDICE:

Contexto:.....	3
Consideraciones:.....	4
Implementación:	5
Contenido:	6
¿Cómo puedo modificarlo en Unity ?	10
Cámara:	10
El jugador:.....	11
El "Controlador de Datos":	11
El Canvas:	12
Contenedor de Modificadores:	13

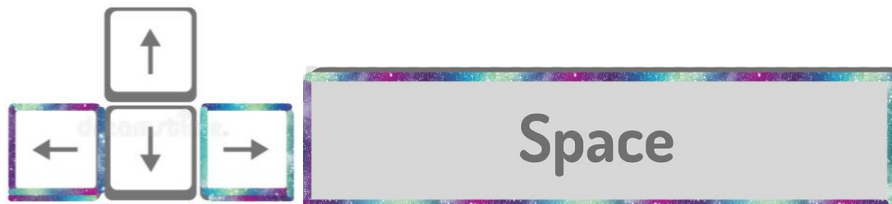


Contexto:

Este sistema es una herramienta que guardará diferentes datos de tu juego, el cual funciona con Visual Studio y Unity 2D.

Para este sistema se utilizó la herramienta de Slider (herramienta que nos proporciona Unity) como barras de vida, armadura y energía. Por otro lado, también se utiliza la herramienta de Text Mesh Pro para mostrar la posición de nuestro personaje en pantalla. Por último, otro de los elementos que se utilizaron fueron botones que unity proporciona (cada uno de los elementos puedes personalizarlos a tu gusto).

Este sistema proporciona un sistema básico de movimiento para nuestro personaje, el cual es importante para que el sistema también funcione (pues guarda también la posición del personaje) y funciona con las tecla de espacio y flechas de izquierda y derecha.



En pocas palabras, en este Sistema de Guardado podrás guardar elementos como:

- ✓ Barra de vida
- ✓ Barra de energía
- ✓ Barra de armadura
- ✓ Posición de nuestro personaje.

Dicho sistema funciona en base a las teclas **G** y **C**.

G: Guardar los datos.

C: Cargar los datos.

Consideraciones:

- ✓ Este proyecto trabaja en base a archivos **Json**
- ✓ Recuerda agregar los diferentes elementos correspondientes en cada una de las casillas nuevas que proporcionen los scripts en unity (una vez asignados los scripts a los objetos como jugador, etc).
- ✓ Recuerda que debes tener instalada o instalar la paquetería de **Cinemachine** (para que la cámara proporcionada funcione bien) y **Text Mesh Pro** para que se pueda mostrar la posición de personaje en pantalla (pues en uno de los scripts se creará una variable para colocar dicho texto).
- ✓ Para este proyecto se descargaron otros proyectos (para el uso de Sprites y ambientación) de la Asset Store (en Assets aparecen como **BayatGames** y **Simple 2D Platformer BE2**, pues en el caso de no quererlos en tu proyecto eres libre de eliminarlos, pero esto podría afectar al lado visual del proyecto (en dado caso, los materiales utilizados están convertidos en prefabs).

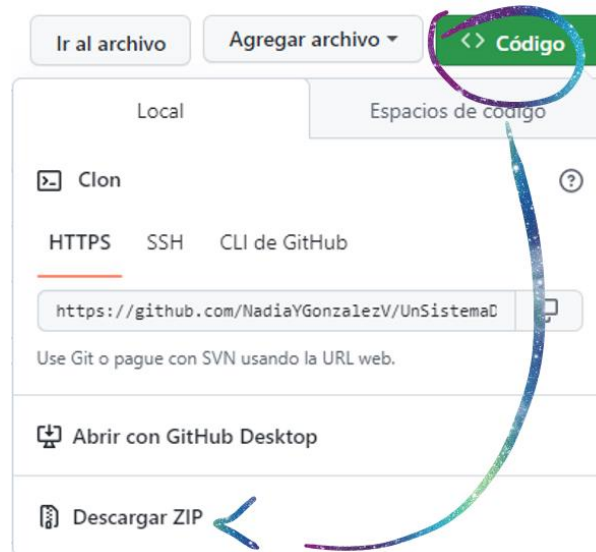


Implementación:

- ✓ Deberás descargar todo el proyecto para que funcione correctamente en Unity desde GitHub, el repositorio con el sistema esta en el siguiente enlace:

<https://github.com/NadiaYGonzalezV/UnSistemaDeGuardado.git>

- ✓ Una vez que te encuentres en el repositorio, le darás clic al botón de “Código” y posterior mente al de “Descargar ZIP”



- ✓ Una vez descargado el archivo lo ubicas en tu dispositivo (normalmente en la carpeta de descargas)
- ✓ En la carpeta donde se encuentre el ZIP descargado , crearas una nueva carpeta para mover el Zip dentro de ella (esto para tener orden,opcional).
- ✓ Una vez dentro de la nueva carpeta ,extraeras los archivos de la crpeta dando clic derecho y en “Extraer aqui”
- ✓ Listo, tendras acceso a todos y cada uno de los archivos del proyecto, junto al Unity package file.
- ✓ Si quieres probar la herramienta, se encontrara en la carpeta de “Ejecutables” y contendra este logo:



Contenido:

Este documento te ayudará a crear un sistema de guardado para tu juego basado en archivos Json.

En este sistema se proporcionan varios elementos:

- ✓ 11 scripts que serán los responsables de que nuestro sistema funcione la perfección:
 - 2 scripts que tendrán la función de guardar los datos (En la carpeta de Scripts aparecerán con el nombre de “ControladorDatos” “DatosGuardados”)
 - 2 Scripts que mostrarán la posición y el aviso de que los datos han sido guardados o cargados (En la carpeta de Scripts aparecerán con el nombre de “AvisoDeGuardadoCarga” y “MostrarElementos”)
 - 3 scripts sobre los elementos a mostrar en pantalla con Slider (En la carpeta de Scripts aparecerán con el nombre de “ArmaduraJugador” “EnergíaJugador” y “VidaJugador”)
 - 3 scripts para modificar cada uno de los elementos anteriores (En la carpeta de Scripts aparecerán con el nombre de “ModificarE” “ModificarVida” “ModificarSlideAr”)
 - Por último, un script que le otorgará movimiento a nuestro personaje.

Nota: Los scripts de los elementos de armadura, vida y energía se pueden colocar en un solo script, por si no te gusta tener tantas ventanas de código abiertas en tu visual studio.

- ✓ Para este proyecto se utilizaron dos paquetes de la Asset Store para la ambientación (como mencionábamos anterior mente, BayatGames y Simple 2D Platformer BE2)
- ✓ Se proporcionan los Prefabs de todos los elementos utilizados.

A continuación se te presentarán los diferentes códigos (de forma resumida) que podrás encontrar en este sistema (todos los códigos en visual studio se encuentran comentados):

-Primer script que guardará los datos-

```
public class DatosGuardados
{
    //aqui son todos los datos que seran guardados en Json
    public Vector3 posicion;
    public int Vida;
    ... }
```

-Segundo script que tendrá la función de guardar y cargar los datos-

```
using UnityEngine;
using System.IO; //Recuerda agregar esta funcion para que tu sistema funcione
bien.
public class ControladorDatos : MonoBehaviour
{
    public GameObject jugador; //objeto a quien le guardaran los datos
    public string archivosdeGuardado;
    //Nombre del script de "datosguardados"
    public DatosGuardados datosjuego = new DatosGuardados();

    private void Awake()
    {
        archivosdeGuardado = Application.dataPath + "/datosJuego.json"; //como
se guardaran los datos
        jugador = GameObject.FindGameObjectWithTag("player");
        //ayudara a que cuando inicie el juego nuestro personaje aparecera en
la ultima posicion guardada
        CargarDatos();
    }
    private void Update()
    //las teclas que haran la funcion de guardado y carga
    {
        ....
    }

    //Los datos que seran guardados
    private void CargarDatos()
    {
        if (File.Exists(archivosdeGuardado))
        {
            //ayuda a actualizar la posicion de nuestro jugador cuando le damos
la tecla C (de cargar)
            jugador.transform.position = datosjuego.posicion;

            //Asi mismo, los siguientes parrafos cargaran los datos de vida,
armadura y energia
        }
        else
        {
            //en el caso de no haber guardado nada aparecera el siguiente
mensaje en la consola
            Debug.Log("El archivo no existe");
        }
    }
    //Aqui sera la forma en la que se guardaran los datos de todos nuestros
elementos
    private void GuardarDatos()
    {
        //Aqui sera la forma en la que se guardaran los datos de todos nuestros
elementos

        }; ...
    }
    //Se guardan en Json y aparecera un mensaje que aparecera al momento de guardar
los datos y haberlos modificado
}
```

```
//Este script nos ayuda a que cada vez que guardemos nuestro personaje
repareciera en dicha posicion donde guardamos
//pues al momento de cargar esa posicion se respetara y nuestro personaje
siempre reaparecera en los puntos donde guardemos .
```

-Script de elementos a mostrar en pantalla, este código es el mismo para los 3 elementos (armadura, vida y energía) en este caso , se muestra el script de Armadura-

```
using UnityEngine;
using UnityEngine.UI; //recuerda agregar esta fucion para que todo vaya bien.

public class ArmaduraJugador : MonoBehaviour
{
    public int CantidadArmadura; //Cantidad que se podra visualizar y modificar
    desde el inspector
    public Slider barradearmadura; //lugar donde se asignara el Slider en unity

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        barradearmadura.value = CantidadArmadura; //Sirve para conectar el
        slider con la cantidad que buscas representar en pantalla
        //con este mismo Slider.

        //Este ultimo fragmento de codigo convertido en comentario sirve para
        que al momento de llegar a cero, tu personaje desaparezca
        //puedes implementarlo en lgun elemento que necesites.

        //if (CantidadArmadura <= 0)
        //{
        //    Destroy(gameObject);
        //}
    }
}
```

-Este script nos mostrará el aviso de que los archivos fueron guardados o cargados-

```
public class AvisoDeGuardadoCarga : MonoBehaviour
{
    public KeyCode tecla1; // La primera tecla que activa el primer sprite
    public KeyCode tecla2; // La segunda tecla que activa el segundo sprite
    public SpriteRenderer spriteRenderer1; // El objeto SpriteRenderer que
    muestra el primer sprite
    public SpriteRenderer spriteRenderer2; // El objeto SpriteRenderer que
    muestra el segundo sprite
    public bool spritesVisiblesAlInicio = false; // Indica si los sprites están
    visibles al inicio
```



```

void Start()
{
    ... // Configurar la visibilidad del primer sprite al inicio
    ... // Configurar la visibilidad del segundo sprite al inicio
}

void Update()
{
    if ... // Si la primera tecla se presiona
    {
        ... // Activar el primer sprite
    }

    if (Input.GetKeyUp(tecla1)) // Si la primera tecla se deja de presionar
    {
        spriteRender1.enabled = false; // Desactivar el primer sprite
    }

    if (Input.GetKeyDown(tecla2)) // Si la segunda tecla se presiona
    {
        spriteRender2.enabled = true; // Activar el segundo sprite
    }

    if (Input.GetKeyUp(tecla2)) // Si la segunda tecla se deja de presionar
    {
        spriteRender2.enabled = false; // Desactivar el segundo sprite
    }
}
}

```

-Este script es el encargado de modificar en pantalla nuestras barras de vida, energía y escudo, por lo que el script es el mismo para todos estos elementos, pero aquí te mostraré el de Armadura-

```

using UnityEngine;
using UnityEngine.UI;
public class ModificarSlideAr : MonoBehaviour
{
    public GameObject objetoModificadorA; // referencia al objeto con el script
    ArmaduraJugador
    public Slider barradearmadura; // referencia al slider en el script de
    ArmaduraJugador
    public GameObject botonAumentar; //botones para aumentar y disminuir en
    pantalla
    public GameObject botonReducir;

    public void AumentarArmadura()
    {
        objetoModificadorA.GetComponent<ArmaduraJugador>().CantidadArmadura +=
1; // aumenta la cantidad de armadura
        barradearmadura.value =
objetoModificadorA.GetComponent<ArmaduraJugador>().CantidadArmadura; //
actualiza el valor del slider
    }
    public void ReducirArmadura()
    {
        ... // reduce la cantidad de armadura
        ... // actualiza el valor del slider
    }
}

```

¿Cómo puedo modificarlo en Unity ?

Este proyecto se puede modificar y personalizar de distintas maneras:

Cámara:

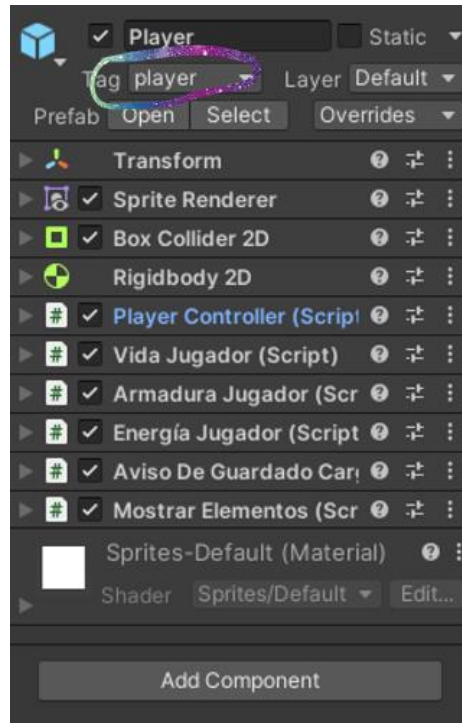
En Follow: Es al objeto que va a seguir

Bounding Shape 2D: se asigna un objeto vacío que contenga un Polygon Collider 2D y se modifica su Collider en los márgenes de nuestra escena para que la cámara no se vaya a salir de esos límites y no enfoque otra cosa que no queremos que vea nuestro jugador.



El jugador:

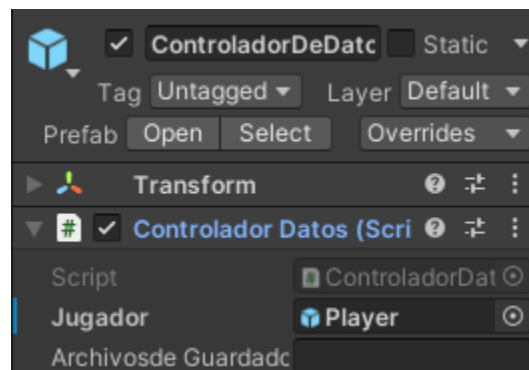
A nuestro jugador se le asignaran los diferentes scripts mostrados a continuación se le crearemos un Tag de "Player"



Cada uno de los scripts (abiertos desde unity) pedirá diferentes elementos para funcionar (como teclas, botones , etc) . Así que en cada casilla correspondiente podrá colocar la tecla que guste para guardar , el slider que usted haya diseñado para la vida y los SpriteRender que haya diseñado para los carteles de guardado y carga.

El "Controlador de Datos":

Deberá crear un objeto vacío para colocarle el script correspondiente y asignarle al jugador (pues será de quien guarde los datos)



El Canvas:

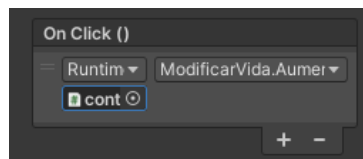
Es aquí donde deberá colocar toda su ambientación , los slider y los botones correspondientes a cada slider (pues cada uno subirán o disminuirán la vida, energía o armadura)

De ejemplo se tomo el Slider de vida, el cual contiene dos botones (nombrados como “Mas V” y “Menos V”, contiene una imagen (el cual funciona como el icono del jugador y un Texto de “Vida” .

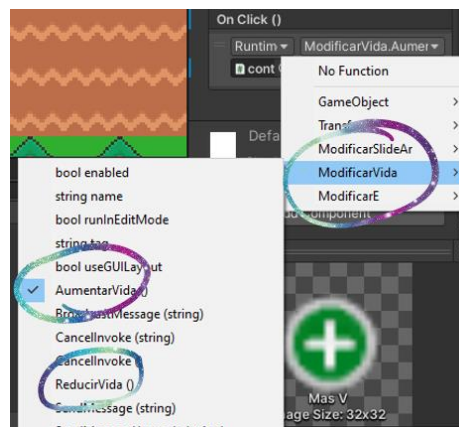
Los demás slider tienen casi lo mismo (A excepción de la imagen, pero si usted gusta igual puede agregarle una especie de icono a cada uno)



No olvide a cada botón asignarle su función correspondiente (en este caso tiene el objeto vacío llamado “Contenedor Modificadores” el cual, como su nombre dice, es quien contiene los 3 scripts que permiten modificar los valores de los slider con los botones)

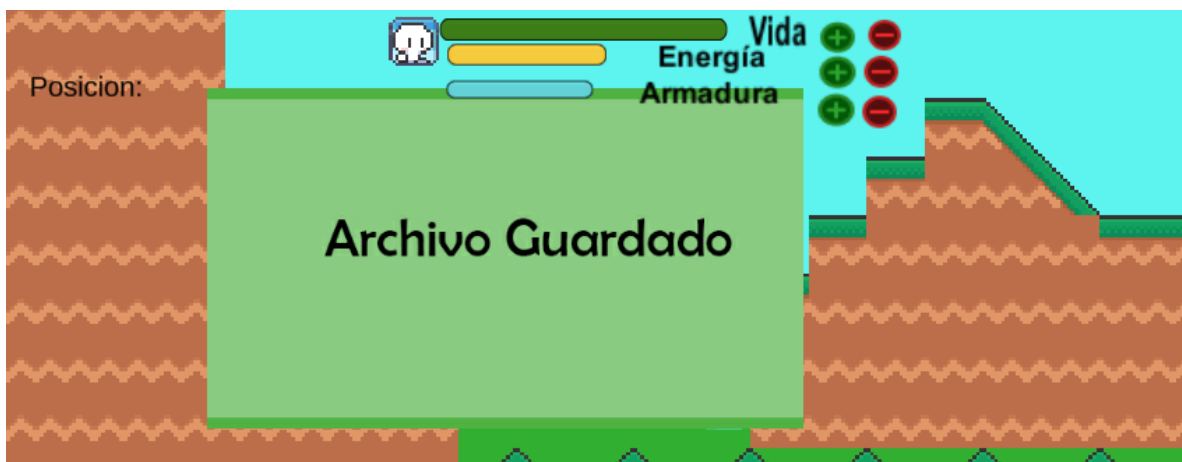
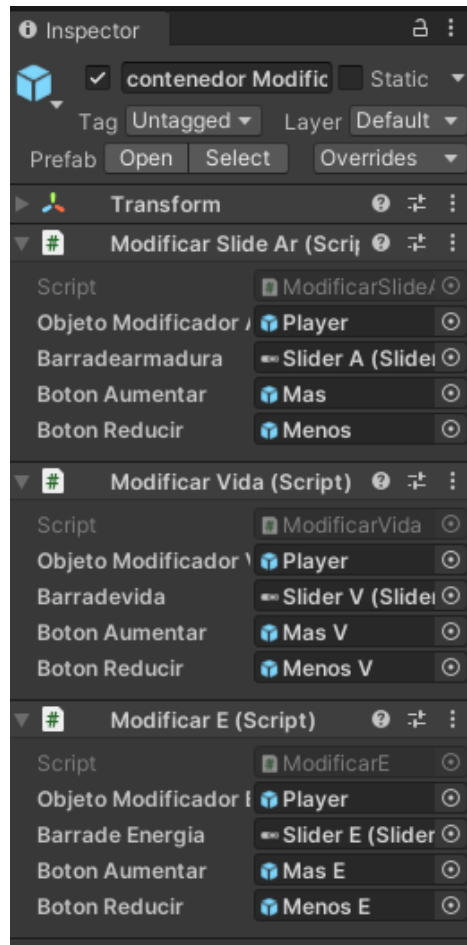


Dependiendo a que botón le quiera agregar la función será el Modificador que usara(en este caso pondremos el ejemplo de uno de los botones de Vida, pero todos los demás modificadores tienen la misma función de reducir y aumentar su elemento correspondiente)



Contenedor de Modificadores:

Este es un objeto vacío (Create Empty) que debes crear para asignarle los siguientes scripts, en los cuales deberás colocar cada uno de sus slider, botones y al jugador correspondientes y que usted haya diseñado:



Espero que esta documentación
te haya sido de ayuda y que
puedas disfrutar al máximo mi
herramienta para sistema de
guardado.

Contacto: yologonzalezv@gmail.com

