

Sistema de Aumento de Habilidades (Juego 2D en Unity).

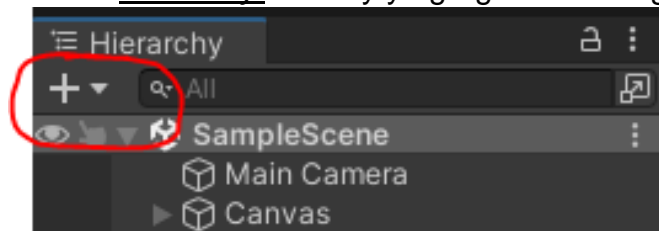
Implementación.

Este documento te ayudara a crear e implementar esta herramienta en tus video juegos.

En este sistema existen 4 scripts encargados del funcionamiento de este sistema de aumento de habilidades y otros 2 scripts encargados de un Timer básico y el movimiento de nuestro personaje.

Para utilizar el sistema de aumentos creado en Unity, sigue estos pasos:

- Crea un Canvas en tu Hierarchy en unity y agrega un UI>Legacy>Text.



- Después crea un objeto vacío en el mismo Canvas y agrega el script "Timer".
1. Crea un script en Unity llamado "Timer" y pega el código proporcionado en el script. Este script se encargará de realizar un conteo y mostrarlo en pantalla. Toma en cuenta lo siguiente:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI; //Recuerda que se necesitara agregar este paquete
para que funcione la opción de "Text".

public class Timer: MonoBehaviour
{
    //creamos la variable desde la cual se iniciará el conteo
    public float timer = 0;
    public Text textoTimer;

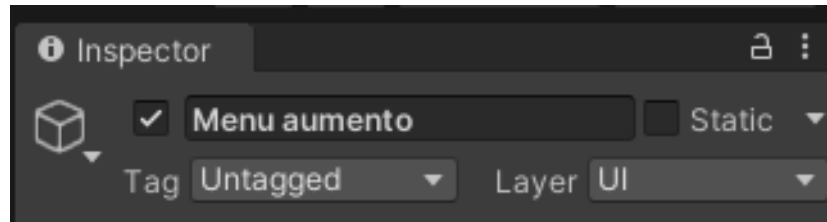
    // Update is called once per frame
    void Update ()
    {
        timer += Time.deltaTime;

        textoTimer.text = "" + timer.ToString("0");
    }
}
```

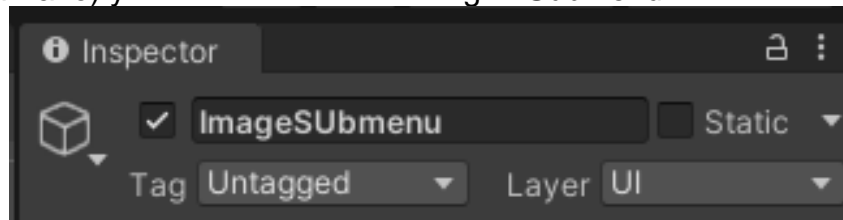
- Agrega a tu personaje (importarlo desde la ventana de Project y Assets con clic derecho) o puedes usar el que viene junto a la paquetería de este proyecto y solo arrastrarlo a la ventana de Hierarchy.
 - Después a nuestro personaje le agregaremos un Rigidbody 2D, un Box Collider 2D(se recomienda poner Gravity scale: 5 , cuando ya hayas ambientado tu juego o simplemente colocado piso) y el script "PlayerController".
2. Crea un script llamado "PlayerController" y pega el código proporcionado en el script. Este será el script para que tu personaje se pueda mover (movimientos de izquierda a derecha y salto). Toma en cuenta lo siguiente:

```
void Salto ()
{
    //GetKeyDown es importante para que el salto sea exitoso
    if (Input.GetKeyDown(KeyCode.Space))
    {
        //no es recomendable usar AddForce pero para prueba es factible.
        rigidbody2D.AddForce(Vector2.up * fuerzaSalto, ForceMode2D.Impulse);
    }
}
```

- Crea un nuevo Canvas y agrégale, UI>Panel (puedes personalizarlo) el cual llevara el nombre de "Menú aumentos"



- Después a ese mismo Panel le vas a crear UI>Imagen (puedes personalizarlo) y llevara de nombre "Imagen Submenú"



- Y a esa misma Imagen le agregaras 5 Botones (UI>Button-Text) los cuales serán 3 de Aumentos/Habilidades, 1 para salir de ese menú y 1 para restar los Aumentos y 3 Textos (UI>Legacy>Text) los cuales dirán que hace cada Botón (les puedes cambiar el nombre y el texto a cada uno seleccionándolos y modificándolos en la ventana de Inspector)

(Este es un ejemplo de cómo se puede ver)



- Al terminar de acomodarlo, apaga la vista por un momento de los objetos que creaste desde el inspector (seleccionando el objeto que lleva el nombre de "Menú aumentos" y quitando la palomita) si organizaste bien tus objetos (botones, textos e imágenes que estaban dentro del panel "Menú de aumentos") estos mismos deben desaparecer también.
3. Crea un script llamado "Menú de Aumentos" y pega el código proporcionado en el script. Este script se encargará de abrir el menú donde se encontrarán tus opciones de aumentos para el personaje. Toma en cuenta lo siguiente:

```
{
    //Este script ayuda a que el menú de habilidades se abra y cierre

    [SerializeField] private GameObject botonAumentos;
    [SerializeField] private GameObject menuAmentos;

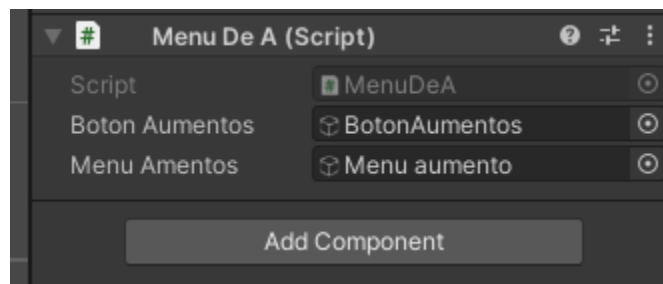
    //ayuda a que el botón de "Aumentos" abra un sub menú
    public void Aumentos ()
    {
        //al entrar al sub menú el juego se detiene como si estuviera en
        pausa
        Time.timeScale = 0f;
        botonAumentos.SetActive(false);
        menuAmentos.SetActive(true);
    }

    //Ayuda a que el botón de "atrás" te regrese al juego
    public void Reanudar ()
    {
        //al salir del submenú el tiempo vuelve a correr.
        Time.timeScale = 1f;
        botonAumentos.SetActive(true);
        menuAmentos.SetActive(false);
    }
}
```

- Al nuevo Canvas que creaste le agregaras el script llamado “Menú de Aumentos”
- Crea un nuevo Botón ahora en el Canvas (UI>Button-Text) y cambia el texto y el nombre a “BotonAumentos”



- Al agregar el script al Canvas, en la ventana del inspector aparecerán las variables que están escritas en el mismo script, solo arrastra el objeto que llamaste “menú aumentos” a la casilla que le corresponde y el nuevo botón que creaste a su casilla.



- Al botón que se encuentra en el Canvas con el nombre “BotonAumentos” le agregaras el siguiente script llamado “DesactivarBoton”
4. Crea un nuevo script en Unity llamado “DesactivarBoton” y pega el código proporcionado en el script. Este script se encarga de bloquear el botón de donde aparece el Menú de los amentos. Toma en cuenta lo siguiente:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI; //Recuerda agregar esta opción para que funcione la función de "Button"

public class DesactivarBoton : MonoBehaviour
{
    //al hacer clic en el botón se desactiva

    public GameObject optionsMenu;
    public Button optionsButton;

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

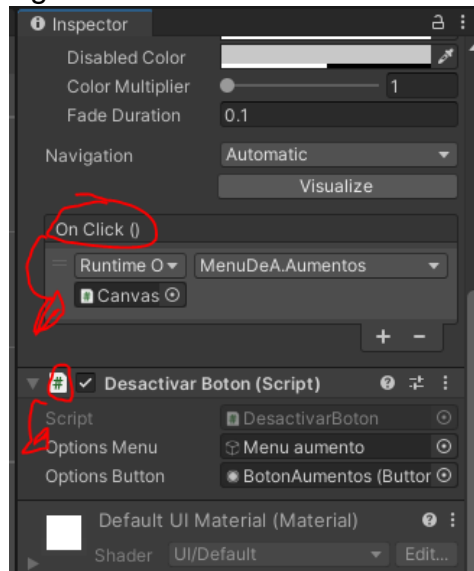
```

        optionsButton.onClick.AddListener(OpenOptionsMenu);
    }
    private void OpenOptionsMenu ()
    {
        optionsMenu.SetActive(true);
        optionsButton.interactable = false;
        //deshabilita el botón después de 5 segundos o el tiempo que le
        //menciones
        Invoke("ReenableButton", 5f);
    }

    private void ReenableButton()
    {
        optionsButton.interactable = true;
    }
}

```

- Al agregarle el script “DesactivarBoton” al botón en la vista del inspector aparecerán nuevas opciones gracias al script , modifícalas para que queden igual a la siguiente imagen.



- Ahora en nuestro Canvas agregaremos con Texto (UV>Legacy>Text) los nombres de nuestras habilidades y con UI>Text-textMeshPro crearemos 3 tipos de este texto para escribir un numero “0” y que ese mismo número vaya cambiando gracias a nuestros botones de aumentos. (recuerda cambiar el nombre del objeto en el inspector)



5. Crea un nuevo script en Unity llamado "Aumentos" y pega el código proporcionado en el script. Este script se encarga de deshabilitar los botones de cada uno de los aumentos. Toma en cuenta lo siguiente:

```
using UnityEngine.UI; //Recuerda agregar esta opción para que "Button"
funcione.
{
public class Aumentos : MonoBehaviour
{
    public GameObject aumentoV;
    public GameObject aumentoF;
    public GameObject aumentoVi;
    public Button aumento1;
    public Button aumento2;
    public Button aumento3;

    // Start is called before the first frame update
    void Start()
    {
        aumento1.onClick.AddListener(OpenaumentosV);
        aumento2.onClick.AddListener(OpenaumentosS);
        aumento3.onClick.AddListener(OpenaumentosVi);
    }
    private void OpenaumentosV()
    {
        aumentoV.SetActive(true);
        aumento1.interactable = false;
        //deshabilita el botón después de 5 segundos o el tiempo que le
menciones
        Invoke("ReenableButton", 5f);
    }
    private void OpenaumentosS()
    {
        aumentoF.SetActive(true);
        aumento2.interactable = false;
        //deshabilita el botón después de 5 segundos o el tiempo que le
menciones
        Invoke("ReenableButton", 5f);
    }
    private void OpenaumentosVi()
    {
        aumentoVi.SetActive(true);
        aumento3.interactable = false;
        //deshabilita el botón después de 5 segundos o el tiempo que le
menciones
        Invoke("ReenableButton", 5f);
    }
    //Activar los botones después del tiempo determinado
    private void ReenableButton ()
    {
        aumento1.interactable = true;
        aumento2.interactable=true;
        aumento3.interactable=true;
    }
}
```

6. Crea un nuevo script en Unity llamado “Aumentos2” o “AumentodeHabilidades” y pega el código proporcionado en el script. Este script se encarga de modificar valores ya asignados y mostrarlos en la pantalla del juego. Toma en cuenta lo siguiente:

```
using TMPro; //Recuerda agregar este paquete para que la opción
“TMP_Text” funcione.

public class Aumentos2: MonoBehaviour
{
    //valor a modificar
    public int velocidadPoint = 0;
    public int fuerzaPoint = 0;
    public int vidaPoint = 0;
    //Texto a modificar
    public TMP_Text puntosvelocidad;
    public TMP_Text puntosfuerza;
    public TMP_Text puntosvida;

    // Update is called once per frame
    void Update ()
    {
        //Se le asigna el nuevo valor que estamos asignando

        puntosvelocidad.text = velocidadPoint.ToString();

        puntosfuerza.text = fuerzaPoint.ToString();

        puntosvida.text = vidaPoint.ToString();
    }

    public void IncrementodePuntosVE () //aumento de puntos de cada
    variable por separado.
    {
        velocidadPoint++;

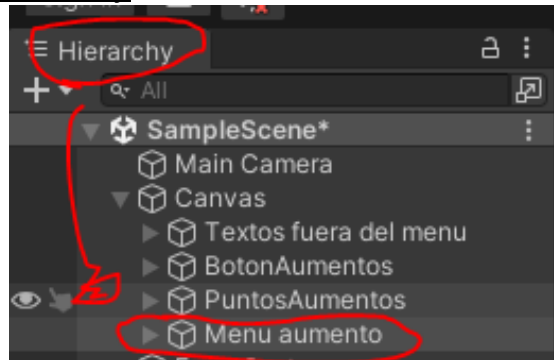
    }

    public void IncrementosdepuntosFuerza ()
    {
        fuerzaPoint++;
    }

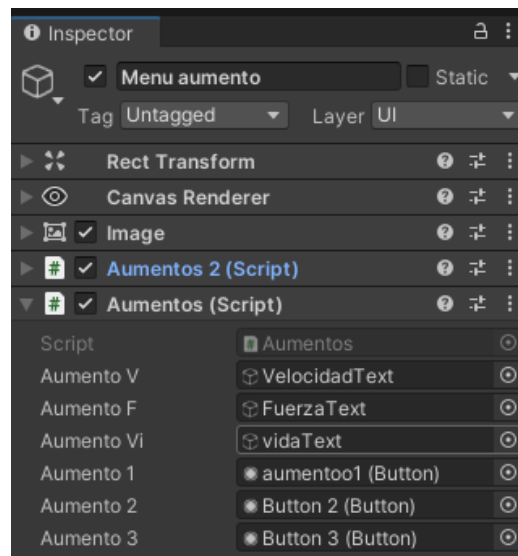
    public void Incrementodepuntosvida ()
    {
        vidaPoint++;
    }

    public void Decrementosdepuntos() //disminuyen puntos de todas las
    variables juntas
    {
        velocidadPoint--;
        fuerzaPoint--;
        vidaPoint--;
    }
}
```

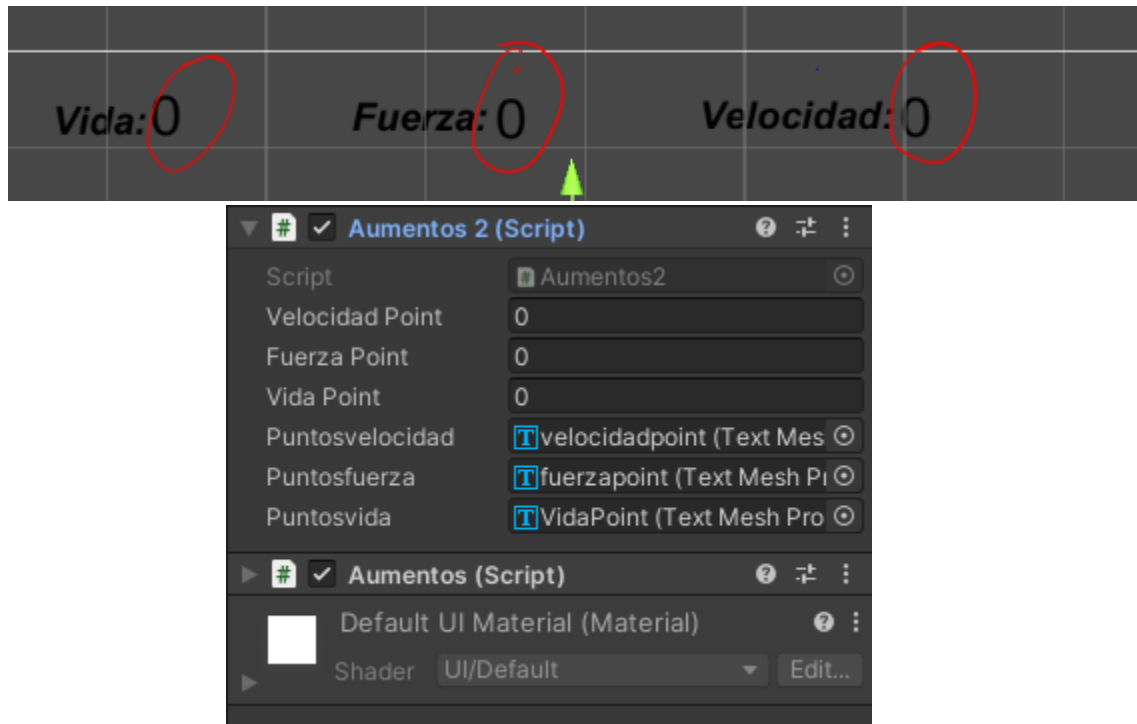
- Ahora activaras de nuevo tu objeto llamado “Menú aumento” (el cual contiene todos los botones y textos de las habilidades) el cual se encuentra en la ventana de Hierarchy.



- A este objeto le agregaremos los dos scripts llamados “Aumentos” y “Aumentos2”
- En el inspector gracias al script “Aumentos” habrá nuevas opciones, en las cuales colocaremos los botones de cada habilidad y los objetos anteriores creados con Text (los que tienen el nombre de las habilidades como texto para mostrar en nuestro Canvas)

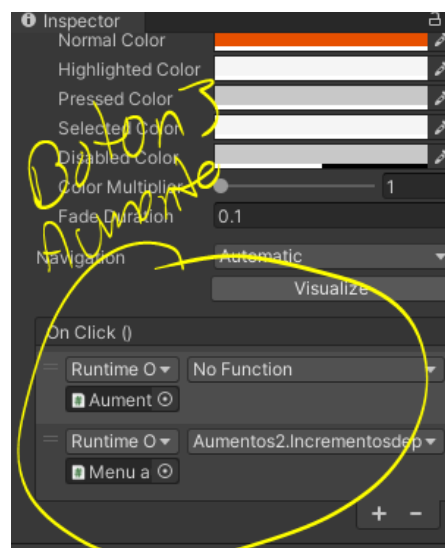
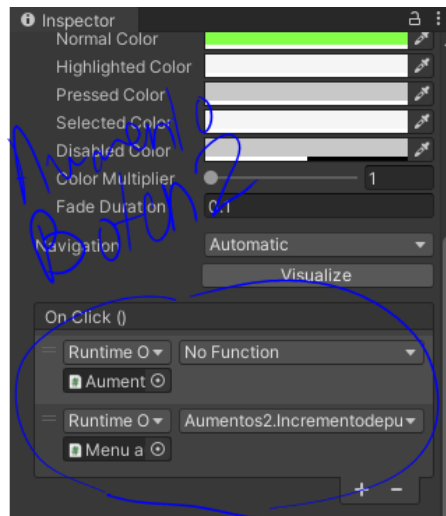
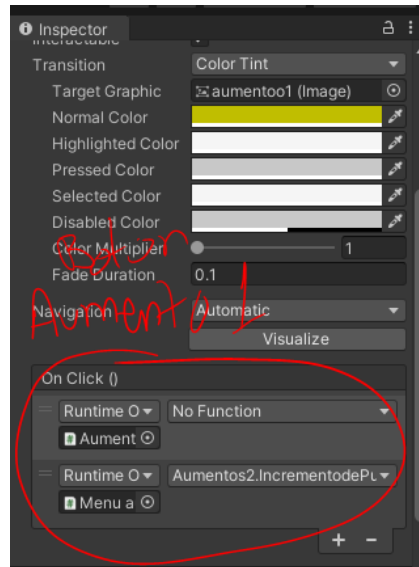


- También gracias al script “Aumentos2” o “AumentodeHabilidades” habrán nuevas funciones , en las cuales colocaremos los objetos creados anteriormente con textMeshPro en su valor correspondiente.

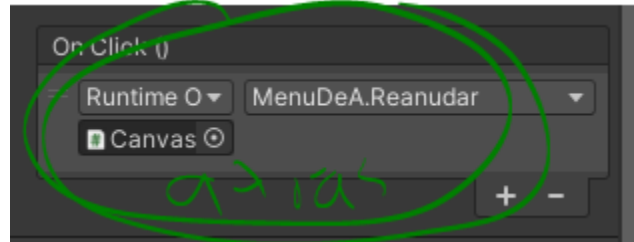


- Por ultimo, a los botones que estan dentro del “Menu aumentos” les colocaras estas nuevas funciones a cada uno.

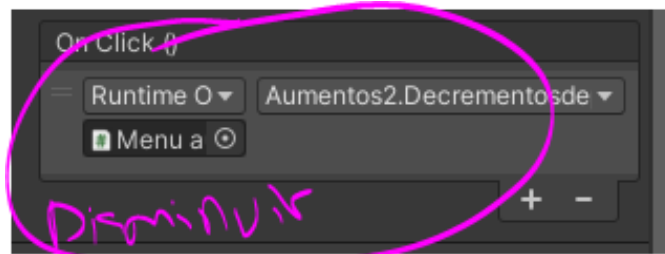




- A este boton se le asignara la funcion de “Atrás” asi podremos regresar a la pantalla principal de nuestro juego.
- Solo debemos agregarle el segundo canvas que creamos (el cual contiene todos nuestros elementos del menu, nombre de nuestras habilidades, etc)



- Y nuestro último botón será el encargado de reducir nuestros valores de todas nuestras habilidades, por lo que sus valores quedarían de la siguiente manera:



Y listo, de esta manera ya estaría completa la herramienta que funciona por medio del tiempo para aumentar las habilidades de tu personaje.

Contexto:

Esta herramienta esta inspirada un poco en la nueva mecánica que la empresa Epic Games implemento a su juego Fortnite. En este caso mi herramienta funcionara por medio del tiempo.

Es decir, que cuando aceptemos aumentos/habilidades, estos se bloquearan en seguida para que después de un tiempo (dependiendo de cada programador, pues ellos podrán asignarles el tiempo que estará bloqueado cierta habilidad) de nuevo logremos subir nuestra habilidad a nuestro personaje.

Consideraciones:

- ✓ Recuerda utilizar “Create Empty” (objetos vacíos) para utilizarlos como una especie de carpetas y organizar bien tu ventana Hierarchy, ya que así tendrás en orden todos tus objetos.
- ✓ Recuerda que a la hora de ambientar tu juego deberás jugar con las “capas” de Unity para que todos tus objetos encajen a la perfección y (en el caso de tu personaje) se caiga al vacío.
- ✓ Recuerda ocultar el “Menú de aumentos” antes de darle play a tu juego para que este desaparezca y aparezca cuando le des clic al botón correspondiente para que aparezca.
- ✓ Recuerda que para el movimiento del personaje en este sistema se utilizan las flechas (izquierda/derecha) y el espacio para saltar.

