

Progresión de personaje para videojuegos en unity 2D

En este documento se explicará la información del código y que hace cada parte del mismo. Esto ayudará al usuario a entender de manera como funciona la interfaz y como podrá modificarla para sus propios proyectos.

El código está hecho para una interfaz de usuario para la mejora del personaje. El código está dividido de la siguiente manera:

1. Partes que el código necesita para funcionar.

```
public int Hp;
public int Hp_Max;
public int Level;
public float Exp;
public float Exp_Max;

public int Vitalidad;
public int Fuerza;
public int Agilidad;
public int Magia;
public int Defensa;

public int Stats;

public Text Hp_Texto;
public Text Level_Texto;
public Text Exp_Texto;

public Text Vitalidad_Texto;
public Text Fuerza_Texto;
public Text Agilidad_Texto;
public Text Magia_Texto;
public Text Defensa_Texto;

public Text Stats_Texto;
```

Estos son los aspectos con los que el código función, el usuario puede modificarlos y cambiarlos a los que el guste.

2. Proveedor de experiencia para el jugador.

```
1 referencia
void Subir_Exp()
{
    if (Input.GetKeyDown(KeyCode.Space))
    {
        Exp += 10;
    }

    if (Exp >= Exp_Max)
    {
        Level++;
        Exp_Max = Mathf.RoundToInt(Exp_Max * 1.5f);
        Stats += 5;
    }
}
```

Esta parte del código da al jugador la experiencia que necesita para subir de nivel, esta modificada para que cada que suba de nivel, sea más difícil subir al siguiente nivel.

El usuario puede modificar los valores de la obtención de experiencia para ajustar cuanto experiencia obtendrá el jugador.

El código está programado para que cada que el jugador de la tecla space, obtenga experiencia. De igual manera se puede cambiar la tecla o el botón.

3. Subir estadísticas.

```
0 referencias
public void Subir_Vitalidad()
{
    if (Stats > 0)
    {
        Vitalidad++;
        Stats--;
    }
}
```

En esta parte se almacenan los puntos de cada uno de los atributos, cada que el jugador suba de nivel tendrá un punto para poder subir una habilidad.

Cada uno de los atributos tiene su propio void, por lo que se pueden modificar de forma independiente.

4. Bajar estadísticas.

```
0 referencias
public void Bajar_Vitalidad()
{
    if (Vitalidad > 50)
    {
        Vitalidad--;
        Stats++;
    }
}
```

En esta parte se almacenan los puntos de cada uno de los atributos, cada que el jugador suba de nivel podrá quitar puntos de las habilidades.

Cada uno de los atributos tiene su propio void, por lo que se pueden modificar de forma independiente.

5. Interfaz y botones.

```
1 referencia
public void Texto_Interfaz()
{
    Hp_Texto.text = "" + Hp + "/" + Hp_Max;
    Level_Texto.text = "" + Level;
    Exp_Texto.text = "" + Exp + "/" + Exp_Max;

    Vitalidad_Texto.text = "" + Vitalidad;
    Fuerza_Texto.text = "" + Fuerza;
    Agilidad_Texto.text = "" + Agilidad;
    Magia_Texto.text = "" + Magia;
    Defensa_Texto.text = "" + Defensa;

    Stats_Texto.text = "" + Stats;

    Hp_Max = Vitalidad;
}
```

Esta parte del código es la que hace que los números en la interfaz cambien y se actualicen a lo que el jugador haga.

Está programado con los botones predeterminados de unity, por lo que hace que se ahorre en código y sea más fácil modificar los valores de cada una de las habilidades.