

Laboratorio 2: Flappy Bird Infinito

En este laboratorio experimentaremos con una nueva forma de manejar el mundo de un juego, a través del objeto GameController. Probaremos una técnica de programación de juegos infinitos reutilizando el fondo y los objetos.

En este laboratorio veremos también los conceptos de Prefabs y Spawners.

Prefabs: Son GameObjects de Unity que pueden utilizarse como plantillas para generar más objetos que se ven y comportan de la misma manera que el objeto base. Son útiles al crear enemigos u obstáculos que tienen el mismo comportamiento.

Spawners: Son objetos cuya función es generar objetos de cierto tipo (basados en un prefab) de manera programática.

Parte 1: Configurando el proyecto

1.1 Clonando el proyecto

1. Diríjase a la dirección <https://classroom.github.com/a/6alArEbu>
2. Acepte la tarea
3. Clone el repositorio
4. Abra el proyecto en Unity
5. Abra la escena Level
6. La escena debería de mostrar un fondo celeste

Si tiene problemas al clonar su repositorio consulte con su profesora.

1.2 Seleccionando la escena y revisando elementos

1. Dentro de Unity, busque el folder de escenas, en el Project inspector y abra la escena Flappy
2. Revise los elementos que contiene la escena. Esta debe contener una cámara y dos Game Objects, llamados Background1 y Background2

Parte 2: Desarrollo del juego

2.2 Moviendo el Background

En esta sección haremos que el fondo se repita infinitamente, como en las caricaturas antiguas. Y así pueda usarse para nuestro juego infinito.

1. Dentro de Unity, busque el folder llamado Scripts
2. Dentro de este folder cree un script que se llame ScrollingBackground
3. Abra el script en VisualStudio o MonoDevelop.

Laboratorio 2: Flappy Bird Infinito

4. Declare una variable, como miembro de la clase ScrollingBackground, de tipo float llamada scrollingSpeed e inicializada con el valor de 5f.
5. El componente transform de los Game Objects contiene la posición de los mismos (transform.position) la rotación (transform.rotation) y la escala (transform.scale). Utilizaremos la función Translate del objeto transform, para mover la posición del fondo.
6. Dentro de la función Update de ScrollingBackground haga que el componente transform de este objeto se mueva, aplicando la función transform.Translate y pasándolo como parámetros el vector que nos indicará la dirección de movimiento (determinado por usted), multiplicado por la velocidad (scrollingSpeed) y multiplicado por la variable Time.deltaTime.
7. Regrese a Unity y asigne el script a los objetos Background1 y Background2
8. Corra el programa y compruebe que el fondo se mueve.

Si tiene problemas al hacer mover el fondo consulte con su auxiliar o profesora.

2.3 Haciendo que el background se repita

Al momento el fondo se mueve pero no se repite. Para que se repita, detectaremos la posición en la cual el fondo se ha salido del rango de visibilidad de la cámara y lo colocaremos nuevamente pegado al final del último fondo, para así crear la sensación de un fondo infinito.

1. En la función Update del script ScrollingBackground poner una condición para cuando el transform.position.x de este objeto haya alcanzado o pasado la posición -20.4f.

```
if (transform.position.x < -20.4f)
```

2. Dentro del condicional, cambie la posición del objeto a la par del segundo fondo.

```
if (transform.position.x < -20.4f) {  
    transform.position = new Vector3(20.4f,  
    transform.position.y, transform.position.z);  
}
```

3. Ejecute el programa en Unity y compruebe que los fondos se repiten

Revise que los fondos se repitan correctamente. Si no es el caso consulte con su auxiliar o profesora.

2.4 Game Controller

Crearemos un objeto, al cual le agregaremos un script llamado GameController. Este script será el encargado de llevar el control del punteo del jugador y de saber cuando el juego ha terminado. Haremos que este script tenga una referencia estática a él mismo, para que pueda ser llamado por los demás objetos, sin necesidad de conocer su instancia.

1. En la vista de jerarquía, Hierarchy View, sin tener ningún objeto seleccionado, de click derecho y seleccione la opción Create Empty
2. Esto creará un objeto vacío.
3. Cree un script llamado GameController
4. Declare dos variables como miembros de la clase GameController, una de tipo booleano que indique si el juego está terminado. Se llamará gameOver y su valor inicial será false. La segunda

Laboratorio 2: Flappy Bird Infinito

variable será de tipo entero y se llamará score, inicializada en cero. En esta variable guardaremos la puntuación del jugador.

5. Agregue una variable estática de tipo GameController llamada instance. En esta variable guardaremos una referencia estática a este objeto. Al hacer la variable estática la estamos haciendo parte de la clase y no de la instancia de los objetos. De esta manera podremos acceder al GameController desde cualquiera de los otros objetos.
6. Dentro de la función Start haga que la variable instance sea igual a this. Para que tenga referencia a este objeto.
7. El código debe verse de esta manera

```
public class GameController : MonoBehaviour {  
  
    public bool gameOver = false;  
    public int score = 0;  
    public static GameController instance;  
  
    void Start () {  
        instance = this;  
    }  
  
    void Update () {  
  
    }  
}
```

8. Agregue el script GameController como componente del objeto GameController en la escena.

2.5 Agregando los obstáculos

1. En la vista de jerarquía, Hierarchy view, ubique el objeto Column, selecciónelo dando click sobre su nombre.
2. En el inspector, al lado del nombre del objeto, encontrará una casilla vacía, a la que se le puede poner un cheque. Este cheque habilita y deshabilita los objetos. Marque la casilla con cheque para habilitar el objeto.
3. Observe que elementos y componentes forman el objeto.
4. Examine el código del script Column, que es componente de este objeto. Reflexione sobre lo que hace el código
5. Ejecute el programa y pruebe como funciona.

2.6 Creando Prefabs

1. En el Project inspector cree un folder llamado Prefabs
2. Arrastre el objeto Column desde el Hierarchy view hasta el folder Prefabs
3. Se creará un objeto de icono azul llamado Column. Este es un Prefab, es decir un GameObject que podemos reutilizar.

Laboratorio 2: Flappy Bird Infinito

4. Elimine el objeto Column del Hierarchy view.

Revise que el prefab se haya creado correctamente, si no es así consulte con su auxiliar o profesora.

6. Note que este objeto tiene un colisionador en medio de las dos columnas. Este colisionador no tiene imagen (sprite renderer) y es de tipo trigger, pues la casilla de trigger está marcada en las opciones del componente. Que un colisionador sea trigger significa que no responderá como los colisionadores normales de física, que frenan o rebotan con el objeto, sino servirá únicamente para detectar cuando un objeto entra en este colisionador. Utilizaremos esto para contar los puntos que haga el usuario.

2.7 Preparando el ColumnSpawner

Crearemos un script que se encargará de generar las columnas automáticamente después de cierto tiempo.

1. En Unity, diríjase al folder de Scripts. Allí cree un nuevo script llamado ColumnSpawner.
2. Abra el script en Visual Studio o Mono Develoop.
3. Declare una variable pública de tipo GameObject y de nombre column. Esta variable alojará nuestro prefab o column template que se utilizará para instanciar las columnas que necesitamos.
4. Diríjase a Unity y asigne el componente ColumnSpawner al objeto GameController. Note que un objeto puede tener más de un script.
5. Arrastre el prefab Column, que se encuentra en el Project inspector, hacia la referencia en el script de ColumnSpawner que se pide en el objeto GameController.
6. Ejecute el programa para ver si no hay ningún error de compilación. Al momento no debería de haber funcionalidad visible en el juego.

Si tiene un error de compilación consulte con su auxiliar o profesora.

2.8 Instanciando las columnas

Haremos que este script genere columnas cada dos segundos. Para detectar cuando hayan pasado dos segundos sumaremos el tiempo transcurrido entre cada frame, este tiempo está dado por la variable Time.deltaTime.

1. Abra nuevamente el script ColumnSpawner en Mono Develop o Visual Studio.
2. Declare una variable de tipo float, como miembro de la clase ColumnSpawner, la variable se llamará spawnTime y será igual a 4f.
3. Declare otra variable, de tipo float, también miembro de la clase ColumnSpawner, esta variable llevará la cuenta del tiempo transcurrido desde la última instanciación y se llamará elapsedTime. Será igual a 0f.
4. Dentro de la función Update coloque una condición que detecte si el elapsedTime es menor que el spawnTime.
5. Si el elapsedTime es menor que el spawnTime, sume la variable Time.deltaTime al elapsedTime

Laboratorio 2: Flappy Bird Infinito

6. Si el `elapsedTime` es mayor que el `spawnTime` instanciaremos una nueva columna y resetearemos la variable `elapsedTime`. El código quedará de esta manera.

```
void Update () {  
    if (elapsedTime < spawnTime)  
    {  
        elapsedTime += Time.deltaTime;  
    }  
    else {  
        float random = Random.Range(-2f, 2f);  
        Instantiate(column, new Vector3(8, random, 0),  
Quaternion.identity);  
        elapsedTime = 0;  
    }  
}
```

7. Reflexione sobre lo que hace el código y sobre la posición donde se instancia la columna. El `Quaternion.identity` indica que el objeto no tendrá rotación.
8. Ejecute el programa en Unity y compruebe que se generen automáticamente las columnas.

Si el programa no se comporta de la manera en que debería consulte a su auxiliar o profesora.

2.9 Activando el player

1. Localice el objeto Bird en el Hierarchy View.
2. Active el objeto marcando el cheque en la casilla junto al nombre del objeto en el Inspector.
3. Examine el código del script Bird y note que esta vez el personaje no se mueve en el eje X.
4. Corra el juego en Unity y compruebe el funcionamiento del mismo.

Si el programa no se comporta de la manera en que debería consulte a su auxiliar o profesora.

2.10 Contando los puntos

Utilizaremos la función `OnTriggerEnter2D` para detectar cuando el personaje ha entrado al colisionador de tipo trigger. Note que esta función es diferente a la función `OnCollisionEnter2D`.

5. Abra el script Bird en Visual Studio o Mono Develop.
6. Agregue la función `OnTriggerEnter2D`, abajo de la función `update`, de la siguiente manera

```
private void OnTriggerEnter2D(Collider2D collision)  
{  
  
}
```
7. Dentro de la función agregue el código `GameController.instance.score++`. Note que podemos acceder a la variable `instance` de `GameController`, pues es una variable de tipo `static`. De esta manera podemos ver y modificar las demás variables contenidas en `GameController`.
8. Agregue una función que imprima en la consola el número de puntos que tiene el jugador.
9. Ejecute el programa y examine como se comporta.

Laboratorio 2: Flappy Bird Infinito

Si tiene algún problema consulte a su auxiliar o profesora.

2.11 Haciendo que el juego se detenga

Haga que el juego se detenga Utilizando la variable pública, gameOver del objeto GameController.instance.

1. Detecte cuando el jugador colisiona contra cualquier objeto (columna o piso). Al detectar la colisión ponga la variable GameController.instance.gameOver como true.
2. Agregue el código necesario para evitar que las columnas y el background se muevan cuando el juego ha terminado, revisando la variable GameController.instance.gameOver
3. Ejecute el programa y compruebe que funcione correctamente

Si el programa no se comporta de la manera en que debería consulte a su auxiliar o profesora.

Si desea aprender más puede tratar de

1. Colocar monedas y que el personaje las colecte
2. Colocar la puntuación en un componente de UI en el Canvas de Unity
3. Exportar su juego como un .exe
4. Investigar acerca de Entity Pooling

Entrega: Vía Github Classroom el 14 de abril a las 23:59

Grabe un video con el juego funcionando debe mostrar todas las funcionalidades requeridas

Suba el video a Youtube

Haga un archivo Readme.md y colóquelo en el directorio raíz de su repositorio. En este archivo coloque el URL al video.

Darle commit y push al código a su repositorio

Puede omitir el paso del video si termina su laboratorio en clase y lo muestra a su auxiliar o profesora