



MOVIENDO UN SPRITE

APRENDE A PROGRAMAR TUS PROPIOS VIDEOJUEGOS

BOOTCAMP UTN 2019

MOVIMIENTO

La posición del sprite cambia con el tiempo

```
override public function update(elapsed: Float):Void
{
    super.update();
    sprite.x = sprite.x + 200;
}
```

MOVIMIENTO



```
public override  
function update(elapsed): Void  
{  
    super.update(elapsed);  
    sprite.x = sprite.x + 5;  
}
```



```
sprite.velocity.x = 100;
```

MOVIMIENTO

Haxeflixel incorpora un motor de física simple
(movimiento + colisiones)

Es mejor que el movimiento lo realice el motor en la
medida que sea posible

- más fácil
- menos código
- manejo de colisiones más preciso

MOVIMIENTO UNIFORME

1. La velocidad es constante.
2. La posición se actualiza en base a la velocidad.

MOVIMIENTO UNIFORME

```
override public function create(): Void
{
    hero.velocity.x = 100;
}

override public function update(elapsed: Float):Void
{
    super.update();
}
```

MOVIMIENTO UNIFORME

velocity controla la velocidad del sprite

Se puede asignar de varias formas

```
sprite.velocity.x = 100;  
sprite.velocity.y = -200;  
sprite.velocity.set(-50, 100.5);
```

La velocidad **permanece** así hasta que su valor se modifique

MOVIMIENTO Y UPDATE()

El comportamiento por defecto de *update()* es recorrer la lista de actores la escena y (entre otras cosas) **actualizar la posición en base a la velocidad.**

Por eso debemos llamar a *super.update()* (el método que estamos redefiniendo), no queremos sobrescribirlo completamente, sino agregar nuestro propio código al comportamiento que ya tenía.

Lo mismo ocurre con *create()* y otros métodos.

FLXG

```
import flxel.FlxF;
```

Global helper class for audio, input, the camera system, the debugger and other global properties.

ENTRADA DE TECLADO

```
if(FlxG.pressed.A){  
    hero.velocity.x = 100;  
}
```

ENTRADA DE TECLADO

- *pressed*: **true** si la tecla está presionada
- *justPressed*: **true** si la tecla **NO** estaba presionada en la actualización anterior pero **sí** en ésta
- *justReleased*: **true** si la tecla **estaba** presionada en la actualización anterior pero **NO** lo está en ésta

ENTRADA DE MOUSE

```
if(FlxG.mouse.pressed){  
    hero.velocity.x = 100;  
}
```

- pressed
- pressedMiddle
- pressedRight
- ...

<https://api.haxeflixel.com/flixel/input/mouse/FlxMouse.html>

MOVIMIENTO UNIFORMEMENTE ACELERADO

1. La aceleración es constante.
2. La velocidad se actualiza en base a la aceleración.
3. La posición se actualiza en base a la velocidad.

MOVIMIENTO UNIFORMEMENTE ACELERADO

Necesitamos varias propiedades

- **acceleration**: aceleración del sprite
- **maxVelocity**: límite para la velocidad (que no crezca indefinidamente)
- **drag**: desaceleración cuando *acceleration* es nula

MOVIMIENTO UNIFORMEMENTE ACELERADO

```
override public function create():Void
{
    super.create();
    hero = new FlxSprite(300, 300, "assets/images/hero.png");
    hero.drag.x = 100;
    hero.maxVelocity.x = 200;
    add(hero);
}

override public function update(elapsed:Float):Void
{
    super.update(elapsed);
    if(FlxG.keys.pressed.A){
        hero.acceleration.x = -200;
    }else if(FlxG.keys.pressed.D){
```

MOVIMIENTO UNIFORMEMENTE ACELERADO

Necesitamos varias propiedades

- Cuando **acceleration.x** = 200 el sprite se acelera
- Hasta que **velocity.x** == **maxVelocity.x**, entonces la velocidad se mantiene constante
- Cuando **acceleration.x** = 0, **velocity** se reduce en función de **drag.x**
- **velocity** llega a 0 y el movimiento se detiene

MOVIMIENTO ANGULAR

- angularVelocity
- angularAcceleration
- maxAngular
- angularDrag

FLXOBJECT

This is the base class for most of the display objects (FlxSprite, FlxText, etc). It includes some basic attributes about game objects, basic state information, sizes, scrolling, and basic physics and motion.

<https://api.haxeflixel.com/flixel/FlxObject.html>

FLXOBJECT Y FLXSPRITE

FlxSprite hereda de FlxObject

- **FlxObject**: movimiento, colisiones, etc
- **FlxSprite**: FlxObject + gráficos