

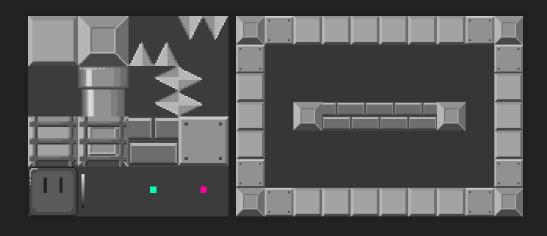
APRENDE A PROGRAMAR TUS PROPIOS VIDEOJUEGOS

**BOOTCAMP UTN 2019** 

Permiten generar imágenes grandes a partir de la repetición de patrones (o mosaicos) más pequeños

- Surgen durante los primeros años del desarrollo de videojuegos
- Permiten ahorrar memoria
- Acelerar el dibujado
- Fácil detección de colisiones
- Dan una estética clásica

```
2,12, 1, 1, 1, 1, 1, 1, 1,12, 2, 12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 12, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 2,11,11,11,11, 2, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 12, 0, 0, 0, 0, 0, 0, 0, 12, 2,12, 1, 1, 1, 1, 1, 1, 1, 2, 2
```



## TILEMAPS EN HAXEFLIXEL

```
override public function create():Void
{
    super.create();

    var map: FlxTilemap = new FlxTilemap();
    map.loadMapFromCSV(mapData, AssetPaths.tileTest2__png, 32, 3 add(map);
}
```

## TILEMAPS EN HAXEFLIXEL

- Se representan con objetos de tipo FlxTilemap
- Hay métodos para cargar la información del mapa desde distintas fuentes
- loadMapFromArray()
- loadMapFromCSV()
- loadMapFrom2DArray()
- ...

https://api.haxeflixel.com/flixel/tile/FlxBaseTilemap.htm

## LOADMAPFROMCSV

- Parámetros
- Cadena de texto con datos en formato CSV (comma separated values)
- Imagen con el tileset
- Ancho del patrón
- Alto del patrón
- Algoritmo de autotiling (no usamos)
- Indice del primer tile colisionable (0: espacio vacío,
   >1 tile sólido)

# COLISIONES CON TILEMAPS

```
override public function update(elapsed:Float):Void
{
   super.update(elapsed);
   FlxG.collide(map, hero);
}
```