

Pyxel Motor Retro para Python

Víctor Suárez García - La jaquería



Contenido



1. Introducción
2. Instalación
3. Hello Pyxel
4. Uso de Pyxel
5. Editor de Recursos
6. Ejemplos de Uso
7. Referencias



01

Introducción



Introducción

Existe un motor que nos permite realizar juegos con dicha estética solo usando 16 colores*.

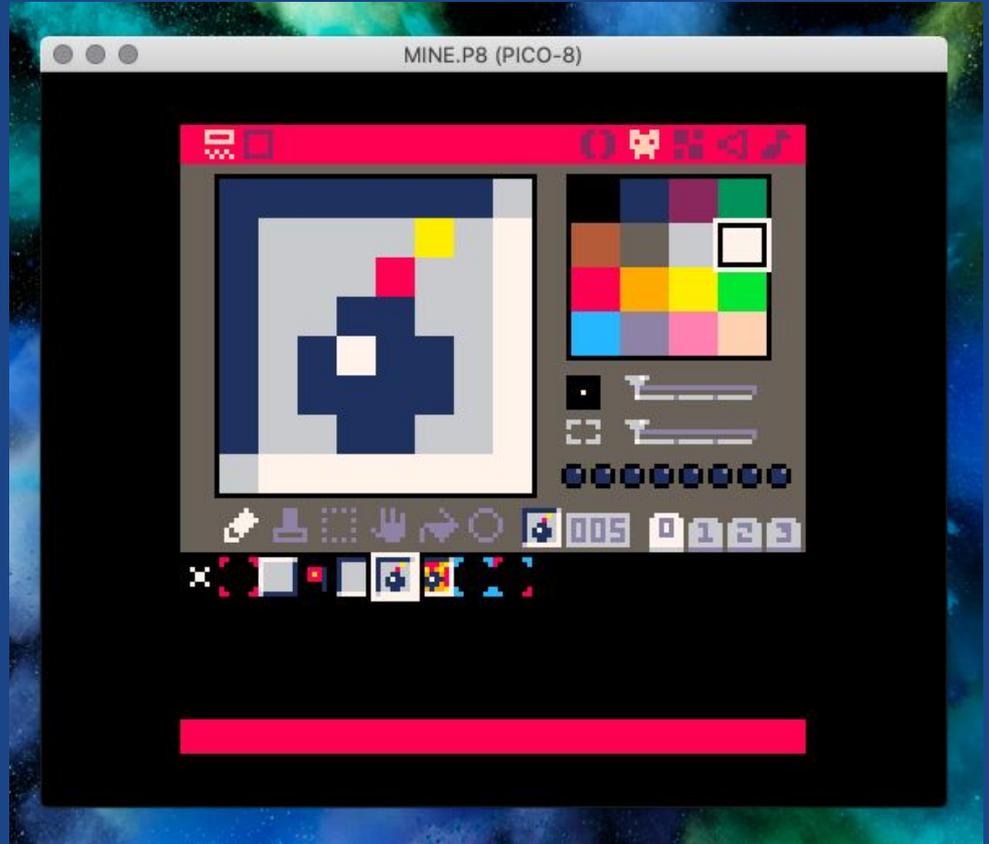
Utiliza el lenguaje LUA y se llama PICO-8.



Introducción

PICO-8 tiene un display de 128x128 píxeles a 16 colores.

Además de tener soporte para varios Sprites, sonido, etc.





Pyxel



Pyxel es un motor retro para crear videojuegos usando Python.

Permite usar Juegos con estética Retro usando solo 16 colores y una serie de características concretas.



Pyxel

Pyxel ofrece las siguientes características:

- Multiplataforma (Windows, MacOs y Linux)
- Programable con Python aunque está escrito en Rust.
- Paleta de 16 colores
- 3 bancos de memoria de 256x256 píxeles cada uno.
- 3 Mapas de 256x256 píxeles cada uno.
- 4 Canales con 64 sonidos
- Música y Sonido Combinable.
- Compatible con Ratón, Teclado y Mando
- Editor de Imágenes, Mapa y sonido Integrado

Pyxel

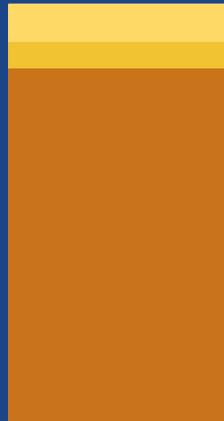
Paleta de Cores de Pyxel:

0	#000000 0, 0, 0	1	#2B335F 43, 51, 95	2	#7E2072 126, 32, 114	3	#19959C 25, 149, 156
4	#8B4B52 139, 72, 82	5	#395C98 57, 92, 152	6	#A9C1FF 169, 193, 255	7	#EEEEEE 238, 238, 238
8	#D41B6C 212, 24, 108	9	#D3B441 211, 132, 65	10	#E9C35B 233, 195, 91	11	#70C6A9 112, 198, 169
12	#7696DE 118, 150, 222	13	#A3A3A3 163, 163, 163	14	#FF9798 255, 151, 152	15	#EDC780 237, 199, 176



02

Instalación



Instalación de Pyxel

Para instalar pyxel, podemos usar el gestor de paquetes Pip, para instalarlo en nuestra instalación python. Se requiere la versión 3.7 o superior para usar Pyxel.

```
>> pip install pyxel
```

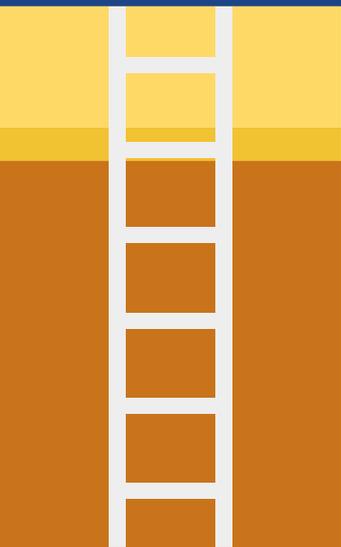
NOTA: para Linux, instalar la dependencia *libsdl2-dev*



03

Hello Pyxel

Ejemplo Inicial



Ejemplo Inicial

```
import pyxel

class Game:
    def __init__(self) -> None:
        pyxel.init(160,120,"Hello Pyxel")
        pyxel.run(self.update,self.draw)

    def update(self):
        pass

    def draw(self):
        pyxel.rect(20,20,20,10,3)

Game()
```

NOTA: Todos los ejemplos están disponibles en:
https://github.com/LaJaqueria/pyxel_workshop

Ejemplo Inicial

Veamos algunas funciones de Pyxel:

- La función *init*, inicializa el motor y la ventana además del título de esta.

```
pyxel.init(160,120,"Hello Pyxel")
```

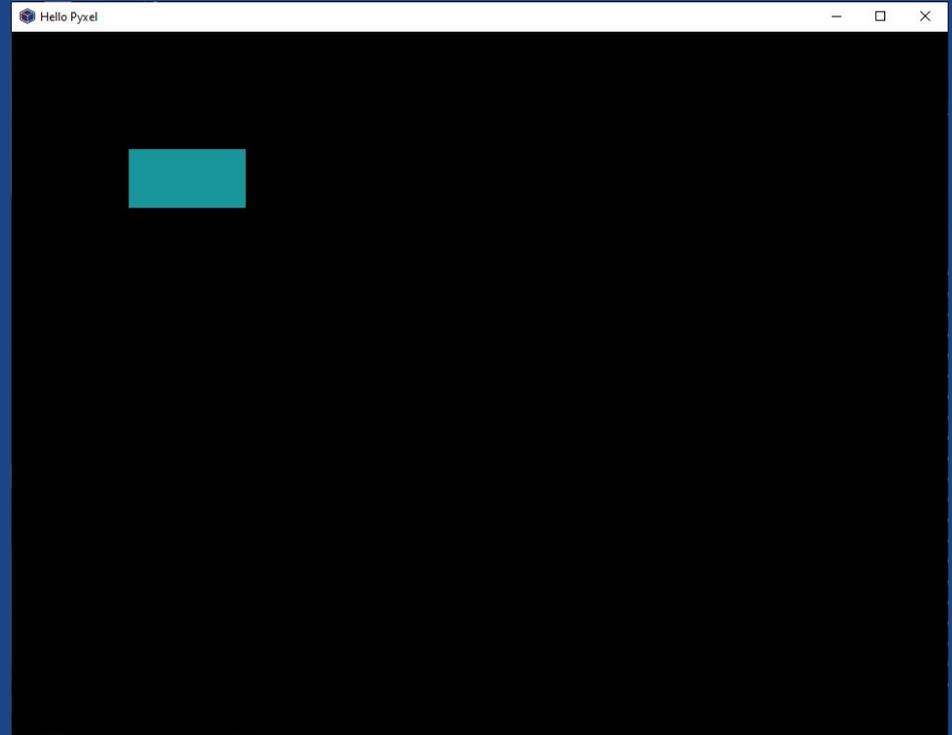
- La función *Run*, arranca la aplicación, pero necesita pasar por parametro 2 funciones, la función *update* y la función *draw*
 - La función *update*, se utiliza para actualizar en cada frame los valores del juego como los controles, física etc... se ejecuta en cada frame.
 - La función *draw*, se utiliza para dibujar todo lo que necesitemos por pantalla. Se ejecuta en cada frame.

```
pyxel.run(self.update,self.draw)
```

Ejemplo Inicial

Para ejecutar este ejemplo, usaremos la siguiente instrucción:

```
>pixel run game.py
```





04

Uso de Pyxel

Comandos e instrucciones para ejecutar Pyxel



Uso de Pyxel

- Ejecutar un Script Python con Pyxel:

```
>pyxel run game.py
```

- Crear o editar un paquete de Recursos (.pyxres)

```
>pyxel edit varaman.pyxres
```

- Generar un Paquete Pyxel (.pyxapp)

```
>pyxel package varaman game.py
```

- Ejecutar un paquete Pyxel (.pyxapp)

```
>pyxel play varaman.pyxapp
```



05

Editor de Recursos

Editor de Recursos

Pyxel, trae integrado un editor de recursos para todos los recursos que necesitará el juego:

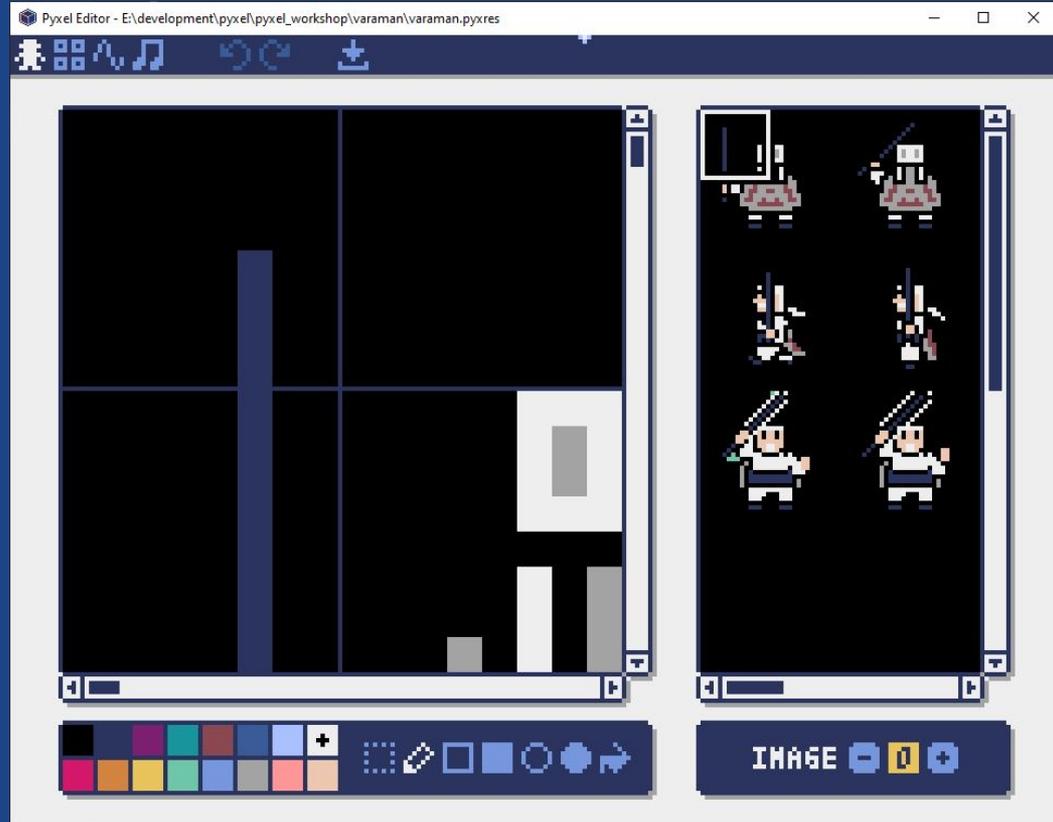
- Editor de Objetos
- Editor de mapas
- Editor de Sonido
- Editor de Música

Editor de Objetos

El editor de Objetos, nos va a permitir editar cada objeto que se encuentre en los 3 bancos de imágenes de 256x256. Recordemos que solo tenemos 16 colores.

Podemos importar una imagen arrastrándola al editor y se almacenará en dicho banco.

NOTA: No olvides guardar!!



Editor de Mapas

El editor de Mapas, permite dibujar un mapa a partir de los elementos que componen el banco de imágenes.

Podemos tener hasta 7 mapas usando los 3 bancos de imágenes. Además de poder importarlo a nuestro juego.

NOTA: No olvides guardar!!

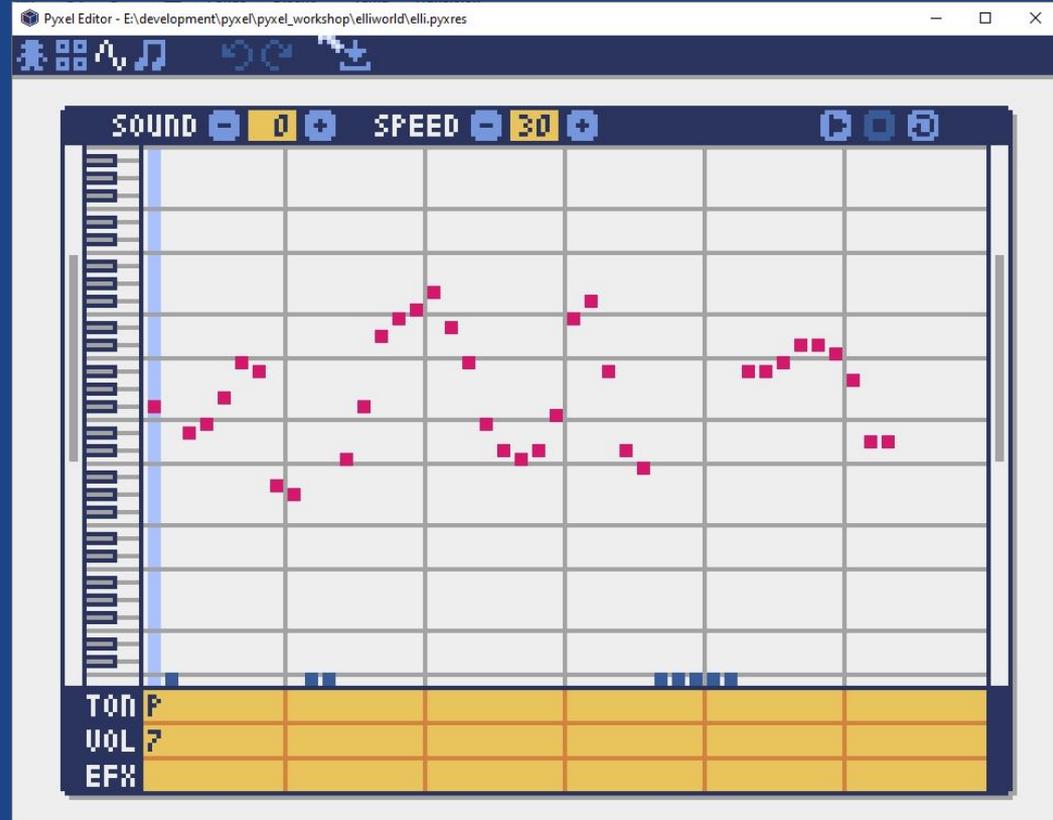


Editor de Sonidos

El editor de Sonidos, permite generar cada uno de los 64 sonidos disponibles para nuestro juego.

Los sonidos son MIDI y permiten generar distintos sonidos con distintos efectos.

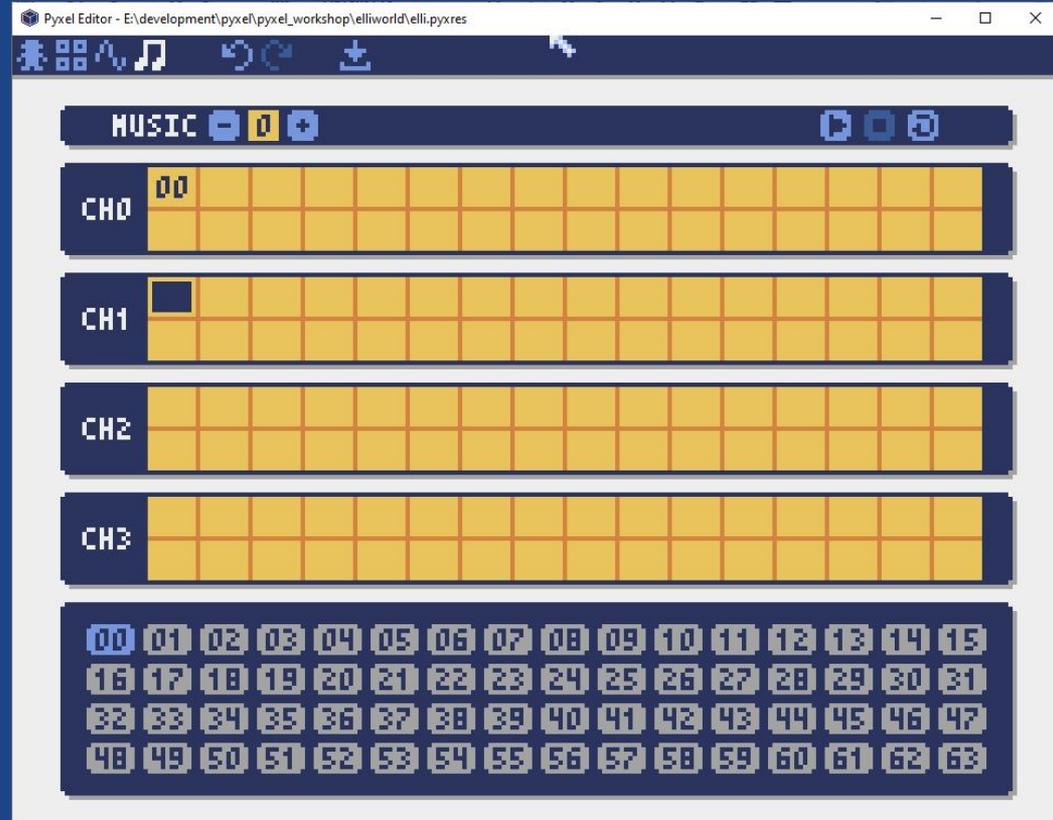
NOTA: No olvides guardar!!



Editor de Música

El editor de música, permite orquestrar los distintos sonidos generados con el editor de sonidos en los 4 canales disponibles.

NOTA: No olvides guardar!!



06

Ejemplos y Uso de la API



Ejemplo 1: texto y colores

En este ejemplo vemos el texto “Hello Jaqueria” y podemos cambiar el color usando las direcciones del teclado.

Podemos ver el uso de las funciones `btnp` y `text`

- La función `btnp` devuelve `True` si se ha pulsado la tecla que se pasa por parámetro. Ej `KEY_RIGHT`.
- La función `text` dibuja por pantalla un texto en un color definido.

```
import pyxel

class App:
    colour=1
    def __init__(self) -> None:
        pyxel.init(160, 120, title="Hello Jaqueria")
        pyxel.run(self.update,self.draw)
        self.colour=1
    def update(self):
        if pyxel.btnp(pyxel.KEY_ESCAPE):
            pyxel.quit()
        if pyxel.btnp(pyxel.KEY_RIGHT):
            self.colour=self.colour+1
        if pyxel.btnp(pyxel.KEY_LEFT):
            self.colour=self.colour-1
            if self.colour < 0:
                self.colour=15
    def draw(self):
        pyxel.text(50,50,"Hello Jaqueria",col=self.colour % 16)

App()
```

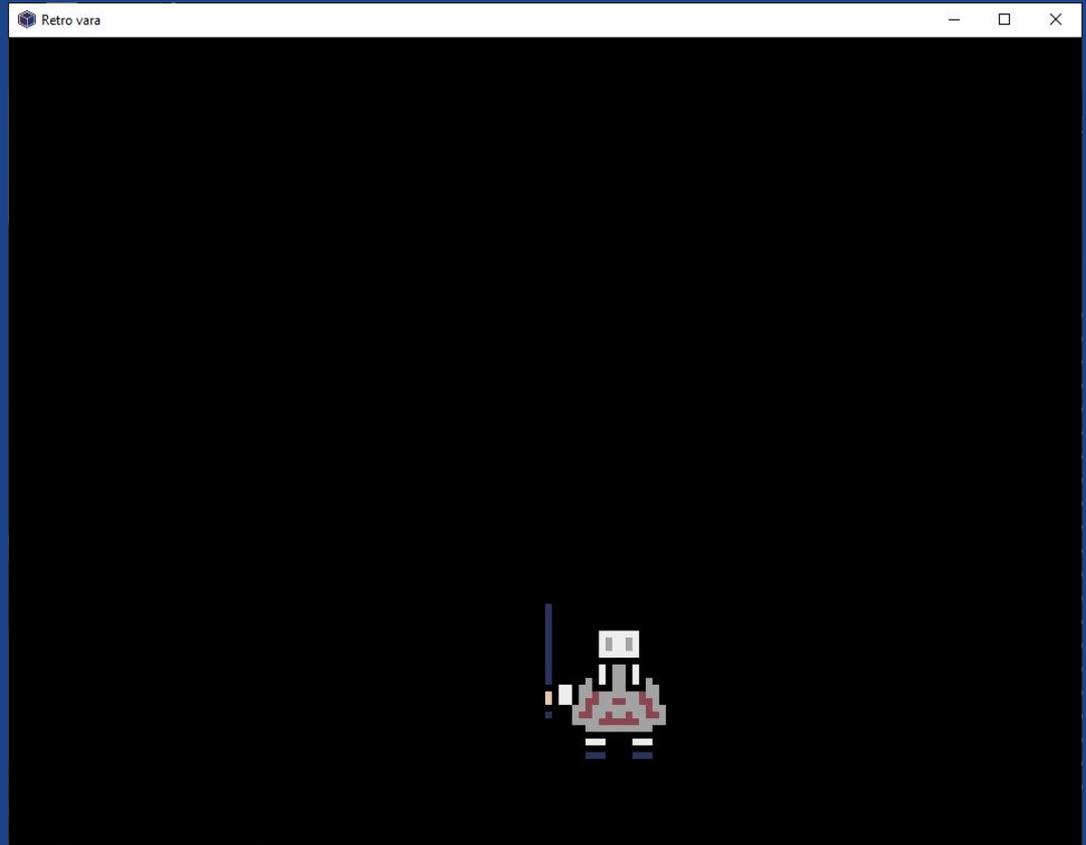
Ejemplo 2: Sprites

En este ejemplo, podemos ver a un personaje que podemos mover de derecha a izquierda.

Este ejemplo se basa en copiar imágenes del banco de imágenes correspondiente formando así una animación.

En este ejemplo se usa la función *blk*: que copia un fragmento del banco de imagen y lo muestra por pantalla.

También se usa la función *btn* que devuelve True mientras se pulse la tecla pasada por parámetro.



Ejemplo 3: Mapa y Sonido

En este ejemplo, podemos ver ya un mapa y un personaje por el que podemos movernos.

En este caso usamos las funciones anteriormente mencionadas y además las siguientes:

- La función *playm*, que permite ejecutar un sonido una vez o en bucle.
- La función *camera*, establece la cámara para poder realizar scroll.
- La función *bltm*, permite dibujar un fragmento de mapa.



Referencias

- <https://lajaqueria.org/>
- <https://www.lexaloffle.com/pico-8.php>
- <https://github.com/kitao/pyxel>
- <https://github.com/sixteenbits/TownQuest>
- <https://slidesgo.com/theme/retro-video-games-newsletter#position-4&related-1&rs=detail-related>
- https://github.com/LaJaqueira/pyxel_workshop



Muchas
Gracias

