

Reproductor de Música con Pygame

Noviembre de 2021

Part I

Librerías

1. Mixer de Pygame (documentación).
2. os (documentación).
3. random (documentación).

Part II

Algoritmo

1 Funciones Mixer

1.1 reproducirCancion()

Recibe la ruta completa como cadena de caracteres y a continuación:

1. Carga la canción con *pygame.mixer.music.load()*
2. Ajusta el volumen con *pygame.mixer.music.set_volume()*
3. Reproduce la canción *mixer.music.play()*

1.2 pausarCancion()

1. Hace uso de *pygame.mixer.music.pause()*

1.3 reanudarCancion()

1. Hace uso de *pygame.mixer.music.unpause()*

1.4 detenerCancion()

1. Hace uso de *pygame.mixer.music.stop()*

1.5 modificarVolumen()

1. Hace uso de *pygame.mixer.music.set_volume()*

Nota: Esta función únicamente acepta valores comprendidos en el intervalo $[0, 1]$.

2 lecturaContenido

La función obtiene los nombres de los elementos contenidos en la ruta dada (carpetas y-o archivos).

2.1 Análisis y especificación

2.1.1 Objetos conocidos

- Cadena de texto con la ruta dada por el usuario $R \in \text{ASCII}$.

2.1.2 Objetos desconocidos

- Lista con los nombres de los elementos contenidos en la carpeta $C \in \text{ASCII}^*$.

2.1.3 Relaciones entre objetos conocidos y objetos desconocidos

$\text{ASCII} \rightarrow \text{ASCII}^*$

Haciendo uso de `os.listdir()` se obtiene la lista.

2.2 Diseño y prueba conceptual

$$\begin{array}{ccc} \text{ASCII} & \longrightarrow & \text{ASCII}^* \\ \text{lecturaContenido: } R & \longrightarrow & \begin{cases} \text{try} & C = \text{os.listdir}(R) \\ \text{except} & \text{FileNotFoundError} \end{cases} \end{array}$$

3 compruebaContenido

Comprueba que hay al menos 1 elemento en la lista.

3.1 Análisis y especificación

3.1.1 Objetos conocidos

- Arreglo de cadenas de caracteres $C \in \text{ASCII}^*$.

3.1.2 Objetos desconocidos

- Valor booleano $X \in \mathbb{B}$.

3.1.3 Relaciones entre objetos conocidos y objetos desconocidos

$\text{ASCII}^* \rightarrow \mathbb{B}$

Existe una lista C_n elementos donde $n > 0$, retorna *True*.

3.2 Diseño y prueba conceptual

$$\begin{array}{ccc} \text{ASCII}^* & \longrightarrow & \mathbb{B} \\ \text{compruebaContenido: } C & \longrightarrow & \begin{cases} C_n | n > 0 & \text{True} \\ & \text{False} \end{cases} \end{array}$$

4 menuBusquedaCarpeta

Obtiene la ruta de la carpeta, realiza la lectura de esta y devuelve la ruta dada por el usuario y la lista de elementos es esa ruta.

4.1 Análisis y especificación

4.1.1 Objetos conocidos

- Ruta de la carpeta donde se encuentra la música dada por el usuario $R \in \text{ASCII}$.
- Lista de nombres de canciones contenida en la ruta $C \in \text{ASCII}^*$.

4.1.2 Objetos desconocidos

- Ninguno.

4.2 Diseño y prueba conceptual

$$\begin{array}{ccc} \text{ASCII}^* \times \text{ASCII} & \longrightarrow & \text{ASCII}^* \times \text{ASCII} \\ \text{menuBusquedaCarpeta}(): & R, C & \longrightarrow \begin{array}{l} R = \text{input}() \rightarrow C = \text{lecturaContenido}(C) \Rightarrow \\ \forall \left\{ \text{compruebaContenido}(C) = \text{False} \quad \text{ERROR} \right. \end{array} \end{array}$$

Por ultimo retorna R, C .

5 imprimirListaCanciones()

Recorre la lista de elementos y los imprime.

5.1 Análisis y especificación

5.1.1 Objetos conocidos

- Lista de nombres de canciones $C \in \text{ASCII}^*$.

5.1.2 Objetos desconocidos

- Impresión de cada elemento de la lista ASCII^n .

5.1.3 Relaciones entre objetos conocidos y objetos desconocidos

$\text{ASCII}^* \longrightarrow \text{ASCII}$

Imprime cada elemento.

5.2 Diseño y prueba conceptual

$$\begin{array}{ccc} \text{ASCII}^* & \longrightarrow & \text{ASCII} \\ \text{imprimirListaCanciones}(): & C_n & \longrightarrow \forall_{i=0}^{n-1} \text{print}((i+1)), \text{print}('/' + C_i) \end{array}$$

6 eligeCancion()

Presenta un menú para que el usuario, elije la canción que desea escuchar y retorna el número de canción elegida respecto a su impresión.

6.1 Análisis y especificación

6.1.1 Objetos conocidos

- Lista de nombres de canciones $C \in \text{ASCII}^*$.
- Número de la canción elegida dado por el usuario $N \in \mathbb{Z}$.

6.1.2 Objetos desconocidos

- Ninguno.

6.1.3 Relaciones entre objetos conocidos y objetos desconocidos

$\text{ASCII}^* \times \mathbb{Z}$

La función retornara el número N dado por el usuario. Este número como $N - 1$ para el nombre de la canción en la lista C .

Para selección aleatoria se hace uso de la librería “random” para elegir un número en el intervalo comprendido entre 0 y el tamaño de C .

6.2 Diseño y prueba conceptual

	$\text{ASCII}^* \times \mathbb{Z}$	\longrightarrow	\mathbb{Z}
eligeCancion:	C, N	\longrightarrow	$\begin{cases} N = 0 & N = \text{random.choice}(0, \text{len}(C) + 1) \\ 0 < N \leq \text{len}(C) & \text{continue} \end{cases}$

7 menuCanciones()

Ingresa la ruta de la carpeta y la lista de nombres de canciones y retorna N .

7.1 Análisis y especificación

7.1.1 Objetos conocidos

- Ruta de la carpeta donde se encuentra la música dada por el usuario $R \in \text{ASCII}$.
- Lista de nombres de canciones contenida en la ruta $C \in \text{ASCII}^*$.
- Número de la canción elegida dado por el usuario $N \in \mathbb{Z}$.

7.1.2 Objetos desconocidos

- Ninguno.

7.1.3 Relacion entre objetos conocidos y objetos desconocidos

$\text{ASCII}^* \times \text{ASCII} \times \mathbb{Z}$

1. Imprime los elementos haciendo uso de `imprimirListaCanciones()`
2. Ordena elegir la canción haciendo uso de `eligeCancion()`
3. Arma la ruta completa en una única cadena de caracteres.
4. Ordena reproducir la canción haciendo uso de `reproducirCancion()`