

**UNIDADE CURRICULAR:** Programação por Objetos

**CÓDIGO:** 21093

**DOCENTES:** Leonel Morgado & Jorge Morais

**A preencher pelo estudante**

**NOME:** Nuno Rodrigues

**N.º DE ESTUDANTE:** 2201022

**CURSO:** Engenharia Informática

**DATA DE ENTREGA:** 22/11/2023

**TRABALHO / RESOLUÇÃO:**

**Descrição geral do projeto:**

Este projeto consiste no desenvolvimento de um *sampler* virtual simples e intuitivo. O seu principal objetivo é permitir que os utilizadores carreguem ficheiros de áudio, como sons de bateria, e os associem a um *pad*, termo usado para designar os botões em *samplers* físicos (**Fig. 1**). Após esta associação, os sons poderão ser ativados através do teclado do computador ou por cliques do rato.



**Fig. 1** - Exemplo de um *sampler* físico

**Descrição das funcionalidades:**

A aplicação inicia-se com a carga automática de uma seleção de *samples* pré-definidos. Para atribuir um *sample* a um determinado *pad*, o utilizador pode clicar no botão “+” situado no canto superior direito de cada *pad*. Esta ação desencadeia a abertura de uma nova janela, que apresenta uma lista dos *samples* já carregados. Aqui, o utilizador pode selecionar o sample desejado e confirmar a sua escolha através do botão “confirmar”, associando, assim, o *sample* ao *pad* em questão. Adicionalmente, é possível adicionar novos *samples* à biblioteca, pressionando o botão “adicionar”. Esta ação abre um explorador de ficheiros, permitindo ao utilizador escolher e carregar novos *samples*. Importante salientar que todos os ficheiros carregados pelo utilizador são automaticamente copiados para a pasta “samples”.

**Descrição das classes:**

A classe **App** desempenha um papel central na aplicação, responsabilizando-se pela criação e gestão da interface gráfica do utilizador.

* **\_\_init\_\_:** Este método inicializa a instância da classe. Cria a janela principal da GUI, e é responsável por iniciar uma série de parâmetros, como o título, etc.
* **\_\_onClosing:** É um método de *callback* que é ativado quando o utilizador tenta fechar a janela da aplicação.
* **\_\_loadWidgets:** Este método carrega os widgets que serão exibidos na interface.
* **\_\_placeWidgets:** Este método posiciona os widgets anteriormente carregados na interface gráfica.
* **run:** Este é o método que inicia a execução da interface gráfica.

As classes dos botões na sua implementação, **PlayButton** e **LoadSampleButton**, são especializações da classe **GenericButton** e possuem características e funcionalidades específicas.

**GenericButton:**

* **\_\_init\_\_:** Este método inicializa o botão genérico.
* **\_\_setStyle:** Para definir o estilo do botão.
* **place:** Posiciona o botão na janela.

**PlayButton (subclasse de GenericButton):**

* **\_\_init\_\_:** Inicializa o botão de reprodução.
* **\_\_setStyle:** Define o estilo do botão de reprodução.
* **\_\_pressAnimation:** Animação reproduzida quando o utilizador usa o teclado.
* **\_\_onKeyPress:** Método que invoca a função de *callback* ao primir uma tecla.

**LoadSampleButton (subclasse de GenericButton):**

* **\_\_init\_\_:** Inicializa o botão de carregamento de samples.
* **place:** Posiciona o botão usando coordenadas absolutas (x, y) em vez de uma grelha.

A classe **Pad** é um componente composto pelos botões correspondentes às classes **PlayButton**, e **LoadSampleButton**.

* **\_\_init\_\_:** Inicializa um objeto Pad.
* **\_\_loadWidgets:** Carrega os widgets necessários para o *pad*.
* **\_\_placeWidgets:** Posiciona os widgets carregados anteriormente dentro do Frame do pad.
* **\_\_loadAudioFile:** Invoca a execução do **SampleLoader**, para permitir ao utilizador escolher e carregar um ficheiro de áudio para o pad.
* **\_\_playAudio:** Um método para reproduzir áudio.
* **place:** Método para posicionar o pad na interface gráfica.

A classe **PadBoard** representa um contentor para vários *pads*.

* **\_\_init\_\_:** Este método inicializa uma instância da classe **PadBoard**.
* **\_\_loadWidgets:** Este método é responsável por carregar os *pads*. Cria um novo **Pad** para cada posição na grelha definida pelas variáveis rowSize e columnSize.
* **\_\_placeWidgets:** Este método posiciona os *pads* na interface gráfica.
* **place:** Este método posiciona o PadBoard na interface gráfica.

A classe **SampleLoader** é responsável por gerir o carregamento e a seleção de *samples* de áudio.

* **\_\_init\_\_:** Inicializa uma instância da classe **SampleLoader**.
* \_\_**loadWidgets:** Este método é responsável por carregar e configurar os *widgets* da interface gráfica.
* **\_\_placeWidgets:** Posiciona os *widgets* na interface gráfica.
* **\_\_loadSamples:** Método responsável por carregar os *samples* de uma pasta pré-definida.
* **\_\_onConfirm:** Método executado ao pressionar o botão 'Confirm', carrega o *sample* para o pad e fecha a janela.
* **\_\_onCancel:** Método executado ao pressionar o botão 'Cancel', decha a janela sem carregar nenhum sample.
* **\_\_addSample:** Abre um diálogo de seleção de ficheiros para adicionar um novo *sample* à lista.
* **run:** Este método inicia a interface gráfica para o **SampleLoader**.

A classe **SampleList** representa uma lista que permite selecionar um *sample* contido no diretório '../samples/'. E fornece uma interface que simplifica a lógica da classe **SampleLoader.**

* **\_\_init\_\_:** Inicializa uma instância da classe **SampleList.**
* **\_\_iadd\_\_:** Subrecarga do operador += para adicionar *samples* novos á lista, é feito desta forma por questões de avaliação.
* **selectedSample(@property):** Getter para recolher o *sample* selecionado.
* **\_\_loadSamples:** Carrega *samples* no diretório '../samples/'**.**
* **\_\_setSample:** Método usado como *callback* pelo *widget* 'Listbox' para selecionar um *sample***.**
* **place:** Posiciona os *widgets* na janela.

A classe **Audio** é responsável por gerir e reproduzir ficheiros de áudio na aplicação.

* **\_\_init\_\_:** Inicializa uma instância da classe **Audio**.
* **name (@property)**: Um método que permite o acesso ao nome do ficheiro de áudio.
* **play:** Um método para reproduzir o ficheiro de áudio.
* **initAudioPlayer(@staticmethod):** Método a ser usado uma vez para iniciar o mixer.

A classe estática **Save** está feita para encapsular as funcionalidades relacionadas relacionadas com gravar e carregar o ultimo estado do programa. O estado está gravado num ficheiro json onde “default” representa os ficheiros por defeito, na pasta “samples/default/”, e “userSave” representa os *samples* carregados pelo utilizador.

* **save:** Grava o *sample* num ficheiro json.
* **load:** Carrega *sample* com base na tecla selecionada, se não existir um *sample* gravado na tecla selecionada, carrega o *sample* por defeito.