Coding workshop: Drag and drop

Introduction

In this worksheet, we will implement the FileDragAndDropTarget interface. This will allow our DeckGUI component to register for and receive drag and drop events. Drag and drop events happen when the user drops a file onto an application. In the end, we will be able to drop audio files onto the application and then play them.

Update the class inheritance and add pure virtual functions

First of all, update the DeckGUI class declaration, so it reads as follows (assuming you have not added any class relationships of your own:

```
class DeckGUI    : public Component,
                   public Button::Listener,
                   public Slider::Listener,
                   public FileDragAndDropTarget
```

Note that we have added the FileDragAndDropTarget interface. Look at the API documentation for this interface class. Does it have any pure virtual functions that we must implement? I found these two:

```
virtual bool isInterestedInFileDrag (const StringArray& files) = 0;
virtual void filesDropped (const StringArray& files, int x, int y) = 0;
```

We can write those into our DeckGUI.h file. Put these into the public section of the class definition:

```
bool isInterestedInFileDrag(const StringArray& files) override;
void filesDropped(const StringArray& files, int x, int y) override;
```

Now put in basic implementations to DeckGUI.cpp:

```
bool DeckGUI::isInterestedInFileDrag(const StringArray& files)
{
    std::cout << "DeckGUI::isInterestedInFileDrag" << std::endl;
    return true;
}

void DeckGUI::filesDropped(const StringArray& files, int x, int y)
{
    std::cout << "DeckGUI::filesDropped" << std::endl;
}
```

Note that isInterestedInFileDrag returns true. If it returned false, then the system would never call filesDropped. Try running the app and then dragging a file over the app from your operating system's file browsing tool. Figure out when it is calling the different functions.

filesDropped will be called when the user releases the file over the DeckGUI component (if isInterestedInFileDrag returns true).

Look at the data you receive from the drop

Now let's look at the data we received from the drag and drop event. Let's just print out the data for now:

```
void DeckGUI::filesDropped(const StringArray& files, int x, int y)
{
    for (String filename : files)
    {
        std::cout << "DeckGUI::filesDropped " << filename << std::endl;
    }
}
```

Notes:

- I am using an iterator pattern to iterate over the items in files
- the files variable is a StringArray containing items of type String
- String is part of the juce API (I am not sure why they use this instead of std::string).
- When you run the code, you will find out that it prints out the full file paths of the files that the user (you, at the moment...) drops onto the Component
- Each DeckGUI component should trigger these events independently

Use the drag and drop data to load a file

The final step is to hook up the filesDropped event to the load function on DJAudioPlayer.

In the DeckGUI::filesDropped function, convert the String file path into an URL:

```
URL fileURL = URL{File{filename}};
```

Note that we convert the String to a File and the File to an URL.

Then send it over to the DJAudioPlayer:

```
djAudioPlayer->loadURL(fileURL);
```

That's it. Except, you probably don't want to process more than one file. If the user drops several files onto the Component, the filesDropped function will be called multiple times. For now, we only want to process the first file, so we can call return after calling load. So the complete DeckGUI::filesDropped function looks like this:

```
void DeckGUI::filesDropped(const StringArray& files, int x, int y)
{
    for (String filename : files)
```

```
    {
        std::cout << "DeckGUI::filesDropped " << filename << std::endl;
        URL fileURL = URL{File{filename}};
        djAudioPlayer->loadURL(fileURL);
        return;
    }
}
```

Try dragging and dropping a different file onto each deck. You should be able to load two files independently.

Conclusion

In this worksheet, we have added drag and drop functionality to the DeckGUI class. Now we can drop audio files onto our decks and play them.