

# COSC346 Assignment 2

## FBRD Media Player

### Team Fire Breathing Rubber Duckies:

Sam Paterson (3175949)  
Max Huang (4427762)

---

### Object Oriented Concepts in Design and Implementation

The biggest concept we used in this project was MVC (Model, View, Controller). MVC is where there is a model which controls how data is processed and decides what should happen when. The view is what the user can see and how the elements are laid out, and the controller controls changing the UI elements (not necessary how). Even though this model was already established by making the back-end separate from the UI front-end, we extended this pattern one level further by making a model singleton class (called "Model") which talks to the backend, holds most of the variables, and tells the ViewControllers what to do so that the ViewControllers can just hold reference to the UI elements and control small things which are unique to that particular ViewController. We made Model a singleton because we needed to store quite a lot of variables and we wanted a class that can be accessed from anywhere easily, we wanted this class to talk to other classes well and act as almost a "live" middleman.

Throughout the project, and in particular the "Model", we use delegates; this made it really easy to keep reference to the active ViewControllers and call functions on the ViewControllers which change the UI elements (e.g. disable or enable, change value, etc). We also used delegates when it came to elements like "NSTableView". We chose to use this method because Sam has some prior knowledge about "TableViews" and had a much better understanding of delegates rather than bindings so in the interest of time we stuck with what we knew, rather than trying to learn something entirely new even if it may be slightly better.

### How To Use The App

When the app first starts up, the user is presented with the main window with most of the functionality disabled (such as navigation arrows and media control buttons) guide the user to import a JSON file, while also reducing the likelihood that they can break out program by clicking on random things. So to start off, the user must import their library collection from a JSON file, this is done by either clicking the "plus" button on the top panel, or going to File > Import... which will trigger a panel to open to allow the user to navigate through their directories to locate their desired JSON file. For the purposes of this assignment, our JSON file ("test.json") is located in the git repository, inside MediaFiles/test.json. Once the user has selected and 'opened' the JSON, they will see that the library manager is now populated with the collection. The images category is selected by default, from here the user can click on one of the files to bring up a preview of it in the right-hand side preview panel which will display the metadata associated with the file, including a small preview of the notes (if any). The user can select another file to preview or double click on any file to open it and view it - the image replaces where the files used to be shown and now the user can make some

notes on the right-hand side panel, or choose to move onto the next photo by clicking the “next” button at the bottom. If they choose to go back a level, they can hit the “left arrow” at top of the window. Our navigation arrows work like Mac OS Finder but going up a level in the “tree” structure. Since the user has “seen” a file and gone back, they can click the “right arrow” at the top and it will take them back to the view of the image.

If the user is viewing an audio or video file, they can bookmark a playback time (with a label) to come back to later. The bookmarked bookmarks can be found in the left-hand side once the user opens the file. If they wish to make it bigger or present the video on a separate screen, they can hit the “decouple” button to ‘pop’ the video or image out and drag or resize that new window as they see fit. To “recouple” the media player back into the main window, the user just needs to close the media window which opened when they first “decoupled” the media. This “decoupling” feature works for any type of media.

The clear library functionality can be found under File > Clear Library.

## Code Testing

Testing was also very tricky, we knew that the backend worked fine, due to thorough testing for the last assignment so it was more a matter of testing the front end and visual aspects. This resulted in a lot of manual testing because testing the individual functions isn’t as useful as in the first assignment because users can do a lot more in entirely different combinations and probably a lot of ways we don’t expect; Functions may work by themselves in a particular order but in a slightly different order could result in catastrophic results. When testing manually we trying to do what we expect the user to do and also random actions to try and break our program. We run out of time to do thorough testing so we have written up the bugs that we know of in the report. If we had more time we would have got some users to test the program to see if the functionality works as expected and the user experience is pleasant and intuitive.

## Role Separation

Role separation was a little tricky because Sam had a much more experience with Xcode when it comes to UI, so we had to resist the urge to just let Sam do most the work because then Max wouldn’t benefit as much from doing this assignment when it comes to learning and experimentation. We decided that Sam should make an initial start and do a bit of the setting up with things he had knowledge about so Max didn’t have to spend unjustifiable amounts of time learning how to do certain things that Sam could do somewhat easily. This worked out well because while Sam was doing this Max could work on bridging the previous assignment (the backend) to the front end which came together nicely because the roles didn’t conflict much. From then on we separated jobs as they came while trying to stick to fields we know to reduce the learning curve.

## Additional Functionality Added

- Clear all files from the media library - helped with testing.
- Implemented persistent storage of bookmarks made for each file and notes.
- Can jump to a bookmarked time for both audio and video media files.
- When navigating the main window of the media player, when the user selects a category, that category is highlighted to give an indication of the current category. This includes highlighting the current selected file the preview panel is displaying.

- Ability to decouple media content from main window while still having the media controls on the main window interacting with the, now, opened presenting window. The presenting window has a mini scrubbing bar with a simple pause/play button that's display on hover.
- Toggling the state of the import buttons (the 'plus' button and the 'import' menu item) depending on whether the user has already imported a JSON file or not. We implemented this feature so that all files in the library are both imported and exported (with persistent storage) to a singular JSON file. If the user wishes to change 'libraries' (JSON files), they can simply just clear the current library, import the other library they wish to manage, and make changes to that. This decoupling allows the user to manage multiple different collections of libraries using this one media library manager, and all the changes made to a particular library is only saved to itself, without affecting the other libraries.
- Search function- we search if terms are contained within words so they don't have to be an exact match, file types can also be searched. When searching the category table deselects and reselects when the search bar gets emptied. Search bar gets disabled when a file is being viewed.
- Aesthetics - Top menu bar made transparent, all artwork is original.
- Add labels for bookmarks using a popover menu to give more detail about the bookmark. Can delete bookmarks as well as delete them. Bookmarks are hidden in images and documents because it doesn't make sense to have bookmarks there.
- We choose to make the main window a fixed size because of the ability to decouple and use a window which can vary in size.

## Bugs

- Numerous issues with media windows decoupling, especially when trying to play another file while one is open.
- Play pause button can get out of sync with player is player play/ pause is used.
- Audio will sometimes not play when recoupled.