OnCue Projector

Project Progress

OnCue PROJECTOR | Project Progress

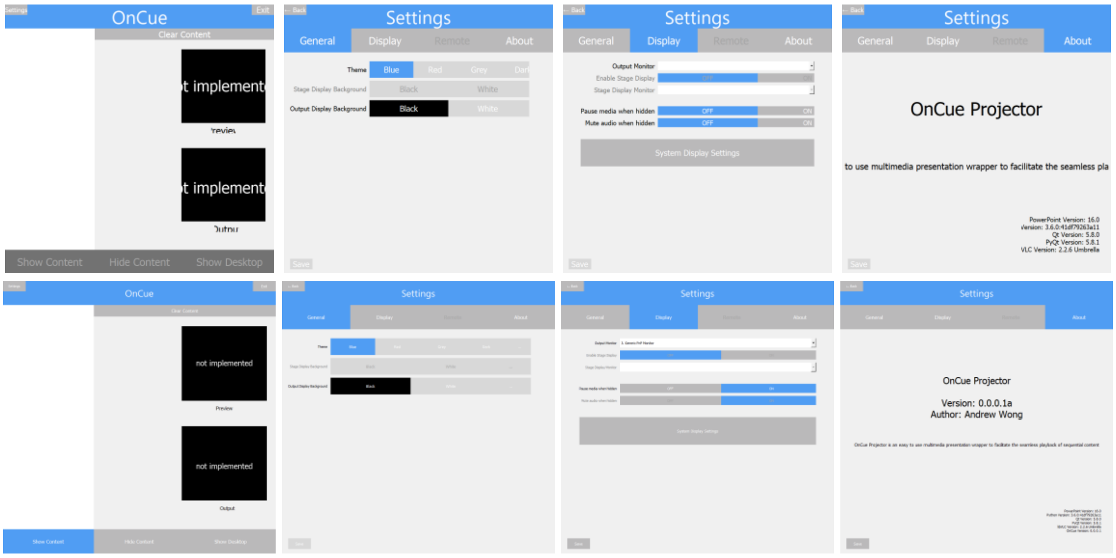
Content is valid as of 5/07/2017

The development of OnCue Projector was the second time I have ever used the Qt framework, and a first time for the PyQt bindings.

> Some issues present in the program is scaling issues, as encountered during development on a Microsoft Surface Book. However, this issue is longstanding and is not fixable on my end, requiring a fix in the Qt framework itself.

Incorrect Look (Scaling Issue)

Correct Look



to use multimedia presentation wrapper to facilitate the seamless pla

PowerPoint Version: 16.0
Version: 3.6.0:41df79263a11
Qt Version: 5.8.0
PyQt Version: 5.8.1
VLC Version: 2.2.6 Umbrella

OnCue Projector
Version: 0.0.0.1a
Author: Andrew Wong

The above screenshots compare the interface look on a scaled environment against its designed look

> In the scope of the program, the ability to collect the machine's display configuration was required (Monitor Name, Current Resolution, Physical Location). In translation to written code in Python, this was quite difficult as the previous methods (prior to the current release) did not return all the required values, but rather only one or two of them. Even those values occasionally didn't return.
Windows is inconsistent.

> PowerPoint integration was painful. The documentation was very bare, and required a lot of external searching to find how to achieve certain functions and methods. The PowerPoint COM (Common Object Module) also did not possess the capability to set the monitor to display a presentation on, so a temporary registry modification had to be made. As there was no 'display only' setting (so the user can't interact with the presentation window), a transparent overlay was implemented to simulate the denial of interaction.

> In terms of memory management / process priority, the progress bar (for media scrubbing) did not update with each change of value. This is because of Qt and Python attempting to optimise the computer's resources, but was easily solved by forcing a redraw of the interface.

> Qt comes with a basic colour picker interface that was something that I needed to get custom theming working. However even as 'basic' as it was, it was still more feature-filled than I needed, so instead of designing my own (which would have taken quite abit of unnecessary effort) I decided to modify the interface to suit my needs.

> I opted to use the libVLC library (what VLC Media Player uses) over the Qt's Media Player and (now-deprecated) Phonon Framework, as libVLC has more support of media formats, and because I have used it before. However I encountered some thread-safe issues with libVLC's event manager (in particular to start and stop the media progress/scrub bar). Though not as efficient, to solve this issue I decided to use a subthreaded timer, rather than a basic timer