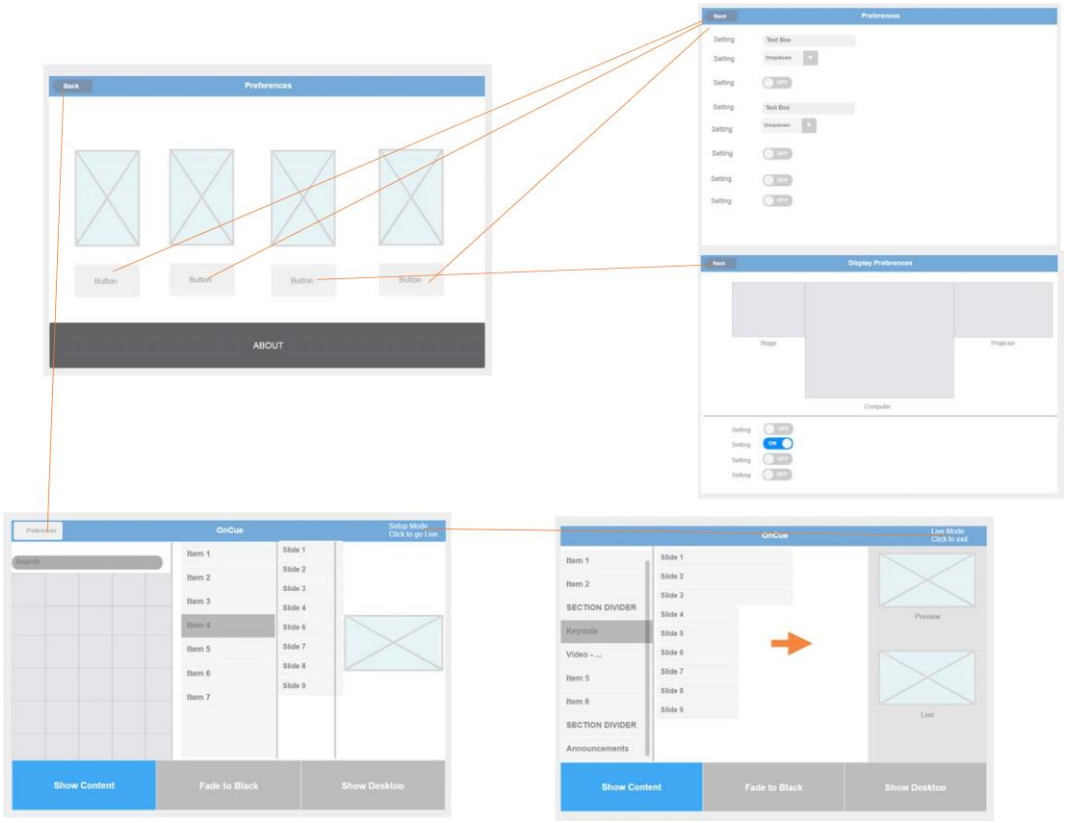
OnCue Projector

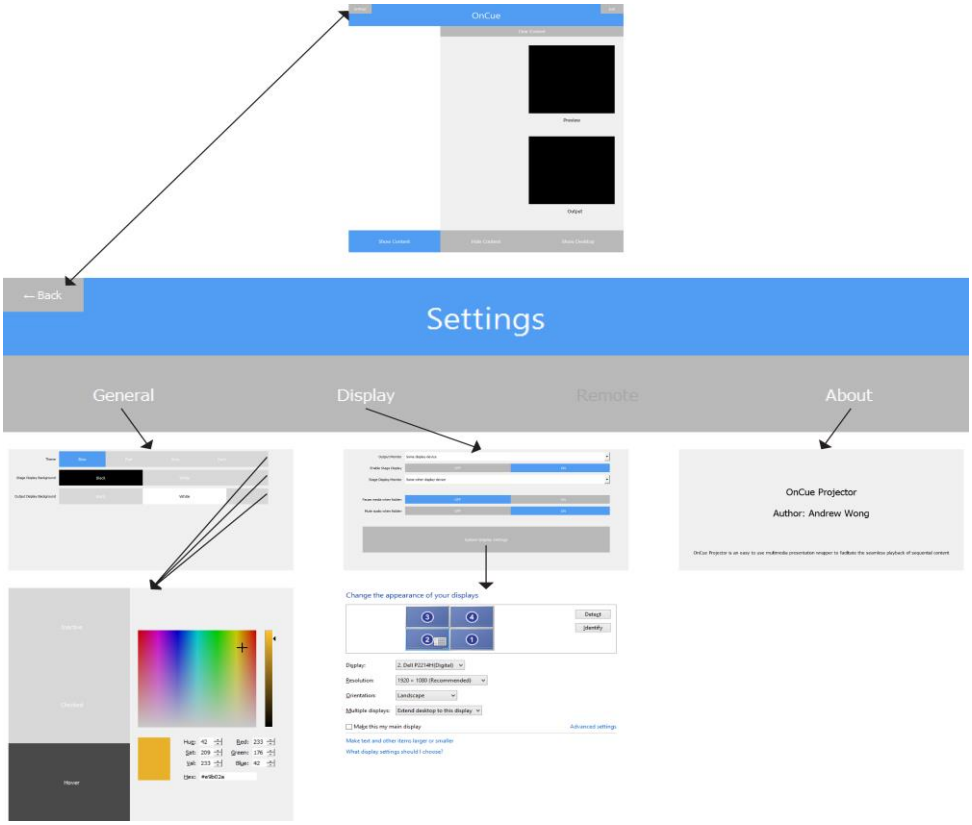
Technical Manual

Content updated 5/07/2017

Original Storyboard and Screen Design

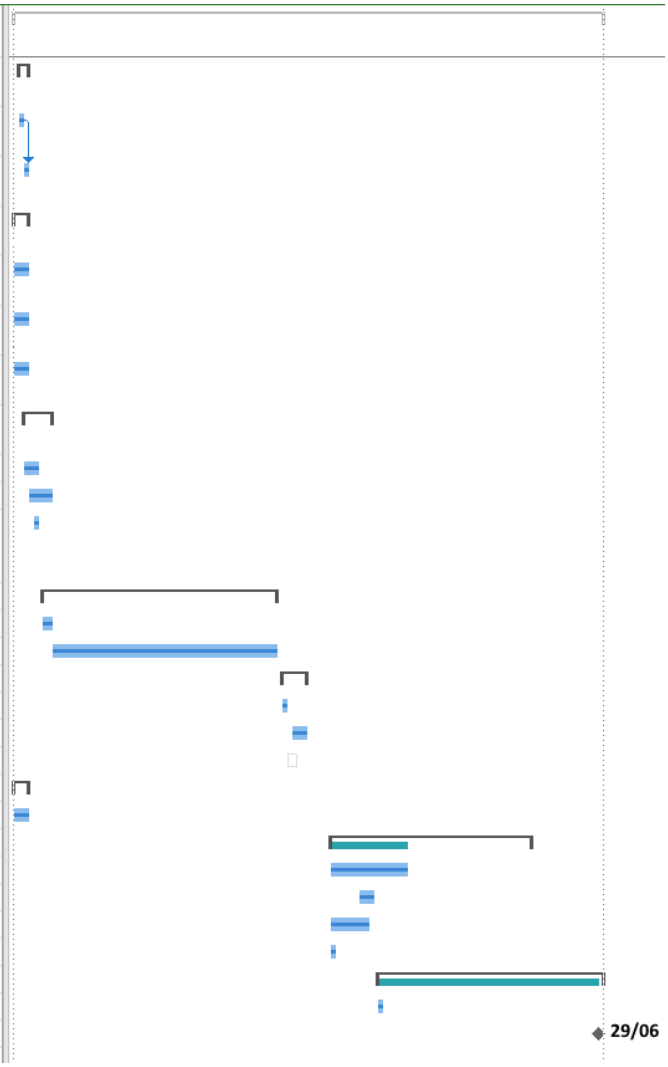


Current Storyboard and Screen Design



Gantt Chart

0	✓	**OnCue Projector**	**94 days?**	**Mon 27/02/17**	**Thu 29/06/17**
1	✓	**Initial Documentation**	**2 days?**	**Tue 28/02/17**	**Wed 1/03/17**
2	✓	Create project proposal	1 day?	Tue 28/02/17	Tue 28/02/17
3	✓	Create gantt chart	1 day?	Wed 1/03/17	Wed 1/03/17
4	✓	**Research and Design**	**3 days?**	**Mon 27/02/17**	**Wed 1/03/17**
5	✓	Find good features of pre-existing solutions	3 days?	Mon 27/02/17	Wed 1/03/17
6	✓	Compile list of features to include	3 days?	Mon 27/02/17	Wed 1/03/17
7	✓	Look at existing interface design and layout	3 days?	Mon 27/02/17	Wed 1/03/17
8	✓	**Planning**	**6 days?**	**Wed 1/03/17**	**Mon 6/03/17**
9	✓	Design interface mockups	3 days?	Wed 1/03/17	Fri 3/03/17
10	✓	Design storyboard	5 days?	Thu 2/03/17	Mon 6/03/17
11	✓	Evaluate availability of software libraries specific to programming language	1 day?	Fri 3/03/17	Fri 3/03/17
12	✓	**Development**	**37 days?**	**Sun 5/03/17**	**Sat 22/04/17**
13	✓	Create interfaces	2 days?	Sun 5/03/17	Mon 6/03/17
14	✓	Create functions	35 days?	Tue 7/03/17	Sat 22/04/17
15	✓	**Testing**	**5 days?**	**Mon 24/04/17**	**Fri 28/04/17**
16	✓	Test PowerPoint Video	1 day?	Mon 24/04/17	Mon 24/04/17
17	✓	Test Video & Audio Media	3 days?	Wed 26/04/17	Fri 28/04/17
18		~~Test Remote~~	~~2 days?~~	~~Tue 25/04/17~~	~~Wed 26/04/17~~
19	✓	**Deployment**	**3 days**	**Mon 27/02/17**	**Wed 1/03/17**
20	✓	Create installer	3 days	Mon 27/02/17	Wed 1/03/17
21	✓	**Documentation**	**31 days?**	**Thu 4/05/17**	**Wed 14/06/17**
22	✓	User Manual	13 days?	Thu 4/05/17	Fri 19/05/17
23	✓	Troubleshooting Guide	3 days?	Wed 10/05/17	Fri 12/05/17
24	✓	Technical Manual	6 days?	Thu 4/05/17	Thu 11/05/17
25	✓	Installation Guide	1 day?	Thu 4/05/17	Thu 4/05/17
26	✓	**Finalisation**	**35 days**	**Sun 14/05/17**	**Thu 29/06/17**
27	✓	Proof read	1 day?	Sun 14/05/17	Sun 14/05/17
28	✓	Submit	0 days	Thu 29/06/17	Thu 29/06/17



Data Dictionary: User Scope (Python)

Field Name	Data Type	Example	Description
display	dict	-	Display settings
display.backgroundaudio	int	0 or 1	Audio behaviour when backgrounded 0: Mute when hidden 1: Play when hidden
display.backgroundmedia	int	0 or 1	Media behaviour when backgrounded 0: Pause when hidden 1: Play when hidden
display.outputbackground	str	“FFFFFF”	Hex code of the output background
display.outputID	int	integer	Monitor for output display
display.stagebackground	str	“FFFFFF”	Hex code of the stage background
display.stageID	int	integer	Monitor for stage display
interface	dict	-	Interface settings
interface.theme	tuple[[str,str,str]]	[“FFFFFF”, “FFFFFF”, “FFFFFF”]	Contains the current application colour scheme

Code Inspection

OnCue Development Launcher

Description: Compiles Qt interface into Python, and launches OnCue

:: helper/OnCue (Compile and Run).bat ::

```
@echo off
set nul=^>nul 2^>nul
cd /d %~dp0\..\source
title OnCue Development

cd OnCue\forms_gen
if %ERRORLEVEL%==1 echo Failed to change directory? && pause && exit
del *.py* %nul%
for %%f in (*.ui) do (
echo ^> Compiling %%f...
pyuic5 "%%f" -o "%%~nf.py"
)
echo # -*- coding: utf-8 -*->__init__.py
echo ^"^"^">>__init__.py
echo OnCue Projector>>__init__.py
echo Copyright 2017 Andrew Wong ^<featherbear@navhaxs.au.eu.org^>>>__init__.py
echo.>>__init__.py
echo The following code is licensed under the GNU Public License Version
v3.0>>__init__.py
echo ^"^"^">>__init__.py
echo.>>__init__.py
echo import pkgutil; __path__ = pkgutil.extend_path(__path__, __name__);
[__import__(modname) for importer, modname, ispkg in
pkgutil.walk_packages(path=__path__, prefix=__name__+'.')]>>__init__.py

cd ..\..

if [%1]==[-norun] exit /b

echo.
echo ^> Executing OnCue
echo.
echo [------ %date% %time% ------]
echo.

"C:\Program Files (x86)\Python\3.6.0\python.exe" OnCue.py

echo.
echo OnCue quit (%ERRORLEVEL%)

echo.
echo [------ %date% %time% ------]
echo.
echo ^> Application terminated... Closing in 3 seconds

timeout /t 3 >nul

exit /b
```

Utility Functions

Description: Provides utility functions to the program (Function explanation in full source code)

:: snippet of oncue/lib/utils.py ::

```
def confine(n, m, M):
    # Confines a value inside a range
    return max(min(M, n), m)

def fourcc(dec):
    # Convert a 4 byte ASCII code into a string
    dec = int(dec)
    return chr((dec & 0XFF)) + chr((dec & 0XFF00) >> 8) + chr((dec & 0XFF0000) >> 16)
           + chr((dec & 0XFF000000) >> 24)

def parseMedia(path):
    # Parse media information of a file
    media = vlc.media_new(path)
    media.parse()

    _title, _artist, _artist = media.get_meta(0), media.get_meta(1), media.get_meta(4)
    m, s = divmod(int(media.get_duration() / 1000), 60)
    h, m = divmod(m, 60)
    _duration = ("%s:" if h else "") + "%02d:%02d" % (m, s)

    _acodec = None
    tracks = list(filter(lambda track: track.type == Vlc.TrackType.audio,
media.tracks_get()))

    if len(tracks) > 0:
        _acodec = fourcc(tracks[0].codec)
        _acodec2 = fourcc(tracks[0].original_fourcc)

    _vcodec = None
    tracks = list(filter(lambda track: track.type == Vlc.TrackType.video,
media.tracks_get()))
    if len(tracks) > 0:
        _vcodec = fourcc(tracks[0].codec)
        _vcodec2 = fourcc(tracks[0].original_fourcc)

    return ...

def identifyFileType(path):
    # Attempts to identify the file type of a given file
    media = vlc.media_new(path)
    media.parse()
    if media.get_duration() > 0:
        return "media"

    matchPatterns = {
        'powerpoint': '^pp[ts]x?$',
    }

    extension = os.path.splitext(path)[1][1:]
    for type in matchPatterns:
        if re.match(matchPatterns[type], extension):
            return type
    return "unknown"
```

Getting Computer Monitor Information

Description: Queries the Windows API for monitor information, and parses it

:: snippet of OnCue.py ::

```
# Enumerate display monitors
monitors = dict([((monitor["Monitor"][0], monitor["Monitor"][1]),(monitor["Device"],
  win32api.EnumDisplayDevices(monitor["Device"]).DeviceString)) for monitor in
  [win32api.GetMonitorInfo(display[0]) for display in win32api.EnumDisplayMonitors()]])

# Populate monitor information
for i in range(app.desktop().screenCount()):
    screen = app.desktop().screenGeometry(i)
    topleft = (screen.left(), screen.top())
    states["screens"][i + 1] = {
        'width': screen.width(),
        'height': screen.height(),
        'physical': monitors[topleft][0],
        'name': monitors[topleft][1],
    }
```

Qt Colour Picker Override

Description: Utilises Qt's QColorDialog class as a basis for OnCue's colour picker

:: snippet of oncue/forms/colorPicker.py ::

```
from PyQt5 import QtCore, QtWidgets

class customQColorDialog(QtWidgets.QColorDialog):
    """
    Color picker override
    """
    def __init__(self):
        QtWidgets.QColorDialog.__init__(self)
        self.setOption(QtWidgets.QColorDialog.NoButtons)
        self.children()[10].children()[16].setText("&Hex:")
        [self.children()[1].setParent(None) for elem in range(7)]
        self.updateColor()
        # Remove elements 1-7
        # Elements 0, 8, 9, 10, 11 are important

    def updateColor(self, colorHex="FFFFFF"):
        foc = self.children()[3].children()[17]
        foc.clear()
        foc.insert("#" + colorHex)

    def getColor(self):
        return self.children()[3].children()[17].text()[1:]
```

Developer Questions

How does OnCue Projector actually work?

OnCue is written in Python (3.6), using the Qt framework through the PyQt bindings.

Audio and video files are handled with the libVLC (VLC SDK) media framework

PowerPoint presentations are handled by Microsoft's PowerPoint COM object

Open Source?

Yes! Check out the source code at github.com/bearbear12345/OnCue

OS X / Linux Version?

Not yet – OnCue Projector uses Windows specific libraries and functions that are not available on other operating system platforms. However, redesign of OnCue for cross-compatibility may happen.

Attribution