

오픈소스 SW – 디지털 사이니지 만들기 팀 프로젝트

5 조

2020071658 김세현

2021006071 정희주

2021029789 황재민

2022004775 정윤아

1. 영상 실행

- 파일을 불러오는 open_file 함수에서 사용자에게 파일을 선택할 수 있도록 하는 부분인
dialog_txt = "Choose media File"
Filename = QtWidgets.QFileDialog.getOpenFileName(self, dialog_txt,
os.path.expanduser('~'))
If not filename[0]: return 부분을 삭제했다.
- 기존의 다운로드한 학교 홍보영상을 video.webm 으로 같은 디렉토리에 저장한 뒤
self.media = self.instance.media_new("video.webm")을 통해 영상을 미리 지정했다.

2. 영상 무한 반복

- self.timer = QtCore.QTimer(self)

self.timer.setInterval(10)

self.timer.timeout.connect(self.update_ui)

self.data_queue = data_queue

self.timer.start()

- **def** update_statusbar(self):

mtime = QtCore.QTime(0, 0, 0, 0)

```
time = mtime.addMSecs(self.mediaplayer.get_time())
```

```
self.statusbar.showMessage(time.toString())
```

```
if self.mediaplayer.get_state() == vlc.State(6):
```

```
    self.open_file()
```

매 초마다 반복 실행되는 `update_statusbar` 함수에서 현재 실행되고 있는 `mediaplayer` 의 상태를 체크하고 영상이 종료되는 시점이 `vlc.State(6)`으로 표현되어 있어서 영상이 종료되는 시점에서 동영상을 다시 재생하기 위해 `open_file` 함수를 재사용한다.

3. Mini_player 가 가장 위에 있도록 설정

```
def __init__(self, data_queue, master=None):
```

```
    QtWidgets.QMainWindow.__init__(self, master)
```

```
    self.setWindowTitle("Mini Player")
```

```
    self.statusbar = self.statusBar()
```

```
    self.statusbar.showMessage("Ready")
```

```
    self.instance = vlc.Instance()
```

```
    self.media = None
```

```
    self.mediaplayer = self.instance.media_player_new()
```

```
    self.setWindowFlag(QtCore.Qt.WindowStaysOnTopHint) 추가
```

`self.setWindowFlags(QtCore.Qt.WindowStaysOnTopHint)`는 PyQt5 에서 제공하는 메소드로 해당 창을 다른 창 위에 고정시키는 역할을 한다.

QtCore.Qt.WindowStaysOnTopHint 는 창의 플래그로, 이 플래그를 설정하면 창이 항상 다른 창 위에 나타나도록 지정해준다. 그러면 다른 창을 클릭하거나 활성화해도 해당 창이 최상위에 유지가 된다.

4. 클라이언트&서버

- #_ = Client("localhost", 10000, data_queue) 주석처리
- 네트워크를 통해 서버와 클라이언트가 굳이 필요없으므로(미리 저장된 파일사용) 처음 import 부분에 ' #from network import Client' 도 주석처리