

 **khw7096** Update load_nk.md

1671941 2 days ago

1 contributor

72 lines (58 sloc) | 2.16 KB

nk 파일 불러오기

작업을 하다보면 반복적인 작업이 발생하는 경우가 많습니다. 반복적인 패턴이라면 패턴 부분을 따로 다른 .nk 파일로 떼어서 저장하고 필요할 때 로딩해서 사용하면 편리합니다. 이러한 패턴을 노드로 제작하고 로딩할 수 있도록 만드는 방법은 라이브러리화 하는 첫단계이기도 합니다.

임시로 .nk 파일을 생성하고 뉴크 Python 창에서 아래 파이썬 코드가 잘 작동되는지 체크합니다.

```
import nuke
nuke.nodePaste("/path/and/path/assets.nk")
```

PySide2를 이용해서 간단하게 라이브러리를 만들고 로딩하는 방법을 배워보겠습니다.

GUI생성

```
from PySide2.QtWidgets import *
from PySide2.QtCore import *

class NkLibrary(QWidget):
    currentItem = ""
    def __init__(self):
        super(NkLibrary, self).__init__()
        self.nklist = QListWidget()
        for i in range(100):
            self.nklist.addItem(QListWidgetItem("/test/"+str(i)+".nk"))
        self.ok = QPushButton("OK")
        self.cancel = QPushButton("Cancel")
        self.oc = QDialogButtonBox(QtCore.Qt.Horizontal)

        # event
        self.ok.clicked.connect(self.bt_ok)
        self.nklist.itemClicked.connect(self.item_click)
        self.cancel.clicked.connect(self.close)

        # layout
        self.setWindowTitle('Nuke Library')
        layout = QGridLayout()
        layout.addWidget(self.nklist, 0, 0)
        layout.addWidget(self.ok, 3, 0)
        layout.addWidget(self.cancel, 4, 0)
        layout.addWidget(self.oc, 5, 0)
        self.setLayout(layout)

    def item_click(self, item):
        self.currentItem = item.text()

    def bt_ok(self):
        print "load %s" % (self.currentItem)
        self.close()

def main():
    # 이미 존재하는 customApp이 있다면 종료시킨다.
    global customApp
    try:
```

```
        customApp.close()
    except:
        pass

# customApp 변수로 NkLibrary를 실행한다.
customApp = NkLibrary()
try:
    customApp.show()
except:
    pass
```

Reference

- https://learn.foundry.com/nuke/developers/11.2/pythondevguide/custom_panels.html