

 kim hanwoong 내용추가함.

4fe61e1 26 days ago

0 contributors

129 lines (110 sloc) | 6.12 KB

경로기반 파이프라인

대부분 회사는 경로기반으로 파일을 관리합니다. 이 방식을 많이 사용하는 이유는 팀 또는 회사가 경로를 규약하고 프로젝트를 진행하면 특정 소프트웨어에 종속될 필요없이 약속된 경로 규칙만으로 작업이 가능하기 때문입니다. 이미 많은 회사들이 이 방식으로 작업하고 있습니다. VFX, 애니메이션 이외에도 언리얼 엔진 역시 파일 메니징은 경로기반을 두고 있습니다.

콘텐츠를 만드는 회사에서 프로그램을 작성할 때 Python같은 언어등을 이용해서 경로를 잘 다룰 필요가 있습니다. 이번시간에는 경로를 다루는 방법에 대해 다루어 보겠습니다.

여러분이 앞으로 일하게될 회사가 어떤 경로구조를 가지게 될 지 기본적으로 상상해 봅시다.

1. VFX, 애니메이션, 게임 모두 에셋 개념이 있을 것 입니다. 특히 에셋 데이터는 자산이 됩니다.(중요도가 높습니다.)
2. VFX는 최소 샷 형태로 작업이 일어납니다.
3. 애니메이션도 최소 샷 형태로 작업이 일어납니다. 필요시 에피소드가 추가 될 수 있습니다.
4. 애니메이션, VFX는 시퀀스 단위로 묶일 수 있습니다.
5. 게임은 시나리오, 장소(레벨) 또는 월드 단위로 묶일 수 있습니다.
6. 오래된 회사일 수록 일반적으로 스토리지 서버가 많을 수 있습니다. 스토리지 서버가 많다는 것은 100% 확신할 수 없지만 일반적으로 경로가 복잡할 수 있습니다.

수년전부터 VFX 시장은 자금이 많은 중국시장으로 부터 데이터가 넘어올 때 F00_0010 , BAR_0010 형태로 샷이 정리되어 넘어왔습니다. 한국의 대부분 VFX 시장은 이미 이러한 이름으로 경로 파이프라인이 만들어져 있습니다.

회사에 따라 다르겠지만 아래 예시와 비슷한 패턴의 경로를 가지게 됩니다. 업무를 하면서 중간에 필요한 경로들은 회사마다 내부 회의를 통해서 다르게 추가됩니다.

보통 많은 할리우드 회사들은 프로젝트 스토리지의 Root 폴더로 show또는 project라는 단어를 많이 사용합니다. <http://opencolorio.org/userguide/contexts.html> 문서를 보면 showcfg 같은 단어는 각 show의 설정파일이라는 의미도 있구요. 우리나라 회사들도 이러한 특징을 많이 반영하며 회사 경로 구성시 따라하고 있습니다.

```
/project/프로젝트명/shot/F00/0010/task/...
/project/프로젝트명/shot/BAR/0010/task/...
```

프로젝트, 샷명, 팀명 이외에 중간에 추가되는 경로로는..

문서파일, 에셋경로, 상대방으로 부터 받은 데이터, 현장데이터, 레퍼런스, 컨셉, 외주로 보낼 데이터, 샷 작업파일, 최종 배포파일, 버전, Proxy 파일, 프로젝트 고유 설정파일등 프로젝트에 필요한 경로가 추가됩니다.

아래는 lazypic에서 사용하는 폴더 구조입니다.



```
├── config
│   ├── ocio
│   │   └── config.ocio
│   └── thumbnail
│       └── shot
│           └── F00_0010_plate_v001.jpg
├── doc
│   └── concept
│       ├── concept01.jpg
│       └── concept01.krita
├── edit
│   ├── edit_v001.edl
│   └── edit_v001.otio
├── in
├── out
├── shot
│   └── F00
│       └── 0010
│           ├── comp
│           │   ├── F00_0010_comp_v001.natron
│           │   ├── F00_0010_comp_v001.nk
│           │   ├── in
│           │   │   ├── source01.jpg
│           │   │   └── source02.jpg
│           │   ├── out
│           │   │   ├── F00_0010_comp_v001
│           │   │   │   └── F00_0010_comp_v001.0001.exr
│           │   └── outproxy
│           │       ├── F00_0010_comp_v001
│           │       │   └── F00_0010_comp_v001.0001.jpg
│           ├── light
│           │   ├── F00_0010_light_v001.blend
│           │   ├── in
│           │   │   └── F00_env.hdri
│           │   ├── out
│           │   │   ├── F00_0010_light_v001
│           │   │   │   └── F00_0010_light_v001.0001.exr
│           ├── matchmove
│           │   └── F00_0010_matchmove_v001.blend
│           ├── plate
│           │   └── v001
│           │       └── F00_0010_plate_v001.0001.exr
│           └── plateproxy
│               └── v001
│                   └── F00_0010_plate_v001.0001.jpg
```

tree 명령어 설치

linux

```
# yum install tree
```

macOS

```
$ brew install tree
```

실습

- tree 명령어의 설치도 최초에 리눅스 셋팅시 자동 설치되도록 github에 추가합니다.
- linux 에는 useradd or adduser, groupadd 명령어가 존재합니다. addproject 명령어를 만들어봅시다.
- addproject 명령어의 역할은 프로젝트 폴더 생성, 내부 필요한 폴더를 생성하는 python 스크립트입니다.
- 에러처리 : project가 존재하는지 체크할 것.
- 프로젝트명의 시작이 숫자로 시작하는지 체크할 것.

- 소문자 영문 이외에 다른 글씨가 존재하는지 체크할 것.
- 윈도우즈에서 사용하면 안되는 글씨가 존재하는지 체크할 것.
 - CON
 - PRN
 - AUX
 - CLOCK\$
 - NUL
 - A~Z
 - COM1, COM2, COM3, COM4
 - LPT1, LPT2, LPT3

Reference

- <https://www.thewindowsclub.com/create-restricted-files-folders-con-aux-nul-click-windows>