

애니메이션 효율적 공정관리를 위한 캐릭터 셋업

'Master Character Set-up' for Effective Management of 3D Animation Pipe Line

채일진

동서대학교 디지털콘텐츠학부

Eeljin Chae(trecine@gmail.com)

요약

기존의 상업적인 3D CG 애니메이션 제작 공정 구조는 부서간의 병목현상으로 모든 부서가 동시에 작업을 할 수 있는 효율적인 구조로 이루어지지 않고 있다. 예를 들어, 애니메이팅 작업을 하기 위해선 선행되어야 할 모델링과 Set-Up 단계가 끝나기 전에는 애니메이팅 작업을 할 수가 없다. 그 결과, 제작 일정의 지연, 제작비의 손실을 발생시키게 되고, 또한 선행된 작업에 수정을 가할 경우 수정이 끝날 때 까지 애니메이팅 작업은 중단된다는 결과를 야기 시킨다. 이런 문제는 표준으로 정해 놓은 "마스터 캐릭터"의 적용으로 도움을 줄 수가 있는데, 프로젝트의 시작과 동시에 모델링과 set-up 그리고 다른 어떤 작업과도 독립적으로 애니메이팅 작업을 시작 할 수 있고 수정 작업도 독립적으로 수행이 가능해서 애니메이션 제작공정을 효율적으로 관리할 수 있다.

■ 중심어 : | 공정 | 제작 관리 | 마스터캐릭터 | 애니메이팅 |

Abstract

The current 3D CG animation pipe line, which is the production system, does not have an effective solution that all animation departments are able to start their jobs independently against other animation departments. It is generally acknowledged that we could not do animating task before the modeling and set-up task. This kind of process has caused loss of labour. We could expect lasting problems which is stopping marking animating process before the previous process done again in case of requiring editing the jobs which are modeling and set-up. We could solve these "bottleneck" situation with "master character" defined as standard, so that animate and retake the job independently against other departments. These new process could accelerate the animation pipe line system.

■ keyword : | Pipe Line | Production Management | Master Character | Animating |

1. 서론

1인 이상의 노동력으로 생산할 제품이 필요할 때에 노동 인력간의 역할 분담 및 시간분배가 결과물의 생산력을 결정하게 될 것이다. 컴퓨터를 사용한 영화의 특

수효과나 3D 애니메이션 제작은 가장 짧은 기간 동안 가장 화려하게 성장한 산업중의 하나라고 할 수 있다. 하지만 그 빠른 성장에도 불구하고 제작 방식 및 운영 방식에는 생각처럼 효율적인 적용이 이루어지지 않아 왔다.

* 본 연구는 산업자원부의 지역혁신센터의 연구결과로 수행되었습니다.

접수번호 : #070315-001

접수일자 : 2007년 03월 15일

심사완료일 : 2007년 04월 16일

교신저자 : 채일진, e-mail : trecine@gmail.com

2D 애니메이션의 하청 용역 작업을 수행하면서 얻은 기술로 창작 애니메이션 제작에는 실패를 거듭할 수밖에 없었던 우리 내부의 문제가 상위의 기술 개발과 이전이 없었다는 것을 감안할 때 IT의 발달로 성장한 3D 애니메이션은 역사가 짧은 분야여서 비교적 동시대에 기술들을 공유하며 시작할 수 있었다. 네트워크와 IT가 발달한 상황에서 CG를 이용한 디지털 콘텐츠 분야에 정책적인 지원 자금들이 늘어나며 제작을 하고자 하는 회사가 많이 생겨나고 그중에선 막대한 투자비를 받아 대규모 3D 애니메이션 영화제작 및 TV 시리즈 제작에 들어갔다. 대표적인 예로 영화제작을 진행하던 '디지털 드림 스튜디오(DDS)'의 '아크-미개봉', '빅필름'의 '엘리시움-2005년 상영', 그리고 TV시리즈인 '디지털 드림 스튜디오(DDS)'의 '런담-2001년 방영', '시네픽스'의 '큐빅스-2002년 방영'라는 애니메이션이다. 그러나 아직도 TV시리즈를 제외한 장편영화들의 제작 계획이 무산되거나 출시 기간이 늦어지는 이유는 투자비를 회수할 만큼이라도 작품의 퀄리티가 창출되지 못해서이다. 그렇다면 그 이유는 무엇일까, 단순히 비효율적인 인력구조나 방만한 경영, 미숙한 마케팅 기법 또는 부족한 제작 경험이 아닌 근본적인 원인이 있다. 그것은 애니메이션을 일정한 수준으로 정해진 스케줄에 맞게 제작하는 애니메이션 공정관리 시스템의 기반 하에서 제작을 할 수 있는 환경 구축이 미비하였었기 때문이다.

이에, 본 연구에서는 제작파이프라인에서의 작업공정과 마스터캐릭터의 효율성에 관하여 그 생성과정과 적용 사례를 들어 키 애니메이션의 효율성을 높이기 위한 연구와 키 필터링과 정렬을 위한 프로그램화를 통해 적용범위를 확대할 수 있는 연구를 제안한다.

II. 본 론

1. 제작 공정 관리의 부재

애니메이션 제작자인 디지털 드림 스튜디오(DDS, 2004년에 'The digo'로 개명함)는 '아크, Ark'라는 장편 3D 애니메이션 영화제작 작업을 통해 많은 기술을 쌓

아온 것으로 알려져 있다. 서울대학교 및 KAIST와 함께 연구한 립싱크 관련 기술이나 다양한 캐릭터의 Set-Up기술, 많은 투자를 통해 그 어떤 회사보다도 관련 기술을 많이 보유하고 있었다. 그런 회사가 프로젝트들을 취소하고 대규모 감원을 하는 점은 모두를 어리둥절하게 만들었지만, 결국 애니메이션 기획, 제작사는 기술 보유보다는 얼마나 수익을 창출할 수 있느냐가 회사의 존립에 기초라는 것이 자명한 이치이다. 다른 산업과 달리 애니메이션 산업이 아직 이렇다 할 성과와 성공 사례가 없다는 것은 막대한 정책 지원 자금이 들어간 수 년이 지난 현재에도 관련 분야에 종사하는 전문가들에게 당면과제가 되어 있다. 한 두 가지의 원인으로만 그 전체적인 책임 소재를 밝혀 낼 수는 없을 것이다. 결국 기획력의 부재와 제작 경험의 부족, 마케팅의 부진, 산업자본 채투자 불가능의 악순환이 계속되는 것이다. 다른 분야야 외적인 환경에서 그 원인을 찾아낼 수 있겠지만, 제작 경험은 그동안 전문제작사에서 많은 시간과 인력을 동원하여 내적으로 일정부분 따라잡을 수 있었을 터이나 아직도 요원한 이유가 바로 효율적인 공정관리의 부재에 그 원인이 있다고 할 수 있는 것이다.

2. 일반적인 제작 파이프라인

제작 파이프라인이란 영화 특수효과나 애니메이션을 제작하기 위한 프로덕션 단계를 효율적으로 운영할 수 있는 무형의 역할 분담 및 스케줄이다. 미국의 3D 제작 회사들은 100년 넘는 전통의 2D 애니메이션 제작 및 영화 제작을 경험으로 제작 파이프라인을 3D 애니메이션에 맞게 변형, 적용하여 사용해 온 반면에, 한국은 CG 콘텐츠 분야를 전혀 새로운 분야라고 생각해 기존의 아날로그 식 제작 방법의 이전을 거부하거나 작업의 전문화 및 기술 이전에 소극적인 태도들이 이런 제작 공정을 원활히 돌아가게 하는 원활제 역할을 못해 몇몇 부서들의 정체현상을 가져와 작업의 효율성을 떨어트린 점이 가장 큰 인력의 부실적인 운영의 원인이 되었다.

제작 파이프라인의 구성과 역할은 다음과 같이 조직도를 그려 볼 수 있는데, 애니메이션을 제작하는 프로덕션 단계에 있어서 크게 모델링, 캐릭터 Set-Up, 레이

아웃 및 애니메이션, 텍스처, 매핑, 라이팅, 컴프의 7가지 부서가 필요하다.

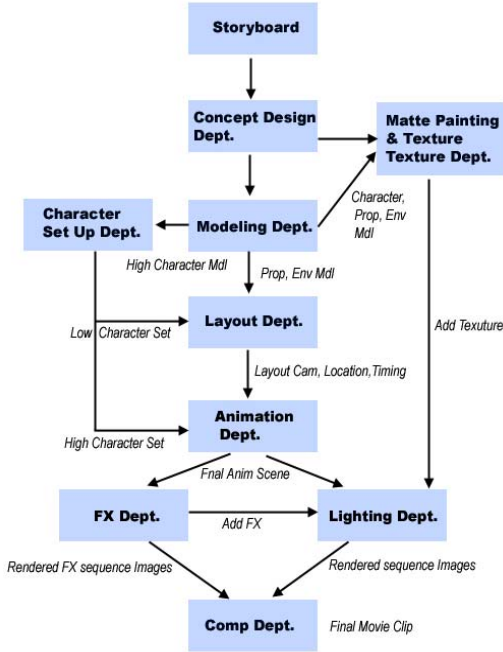


그림 1. 애니메이션 파이프라인 조직도

애니메이션 프로덕션의 제작 순서는 다음과 같다.

표 1. 프로덕션의 제작 순서

1. 감독이나 클라이언트가 스토리보드를 확정짓게 된다.
2. 그 스토리보드를 바탕으로 컨셉 디자이너가 디자인을 결정하게 된다. 컨셉 디자인에는 캐릭터의 디자인 외에 칼라맵, 배경 디자인이 포함된다.
3. 컨셉디자인이 결정되면 텍스처 부서에서는 곧바로 매트 페인팅이 필요한 환경 매핑 작업과 캐릭터 및 도구들의 매핑작업을 시작하게 된다.
- 3a. 컨셉디자인을 바탕으로 모델러는 캐릭터 모델링, 배경모델링, 소도구 모델링을 하게된다.
4. 모델러가 캐릭터 모델링을 마치게 되면 캐릭터 Set-Up 부서에서는 모델링에 맞춰 뼈대를 구축하고 스키닝 대신 mesh들을 조인트의 길이에 맞게 잘라 조인트에 Parenting하는 것으로 로우 Set-Up을 하여 레이아웃 및 애니메이션 부서로 보내게 된다.

5. 레이아웃 부서는 로우 Set-Up 캐릭터와 소도구 모델링을 3D 환경에 배치하고 촬영 카메라와 Cut 타이밍을 결정하게 된다.
6. 레이아웃 부서에서 결정된 카메라와 타이밍의 scene에 애니메이션부서에서는 Blocking Animation 작업을 하게 된다.
7. 캐릭터 Set-Up 부서에서 모델링 부서로부터 받은 하이모델로 조인트와 mesh간의 스키닝을 하여 하이 캐릭터 Set-Up 모델을 만들게 된다.
8. 애니메이션 부서에서 애니메이션 작업된 로우 캐릭터 Set-Up 모델을 하이 캐릭터 Set-Up 모델과 교체하여 디테일 애니메이션 작업을 마치게 된다.
9. FX부서와 라이팅 부서는 디테일 애니메이션이 완료되기 전까지 필요한 효과를 위한 R&D작업을 하게 된다. 애니메이션 부서의 디테일 애니메이션이 끝나면 필요한 다이내믹 효과를 씬에 적용하고, 라이팅 효과가 필요 없는 다이내믹 효과들은 직접 렌더링을 완료해 결과물을 컴프 부서로 넘기게 된다.
10. 라이팅 부서는 애니메이션 디테일이 끝난 씬 또는 FX 부서에서 넘긴 씬을 받아 효과적인 라이팅을 배치하고 렌더링을 지시하여 결과물을 컴프 부서에 넘긴다.
11. 마지막으로 컴프 부서에서는 렌더링된 이미지 레이어들을 합성, 편집하여 사운드와 합성하여 최종 결과물을 만들게 된다.

2.1 일반적인 파이프라인에서의 병목현상

위와 같은 순서에 의해 CG를 이용한 3D 애니메이션이나 영화특수효과가 제작이 된다. 서로 다른 부서들의 일정이 얽혀 있기 때문에 단 한 번이라도 문제가 발생하면 병목현상이 쉽게 생기게 된다.

2.1.1 작업공정

첫째, 모델링 부서가 콘셉 디자인 부서에서 캐릭터와 도구, 배경에 필요한 건물들의 디자인이 나오게 되면 먼저 3D 모델링을 시작하게 된다.

둘째, 모델링이 클라이언트로부터 컨펌이 나기 전까지는 직접적인 작업이 연결되는 Set-Up과 레이아웃 및 애니메이션 부서들은 쉴 수밖에 없는 상황이 된다.

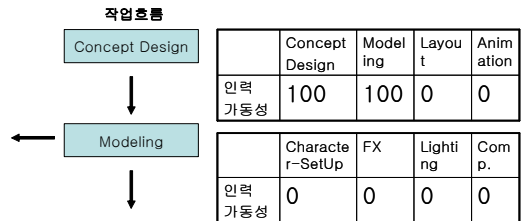


그림 2. 둘째 단계의 작업 흐름과 인력 가동성

셋째, 모델링 부서에서 캐릭터 모델링이 끝나 Set-Up 부서로 로우모델 Set-Up이 진행되고 있는 동안에도 레이아웃 및 애니메이션 부서는 공백이 계속된다. 하이 모델을 로우모델로 변형시키고 조인트를 생성하여 모델의 사이즈에 맞춰 길이를 맞추는 시간이 필요하기 때문이다.

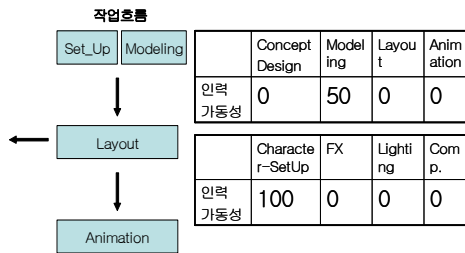


그림 3. 셋째 단계의 작업 흐름과 인력 가동성

넷째, 로우모델 Set-Up이 끝나고 레이아웃 부서에 전해지면 이제 모델링 부서의 할일은 대폭 줄어들기 시작하고 도구나 배경 모델링이 끝나게 되는 작업 중반쯤 되면 스케줄의 공백이 예상된다. Set-Up 부서도 마찬가지로 작업 중반의 모델링이 끝나는 시점부터 애니메이션 부서의 디테일 애니메이션이 들어가는 시점까지 외에는 공정 초기와 중반이후에는 공백이 예상된다.

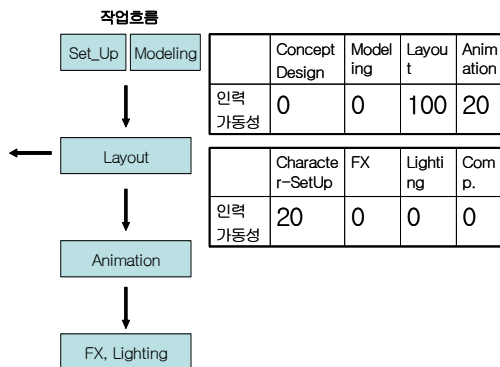


그림 4. 넷째 단계의 작업 흐름과 인력 가동성

2.2 수정작업 시 문제점

위 작업의 흐름에서 인력의 효율적인 이용뿐만 아니라 애니메이션 작업의 필연적인 Retake 작업에 들어가

면 문제는 더 심각하다. 일반적으로 Retake은 스케줄의 여유가 매우 없으므로 작업의 효율이 어떤 애니메이션 기술보다 중요한 요소가 된다. 그러나 위의 흐름대로라면 예를 들어 클라이언트가 디자인을 바꾸길 요구한다면 모든 작업을 작업에 흐름에 따라 다시 해야 한다. 때에 따라선 그 정도가 심각하다면 수정된 디자인에 따라 모델링 작업부터 시작하여 texture, set-up, (joint의 수정 시 애니메이션까지 수정이 필요할 수도 있음) 그리고 rendering 까지 작업의 흐름에 따라 수정이 되어야 한다.

지금까지 CG 영상 제작을 하는데 있어서 7개의 부서의 인력들을 효과적으로 배치하여 일정대비 공정의 효율성을 높이기 어려운 이유가 제작 부서들 간의 작업이 동시에 이루어질 수 없다는데 원인이 있는 것을 알아보았다. 제작 공정을 효율적으로 관리 및 운영하기 위해 필요한 프로그램으로서 마스터 캐릭터의 활용에 대해서 알아보자.

3. 마스터 캐릭터 Set-Up

모델링과 Set-Up 그리고 레이아웃을 포함한 애니메이션 부서의 일정별 공정을 맞추기 위해 첫째, 제작의 흐름을 원활하게 하고 둘째, 클라이언트의 수정사항을 빨리 수행하기 위함이다. 모델링, Set-Up, 애니메이션 부서의 동시 작업을 가능하게 하기위해서 관련 데이터의 피드백을 원활하게 할 수 있는 중간 매체의 Set-Up을 '마스터 캐릭터 Set-Up' 이라고 하자.

3.1 개요

마스터 캐릭터 Set-Up은 모델링 부서에서 모델링이 제작되기 시작하는 것과 동시에 Set-Up 부서에서 콘셉 디자인에 맞게 캐릭터의 사이즈를 변형하여 레이아웃 및 애니메이션 부서에서 레이아웃 및 애니메이션 블리킹 작업을 시작할 수 있게 해주는 중간 매체이다. 이것이 레이아웃 부서에서 사용하는 로우 캐릭터 set의 대용으로 사용되며 최종적으로 하이 캐릭터 set과의 교체 대상 set이 되는 셈이다. 캐릭터의 사이즈는 키, 어깨의 넓이, 팔 길이, 그리고 다리 길이로 정해지는데 콘셉 디자인에 의해 사이즈들이 정해지므로 Set-Up 부서에서

모델링 부서와 동시에 마스터 캐릭터를 이용하여 간단히 사이즈를 정하여 레이아웃 부서로 전달하여 로우 캐릭터 set 대응으로 사용할 수 있으므로 모델링, Set-Up, 레이아웃을 포함한 애니메이션 부서가 동시에 작업이 이루어지는 기반이 된다. 마스터 캐릭터란 캐릭터의 뼈대를 구성하는 조인트들의 Naming과 축의 각도 등을 통일한 표준적인 캐릭터 Set이므로 갖고 있는 데이터들 또는 조인트에 주어진 키 값들은 레이아웃용이든, 로우 캐릭터나 하이 캐릭터의 Set-Up과도 한 번에 전체 키복사에 의해 마스터 캐릭터 set으로부터의 전이가 가능하게 된다[1].

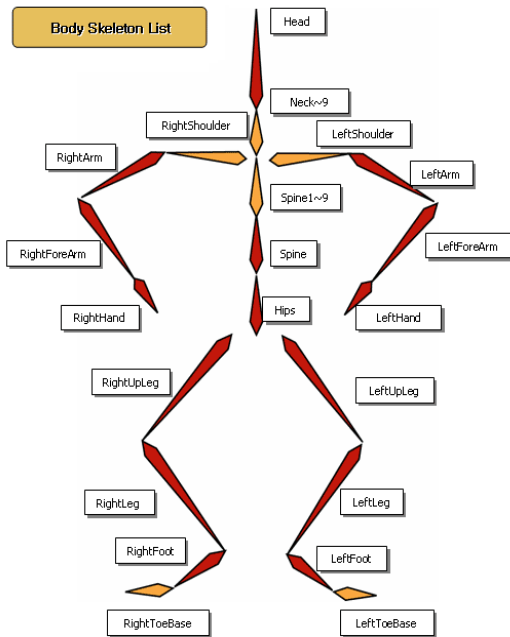


그림 5. 마스터캐릭터의 구조와 Naming

3.2 캐릭터 생성의 규칙성

마스터 캐릭터는 기본적으로 로우 및 하이 모델 Set-Up에 들어가는 캐릭터의 뼈대를 구성하는 조인트들과 동일한 개수 및 Naming을 갖추어야 한다. 많은 애니메이션 제작회사들이 서로 다른 Naming 규칙을 가지고 있기 때문에 그렇게 구성이 이루어지면 캐릭터 Set-Up에 관한 노하우 및 데이터들이 재사용이 될 수

없다. Naming의 공통화는 데이터들의 호환성을 높이는 데 사용될 수 있기 때문에 캐릭터 set의 호환 문제 뿐 아니라 차후의 애니메이션 액션의 데이터베이스 구축에도 도움이 되어 제작 파이프라인의 효율성을 극대화하는데 큰 도움이 될 수 있다. [그림 5]와 [그림 6]은 호환성을 높일 수 있는 구조의 한 예이다. 예를 들면 소니 엔터테인먼트 스튜디오의 몬스터 하우스(2006년 개봉)를 비롯해 많은 애니메이션 제작회사들이 모션 캡처를 이용하는 방향으로 제작이 진행되므로 모션 캡처에서 규정하는 캐릭터 뼈대의 조인트 개수와 Naming을 기반으로 삼아 마스터 캐릭터 Set-Up의 Naming 구조를 만들었다.

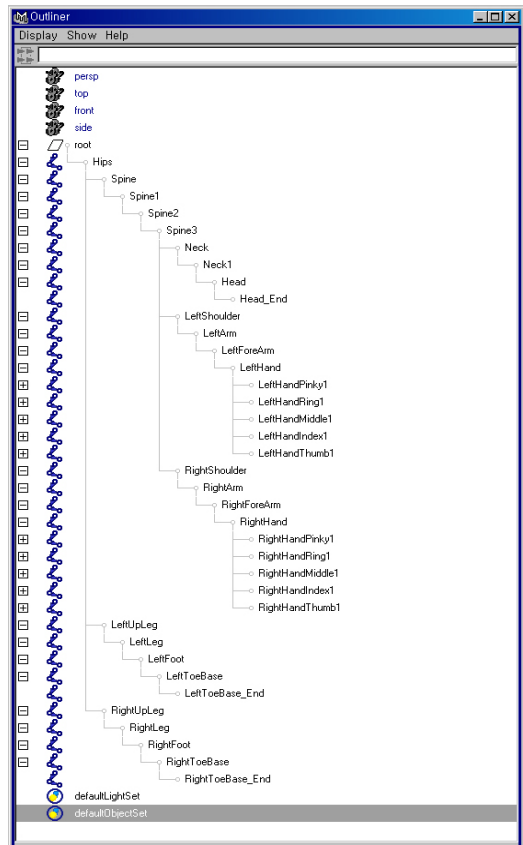


그림 6. 마스터캐릭터의 Joint의 Tree 구조화한 Naming

3.3 마스터 캐릭터 set 자동생성 Mel 프로그램 개발

마스터 캐릭터 Set-Up을 위한 Naming 구조를 규정 한 후 이를 매번 수작업으로 캐릭터를 만들기는 오류 발생의 경우가 있을 가능성이 있음과 레이아웃 부서로 마스터 캐릭터 Set-Up을 빨리 전달하기 위해서는 빠른 생성 및 키 이전 틀이 필요하다[4]. 마야의 Mel 스크립트를 사용하여 Naming 규칙에 맞게 지정된 조인트 자동 생성 프로그램 개발을 다음과 같이 하였다.

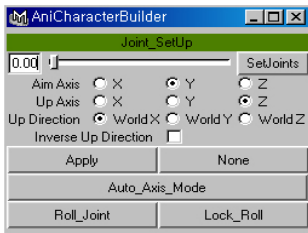


그림 7. 마스터캐릭터 자동생성 툴

3.4 마스터 캐릭터 비율 결정과 그 비례에 따른 Joint 자동 생성

Maya에서 적당한 크기의 마스터 캐릭터용 Joint를 수작업 한다. 그러면 마야 스크립트 창에 수행된 작업에 관한 Mel이 기록된다. 아래 그림은 기록된 Mel에 각 Joint의 Naming만 추가 하였다[3].

```

7 joint -n ("Spine3") -p 0 6.891055 -0.152886 ;
8 joint -n ("Neck") -p 0 7.186518 -0.159495 ;
9 joint -n ("Neck1") -p 0 7.478675 -0.167178 ;
10 joint -n ("Head") -p 0 7.783262 -0.167077 ;
11 joint -n ("Head_End") -p 0 8.790487 -0.165196 ;
12
13 //----Arm-----//
14
15 select -cl ;
16 select -r ("Spine3");
17 joint -n ("LeftShoulder") -p 0 4.16682 7.048957 -0.0503493 ;
18 joint -n ("LeftArm") -p 0 0.979334 6.860268 -0.185754 ;
19 joint -n ("LeftForeArm") -p 2.339154 6.860268 -0.335231 ;
20 joint -n ("LeftHand") -p 3.529149 6.860268 -0.218778 ;
21
22
23 //----Make fingers--//
24 select -r ("LeftHand");
25 joint -n ("LeftHandMiddie1") -p 3.946921 6.825509 -0.421468 ;
26 setAttr -type float3 ("LeftHandMiddie1.jointOrient") 0 180 0;
27 joint -n ("LeftHandPinky2") -p 4.08019 6.778645 -0.444994 ;
28 joint -n ("LeftHandPinky3") -p 4.174916 6.744305 -0.462079 ;
29 joint -n ("LeftHandPinky4") -p 4.276439 6.708378 -0.479735 ;
30
31 select -r ("LeftHand");
32 joint -n ("LeftHandRing1") -p 3.986748 6.855872 -0.288302 ;
33 setAttr -type float3 ("LeftHandRing1.jointOrient") 0 180 0;
34 joint -n ("LeftHandRing2") -p 4.116817 6.838737 -0.299698 ;
35 joint -n ("LeftHandRing3") -p 4.246894 6.8013 -0.309414 ;
36 joint -n ("LeftHandRing4") -p 4.384995 6.765012 -0.319676 ;
37
38 select -r ("LeftHand");
39 joint -n ("LeftHandMiddle1") -p 3.981062 6.855872 -0.170268 ;
40 setAttr -type float3 ("LeftHandMiddle1.jointOrient") 0 180 0;
41 joint -n ("LeftHandMiddle2") -p 4.13189 6.817004 -0.16469 ;
42 joint -n ("LeftHandMiddle3") -p 4.259473 6.785531 -0.161087 ;
43 joint -n ("LeftHandMiddle4") -p 4.415801 6.743446 -0.156166 ;
    
```

그림 8. 스크립창에 기록된 것에 네이밍만 추가한Mel[5-7]

위의 Mel은 수행된 결과물로 나온 것이므로 캐릭터의 크기를 조정 할 수가 없으므로 프로그램적으로 조정 할 수 있도록 수정 한다. 물론 위의 Mel을 실행하면 처음 만든 Joint를 똑 같이 만들어 낸다.

```

1028 joint -n ("LeftHandIndex1") -p (3.968257*SizeJoint) (6.855872*SizeJoint) (-0.0539687*SizeJoint) ;
1029 joint -e -zso -oj xyz -sao yup LeftHand;
1030 string $select4[];
1031 $select4 = 'ls -sl';
1032 setAttr -type float3 ($select4[0] + ".jointOrient") 0 180 0;
1033 joint -n ("LeftHandIndex2") -p (4.101582*SizeJoint) (6.807148*SizeJoint) (-0.0203013*SizeJoint) ;
1034 joint -e -zso -oj xyz -sao yup LeftHandIndex;
1035 joint -n ("LeftHandIndex3") -p (4.206101*SizeJoint) (6.771573*SizeJoint) (0.00488898*SizeJoint) ;
1036 joint -e -zso -oj xyz -sao yup LeftHandIndex;
1037 joint -n ("LeftHandIndex4") -p (4.320893*SizeJoint) (6.730081*SizeJoint) (0.0340428*SizeJoint) ;
1038 joint -e -zso -oj xyz -sao yup LeftHandIndex;
1039 for($i=0; $i < 4; $i++)
1040 {
1041     pickWalk -d up;
1042 }
1043 joint -n ("LeftHandThumb1") -p (3.626183*SizeJoint) (6.856105*SizeJoint) (-0.0846037*SizeJoint) ;
1044 joint -e -zso -oj xyz -sao yup LeftHand;
1045 string $select5[];
1046 $select5 = 'ls -sl';
1047 setAttr -type float3 ($select5[0] + ".jointOrient") 102.820 -30.010 -148.344;
1048 joint -n ("LeftHandThumb2") -p (3.727409*SizeJoint) (6.746502*SizeJoint) (0.0289405*SizeJoint) ;
1049 joint -e -zso -oj xyz -sao yup LeftHandThumb;
1050 joint -n ("LeftHandThumb3") -p (3.804588*SizeJoint) (6.672021*SizeJoint) (0.111667*SizeJoint) ;
1051 joint -e -zso -oj xyz -sao yup LeftHandThumb;
1052 joint -n ("LeftHandThumb4") -p (3.878619*SizeJoint) (6.597383*SizeJoint) (0.198578*SizeJoint) ;
1053 joint -e -zso -oj xyz -sao yup LeftHandThumb;
    
```

그림 9. 네이밍 추가와 크기 조절을 위한 변수 추가[5-7]

위 그림은 단순한 수치에 변수를 추가하여 크기를 조정 할 수 있도록 하였다. 마스터캐릭터 자동 생성 툴에서 슬라이더 바가 크기를 조절 할 수 있는 곳이 된다.

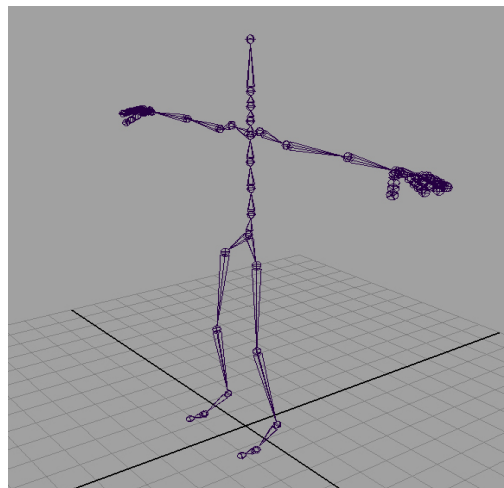


그림 10. 마스터 캐릭터의 Maya에서의 Joint 구조

위 그림은 Maya에서 자동 생성된 마스터 캐릭터의

Joint구조를 보여준다.

3.5 Mesh의 추가

추가로 조인트에 비주얼을 위한 mesh를 추가시킬 수 있게 프로그래밍이 진행된다.

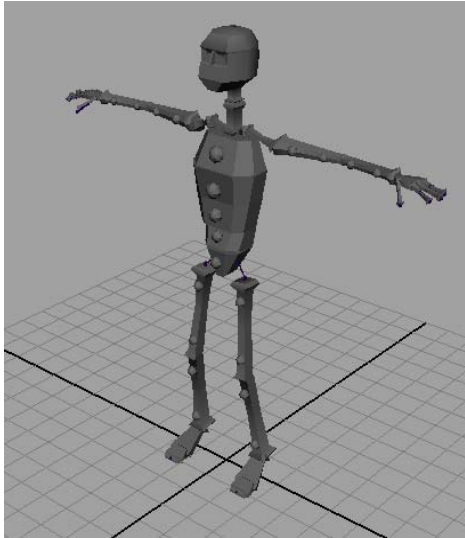


그림 11. Mesh가 캐릭터의 비례에 맞게 자동 생성

```

55
56 string[] jointMesh[30] = { "Head", "Neck", "Spine", "Spine", "LeftShoulder", "LeftArm", "LeftForearm", "LeftHand", "LeftChest", "LeftShoulder", "LeftForearm", "LeftHand", "LeftChest", "LeftShoulder",
57 "LeftForearm", "LeftHand", "LeftChest", "Spine", "RightShoulder", "RightArm", "RightForearm", "RightHand", "RightChest", "RightShoulder", "RightForearm", "RightHand", "RightChest",
58 "RightShoulder", "RightForearm", "RightHand", "RightChest", "Spine", "Spine", "Spine", "Hip", "Hip", "LeftLeg", "LeftLeg", "LeftFoot",
59 "LeftFoot", "LeftForearm", "RightLeg", "RightLeg", "RightForearm", "RightForearm" };
60
61
62 string[] jointMesh[30] = { "Head", "Neck", "Spine", "Spine", "LeftShoulder", "LeftArm", "LeftForearm", "LeftHand", "LeftChest", "LeftShoulder", "LeftForearm", "LeftHand", "LeftChest", "LeftShoulder",
63 "LeftForearm", "LeftHand", "LeftChest", "Spine", "RightShoulder", "RightArm", "RightForearm", "RightHand", "RightChest", "RightShoulder", "RightForearm", "RightHand", "RightChest",
64 "RightShoulder", "RightForearm", "RightHand", "RightChest", "Spine", "Spine", "Spine", "Hip", "Hip", "LeftLeg", "LeftLeg", "LeftFoot", "LeftFoot",
65 "LeftForearm", "LeftForearm", "RightLeg", "RightLeg", "RightForearm", "RightForearm" };
66
67
68 string[] jointMesh[30] = { "Head", "Neck", "Spine", "Spine", "LeftShoulder", "LeftArm", "LeftForearm", "LeftHand", "LeftChest", "LeftShoulder", "LeftForearm", "LeftHand", "LeftChest", "LeftShoulder",
69 "LeftForearm", "LeftHand", "LeftChest", "Spine", "RightShoulder", "RightArm", "RightForearm", "RightHand", "RightChest", "RightShoulder", "RightForearm", "RightHand", "RightChest",
70 "RightShoulder", "RightForearm", "RightHand", "RightChest", "Spine", "Spine", "Spine", "Hip", "Hip", "LeftLeg", "LeftLeg", "LeftFoot", "LeftFoot",
71 "LeftForearm", "LeftForearm", "RightLeg", "RightLeg", "RightForearm", "RightForearm" };
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
    
```

그림 12. Mesh 자동화 스크립트[5-7]

III. 결론

마스터캐릭터의 자동화 프로그램으로 콘셉 디자인 완료에 따라 모델링 부서의 모델링 제작과 캐릭터 Set-Up 부서의 마스터 캐릭터 Set-Up의 자동 생성에 의한 캐릭터 Set-Up이 동시에 진행될 수 있게 된다. Set-Up 부서에서는 마스터캐릭터 set을 생성하여 콘셉 디자인의 사이즈에 맞게 키와 어깨너비를 지정하여 레이아웃부서와 애니메이션 부서에 전해주면 한 두 시간의 시간차이는 있지만 모델링 Set-Up 부서와 비슷한 시간에 작업을 시작할 수 있게 되어 작업 시간의 효율성을 높일 수 있게 된다. 다음 그림은 마스터 캐릭터 Set-Up을 이용하여 모델링, 캐릭터 Set-Up 및 레이아웃을 포함한 애니메이션 부서에서 거의 동시에 제작이 진행될 수 있는 조직도를 보여준다. 이 세 부서 이외의 부서 중 Comp를 제외한 부서 중 텍스처 부서는 역시 처음부터 텍스처 및 매핑 제작이 들어가게 되고 FX 부서 및 라이팅 부서도 R&D 테스트가 계속 진행 되므로 후반작업에 투입되는 컴프를 제외하고 총 6개의 모든 부서들이 동시에 프로젝트를 진행하는 결과를 가져온다.

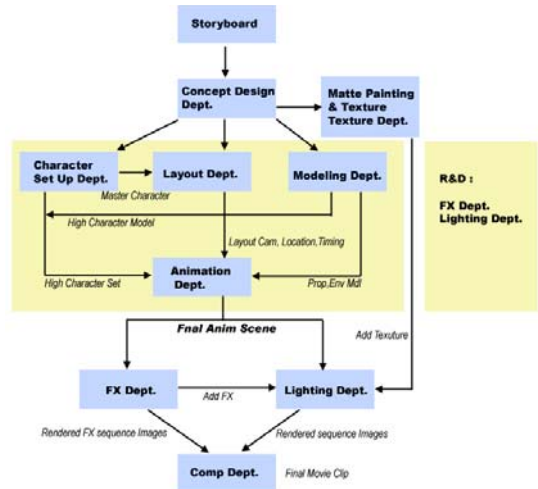


그림 13. 마스터 캐릭터 Set-Up을 이용한 파이프라인 조직도

애니메이션 제작 공정상의 파이프라인의 작업 능력 향상을 위해서는 각 제작 부서들이 동시에 작업이 진행

될 수 있는 과정이 필요하다. 가장 작업의 병목현상이 집중되는 모델링과 캐릭터 Set-Up, 레이아웃을 포함한 애니메이션부서의 문제점을 파악하여 그에 대한 해결책으로 '마스터 캐릭터 Set-Up'의 생성 및 프로그램화하여 효율성을 높이고 그 적용과정을 알아보았다. 이를 활용한 프로젝트로 동서대'Digital Content Center'에서 제작을 하여 완성한 Charlie and Chocolate Factory (2005, W.B 배급, 팀 버튼 감독), PC 게임[8] 애니메이션의 경우 6주 만에 9개 무비 클립, 14여분을 완성(일반적인 경우 15분 full 3D CG 경우 최소 3~4개월 소요)하였다. 앞으로는 마스터 캐릭터 Set-Up의 키 애니메이션 효율성을 높이기 위해 IK 자동생성 및 모션 캡처 데이터의 키 값 수정의 효율성을 높이기 위해 키 필터링과 정렬을 위한 프로그램화를 통해 적용범위를 확대할 수 있는 연구가 진행되어야 할 것이다.

저 자 소 개

채 일 진(Eeljin Chae)

정회원



- 1998년 2월 : 중앙대학교 사진학과 (예술학사)
- 2001년 5월 : Academy of Art Univ.(USA) Motion Picture & Television (MFA)
- 2001년 6월 ~ 2002년 10월 : (주) 튜브 픽처스, 프로듀서

- 2002년 11월 ~ 2004년 8월 : (주) DCD Korea, 대표
- 2004년 9월 ~ 현재 : 동서대학교 디지털콘텐츠학부 게임 전공 교수

<관심분야> : 멀티미디어, 소프트웨어, 문화콘텐츠, 기획, 마케팅

참 고 문 헌

[1] <http://www.keymake.com/index.php?pl=97>

[2] R. Stuart Ferguson, *Practical Algorithms for 3D Computer Graphics*, A K Peters, Ltd. 2001.

[3] John Edgar Park, *Understanding 3D Animation Using Maya*, Springer, 2004.

[4] Brad Clark Jofn Hood Joe Harkins, *inspired 3D Advanced Rigging and Deformations*, HOMSON, 2005.

[5] A. D. David, Gould Complete Maya Programming, 2000.

[6] A. D. David, Gould Complete Maya Programming Volume II, 2005.

[7] M. R. Wilkins and C. Kazmier, *Mel Scripting for Maya Animators*, 2003.

[8] http://www.amazon.com/Take-71042521714-Charlie-Chocolate-Factory/dp/B0009R5KM2/ref=sr_1_3/002-1887979-8583243?ie=UTF8&s=software&qid=1176973014&sr=8-3