

# CGI (Computer Generated Image)는 예술인가 ? 기술인가 ?

(부제: Inhouse Programmer, 그들은 누구인가?)

(주)미디어팩토리 대표이사

홍익대학교 디지털미디어학부 겸임교수 김종보

## 차례

- I. 서론
- II. 본론
- III. 결론

## I. 서론

CGI의 초창기 세월을 보낸 사람들이라면 지금의 CGI 기술이 초기의 기술과는 하늘과 땅 차이라는 것을 알 것이다. 최근 각광을 받고 있는 3D CGI 뿐 아니라, 2D CGI역시 예외는 아니다. 새로운 기술의 CGI를 이용하여 각종 Contents를 만들면서 자신들의 기술을 과시하려던 시절도 있었다.

예를 들면 AKIRA의 경우 오토바이를 타고 달릴 때 뒤에 남는 붉은색 조명의 Trail 은 그 이전 시대에는 없었던 장면이다. 디즈니의 알라딘의 경우 양탄자를 타고 나르는 장면은 오토바이처럼 생긴 VR(Virtual Reality) 장비를 이용해 만들었다는 사실 역시 많이 알려진 사실 일 것이다.

과거의 작업들은 프로그래머와 아티스트들의 공동 작업에 의해 만들어져왔다. 또한 매우 복잡한 여러 종류의 프로그램들을 같이 사용하여 최상의 결과물을 만들기 위해 노력해왔다. 과거 헐리우드의 전통적인 CGI제작 양식은 Alias (MAYA의 이전 이름)를 이용하여 Modeling 을 하고 SoftImage (XSI의 이전이름)을 이용하여 Animation Key작업을 하고 최종 렌더링은 Renderman 을 이용하여 작업을 하는 방식이었다. 서로 다른 종류의 프로그램들을 사용하다보니 프로그램간의 형식이 달라 이를 조절할 수 있는 별도의 프로그램이 필요했고, 이런 프로그램의 자체개발이 필요하게 되었다. 이런류의 Artist의 영역외의 일위해 만들어진 새로운 Job이 Inhouse Programmer이다.

이들과 프로그램 개발사들간의 유기적인 관계가 이루

어졌고 이를 토대로 지금의 프로그램들이 완성되어졌다.

## II. 본론

### 한국의 현실

실질적인 영화의 이름을 거론하진 않겠다. 그러나 영화에 쓰이는 Source를 비디오 편집에 쓰는 장비로 작업 한후 다시 Film으로 Convert하여 스크린에 건다면 가로 Pixel의 Size가 720 \* 480 Pixel이니 가로가 10M인 스크린의 경우 1 Pixel의 크기가 무려 1.4CM나 되는 것이다. 간단한 산수계산만 미리 한다면 미연에 방지할 수 있었던 일을 불필요한 작업을 하고 결국은 쓸모없는 일에 시간과 비용을 낭비했던 일도 많았다. 당시한국상황은 CGI 관련 회사에 TD (Technical Director)가 거의 전무하던 상황이었다. 당시 CGI 영화의 대표라 할수있는 터미네이터의 경우 해상도가 2000 \* 2000, 토이스토리의 경우 가로가 1600에 이르는 Cineon 포맷이었던 것을 보면 정말 아찔한 작업이었던 것이었다.

또 다른 현실은 만약 Inhouse Programmer 가 있더라도 그들이 일할 자리나, 그에 맞는 보수를 줄만한 회사가 많지 않다는 것도 만만치 않은 문제 중 하나이다.

한국의 대표적인 CGI회사의 예를 들어봐도 영구아트 무비의 경우는 Artist는 40명에 이르나 Inhouse Programmer 는 단 1명에 불과하다. 물론 특수한 큰 프로젝트가 있을 경우 예는 별도의 프로그래머들을 추가로 고용한다고 한다. 또한 인디펜던스의 경우도 총 50명의 Artist가 있으며 Inhouse Programmer 는 1명뿐이다.

매우 많고 다양한 Digital Character 들이 유기적으로 움직이는 군중씬의 변천 과정을 보면, 노트담의 곱추 -> 라이언킹 -> Matrix -> 반지의 제왕 -> 300에 이르기 까지 수많은 진보에 진보를 넣고 있다. 이렇듯 수많은 Character를 동시에 움직일 수 있는 프로그램의 경우 Massive라는 별도의 비싼 프로그램이 나와 있기는 하지만, 어찌 플러그인이 Massive만 필요하겠는가? 하드웨어, 소프트웨어, 각종 프로그램간의 유기적인 자료의 교환 등 수많은 산재한 문제들을 풀기에는 그에 걸맞은 엔지니어가 필요한 게 현실이다.

간단한 예로 요즘 초등학교들도 즐겨 쓰는 포토샵의 경우도 RGB 컬러와 CMYK 컬러간의 변화만 해도 100% 호환이 되지 않음은 전문가라면 알 수 있는 얘기이다. 수많은 그래픽 포맷, 3D 포맷, 기타 잡스런 파일의 변환만 하더라도 그 일은 끝이 없을 것이다.

국내에서는 “비”가 출연한 카드회사 광고로 익숙하지만 영화 “Next”에서 보여준 객체복사기술 및 최근 개봉한 JUMPER에서 보여준 합성의 경우 CGI의 끝없는 가능성을 보여주고 있다. 예전에 포레스트 검프에서 잠깐 보여줬던 탁구장면의 완성은 코미디 영화인 Balls of Fury (분노의 핑퐁)에서 이루어졌다.

이렇듯 하루가 다르게 변하고 발전해가는 CGI기술의 진보에는 Artist 만이 존재하는 것은 아니다. 그들 뒤에

는 시나리오작가, 연출감독, 특수효과감독, 그래픽감독 등의 이름 이 거론되지만, 그들뒤에 숨어있는 Inhouse Programmer 들은 항상 그림자 뒤에 있고 그 존재가 잘 드러나지 않고 있다. 그에 따라 존재의 가치는 물론 필요성조차도 모르는 이들이 많은 게 사실이다. 그러나 그들의 존재를 무시한다면 수많은 일들을 진행함에 엄청난 추가 비용과 시간이 필요로 함은 물론, 이것을 통한 경쟁력 저하가 산업의 도태까지도 이룰 수 있음을 명심해야 할 것이다.

그러면 예술과 기술이 만나는 정점을 한번 살펴보자. 과거 미국식의 작업을 예를 들면 모델링의 최고인 Alias (현재 Maya) 로 모델링 한 것을 SoftImage에서 Animation작업을 할 경우 기본적인 그래픽에 대한 법칙이야 같겠지만, 서로 다른 회사에서 만든 Format이 다른 화일들 간의 호환성이 이루어 져야한다. 이는 마치 TGA, TIFF, JPG, PSD등의 각기 다른 그래픽 파일 포맷을 서로 호환해주는 프로그램이 별도로 필요한 것과 마찬가지로 이다. 그래픽은 일반인들도 사용하는 범용의 프로그램이 된지 오래고 대표적으로 Photoshop 등의 프로그램이 있어 손쉽게 이런 작업을 할 수 있으나 3D는 그렇지 않다.

아래의 표 1은 Lightwave3D 의 Modeling 파일의 Binary 파일의 예이다.

표 1. Lightwave3D 의 Modeling 파일의 Binary 파일의 예

```

0000000h: 46 4F 52 4D 00 00 12 34 4C 57 4F 32 54 41 47 53 ; FORM..4LWO2TAGS
0000010h: 00 00 00 04 42 6F 78 00 4C 41 59 52 00 00 00 12 ; ...Box.LAYR...
0000020h: 00 00 00 00 00 00 00 00 3F 00 00 00 00 00 00 ; .....?.....
0000030h: 00 00 50 4E 54 53 00 00 04 C8 3E 0F 5C 29 3D 0F ; ..PNTS...?.\)=.
0000040h: 5C 29 BF 38 51 EB BE 0F 5C 29 3D 0F 5C 29 BF 38 ; \?Q毓\)=.\?
0000050h: 51 EB BE 0F 5C 29 3E 91 EB 85 BF 38 51 EB 3E 0F ; Q毓\>뽕뽕8Q?.
0000060h: 5C 29 3E 91 EB 85 BF 38 51 EB 3D F5 C2 8F 3D 0F ; \>뽕뽕8Q?慮?.
0000070h: 5C 29 3E A3 D7 0C BD F5 C2 8F 3D 0F 5C 29 3E A3 ; \>W.쌘뽕=. \>?
0000080h: D7 0C BD F5 C2 8F 3E 91 EB 85 3E A3 D7 0C 3D F5 ; ?쌘뽕>뽕?W.=?
0000090h: C2 8F 3E 91 EB 85 3E A3 D7 0C BE 0F 5C 29 3D 0F ; 뽕>뽕?W.?)=.
00000a0h: 5C 29 BF 38 51 EB BE 0F 5C 29 3E 91 EB 85 BF 38 ; \?Q毓\>뽕뽕8
00000b0h: 51 EB BD F5 C2 8F 3D 0F 5C 29 3E A3 D7 0C BD F5 ; Q鬱慮?. \>W.쌘
00000c0h: C2 8F 3E 91 EB 85 3E A3 D7 0C 3E 0F 5C 29 3D 0F ; 뽕>뽕?W.>.\=.
00000d0h: 5C 29 BF 38 51 EB 3D F5 C2 8F 3D 0F 5C 29 3E A3 ; \?Q?慮?. \>?
00000e0h: D7 0C 3E 0F 5C 29 3E 91 EB 85 BF 38 51 EB 3D F5 ; ?>.\>뽕뽕8Q??
00000f0h: C2 8F 3E 91 EB 85 3E A3 D7 0C BE A9 51 82 3D 94 ; 뽕>뽕?W.쌘Q??
0000100h: E3 BD BF 17 0A 3D BE A9 51 82 3E 35 F6 FD BF 17 ; 絨?.=쌘Q?5埠?
0000110h: 0A 3D BE AE 63 1F 3D 94 E3 BD 3E 75 C2 92 BE AE ; .=쌘c.=뽕?u뽕뽕
0000120h: 63 1F 3E 35 F6 FD 3E 75 C2 92 3E A9 51 82 3D ; c.>5埠>u뽕>쥬?
.....
    
```

첫부분에 있는 Header를 시작으로 일정한 Pattern에 의하여 기록된 파일이다 모든 프로그램의 자료들은 이런 형태의 2진수 형태의 파일(Binary File이라부름)로 저장되거나 ASCII 형태의 파일로 저장된다(표 2).

이런 류의 파일을 다루는 일에는 훌륭한 Artist들의 경우에도 일단 쳐다보기도 싫을 것은 물론 무슨 내용인지를 파악하는데 만도 오랜 시간이 걸릴 것이다. 그러나 프로그래머의 경우는 매우 친숙한 파일들일 것이다, 다만 그들이 그래픽을 모른다는데 문제가 있을 뿐이다.

매우 실질적인 예를 하나 들어보자. 2007년 대단한 흥행을 몰고 왔던 마이클베이 감독의 Transformer 의 경우 Autobot의 리더인 Optimus Prime 한 가지 객체만도 총 10,108개의 독립적인 객체에 의해 이루어졌다. 이 각각의 객체들이 서로 유기적으로 링크되어 만들어 지는 장면은 정말 장관이 아닐 수 없다. 그러나 이런 10,000개

가 넘는 객체들을 단순하게 Bone이나 Joint 기능만으로 만든다고 가정하면 2007년 개봉은 물론이고 지금까지도 Rigging을 하고 있을지도 모른다.

Rendering의 경우도 예외는 아니다. Renderfarm은 기본이고 개별적인 Studio에서 사용 가능한 Renderblade, RenderBoXX 등 수도 없이 쏟아지는 다양한 장비들을 사용하자면 Artist들은 본업을 제쳐놓고 기술적인 공부만을 하기도 바쁠 것이다.

가장 이상적인 방법은 Great Artist 가 Great Engineer를 겸하면 되는 것인데, 역사적으로 보면 레오나르도 다빈치 이후로 몇명 나오지 않았던 것으로 기억한다. 예전 UTOPIA라는 세계적으로 유명한 Progressive Rock그룹을 이끌었던 그룹의 리더 Todd Rundgren을 만났던 것은 1995년 Florida에서 있었던 Siggraph 세미나 예서였다.

표 2. ASCII 파일의 예

|  |  |
|--|--|
| <pre>LWSC 3  FirstFrame 1 LastFrame 300 FrameStep 1 PreviewFirstFrame 0 PreviewLastFrame 300 PreviewFrameStep 1 CurrentFrame 0 FramesPerSecond 30  LoadObjectLayer 1 Objects/tunnel.lwo ShowObject 6 3 ObjectMotion NumChannels 9 Channel 0 { Envelope 5 Key 0 0 0 0 0 0 0 0 0 Key 0 25 0 0 0 0 0 0 0 Key 0 5.9666667 0 0 0 0 0 0 0 Key 0 7.9333334 0 0 0 0 0 0 0 Key 0 10 0 0 0 0 0 0 0 Behaviors 1 1 }</pre> | <pre>Channel 1 { Envelope 5 Key 0 0 0 0 0 0 0 0 0 Key 0 25 0 0 0 0 0 0 0 Key 0 5.9666667 0 0 0 0 0 0 0 Key 0 7.9333334 0 0 0 0 0 0 0 Key 0 10 0 0 0 0 0 0 0 Behaviors 1 1 } Channel 2 { Envelope 5Key 0 0 0 0 0 0 0 0 0 Key 0 25 0 0 0 0 0 0 0 Key 0 5.9666667 0 0 0 0 0 0 0 Key 0 7.9333334 0 0 0 0 0 0 0 Key 0 10 0 0 0 0 0 0 0 Behaviors 1 1 } Channel 3 { Envelope 5 Key 0 0 0 0 0 0 0 0 0 Key 0 25 0 0 0 0 0 0 0 Key 0 5.9666667 0 0 0 0 0 0 0 Key 0 7.9333334 0 0 0 0 0 0 0 Key 0 10 0 0 0 0 0 0 0 Behaviors 1 1 }</pre> |
|--|--|

UTOPIA 이후 솔로로 독립하여 작곡, 작사, 노래, 연주 등을 혼자서 하는 Singer Song Writer 로서 활동을 하던 중, CGI기반의 Music Video까지 혼자서 작업을 한 이후로 CGI강사로 Siggraph로 나왔었다. 물론 CGI를 도와준 동료도 있었다고는 하지만, 가수가 CGI제작은 물론 Network Rendering을 위한 프로그램까지 했다하더라도 당시로는 굉장한 센세이션이 일어났었다.

2001년도 Siggraph에서 상을 받은 Animusic의 Pipe Dream이라는 영상은 바로 이런 기술과 예술의 협력에 있어서는 좋은 예가 될 수 있다고 생각된다. 이들은 직접 음악화일인 MIDI와 MAYA를 접목시키는 프로그램을 만들어 CGI제작에 활용했다. 이런 관점에서 수도 없이 많은 국내의 CGI를 가르치는 각종 교육기관들은 Art와 Science가 너무 동떨어져 있지 않은가 하는 우려가 있다.

Lightwave3D의 LScript (Lightwave Script Language)를 비롯하여 최근 각광을 받고있는 MEL (Maya Embedded Language)를 시작으로 C++ 기반의 언어로 프로그램을 제작하여 Plug-in시키는 API까지 다양한 개발용 도구들이 있으며 이들을 이용하여 모델링을 시작으로 Motion Capture 또한 Animation 제작 및 Rendering에 이르기 까지 매우 다양하게 이용되고 있는게 현실이지만, 이에 대한 대비가 한국에는 얼마나 이루

어져 있는지 의문을 던져본다.

### III. 결론

Artist가 공학을 공부할 것인지 아님 Engineer가 Art를 공부할 것인지에 대한 논의는 마치 닭이 먼저냐, 달걀이 먼저냐에 대한 얘기처럼 정해진 해답은 없는 것 같다. 오래전 자동차의 경우는 Artist가 공학을 공부하는 게 좀 더 나은 결과가 나온다는 사례가 있었다고는 하지만, 모든 분야가 같지는 않을 것이다. 기존의 CGI관련 프로그램들이 막강해 지면서 CGI회사들의 경쟁력은 Inhouse Programmer를 통한 기술력이 경쟁력의 중심이 되는 시기에 이르렀다.

왜 한국에는 In-House Programmer가 없을까? 스스로 이 질문을 수도 없이 해봤다.

첫 번째 이유로는 인식의 부재일 것이다. 무슨 일을 하는지, 그 일에 대한 가치는 어떻게 되는지 CGI에 종사하는 사람들조차 관심이 없는 것은 정말 큰 문제가 아닐 수 없다.

둘째로는 교육기관의 부재이다. 아무리 공부를 하고 싶어도 적합한 교재 및 교육기관이 없다. 미국에 대한 전문책이 5~6종(종합적인 책은 제외)이 있지만, 한국에는



▶▶ 그림 1. <http://www.animusic.com> Pipe Dream의 한 장면

단 한편도 없다는 것이 그 단면일 것이다.

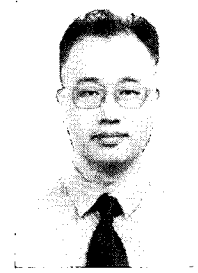
마지막으로 불투명한 미래를 들 수 있다. CGI전문 업체에 취업 및 기타 미래를 보장할 수도 없을 뿐더러 일반 프로그래머들이 병역특례업체에 많이 가는 것에 비하면 Inhouse Programmer는 그런 기회조차 많지 않다는 것도 매우 심각한 문제이다. 물론 군대를 가느냐? 아님 병역특례를 하느냐에 대해 어느 것이 좋은가에 대한 논쟁은 아니다, 다만 모든 일에 대한 공평한 기회의 제공에 대한 문제를 말하는 것뿐이다.

이에 대한 대안으로 그래픽 계통에서의 교육에 있어서 Art적인 교육 외에 Programmer로서의 자질을 같이 키우는 커리큘럼이 절대적으로 필요하다. 이에 걸맞는 교원과 교재의 개발역시 필요함은 물론이다. 또한 이들에 맞는 적합한 일자리들이 확보되어야 한다. 이를 위하여 CGI를 경영하는 경영자들부터 마인드를 바꿀 필요가 있다고 판단된다. Inhouse Programmer의 자질이 아무리 뛰어나도 이들의 가치 및 활용에 대한 이해가 없다면 이들은 진흙속의 진주가 될 뿐이다.

한국의 CGI기술력은 이미 세계수준이라고 말할 수 있다. 그러나 여기서 말하는 기술력이라 함은 Artist들의 역량을 말하는 것일 것이다. Inhouse Programmer의 수준은 뛰어난 몇 명은 있을 수 있겠지만, 국가적인 차원에서 본다면 아마도 형편없을 것이다. 진정한 의미의 CGI강국이 되기 위해선 Artist의 배양 못지않게 Inhouse Programmer의 배양이 필요함을 인지하고 이를 위한 산업계 전반의 이해와 뒷받침이 필요한 시기이다.

### 저자소개

#### ● 김 중 보(Jong-Bo Kim)



- 1988년 2월: 한국 항공대학교 항공기계공학과(공학사)
- 2000년 7월: 숭실대학교 정보과학대학원 정보미디어학과(공학석사)
- 2001년 8월 ~ 현재: (주)미디어팩토리 대표 이사
- 2003년 ~ 현재: 홍익대학교 조형대학 디지털미디어학부 겸임교수

<관심분야> : Racing, 음악, Physical Media, 그래픽알고리즘, Acoustic Theory

### 참고문헌

- [1] [www.animusic.com](http://www.animusic.com)
- [2] <http://vfxworld.com/?sa=adv&code=1e242f07 & atype=articles&id=3337>
- [3] [www.newtek.com](http://www.newtek.com)
- [4] [www.alias.com](http://www.alias.com)
- [5] [www.pixar.com](http://www.pixar.com)