
게임제작사례를 통한 제작파이프 라인 관리체계의 발전방향

- 텍스처 매핑 제작 파이프라인 중심으로 -

The Development Direction of the Game Production Pipeline Management System-based
on the Example of Game Production - Focus on the texture

김현조

경기대학교 다중매체영상학부 애니메이션 전공

Hyun-Jo Kim(aliaskim@hanmail.net)

요약

세계적으로 게임 산업은 문화 콘텐츠 산업의 핵심으로 인정받으면서 2008년경에는 전체규모가 1,000억 달러를 넘어서며 문화 콘텐츠 산업 성장을 실질적으로 주도할 것으로 예상된다. 이러한 추세에 맞추어서 한국 게임 산업의 국제적인 경쟁력을 높이고 지배적인 위치를 점하기 위해서는 현재 세계적인 게임 시장 내에서 우위를 점하고 있는 요소들에 대한 재점검과 발전 방향에 대한 연구가 필요한 시점이다. 본 논문은 IT 기술의 급속한 발전과 그에 따른 홈 네트워킹, 유비쿼터스(Ubiquitous) 환경이 본격적으로 조성되면서 앞으로 세계시장에서 더욱 더 강한 경쟁력을 갖추기 위한 요소들 중 게임 그래픽 제작 파이프라인의 하나인 텍스처 매핑 제작 체계에 대한 효과적인 파이프라인의 구성에 대해 연구하고 발전방향을 제시하는데 그 목적이 있다.

■ 중심어 : | 텍스처 매핑 | 유비쿼터스 | 게임그래픽 제작파이프라인 |

Abstract

The Worldwide Game Industry is getting recognized in the center of the culture contents industry, the total amount of the worldwide game industry is going to be over a hundred billion dollars in 2008 and the worldwide game industry will play a role of leading the growth of culture contents industry. To survive under the current circumstance, it is time to consider the research of the development direction and inspection of factors which Korea game industry has the superiority in the current worldwide game industry. As the result of the rapid growth of IT technology, home networking and ubiquitous environments are getting made up and the competition power of the worldwide game market is also getting hotter. The purpose of this thesis is to show the research and development direction of the organization system that has a effective texture mapping production pipeline among the whole game graphic production pipelines in order to have a international competition power and the future of Korea game industry.

■ keyword : | Texture Mapping | Ubiquitous | Game Graphic Production Pipeline |

1. 서론

한국게임시장은 2003년까지 초고속 인터넷을 기반으로 정부의 적극적인 지원 아래 온라인 게임을 중심으로 하여 많은 게임들이 출시되어 국내에서는 물론 해외에도 게임을 수출할 만큼 한국 게임 산업은 세계 게임시장에서 우위를 점하고 있다. 지금까지 IT 기술력에 의존하여 고속 성장한 한국 게임제작환경으로는 더 이상 국제적인 경쟁력을 갖추는데 한계점에 도달해 있는 것이 현 시점임을 숙지해야한다. 따라서 앞으로 더욱 더 빠르게 변화되어지는 IT 기술력으로 인한 게임문화는 더 이상 과거 놀이문화 및 특정한 계층이 즐겨하던 개념에서 벗어나 사람들의 다양한 감성과 욕구를 반영하고 표현할 수 있는 요소들을 개발하여 게임제작 실무에 바로바로 적용을 시켜야지만 한국 게임 산업의미래가 있다고 생각한다. 이러한 중심이 되는 요소들인 글로벌(Global)기획능력, 전문적 콘텐츠 개발인력, 마케팅 등, 한국 게임 산업의 부재요소들을 정부/산업체/교육기관에서 공동으로 인식을 하여 지금까지 투자와 교육이 이루어지고 있다. 하지만 이러한 요소들 이외에 우리가 잡고 넘어가야 할 점은 바로 게임제작 파이프라인의 효율적인 관리체계이다. 즉 아무리 좋은 기획능력, 고급 콘텐츠 개발인력과 마케팅 능력이 있다고 하더라도 이러한 요소들은 담고 게임타이틀이 시장에 출시하려면 게임제작 파이프라인이 효율적으로 갖추어져야 비로소 빠르게 변화되는 사람들의 욕구와 다양성을 게임에 반영하여 제작할 수 있기 때문이다. 본 논문에서는 본인이 실무재직시절에 참여되는 해머보이라는 피씨/콘솔용 게임제작 사례를 통하여 게임제작 파이프라인 중 텍스처 제작 파이프라인의 효율적인 관리 체계에 대한 발전방향을 모색하고자 한다.

1. 텍스처 제작파이프라인의 효율적인 관리체계 필요성

우리가 흔히 색의 3원색이라고 하면 빨강(Red), 녹색(Green), 파랑(Blue)을 나타내고 이색의 삼원색의 조합에 따라 수십만 가지의 색을 통하여 다양한 이미지들을 연출한다. 이러한 기본적인 개념은 누구라도 알고 있는 상식이다. 정작 중요한 사항은 이러한 기본적인 상식을

단지 그 자체 개념에 대한 인지로써 끝나는 것이 아니라 본인들이 속한 제작부서(그래픽부서)와 연결된 타 제작부서(프로그램부서)에 대한 기본적인 개념을 숙지하는 것이 전반적인 프로젝트 파이프라인의 효율적인 작업에 기초가 되는 것이다. 즉 피씨 게임(pc game), 온라인(on-line) 게임 그리고 콘솔(console) 게임 개발업체의 제작파트는 크게 그래픽제작 파트와 프로그램 파트로 나뉘어져 있으며, 모든 게임 상의 나타나는 캐릭터 및 배경 그리고 오브젝트에 대한 질감 구현은 그래픽파트에서 이루어지고 프로그램 파트에서 게임화면 안에 표현해 주는 역할을 한다. 따라서 그래픽파트와 프로그램파트 사이에는 수많은 양의 리소스(resource)들의 교류가 있다. 이러한 리소스들(resources)의 전달과정에서 부정확하고 부정확한 오류 있는 리소스들(resources)의 전달은 곧 전반적인 프로젝트 효율을 떨어트린다. 즉 정해진 론칭(Launching) 시간의 지연 그에 따른 마케팅에 대한 계획 수정 및 자금 소비 등 많은 문제점을 야기 시킨다. 본인이 속한 부서에 대한 전문성 및 완성도에 대한 중요성도 중요하지만 연계성을 가진 타 부서와의 데이터 및 리소스(resource) 전달의 효율적인 관리능력 및 커뮤니케이션(communication)능력이 더더욱 요구되어지는 것이다. 게임 속에 보여주는 캐릭터 및 배경 기타 오브젝트에 대한 질감의 표현은 바로 텍스처 매핑으로 구성되어져 있으며, 이러한 게임 매핑 소스가 게임엔진으로 곧바로 입력이 되는 것이 아니라 디지털화 된 기호에 의해 변환되어 게임엔진에서 받아들인다. 본인이 PC용/콘솔(Console)용 게임으로 개발된 해머보이 게임제작에 참여했을 당시 대부분의 그래픽파트 아티스트들은 이미 규격화 된 매핑 제작기법에 대한 스킬을 중요시한 채 실질적으로 어떻게 매핑 소스들이 엔진으로 들어가는지에 대한 상식과 리소스(resource)데이터의 효율적인 관리체계에 대한 인지와 이해 능력들이 상당히 부족한 편이었다. 이러한 기본적인면서 중요한 개념을 습득했다면 해머보이 게임 프로젝트에 있어 많은 제작시간을 줄일 수 있었을 것이고, 세계적인 게임문화의 흐름 및 사람들의 원하는 욕구와 다양성을 충족시키기 위한 게임소재 개발에 대한 집중 투자 등 게임 전반에 걸친 R&D(Research and Development)를 할 시간이 있었을 것이다.

1.1 게임엔진의 RGB 개념

그래픽파트에서 제작한 게임 매핑 리소스(resources)들은 게임엔진 상에 전달되어질 때 다음과 같은 RGBA의 Bit수로 처리가 된다. 즉 맵 사이즈는 2n*2n승의 해상도로만 받아들여진다.

R : Red (8 Bit), G : Green(8 Bit)
 B : Black(8 Bit), A : Alpha(8 Bit)

흔히 우리가 포토샵에서 매핑 작업 시 Layer에 작업하는 소스들은 RGB(8+8+8=24Bit) 모드에서 작업이고, RGB 채널(Channel)에서 새로운 레이어는 알파(Alpha) 값(8 Bit)을 의미한다.

- 1 Bit = 2n승 칼라수를 가진다. 2 Colors
- 4 Bit = 2_4승 = 16 Colors(1 Pixel)
- 8 Bit = 2_8승 = 256 Colors
- 16 Bit = 2_16승 = 65536 Colors
- 24 Bit = 2_24승 = 16777216 Colors
- 32 Bit = 2_24승 = 16777216 Colors

단 알파채널(Alpha Channel) 제외하고 같은 수의 칼라 값을 가진다.

II. 텍스처 제작파이프라인의 효율의 극대화 방안

1. 효과적인 텍스처 맵 제작의 이해

해머보이 게임은 초창기 PC게임용 타이틀로 개발하기로 설정되어있었고, 실제 PC게임 포맷에 맞는 매핑 리소스들이 제작되어졌다. 그러나 제작기간 도중 콘솔 게임의 강세로 회사에서는 해머보이 XBOX용 게임타이틀 제작도 시작하게 되었다. 그러나 말이 콘솔게임이지 실제적으로 화면에 보이는 해머보이의 피씨 게임과 콘솔게임의 그래픽 표현력은 거의 차이가 없었다. 따라서 이미 비디오 게임시장에 출시된 타 회사의 게임들과의 경쟁력에서 뒤질 수밖에 없는 처지였다. 이렇게 될 수밖에 없는 요소로서 여러 가지 요소가 있지만 우선

게임개발 제작에 들어가기 전에 앞서 국내/국제 게임시장의 흐름을 분석하고 향후 게임동향에 대한 예측을 하는 작업을 간과한 점이다. 또한 기술적으로 피씨와 콘솔 게임버전(version)에 사용되어지는 텍스처 리소스(resource)제작방법이 다르다는 것은 이미 게임개발 제작진들 또한 당연히 알고 있는 점이나 문제는 게임시스템에 따른 매핑 리소스(resource)의 컨버전(conversion) 방식과 데이터화에 대한 연구의 선행과정 없었다는 점이다. 이러한 요소들은 결국 게임제작 전반에 걸친 비효율적인 제작파이프 라인의 흐름을 이끌어낸다.

1.1 피씨와 콘솔 게임의 매핑 시스템 분석

피씨와 콘솔게임에 사용되어지는 텍스처 제작 시스템은 결국 텍스처 데이터양과 하드웨어적으로 적용될 수 있는 텍스처 종류에 큰 차이점을 두고 있는 것이다. 콘솔게임용으로 제작된 텍스처 리소스들이 피씨 게임용 텍스처 리소스들(resources)로 사용되어질 수 있으나 하드웨어적인 문제로 비효율적이며, 피씨 게임용 텍스처들이 콘솔게임용 텍스처 리소스들(resources)로 사용되어 질수 있으나 그래픽 표현력에 있어서 비효율적이다. 따라서 게임시스템에 따른 텍스처 리소스(resource) 사용의 효과적인 데이터 구축이 필요하다. 지금 세계적인 게임시장의 추세는 멀티 플랫폼으로 제작되어지고 있다. 이젠 피씨 게임이니 콘솔게임이 따로 분리된 게임 제작체계가 아닌 언제든지 원하는 게임 플랫폼을 제작하려면 텍스처 리소스들(resources)의 효율적인 데이터 제작방식과 관리가 필요한 것이다. 본 논문에서는 피씨용 해머보이 게임에서 사용된 텍스처 매핑 데이터들을 XBOX용 콘솔게임으로 전환하는 과정에서 매핑소스 데이터들의 유형을 열거하여 각각의 특징에 대해 알아보고, 이들이 XBOX 콘솔 상에서의 차지하는 메모리양과 엔진과부하에 대한 분석을 아래와 같은 기준사항을 통하여 분석한 결과를 [표 1]에 나타내었다.

- * 해머보이 게임 엔진
 (회사자체 개발한 게임엔진 업그레이드 버전)
- * XBOX용 해머보이 텍스처 데이터 사용을 위해 할당된 메모리용량

- : 10 - 12Mbyte (총 64Mbyte중)
- * 총 텍스처 매핑 데이터 파일 개수
 - : 50개의 텍스처 데이터
 - (35개의 텍스처 데이터는 Transparency 맵을 포함)
- * 텍스처 맵 데이터 Compression 포맷 : DXT1
- * 게임플레이가 원활하기 위한 초당 프레임수
 - : 60 FPS(Frame Per Second)

(1) Color Map 과 Bump Map 사용 시

- * 해상도 512*512일 경우
 - Color Map 용량 : 1024Kb(32bit Alpha),
768Kb(24bit)
 - Transparency Map : Color Map의 Alpha채널
 - Bump Map 용량 : 1024Kb(32bit)
 - Specular 맵은 Alpha 채널

장점 : Bump Map의 Quality 높다.
 단점 : 메모리 용량을 많이 차지한다. 게임 엔진에 부하를 줄 수 있다.

위와 같은 형식으로 게임엔진 상에서 Alpha 채널을 Transparency와 Specular를 인식을 한다.

(2) Color Map, Gray Bump, Specular Map 사용시

- * 해상도 512*512일 경우
 - Color Map 용량 : 1024Kb(32bit Alpha)
768Kb(24bit)
 - Transparency Map : Color Map의 Alpha채널
 - Bump Map 용량 : 256Kb(8bit)
 - Specular Map 용량 : 256Kb(8bit)

이 방식은 Bump와 Specular 맵을 알파채널의 8Bit 데이터로 저장하는 것이 아니라 각각 독립된 데이터 파일 형식으로 저장하는 것이다.

장점 : 메모리 용량을 줄 일수 있고 엔진에 과부하현상을 줄일 수 있다
 단점 : Bump Quality가 낮다.

(3) Bump와 Specular의 재질을 나타낸 이미지를 Color Map 사용시

- * 해상도 512*512일 경우 :
 - Color Map 용량 : 1024Kb(32bit Alpha)
768Kb(24bit)

장점 : 현재 쓰이고 있는 방식으로 어떤 엔진이던지 가장 부하가 덜 간다.

단점 : 실사를 위주로 하는 프로젝트의 경우에는 사실감 결여 및 Quality가 떨어진다.

표 1. 텍스처 맵 데이터 유형별 분석결과

	(1)	(2)	(3)
총 텍스처 메모리 차지 용량	12.5Mbyte	10.6Mbyte	6.4Mbyte
평균 게임속도(FPS)	65 FPS	60 FPS	58.5 FPS
게임엔진과부하	과부하	없음	최적
사용된 압축포맷	DXT1	DXT1	DXT1

2. 텍스처 맵 네이밍(Naming)화 체계구축

게임 그래픽 제작파트와 프로그래머파트사이에는 수많은 양의 데이터를 오고간다. 가능한 새로운 데이터를 새롭게 만들기 보다는 기존에 있는 데이터에 대한 활용을 한다. 그러나 데이터양이 많아질수록 이에 대한 데이터 분류체계가 제대로 갖추기 어렵다면 실 작업에 소요되는 시간보다 데이터를 찾는 데 더 많은 시간이 들어가는 것은 당연한 결과이다. 현재 한국 게임개발업체는 나름대로 데이터 관리를 하고 있지만 좀더 세밀하고 구체적인 데이터 관리 체계를 갖추어야만, 자그마한 시간에 대한 비효율적인시간성 소모로 인한 전체 게임제작 파이프라인의 비효율성을 막을 수 있을 것이다. 본 연구에서는 해머보이 게임에 사용되어진 데이터를 기준으로 효과적인 데이터 네이밍(Naming)화에 대해 알아보자.

2.1 텍스처 데이터 네이밍(Naming)화

- 1) 배경파트 : LvNumber_Texture Type
 ex) LV1_COL, LV1_BUMP

2) 캐릭터 파트 :

The Name of the Character_TextureType
ex) HammerBoy_COL, HammerBoy_BUMP

3) 오브젝트 파트 :

LvNumber_ObjectName_TextureType
ex) LV1_기중기_COL, LV1_기중기_BUMP

4) 애니메이션 파트 :

Name(Character, Object)_LvNumber_Type_An
ex) 망치_LV공통_Dead_Ani

5) F/X 파트 :

Name(Character, Object)_LvNumber_Type_FX
ex) 망치_LV공통_Dust_FX

2.2 셰이더(Shader) 데이터 네이밍(Naming)화

1) LVName_OPAQUE :

Transparency 맵이 없는 경우:
ex) LV3_OPAQUE

2) LVName_TRANS :

Transparency 맵이 있는 경우 :
ex) LV3_TRANS

3) LVNumber_Spec_Type :

Specular 맵이 있는 경우 :
ex) LV3_SPEC_OPAQUE, LV3_SPEC_TRANS

4) LVNumber_Bump_Type :

Bump 맵이 있는 경우 :
ex) LV3_BUMP_OPAQUE, LV3_BUMP_TRANS

3. 텍스처 맵 하드웨어적 알고리즘의 이해

게임그래픽파트에서 게임 매핑 리소스를 제작하는 아티스트들 대부분이 그래픽 툴 사용이해로만 텍스처 매핑 소스들을 제작하다보니, 텍스처 맵 데이터들이 프로그래머 파트에서 문제가 되어지는 데이터들의 양들이 상당하여, 이를 재수정하는 작업에 많은 시간이 소모되어진다. 분명 게임제작 파트별간의 문제점들이 발생될 수 있으나, 가능한 한 번에 끝마칠

일을 여러 번 반복해서 한다는 것은 분명히 비효율적

이다. 따라서 게임 그래픽을 담당하는 아티스트들은 전문적인 프로그래밍 능력에 대한 이해가 아닐지라도 어떻게 그래픽 데이터들이 게임엔진 상에서 적용 되어지는 것에 대한 기본적인 알고리즘에 대한 이해능력이 있다면 결과적으로 게임제작 파이프라인의 효율성을 높일 수 있는 요소이다.

3.1 Texture Size 와 File Size 관계

1K(1024*1024) 해상도를 가진 8 Bit 데이터의 파일 사이즈는 1024Kb이다.

8Bit = 1024Kb, 24Bit = 3073Kb, 32Bit = 4096Kb

그렇다고 하면 512K(512*512) 해상도를 가진 아래 [그림 1]과 같이 8 Bit 데이터 파일 사이즈는

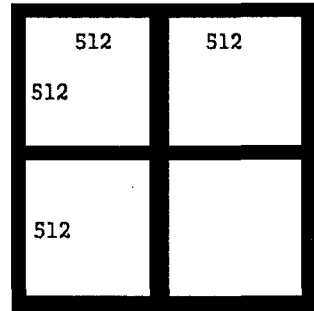


그림 1. 텍스처 해상도 사이즈(Size)

위의 [그림 1]에서 보는 것과 같이 1K를 4등분한 사이즈가 되는 것이다. 즉 1024/4 = 256인 것이다.

따라서 512K(512*512) 해상도의 8 Bit 데이터 파일 사이즈는 256 Kb이다.

- * 8 Bit(512) = 256Kb(1024/4), 24Bit = 768Kb, 32Bit = 1024Kb
- * 8 Bit(256) = 128Kb(512/4), 24Bit = 384Kb, 32Bit = 512Kb
- * 8 Bit(128) = 64Kb(256/4), 24Bit = 192Kb, 32Bit = 256Kb

위에서 보는 거와 같이 어떤 사이즈 해상도를 쓰느냐에 따라 파일사이즈는 4배 차이가 난다.

이 의미는 게임엔진에서의 FPS(Frame Per Second)에 크게 영향을 주는 요소로 작용한다.

3.2 Compression Format (DDS)의 파일 Size에 대한 이해

해머보이 게임제작에 사용된 모든 이미지 소스들은 프로그래머 파트로 넘어갈 때 DDS 형식으로 압축되어 보내진다. 보통 DTX1 이나 DTX3 압축형식을 선택을 한다.

DXT1 : 24Bit 파일은 6:1로 압축, 32Bit 파일은 8:1로 압축

ex) * 512K의 24Bit 파일 사이즈
 $768/6 = 121Kb$

* 512K의 32Bit 파일 사이즈는
 $1024/8 = 128Kb$

DXT3 : 24Bit 파일은 3:1로 압축, 32Bit 파일은 4:1로 압축

ex) * 512K의 24Bit 파일 사이즈
 $768/3=256Kb$

* 512K의 32Bit 파일 사이즈
 $1024/4=256Kb$

4. 색상보정 시스템 구성

해머보이 피씨용 게임을 콘솔 게임용으로 전환하는 과정에 있어서 텍스처 리소스(resource) 부분들은 모니터에서 본 이미지와 실제 TV에서 보여주는 색상 값이 많이 틀려서 TV에서 이미지에 대한 색상을 확인 후 이미지 색상 값을 조절하는 등 많은 시간을 투자하였다. 왜냐하면 모니터와 TV는 색상을 화면에 뿌려주는 방식이 틀리기 때문이다. 즉 모니터에서 볼 수 있는 색상 값이 TV에서 뿌려줄 때 이미지에 들어간 색상 값 중 잃어버리는 칼라 값들이 있다. 따라서 이러한 색상들은 다른 느낌으로 보이는 것이다. 게임 실 제작에 들어가기에 앞서 미리 텍스처 소스에 대한 색상보정 시스템이 구축

이 되었더라면 게임제작 전반적인 효율성을 높였을 것이다. 이러한 문제점을 해결하기 위해서 가장 좋은 방법은 아티스트마다 TV를 가지고 있다면서 작업 시 모니터와 TV사이에서 동시에 색상을 보정시킬 수가 있다. 만약 여건상 힘들다면 나름대로 TV에 보이는 색상과 모니터 상에 보이는 색상을 보정시키는 방법을 알아보자.

4.1 색상보정작업을 위한 모니터와TV간의 셋팅 방법

1) 각 아티스트의 모니터 칼라 셋팅을 동일한 값을조정 한다. 각 모니터 제품마다 색상이 다르게 보여준다. 더군다나 같은 회사에서 나온 모니터라도 색상이 다르다. 따라서 통일된 모니터 셋팅으로는 보정하기가 곤란하다. 따라서 포토샵에서 칼라 셋팅을 해주어야한다.

2) 위의 통일된 모니터 셋팅 후 대부분 이미지 리터칭된 포토샵을 사용한다. 따라서 포토샵의 ColorSetting 메뉴에서 통일된 프로파일(Profile)로 셋팅을 한다. 즉 포토샵 메뉴인 Edit >Color Setting에서 프로파일(Profile) 셋팅 값인 TV를 선택한다.

3) 모니터 칼라 셋팅이 끝나면 우선 모니터용 RGB 팔레트를 만들고, TV 스크린에 모니터용 RGB 팔레트를 열어 TV 스크린용 RGB 팔레트를 만든다. 모니터용과 TV용 RGB팔레트의 가장 큰 차이점은 대부분은 색상은 같은 값을 가지고 있으나 특정한 영역에 있어서는 TV와 모니터 색상 값이 다르다는 것을 알 수가 있다. 또한 TV의 전체적인 팔레트의 채도(Saturation)와 명도(Luminance) 값이 모니터의 팔레트(Palette)의 채도(Saturation)와 명도(Luminance) 값보다 높다. 따라서 모니터에서 보여주는 색상 값을 TV상에 그대로 보여주고 싶다면 본래 색상의 채도(Saturation)와 명도(Luminance) 값을 낮추어 주어야만 TV에서 모니터에서 본 이미지 색상대로 표현되기 때문이다.

4.2 색상보정작업을 통한 텍스처 제작기법

1) 새로운 텍스처 소스를 2D 소프트웨어인 페인터 로

제작시에는 칼라 값 선택을 TV 칼라 팔레트를 사용한다.

- 2) 이미 존재하는 이미지를 Reference로 새로운 텍스처 소스를 제작 시에는 우선 TV 칼라 팔레트를 사용하여 칼라링 작업을 하고 소스 작업이 끝나면 채도(Saturation)이나 명도(Luminance) 값을 낮춘다.
- 3) 기존에 이미 만들어 놓은 이미지를 TV용 색상으로 교정하려면 포토샵에 채도(Saturation)값이나 명도(Luminance) 값을 낮춘다. 그러나 정확히 기준이 되는 채도(Saturation)이나 명도(Luminance) 값이 없으므로, 반복적인 작업을 할 수가 있다. 이러한 문제는 어느 정도의 기준이 되는 값을 설정해야 하는 것이 필수적이다.

5. 텍스처 리소스 재활용 방안 및 맵 타이일 제작 기법

해머보이 게임의 텍스처 소스들을 제작할 때 문제가 되었던 점은 바로 리소스 재활용 및 Map Tiling 문제였다. 즉 같은 텍스처 이미지를 여러 개의 오브젝트에 같이 쓰다 보니 특정한 것에 쓰였던 텍스처 소스를 수정 시 그 소스이미지를 쓴 나머지 부분까지도 바뀌어야 하는 문제로 그에 따른 것을 해결하기 위해 시간적인 투자를 많이 하였다. 게임제작 시스템상 가능한 이미 사용되어진 텍스처 소스의 재사용은 필수적이며 또한 효율적이지만 재사용되어질 텍스처 데이터의 효율적인 관리 없이는 결과적으로 제작라인에 큰 문제점을 야기 시키는 것이다. 또한 텍스처 이미지들의 해상도는 게임 시스템에서 요구하는 제한된 텍스처 사이즈를 요구되어, 커다란 오브젝트에서 매핑시 “맵 타일링”이라는 기법으로 적용된다. 그러나 맵 타일링 기법이 제대로 사용되지 못하면 오히려 텍스처 이미지의 질을 저하 시킬 수 있다. 따라서 재사용되어질 텍스처 데이터에 대한 효과적인 관리 방법 및 맵 타일링 제작기법에 대하여 알아보자.

5.1 리소스(Resource)의 효율적인 재활용

해머보이 매핑작업 당시 같은 매핑 데이터를 여러 개

의 오브젝트에 같이 쓰다 보니 특정한 것에 쓰였던 맵 소스를 수정할 경우 그 소스이미지를 쓴 나머지 부분까지도 바뀌어야 하는 문제점들이 발견되어졌고, 이에 따른 문제점들을 해결하기 위해 많은 시간적인 투자를 하였다. 이러한 문제가 된 제작라인을 개선하는 방안으로 다음과 같은 파이프라인 구조를 가져야 한다. 한 사람이 자신이 맡은 모든 Lv의 모델링과 텍스처 작업을 할 것이다. 그러면 각각 맡은 레벨에 자신이 만들어 놓은 이미지 소스를 다른 아티스트들이 수정하여 쓸 수 있는 경우가 있을 것이다. 즉 다른 아티스트들이 만들어 놓은 소스를 쓸 수가 있고 혹 수정하여 쓸 수도 있다는 것이다. Master Resource Image 폴더에는 각 아티스트가 최종적으로 텍스처 소스에 쓰일 파일 올려놓고 실제 쓰일 이미지 파일은 다른 이름으로 저장한다. 즉 [그림 2]에서와 같이 Master Resource Image 폴더 안에 있는 텍스처 이미지 파일과 실제 오브젝트에 적용된 텍스처 이미지 파일은 이미지는 동일하지만 파일 명이 틀리다는 것이다. 위와 같은 파이프라인은 동일한 리소스(resource)를 여러 개 오브젝트에 사용함에 따른 나타났던 텍스처 소스의 선택과 배경 등 여러 가지 문제점들 해결 하고, 또한 전반적인 맵 분위기 통일감에 있어서도 도움이 될 뿐 만 아니라 맵 제작 시 소요되는 시간을 단축시키는 결과를 가져온다.



그림 2. Master Resource Image 폴더 구조

5.2 효율적인 맵 타일(Map Tile) 제작기법

텍스처 소스 타일 제작 방법은 흔히 포토샵의 오프셋(Offset) 기능을 이용하여 양쪽의 엣지(Edge) 부분을 리터칭하여 연속적인(Seamless) 상태로 만들어주는 식의 방법을 많이 사용하고 있다. 그러나 포토샵의 하이패스 필터(High Pass Filter) 기능을 사용하면 보다 효과적인 타일링 맵(Tiling Map) 작업을 할 수가 있다.

5.2.1 하이패스 필터(High Pass Filter)의 정의 및 제작기법

간단히 이야기해서 가우시안 블러(Gaussian Blur)와 반대되는 효과이다. 이미지에서 크게 블러(Blur)진 영역을 이미지상에서 빼고, 작은 날카롭고, 정교한 영역을 보존하는 기능이다. 하이패스 필터(High Pass Filter)의 의미는 가우시안 블러(Gaussian Blur)와 정반대인 기능이다. 따라서 가우시안 블러(Gaussian Blur)는 로우 패스 필터(Low Pass Filter) 부르기도 한다.

일단 포토샵에서

- 1) Filter>Render>Cloud 사용하면 생성되는 그림은 아래 [그림 3]과 같다.
- 2) 이미지를 복사(duplicate) 한다.
- 3) Filter > Other > High Pass Filter를 적용한다.
- 4) 반경(Radius) 값을 적절히 조절한다.
- 5) 오리지널(Original Image)의 레이어(Layer)를 카피한 다음 Invert를 시킨 이미지들은 아래 [그림 3]과 [그림 4]와 같다.

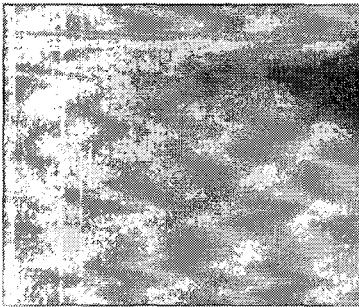


그림 3. Original 이미지

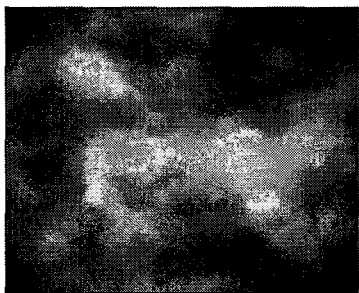


그림 4. Invert된 이미지

- 6) Invert된 이미지의 레이어(Layer)의 투명도

(Opacity) 는 50%를 놓으면 그레이(Gray) 색 상이 된다.

- 7) Filter> Blur > Gaussian Blur를 선택하고 반경(Radius) 값을 하이패스 필터(High Pass Filter Radius) 값을 기준으로 조절한다.
- 8) 오리지널(Original) 이미지에 하이패스 필터(High Pass Filter)를 적용한 이미지와 가우시안 블러(Gaussian Blur)를 적용한 이미지를 비교하면 거의 똑같다고 할 수가 있다.

위와 같이 예제를 든 이유는 로우 패스 필터(Low Pass Filter)와 하이 패스 필터(High Pass Filter)는 같은 원리의 개념에서 나온 것이라는 것을 알 수가 있다. 게임을 위한 거대한 지형의 맵을 만드는데 있어서 대부분의 아티스트들은 타일링 맵(tiling map)으로 텍스처 소스를 만들 것이다. 그러나 포토샵 이미지를 기반으로 연속성(seamless)을 고려해서 만들지라도 타일(tile)되어진 전체적인 맵 자체에 눈으로 보기에 반복적인 패턴 효과를 볼 수가 있다. 하이패스 필터(High Pass Filter)는 이러한 문제들을 해결 하는데 많이 사용되고 있다.

5.2.2 하이패스 필터(High Pass Filter)의 적용

하이패스 필터(High Pass Filter)의 적용시점은 연속성(Seamless) 텍스처 작업에 앞서 이미지에 적용해야 한다. 하이패스 필터(High Pass Filter)는 텍스처의 모든 경계(Edge)들은 대략 같은 색채의 밝기(Luminosity)를 갖도록 해주는 역할을 한다. 그러나 모든 이미지에 이러한 하이패스 필터(High Pass Filter)의 적용이 되는 것은 아니다. 선행되어야 할 조건은 상대적으로 텍스처의 동일성이 있어야 한다는 것이다. 예를 들어서 날카로운 양쪽 경계(Edge)의 선명도(Brightness)가 현저하게 다를 경우나, 디테일 측면에서 현저하게 다르다면 하이패스 필터(High Pass Filter)효과는 없다. 아래 [그림 5]와 [그림 6] 두 이미지를 비교할 때 [그림 6]이 매칭이 안 되는 두 경계(Edge)를 자연스럽게 보이는 것을 알 수가 있다. 따라서 하이패스 필터(High Pass Filter)를 적용하지 않은 [그림 6]보다 연속성(Seamless) 맵을 만들 수 있다.



그림 5. High Pass Filter 적용시

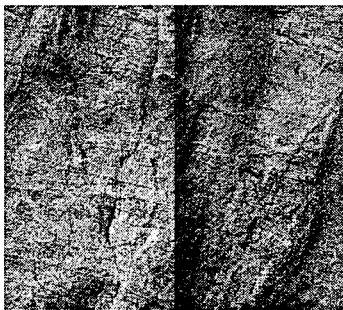


그림 6. High Pass Filter 적용하지 않을시

6. 아티스트 별 칼라 팔레트제작

뛰어난 텍스처 이미지를 제작하기 위해서는 아티스트 별 제작 능력도 중요하지만, 원화부분에서 넘어온 그림들에 대한 색상을 제대로 텍스처 재질로 표현을 하는 것 또한 중요시 여겨진다. 그러나 게임개발업체 그래픽 아티스트 대부분이 개개인의 고유 칼라 팔레트를 가지지 않고, 원화파트에서 지정된 색상을 그대로 표현한다고 하지만 그래픽아티스트들로 하나의 새로운 것을 창조하는 작가정신들이 있어서 어떠한 기준이 되는 색상 체계 없이 원화파트의 지정한 색상과는 다르게 개인적인 취향대로 색상을 입힌 텍스처를 제작하는 경우가 많다. 게임제작 컨셉에 맞는 색상을 가진 텍스처 소스들로 인한 게임전체 색상의 불일치 및 원화파트부분과의 여러 가지 갈등들은 전체적인 게임제작 파이프라인의 문제점을 야기 시켰다. 따라서 텍스처 이미지를 제작할 때 원화파트에서 지정한 색상을 표현함과 동시에 그래픽 아티스트들의 작가정신이 들어갈 수 있도록 공통의 칼라 팔레트가 필요한 것이다. 즉 전체 게임 색상이 기준

이 되는 칼라 팔레트 선정과 각 아티스트별 고유의 칼라 팔레트 구성이 되어진다면 지금까지 빈번하게 발생되어진 원화 파트와 그래픽 파트의 색이원화 현상을 줄일 수 있으며, 결과적으로 게임 텍스처 제작라인의 효율성 및 아티스트들의 의욕을 더욱 고취 시킬 수 있다.

- 1) 원화에서 나온 이미지의 색감을 충분히 반영할 수 있으므로 각 레벨간의 전반적인 색감을 통일하는데 용이하다. 모든 레벨의 모델링 및 텍스처 작업을 시작하기 전 원화 팀에서 각 레벨별의 칼라링 이미지가 나온다. 그래픽 아티스트들은 이 그림을 토대로 각 레벨을 디자인해 나간다. 예를 들면 칼라 팔레트 없이 같은 진한녹색이라도 각각의 아티스트마다 다른 칼라 값을 가진 진한 녹색을 사용하게 된다. 따라서 전체적인 색감이 라든지, 빨간의 연결 느낌이 제대로 표현되지 않는 경우가 많다. 이때 칼라 팔레트가 있으면, 팔레트에 표시된 각각의 칼라 수치를 토대로 원화의 색감이 충분히 표현된다.
- 2) 원화를 기반으로 색감의 변화(Variation)가 용이하다. 그래픽 아티스트들이 원화에 나온 색감을 토대로 그 색감을 충분히 살릴 수도 있지만 경우에 따라서는 색감을 바꾸는 경우가 있다. 이때 기준이 되는 칼라 팔레트가 있다면 칼라 팔레트를 통해서 여러 가지 색깔을 시각적으로 인지하면서, 자신이 생각하는 색상에 대한 느낌을 설정하기에 용이하고, 전반적인 색상과 바꾸고자 하는 부분의 색상에 대한 색상보정을 칼라 팔레트에서 표시되는 색상들을 통하여 쉽게 보정할 수 있다.
- 3) 자신만의 칼라 팔레트를 만들자. 아무리 칼라 팔레트가 있다 하더라도 수많은 색상을 모두 끌고루 쓰이지 않을 수도 있다. 즉 아티스트마다 자신이 주로 많이 사용하는 색상들이 있을 것이다. 예를 들어 산에 쓰일 색상들, 건축물에 쓰일 색상들 등등 이것에 대한 작은 칼라 팔레트를 만들어 놓으면, 텍스처 제작 시 상당한 시간적인 효율성을 얻을 수 있을 것이다. 어찌 보면 상당히 손이 많이 가는 과정일 수 있지만 조금씩 용도별로 신만의 칼라 팔레트를 만들면, 나중에 그것이 모여 여러 가지 용

도로 사용된다.

위의 세 가지 경우 외에 칼라 팔레트의 용도는 각자가 어떻게 활용하느냐에 따라 여러 가지 측면에 적용될 수 있고 유용하게 사용되어질 수 있다. 모델을 만들기에 앞서 이 모델에 대한 개념의 정확한 이해 그에 따른 색상에 대한 생각들은 모든 3D 작업에 있어서 가장 기본적으로 중요하게 고려해야 할 사항이다.

7. 효율적인 텍스처 매핑 방안 연구

기획파트에서 원하는 모델의 정밀함과 게임엔진 상에서 어느 정도의 폴리곤 개수를 지원해 주는가에 따라 모델링방식이 달라지고, 게임 상 속도는 캐릭터 폴리곤의 최대 개수, 건축물의 폴리곤의 최대 개수, 그 외의 오브젝트 폴리곤 개수가 영향을 준다. 해머보이 게임은 한신(Scene)안에 캐릭터, 배경, 오브젝트를 모두 포함하여 1만 5천개의 폴리곤을 사용하고 있다. 여기에 텍스처와 특수효과 등이 함께하기 때문에 게임진행상 한 씬(SCENE)에 대한 폴리곤의 지원되는 개수가 정해져 있는 것이고, 게임 엔진 상에서 표현할 수 있는 폴리곤 안에서 최대한 모델링을 해야 되기 때문이다. 물론 텍스처 크기와 개수, 파일크기 등도 문제가 된다. 다른 예를 들어 실시간 LOD생성을 하는 방식이 있다. 그 대표적인 예는 퀘이크3(Quake3)이다. LOD(Level of Detail)란 먼 거리에 있는 물체는 적은 수의 폴리곤으로 표현하고 가까이 있는 물체는 많은 수의 폴리곤으로 표현하는 기술이다. 즉 몇 단계의 모델을 거리별로 직접 제작하는 방식인데 퀘이크3(Quake3)에서는 각각의 거리와 사양에 따라 세 종류의 모델이 준비돼 있다고 한다. 위의 예와 같이 어떤 방식의 모델링과 텍스처 매핑을 할 것인가는 우선 기획 쪽에서의 레벨 디자인 시 그 스테이지의 레벨에 맞는 기본적인 샘플링 셋(Sample set)을 만들어 카메라와 빛을 설정해 놓는다. 그 다음 카메라의 시점을 고려하여 가까이 보이는 곳과 멀리 보이는 곳의 모델의 폴리곤 개수의 차이를 주고, 텍스처 역시 샘플링 셋(Sample set)에 맞게 카메라와 빛의 방향을 고려하여 가까이 보이는 것과 멀리 보이는 텍스처의 차이를 주고 움직임에도 가만히 고정돼 있는 배경의 그림자는 이제

눈이 높아질 대로 높아진 사용자들에게 신선한 효과로와 닿지 않기 때문에 빛의 방향에 맞추어 가장 기본적인 그림자를 넣어준 후 실시간 빛과 멀티 텍스처링(동일한 메쉬(Mesh)에 대하여 여러 개의 텍스처를 동시에 적용할 수 있는 기능), 라이트 맵핑(기본 텍스처 위에 밝기를 나타내는 또 하나의 텍스처(= 라이트 맵)를 준비해 이를 멀티 텍스처링하는 것과 같다), 범프 맵핑 등에 대한 연구를 더욱 효과적으로 한다면 현재보다 훨씬 뛰어난 그래픽 표현이 가능하다.

III. 결론

앞으로 세계는 무한경쟁 체제 속에 각국의 다양한 문화들이 경계 없이 교류되어지고 있고, 재빠르게 다양한 형태의 매체들이 시장에 나오고 있으면서 이에 반응하는 소비자 기호와 욕구도 또한 빠르게 변화 되어지고 있다. 이젠 더 이상 기술력 하나만으로 무한경쟁 시장에서 소비자들의 기호에 맞는 욕구를 충족시켜줄 수 없다. 따라서 소비자들의 기호와 대중 성향들을 따라가지 못하면 무한경쟁체제 속에서 살아남을 수 없는 것이다. 이미 게임 산업 이외에서 전문 컨설팅(Consulting) 업체들이 회사의 효율적인 경영을 위해 회사 전반에 걸친 철저한 관리를 해주고 있으며, 보수적인 성향을 가진 정부조직 조차도 시장 경쟁체제에 맞는 시스템으로 전환되어지는 시점에서 있는 것이다. 이젠 더 이상 한국 게임 산업도 전문적인 게임 제작 파이프라인의 효율적인 관리가 제대로 이루어지지 않는다면 하루가 다르게 바뀌는 사람들의 기호와 대중적, 사회적 성향을 담은 게임을 시장에 내놓을 수 없다. 이미 고착화된 게임 제작 시스템 라인의 재점검과 국제적인 경쟁력을 갖추기 위한 효율적인 제작 파이프라인에 대한 모델을 찾아야 할 것이다. 본 연구에서는 해머보이 라는 피씨/콘솔용 게임 타이틀을 모델로 하여, 게임제작시스템 가운데 한부분인 텍스처 매핑 제작라인에 대한 효과적인 제작파이프라인을 갖추기 위한 요소들을 게임제작 기법과 기술적인 측면이 아니라 제작 파이프라인에 대한 관리에 초점을 맞추어 제시하였다.

참고 문헌

- [1] 김덕호, 게임기획과 디자인 고수따라하기, 가남사 (도), 2004.
- [2] 정동현, 게임제작의 알파와 오메가, 영진닷컴, 2004.
- [3] MIKE MCSHAFFRY저 류광 옮김, GAME CODING COMPLETE, 정보문화사, 2004.
- [4] 한국게임개발원, 게임제작프로세스, 정일(도), 2003.
- [5] 정병건, 최정윤, 포토샵의마스터7, 성안당, 2003.
- [6] 김경선 외11인, 게임제작개론, 형설(도), 2003.
- [7] KEVIN HAWKINS \$ DAVE ASTLE, 류광 옮김, 정보문화사, 2001.
- [8] DAVID H. EBERLY저, 민프레스 편집부 옮김, 3D GAME ENGINE DESIGN, 민 커뮤니케이션, 2001.
- [9] 정운철, 오상숙, DIRECTX를 이용한 게임 만들기, 가남사, 2001.

저자 소개

김 현 조(Hyun-Jo Kim)

정회원



- 1992년 : 한양대학교 기계공학과(공학사)
- 2000년 : 미 Academy of Art University Computer Arts (M.F.A 석사)
- 2003년~현재 : 경기대학교 다중매체영상학부 애니메이션 전공 교수
- 2000년~2001년 11월 : 미 Ronin Entertainment 3D Artist
- 2001년 12월 ~ 2003년 1월 : 디지털 드림 스튜디오 게임제작2부 아트디렉터, 영상제작3부 셋업 팀장
- <관심분야> : 애니메이션, 게임 제작 및 콘텐츠 개발