

# 3d Tool 기본 기능을 이용한 범용적 게임엔진용 사물 폭파 이펙트 연구

## A study on to be used for all purposes effect of blasting in the game engine, using 3d Tool basic functions

김상중<sup>1</sup>

Sang-Jung Kim<sup>1</sup>

요 약

게임은 상호작용으로 재미를 느끼는 콘텐츠이다. 많은 상호작용의 효과적인 균형으로 인해 몰입도가 높아진다. 그 중 3D 게임에서 타격감은 몰입을 증대하는 매우 중요한 요소 중에 하나다. 타격감을 구성하는 요소 중 이펙트 효과는 매우 중요한데, 그 중 폭파 애니메이션 이펙트는 게임의 내에서 매우 타격감을 높여 주는 매우 중요한 요소이다. 특히 사물 반응에 의한 상호작용으로 폭파가 이루어질 때 타격감을 극대화 시킬 수 있다. 그러나 게임 제작 프로세스에서는 반드시 게임 엔진을 사용하여야 하는데, 회사마다 혹은 팀들마다 게임의 특성에 맞게 다른 게임엔진을 사용하고 있고 방법도 달라서 매년 엔진 공부와 같이 제작을 진행하여 비효율적인 방식으로 제작이 되고 있다. 심지어 매년 업그레이드되는 엔진 소프트웨어에 대해서도 꾸준히 내용을 숙지해야 한다. 이러한 번거로움과 어려운 부분을 없애고자 하나의 3D Tool만 사용하여 효과를 내고 게임엔진에서 범용으로 사용할 수 있는 새로운 제작 방법을 제공하고자한다.

핵심어 : 타격감, 폭파 이펙트, 게임 애니메이션, 3D 툴, 3ds max, 효율

### Abstract

A game is a content that is fun through interaction. an effective balance of many interactions increases engagement. among such interactions, the feeling-of-hit in 3D games is one of the most critical factors that increase engagement. amid the factors that compose the feeling-of-hit, effect features are incredibly important, and the animation effect of blasting among those acts as a crucial factor. It could particularly maximize the feeling-of-hit when blasting is executed from an interaction due to object response. However, while a game engine must be used in the game making process, companies or teams each use different game engines, leading to an inefficient production process. Even worse, one must constantly cultivate thorough knowledge of engine software that is upgraded yearly. To eliminate such inconvenience and difficulties, this study aims to provide a new production method that can create effects and be used for all purposes in the game engine by only utilizing a single 3D Tool employed.

Keyword : feeling-of-hit, effect of blasting, game animation, 3D tool, 3ds max, efficiency

<sup>1</sup> School of Game, College of Cultural Industries ChungKang, Seoul, Korea [Professor]  
e-mail: happygo51@daum.net

Received(May 10, 2021), Review Result(1st: May 28, 2021), Accepted(June 11, 2021), Published(June 30, 2021)



© 2021 The Authors. Published by NCISS.  
This is an open access article licensed under the Creative Commons Attribution-NonCommercial 4.0 International License.  
To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/4.0/>.

## 1. 서론

게임은 상호작용으로 재미를 느끼는 콘텐츠이다. 많은 상호작용의 효과적인 균형으로 인해 몰입도가 높아진다. 몰입을 높일 수 있는 여러 가지 요소 중에서 타격감이 매우 중요하다. ‘타격’이란 단어의 사전적 정의는 세계 때린다는 의미를 가지고 있다. 여기에 ‘감’이란 단어는 느낌이란 의미를 가진다. 타격감이란 것은 게임 상에서 얻을 수 있는 목적을 달성하는 단계에서 생기는 주요한 감성 정보 중의 하나다 [1]. 타격감은 현실에서 느껴왔던 타격감과 마찬가지로 하나의 쾌감으로 받아들여지며 이는 게임의 재미와 완성도에 큰 영향을 끼치고 있다 [2].

게임 제작 중 그래픽 제작 프로세서에서는 2D, 3D Tool을 사용하여 1차 콘텐츠를 제작하고 나면, 반드시 게임 엔진을 통한 2차 콘텐츠로 변환을 시켜야 한다. 게임엔진이란 이러한 게임 제작에 자주 활용되는 개발 노하우나 기술, 하드웨어 API 사용법 등을 쉽게 활용할 수 있는 소프트웨어 라이브러리나 모듈의 형태로 구현해 놓은 것을 말한다 [3]. 3D 게임에서의 애니메이션은 3D Tool (3ds Max, Maya 등등) 만으로는 유저들이 사용 할 수 없다. 게임 엔진을 반드시 사용하여야 동작이 움직이고 유저들이 사용 할 수 있다. 게임 엔진은 게임을 구동하고 프로그램과 데이터를 관리하며, 효율적인 게임 제작 및 게임 구동을 돕는 독립된 프로그램으로 정의 된다 [4].

게임 산업은 눈부시게 발전하면서 고사양의 게임 제작을 위해 다수의 게임 엔진도 같이 성장하였는데, 현재 국내에서 대중적으로 사용되는 게임 엔진으로는 유니티(Unity), 언리얼 (Unreal) 등을 많이 사용하고 있고 있는데, 게임 엔진인 유니티(Unity)를 이용하여 저렴한 가격대의 PC에서 표현하는 가상 환경을 개발했다 [5]. 게임 엔진 및 GPU를 이용하여 시각화 및 센서 유형별 모델링을 수행하였다 [6]. 유니티(Unity) 엔진은 게임 개발용 소프트웨어로서 게임뿐만 아니라 시뮬레이션, 의료 및 건축용 시각화 등의 상호작용이 가능한 3D 및 2D 콘텐츠를 다양한 플랫폼에서 제작 할 수 있다 [7]. 언리얼(Unreal) 엔진은 고퀄리티 3D 게임 제작에 적합한 엔진으로써 엔진의 소스 코드를 무료로 공유하고 있어 개발자들이 게임엔진을 자신의 스타일에 맞게 커스터마이징하여 사용할 수 있으며 여러 최신 그래픽 기능을 빠르게 제공하는 장점을 가지고 있다 [8]. 그러나 문제점은 고사양의 게임 엔진으로 업그레이드되는 빠른 변화에 게임 그래픽 개발자들은 그 속도를 못 따라가고 있다. 각 회사마다 혹은 개발 팀마다 사용하는 엔진이 모두 다르고, 이러한 게임 개발 환경에서 본인이 사용하는 3D 기본 기능만으로는 게임 엔진에서 동작 및 이펙트 등 구현에는 한계가 있다.

본 연구는 이 한계를 극복하고 하나의 3D 기본 기능만으로 폭파 시뮬레이션 이펙트를 만들고 범용으로 모든 엔진에서 구현이 가능한 새로운 방법론을 제공 하는 것은 매우 의의가 있다.

본 연구의 기술적인 문제와 이슈를 명확히 하고자 게임 이펙트 개발자 20명을 대상으로 1차 FGI를 실시하였다. 첫째, 이펙터로 가장 힘든 것은 무엇인가? 둘째, 이펙트 타격감에 필요한 요소

들은 무엇이 있는가? 셋째, 지금까지 폭파 이펙트를 구현 했던 방식은 무엇인가? 넷째, 만약 폭파 이펙트를 구현할 때 한 개의 3D 툴에서 기본 기능을 사용하여 범용적으로 사용할 경우 도움이 될 것인가? 등으로 질문을 하였다.

첫째 질문에 대해서 이펙터로 가장 힘든 것은 여러 툴을 사용해야만 하는 것이 10명(50%) 정도로 가장 많았고, 급변하는 툴에 대한 공부도 4명(20%), 없는 것을 만들어야 하는 부분 3명(15%), 식상하지 않고 새로운 것을 꾸준히 창조해야 하는 것이 3명(15%) 순으로 어려움에 대해서 답변하였다. 둘째 질문에 대해서 이펙트 타격감에 필요한 요소들은 애니메이션 이펙트 작업이 11명(55%)으로 가장 많았고, 사운드 이펙트 4명(20%), 장면 연출 3명(15%), 속도감 있는 전개 2명(10%) 순으로 답변하였다. 셋째 질문에 대해서 폭파 이펙트 구현은 어떠한 방식으로 진행 하는지에 대해서는 가장 많은 것이 제작을 해서 담당 프로그래머에게 제작 소스를 넘긴다는 9명(45%)으로 가장 많았고, 프로그래머와 시스템 기획자 둘에게 제작 소스만 넘긴다는 5명(25%), 시스템 기획자에게 제작 소스를 넘긴다는 3명(15%) 그리고 본인이 모든 것을 다 처리한다는 3명(15%) 순으로 답변하였다. 마지막 질문으로 한 개의 3D 툴에서 기본 기능을 사용하여 범용적으로 사용할 경우 도움이 되겠는가라는 질문에서 18명(90%)의 개발자가 도움이 된다고 말을 했고, 2명(10%)의 경우 현재 진행하는 방식도 좋다고 답변했다. 즉 여러 툴을 다루는 것이 어렵고, 타격감에는 애니메이션이 가장 중요하며, 프로그래머에게 의존하여 각종 이펙트를 제작하고 있어서 본인이 컨트롤 가능한 방법론이 있으면 매우 유용하다는 결과를 도출 하게 되었다.

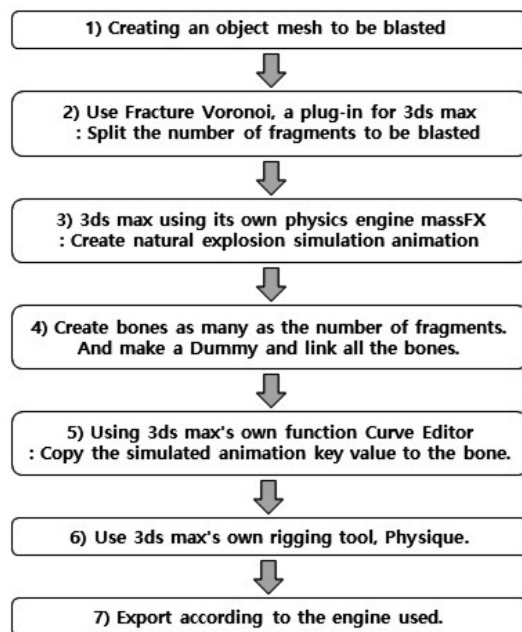
제작 원리를 간단하게 소개한다면, 캐릭터 애니메이션의 경우는 얼굴, 몸, 의복, 장신구 등 소수의 메쉬(mesh)를 애니메이션 하는 경우이기 때문에 애니메이션 작업의 속도를 빠르게 진행 할 수 있다. 참고로 메쉬(mesh)란 3D 모델링 컴퓨터 내부 자료 구조 표현 방법 중 하나로써 모델링 표면 상 존재하는 점들을 표본화 하여 연결구조로 나타내는 삼차원 구조를 말한다 [9]. 그러나 폭파 애니메이션의 경우 많은 파편 메쉬 수만큼 모두 애니메이션 키(Key)를 잡아야 하고 물리 엔진과 같은 자연스러운 동작을 표현하기 위해서 물리 엔진용 변환이 필요하다. 즉 시뮬레이션이 가능하도록 파편 메쉬 수만큼 충돌 장치를 만들어야 하고, 파편 메쉬 마다 물리엔진이 적용되어 충돌 및 반사가 가능하도록 주위 오브젝트도 변환해야한다. 3ds max에는 자체 물리엔진이 제공이 되고 있다. 전에는 마찰이나 충돌 시뮬레이션을 훌륭하게 구현한 하복 물리 엔진 (Havok physical engine) [10]의 플러그 인 형태로 사용 되었다가 하복社 엔진의 다른 이름인 리액터 (Reactor) [11]라는 물리 엔진을 맥스 자체에 탑재 되었다. 그리고는 2012년 이후부터 현재까지 매스에프엑스(massFX) [12]로 진화 되어 사용 할 수 있도록 탑재 되어있다. 엔진마다 시뮬레이션의 원리는 같으나 모두 다른 적용 방식을 가지고 있어서, 물리를 적용하고자 하면 다른 방식의 물리 시뮬레이션 공부를 마친 뒤 특성에 맞게 적용을 해야 하는 매우 어려운 공정을 지나야 한다. 그 후 겨우 폭파 애니메이션 하나를 엔진에서 표현 할 수 있다. 이러한 어려움을 3D Tool(3ds Max) 하나에서 모두 가공하

여 익스포트(Export) 하는 방법을 사용하면 엔진으로 인한 한계도 없애고, 원하는 폭파 시뮬레이션 애니메이션을 쉽게 해결 할 수 있다. 물리 엔진에 대한 장벽으로 인한 제작 시간을 줄 일 수 있을 뿐만 아니라 익숙한 툴의 깊이로 인해 더욱 생동감 있는 시뮬레이션 애니메이션을 쉽게 표현 할 수 있을 것으로 판단된다. 즉 간단한 원리와 하나의 툴만 공부하여 게임 애니메이션의 폭파 연출 장면을 어떠한 게임 엔진에서도 범용으로 모두 사용할 수 있고 마음껏 원하는 대로 조절 할 수 있다면 최고의 타격감을 낼 수 있는 폭파 장면을 연출 할 수 있을 것으로 기대 된다.

## 2. 본론

### 2.1 제작 프로세스

본 연구에서는 3D Tool로 3ds Max 2019를 사용하여 방법론을 제시하고자 한다.



[그림 1] 제작 프로세스

[Fig. 1] Production process

폭파 이펙트 제작 원리는 캐릭터 모델링과 애니메이션 제작 후 익스포트(Export)하는 방식과 동일한 원리를 가지고 응용하여 제작 한다. 즉 모든 게임 엔진에는 캐릭터 모델링과 애니메이션을 익스포트 하여 사용하기 때문에 이 방식을 사용하면 폭파 애니메이션도 당연히 사용 할 수 있게 된다는 원리 이다. 캐릭터 애니메이션의 제작 프로세스는 3D Tool에서 하나의 메쉬(mesh)를 리깅

(Rigging)하여 애니메이션 키(key)를 잡고 익스포트 하여 게임 엔진으로 옮기는 방식을 사용하고 있다.

첫째, 폭발이 진행 될 오브젝트 메쉬를 생성 한다. 둘째, 3ds max 용 플러그 인 Fracture Voronoi [13] 를 사용하여 자연스럽게 메쉬를 나눈다. 나눈 수 만큼 파편의 수가 된다. 물론 맥스의 기본 기능인 Cut과 Shell을 사용하여도 된다. 셋째, 3ds max 자체 물리엔진 기능인 massFX [12]를 사용하여 자연스러운 폭발 시뮬레이션 애니메이션을 만들어서 파편 마다 애니메이션 키 값이 부여 되도록 한다. 넷째, 파편 수만큼 본을 생성 한다. 그리고 더미(Dummy)하나를 만들어서 그곳에 모든 본을 링크(Link)를 한다. 다섯째, 3ds max 자체 기능인 Curve Editor를 사용하여 시뮬레이션 되어져 있는 메쉬(mesh)키 값을 본(Bone)을 파편 수만큼 만들고 본에 키 값을 복사 한다. 여섯째, 리깅 툴인 Physique을 사용하여 전체를 완성 한다. 일곱째, 엔진에 맞게 익스포트 한다. [그림 1]은 제작 프로세스에 관한 그림이다.

#### 1) Creating an object mesh to be blasted

예를 들고자 맥스의 기본 오브젝트(Geometry) 주전자(Teapot)을 생성한다.

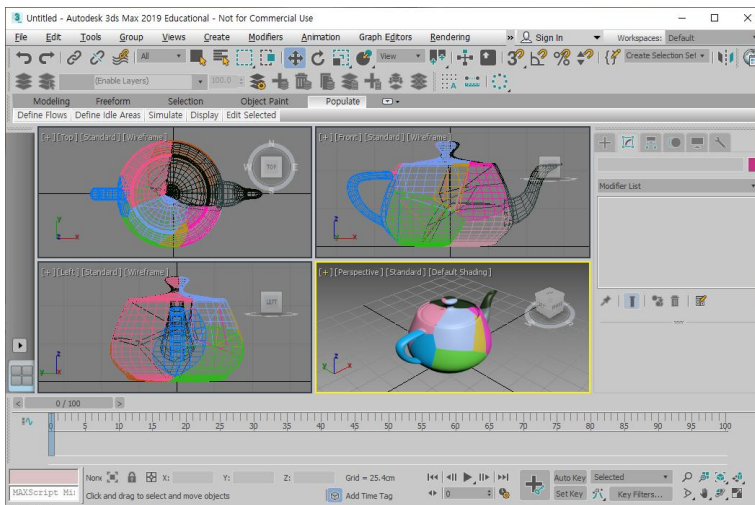
#### 2) Use Fracture Voronoi

생성 된 오브젝트를 파편 오브젝트로 만들고자 한다. 맥스의 기본 기능 등을 사용하여 파편을 만들 수 있지만 가장 쉽고 빠르게 적용 할 수 있는 맥스 와 호환이 되는 플러그 인(Plug-In)을 활용 하고자 한다. 물론 플러그인을 사용하지 않고 맥스의 기본 기능의 Cut과 shell 기능을 이용해서 만들어도 되지만 시간을 줄이기 위해서 무료 플러그 인을 사용 한다. 첫째, Scriptspot 사이트에서 Fracture Voronoi를 검색 후 Attachment에서 Fracture Voronoi\_v1.1.ms를 다운 받는다. 둘째, Fracture Voronoi\_v1.1.ms 파일을 맥스 화면 창에 드래그 해서 적용 시킨다. 다음 Fracture Voronoi의 오브젝트를 선택 후 / 맥스 에 있는 오브젝트를 선택하면 활성화가 되는데, 이때 Nb Parts에서 파편 조각 수를 맞춘 다음 Break in 10을 누르면 된다. 예를 들기 위해 10개의 조각으로 하고자 한다. 10개의 조각으로 Voronoi 방법을 통해 자동으로 파편 조각을 만들어 준다. [그림 2]는 Fracture Voronoi 사용에 관한 그림이다.

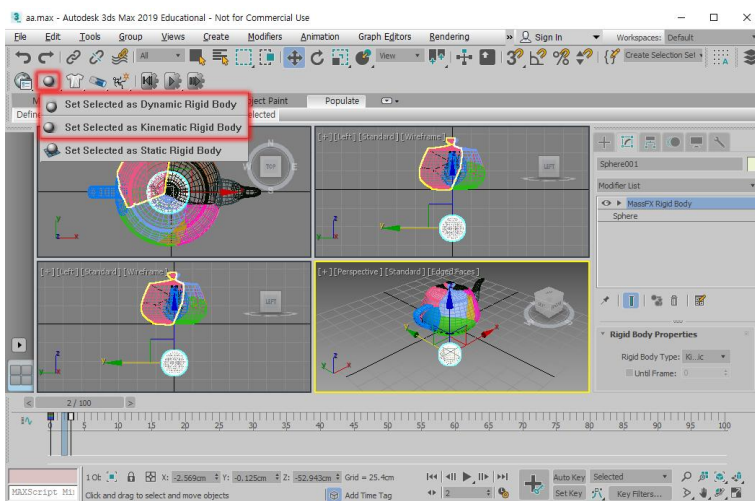
#### 3) Use massFX

10개의 조각을 폭발 시뮬레이션 진행을 하고자 한다. 3ds max 자체 물리 엔진인 massFX를 사용하고자 한다. Animation/ massFX가 있으나 가급적 아이콘 기능을 사용하면 쉽게 더 빨리 진행 할 수 있다. 첫째, 폭발 시뮬레이션을 연출하기 위해서는 주전자 밑에 오브젝트 애니메이션을 따로 만들어서 주전자에 충격을 가하면 쉽게 폭발하는 시뮬레이션을 만들 수 있다. 즉 주전자 밑에 공(Sphere)를 생성 한 뒤 밑에서 위로 주전자 파편까지 애니메이션을 주면 된다. (0프레임 공 Z축 -150cm에서 3프레임 뒤 공 Z축 10cm까지 위로 애니메이션을 준다.) 둘째, 10개의 조각 메쉬(mesh)를 전부 선택하고 Set Selected as Dynamic Rigid body를 클릭하여 적용한다. 공(Sphere)를 선택하고

는 Set Selected as Kinematic Rigid body를 클릭하여 적용 하면 된다. [그림 3]은 massFX 사용에 관한 그림이다.



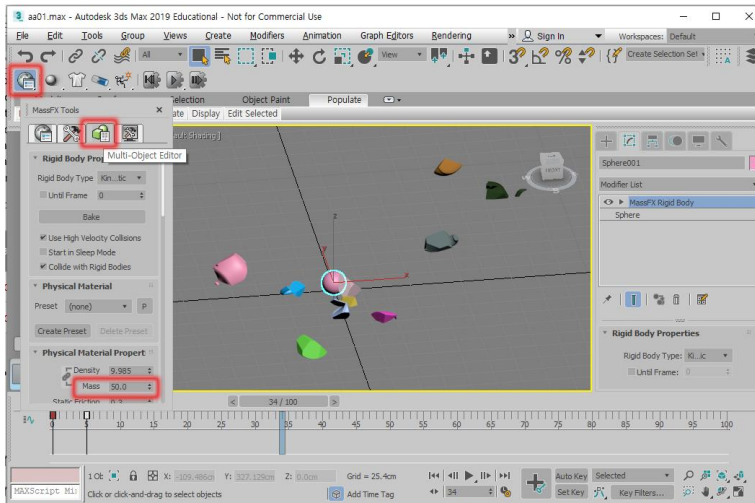
[그림 2] Fracture Voronoi 사용  
[Fig. 2] Using Fracture Voronoi



[그림 3] massFX 사용  
[Fig. 3] Using massFX

셋째, Star Simulation 으로 플레이를 해본다. 원하는 파편 시뮬레이션이 나올 때까지 MassFX Tools에서 중력을 나타내는 Ground Height 조절, 오브젝트나 주전자 파편을 각각 선택해서 Multi-Object Editor/ Physical Material Propert에서 무게를 나타내는 mass 값을 조절 하면 된다. [그림

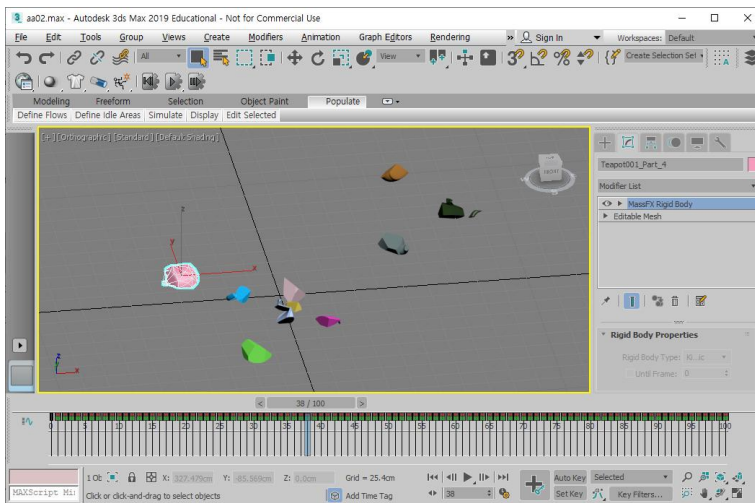
4]는 massFX 사용에 관한 그림이다.



[그림 4] massFX 사용 (1)

[Fig. 4] Using massFX (1)

넷째, 시뮬레이션 된 파편들에 애니메이션 키(Key)값을 주기 위해서는 MassFX Tools에서 Simulation Tools의 Bake All을 클릭해서 키 값을 준다. 이 후 원하는 시뮬레이션이 나오면 공 (Sphere)는 삭제 시켜서 파편 시뮬레이션만 남긴다. [그림 5]는 massFX 사용에 관한 그림이다.

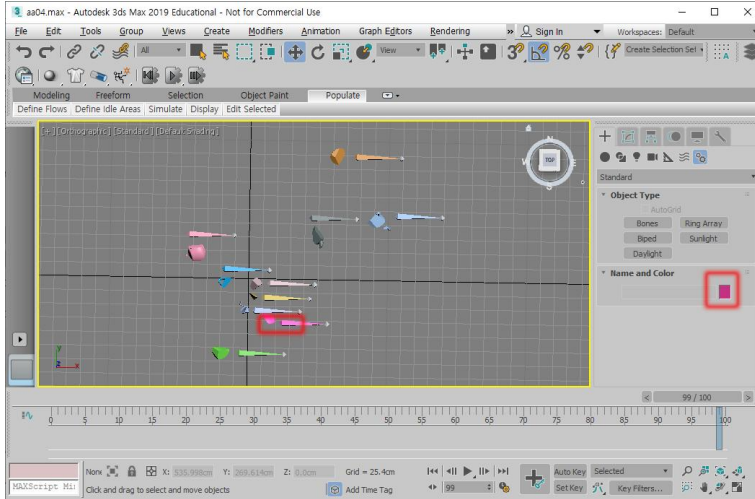


[그림 5] massFX 사용

[Fig. 5] Using massFX

#### 4) Creating bones and dummy

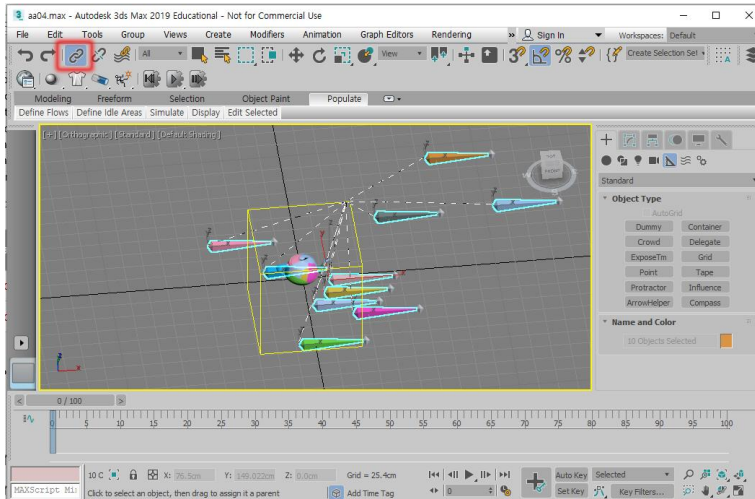
파편의 수만큼 본을 생성 하고 더미 한 개를 만들면 된다. 예제용으로는 10개의 파편 오브젝트가 있기에 10개의 본을 생성 하고 더미 1개를 만들 면된다. 첫째, 파편 숫자가 적어서 파편 색과 본 색을 통일하면 용이한데, [그림 6]은 본과 더미 생성 사용에 관한 그림이다.



[그림 6] 본과 더미 생성

[Fig. 6] Creating bones and dummy

둘째, 더미를 주전자 위치에 만들고, (상위) 본 모두를 선택해서 더미에 링크를 건다. [그림 7]은 본을 더미에 링크 관한 그림이다.

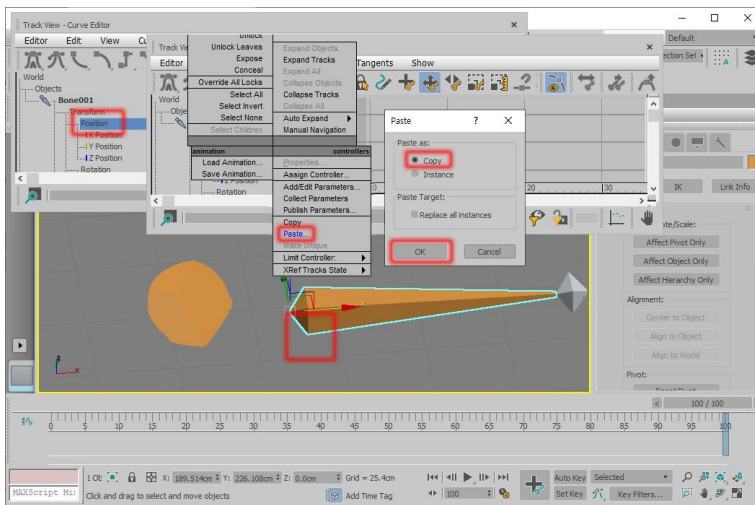


[그림 7] 본을 더미에 링크

[Fig. 7] Link Bones to Dummy

### 5) Use Curve Editor

주전자 파편의 애니메이션 키 값을 본에 복사를 하고자 하는데, 이는 3ds max 자체 기능인 커브 에디터(Curve Editor)를 사용하면 된다. 첫째, 커브 에디터(Curve Editor)를 오픈 한다. 둘째, 주전자 파편 한 개를 선택 한 뒤 커브에디터에서 MassFX Baked Position 선택하고 오른쪽 마우스 클릭을 하여 controllers의 copy를 눌러서 파편 키 값을 복사한다. 셋째, 같은 색 본을 선택 한 뒤 커브에디터에서 Transform 의 Position을 선택 후 오른쪽 마우스 클릭을 해서 controllers의 Paste를 클릭 뒤 최종 OK를 클릭해서 파편 키 값을 복사하면 된다. [그림 8]은 커브에디터 사용, 복사 및 붙여넣기에 관한 그림이다.



[그림 8] 커브에디터 사용, 복사 및 붙여넣기

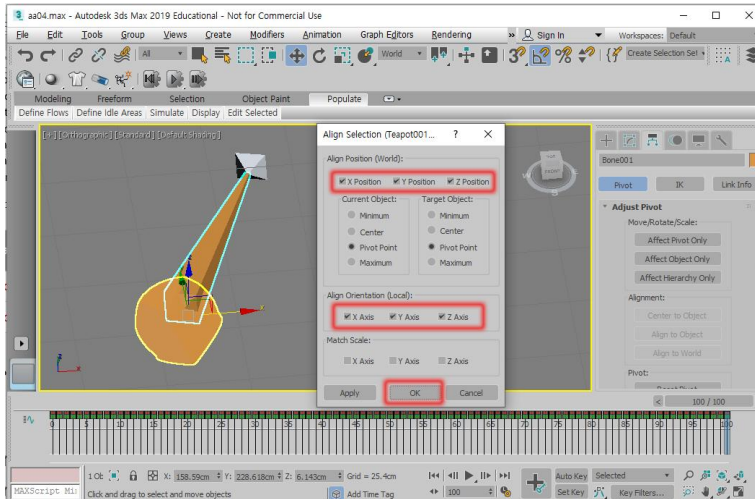
[Fig. 8] Use Curve Editor, Copy and Paste

넷째, 주전자 파편 한 개를 선택 한 뒤 커브에디터에서 MassFX Baked Rotation 선택하고 오른쪽 마우스 클릭을 하여 controllers의 copy를 눌러서 파편 키 값을 복사한다. 다섯째, 같은 색 본을 선택 한 뒤 커브에디터에서 Transform 의 Rotation을 선택 후 오른쪽 마우스 클릭을 해서 controllers의 Paste를 클릭 뒤 최종 OK를 클릭해서 파편 키 값을 복사한다. 마지막으로 맥스 Align (단축키 Alt+A)를 클릭해서 파편 메쉬와 본의 축을 똑같이 정렬 하면 된다. [그림 9]는 정렬에 관한 그림이다.

즉 간단히 정리하면 파편 메쉬(Mesh) 선택 커브에디터에서 위치(Position)값과 회전(Rotation)값을 복사 하고 본(Bone)을 선택하여 커브에디터에서 위치(Position)값과 회전(Rotation)값을 페이스트(Paste)를 한다. 파편 수만큼 반복하여 모든 메쉬 파편 애니메이션 값을 본에 복사하면 된다.

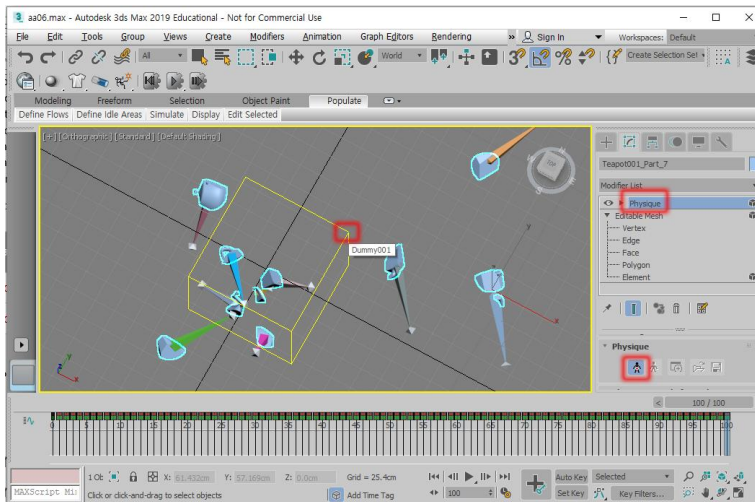
### 6) rigging Physique

파편 메쉬(Mesh)의 모든 키 값은 이미 본에 다 복사가 된 상태이기에 이제 여러 조각의 메쉬는 하나의 메쉬(Attach)로 만들어서 각각의 본에 귀속이 되도록 리깅을 하면 완성이 된다. 3ds max에서는 스킨(Skin)과 피직(Physique) 두 가지 방법으로 리깅을 할 수 있는데, 흩어져 있는 메쉬를 리깅할 때에는 피직(Physique)이 더 편리하여 피직으로 사용하고자 한다. 첫째, Attach 된 하나의 메쉬를 선택하고 Modify에서 Physique을 적용 한다. 둘째, Physique에서 Attach to Node를 클릭 한 뒤 더미(Dummy001)를 클릭 하면 된다. [그림 10]은 피직으로 리깅에 관한 그림이다.



[그림 9] 정렬

[Fig. 9] Align



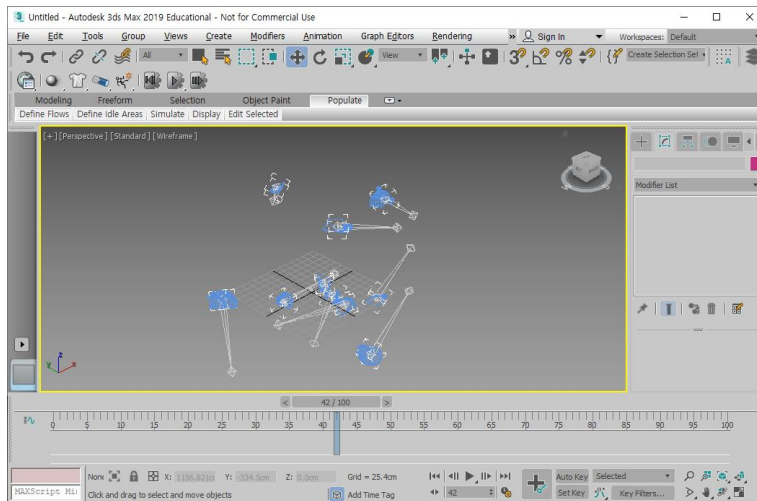
[그림 10] 피직으로 리깅

[Fig. 10] Rigging Physique

셋째, Physique Initialization에서 Blending Between Link에서 No Blending으로 선택하고 Initialize를 클릭하면 된다. 넷째, Modify List에서 Physique / Vertex를 선택하고 Vertex type 3개 모두 선택 한 뒤 / Blending Between Links에서 No Blending으로 선택하고 / 파편 조각 메쉬 하나를 선택하면 된다. 이후 Assign to Link를 클릭하고 Lock Assignments / Hide 순으로 클릭하면 리깅이 된 메쉬는 깨끗하게 버텍스(Vertex)가 안보이고 리깅 된 메쉬와 안된 메쉬 구별이 쉽게 된다. 이렇게 모든 파편 메쉬들 마다 Physique으로 리깅을 하면 완성이 된다.

#### 7) Export 3ds max file

3D Tool에서 제작한 파일을 게임 엔진에서 적용하려면 변환을 해야 하는데 대중적인 게임 엔진인 유니티(Unity)로 익스포트(Export)하기 위해서 FBX파일로 변환하고자 한다. 첫째, 파일(file) / 익스포트(export) / 익스포트(export)를 클릭 하여 저장하고 싶은 위치에 파일 명을 적고 Save 클릭 한다. 둘째, FBX Export 창에서 Bake Animation을 체크 한 뒤 OK 클릭 하면 fbx 파일이 만들어 진다. 마지막으로 fbx 파일을 임포트(Import)해서 보면 완성된 fbx파일을 확인 할 수 있다. [그림 11]은 익스포트 맥스 파일에 관한 그림이다.



[그림 11] 익스포트 3ds 맥스 파일

[Fig. 11] Export 3ds max file

### 3. 결론

캐릭터 하나에 계층구조로 된 본을 리깅하여 캐릭터 모델링 및 애니메이션을 익스포트 하는 원리를 이용하여 파편(폭파) 애니메이션을 만들어 보았다. 물리 엔진을 적용한 키 값으로 애니메이션

되어서 실제 시뮬레이션 효과와 동일하다. 3D Tool에서 제공 되는 기본 기능만을 가지고 어느 게임 엔진에서든 사용 할 수 있는 폭파 시뮬레이션 (애니메이션)이펙트를 만드는 방법을 제공 하였다. 지금 제시한 제작 방법은 기존에 선행 연구가 전무하고 게임 업계에서도 방법론을 찾지 못해서 사용하지 못하는 기법이다. 필자의 개발 경험과 기본 원리를 응용하여 만들어낸 독창적인 제작 방법임을 말하고 싶다.

이를 확인하고자 게임 이펙트 개발자 20명을 대상으로 2차 FGI를 실시하였다. 질문은 다음과 같다. 첫째 질문, 피직을 이용한 폭파 애니메이션이 독창적인가? 둘째 질문, 피직을 이용한 폭파 애니메이션이 유용한가? 셋째 질문, 방식을 습득하면 게임 제작에 직접 활용 할 것인가?

첫째 질문에 대해서 18명(90%)이 “독창적”이라고 대답했고, “보통” 1명(5%)과 “나쁘지 않다” 1명(5%)이 대답 하였다. 둘째 질문에 대해서 18명(90%)이 “유용하다”라고 대답하였고, “보통” 2명(10%)이라고 대답 했다. 셋째 질문에 대해서 18명(90%)이 “활용한다”라고 대답하였고, “본인이 쓰는 방식이 좋다”라고 2명(10%)이라고 대답 하였다. 이와 같이 독창적이고 유용하며 직접 활용 하겠다고 하는 개발자들이 평균 90%이기에 지금의 이 연구는 가치가 있다라고 판단 한다. 단순한 튜토리얼이 아니라 원리를 이해하고 원리를 응용하여 범용 사용이 가능한 방법을 제시했다는 점에서 의의가 있다. 예를 들고자 3ds max와 Unity라는 게임 엔진만 설명하였는데, 마야나 언리얼 등 원리는 동일하기에 응용하여 사용하면 타격감 높은 파편 이펙트 장면을 연출 할 수 있다. 이 연구를 통해 타격감 있는 사물 애니메이션이 게임 엔진에서 유용하게 사용되고 시간도 단축하여 제작 하는데 도움이 되길 바란다.

## References

- [1] N. H. Kim, T. W. Kim, "Advance Effect of Hitting Sense in Shooting Game-Center for the Beetlewing and 1945 Plus", *Journal of Korea Multimedia Society*, Vol.7, No. 2, February 2004. PP. 223-230.
- [2] S. J. Moon, H. J. Cho, "A Study on Enhancing Efficiency for Feeling-of-Hit in Games", *Korea Game Society*, Vol.12, No.2, February 2012, PP. 4.
- [3] J. S. park, "Study of network game engine technology for distributed processing" Master's thesis, Dept. of internet information, Graduate School, Dankook University, Republic of Korea, 2008.
- [4] E. J. Kim, "3D Game Engine for Interactive Snowboard Game in Virtual Reality Environment", Doctoral dissertation, Computer Engineering, Myongji University Graduate School, Republic of Korea, 2006.
- [5] F. Chucholowski, C. Gndt, C. Hepperle, S. Hafner, "Close to reality surrounding model for virtual testing of autonomous driving and ADAS", *Proceedings of the 13th International Symposium on Advanced Vehicle Control (AVEC'16)*, September 13-16, 2016, Munich, Germany, pp. 85-92, doi: <https://doi.org/10.1201/9781315265285>.
- [6] F. Chucholowski, C. Gndt, C. Hepperle, "GPU-supported real-time sensor modelling for ADAS development in Simulink", *mathworks.com*, <https://www.mathworks.com>, (accessed May 1, 2021).
- [7] C. Y. Woo, "Implementation of Physical Fitness Promotion System for Senior Generation using Kinect Sensor and Unity3D Engine", Master's thesis, Dept. of Electric, Electronic and Information Communication Engineering, The Graduate School, Tongmyong University, Republic of Korea, 2016.
- [8] H. Chan, "A Study on Network Based VR Competition Game Using Simulator", Master's thesis, Dept. of Game Engineering, The Graduate School of Game Gachon University, Republic of Korea, 2019.
- [9] Y. J. Park, "Mesh segmentation using global geometric characteristics based on cognitive science" Doctoral dissertation, Dept. of Information Industrial Engineering Graduate School, Chungbuk National University, Republic of Korea, 2009.
- [10] H. Won, "(An) Implementation and performance analysis of physics engine for 3D game", Master's thesis, Dept. of Computer Engineering, The Graduate School of Sejong University, Republic of Korea, 2003.
- [11] Reactor, "Reactor-Introducing Dynamics Simulation", *autodesk.com*, <http://docs.autodesk.com>, (accessed March 12, 2021).
- [12] MassFX, "Using MassFX in 3ds Max", *autodesk.com*, <https://knowledge.autodesk.com>, (accessed Dec 12, 2018).
- [13] Fracture Voronoi, "Fracture Voronoi", *scriptspot.com*, <http://www.scriptspot.com>, (accessed Apr 5, 2009).