

□ 기술애설 □

게임 제작 툴의 개발과 활용

단국대학교 박용범*·권경희*

1. 서 론

컴퓨터의 빠른 보급과 하드웨어의 빠른 발달로 인해 가정용 컴퓨터를 이용한 게임은 급속도로 퍼져 나가고 있다. 더욱이 멀티미디어의 도입으로 컴퓨터 게임은 고급화의 계기를 맞이하고 있다. 이제 컴퓨터 게임의 제작은 더 이상 흥미의 대상이나 취미 활동이 아닌 하나의 분야로서 자리잡고 있다. 게임 산업은 영화 산업처럼 엄청난 부가 가치를 올릴 수 있는 산업이며, 또한 첨단 분야이다. 최근 윈도우즈 95는 게임 지원을 운영체제의 일부로 포함시키려 하고 있다. 이는 게임이 컴퓨터에서 차지하는 비중을 산업계에서 어떻게 보고 있는가를 보여주는 예이다. 게임을 지원하지 않고서는 운영체제 시장의 왕좌를 더 이상 지키기 힘들 것이라는 생각에서 윈도우즈 95는 게임 지원을 운영체제의 일부분으로 채택하려고 하고 있는 것이다. 이것만 보더라도 더 이상 게임은 흥미의 대상이나 취미 활동이 아니라는 것을 알 수 있다[2][6].

하지만 이러한 인식이 있기 전까지, 컴퓨터 게임은 소수의 작은 회사를 중심으로 개발되었으며 일부 큰 기업들의 단순한 관심거리였다. 또한 그 제작방식에 있어서도 확실한 체계를 가지지 못하고 가내 수공업처럼 비효율적으로 제작되었다. 이때만 해도 게임을 만들기 위해 또 다른 프로그램이 필요하다는 것을 인지하지 못하였다. 시간이 지나면서 게임 개발의 경험이 생긴 몇몇 업체에서는 이러한 비효율성을 극복하기 위해 게임을 만드는데 도움이 되는

프로그램을 만들었다. 이것이 바로 게임 제작 툴(TOOL)의 시작이다. 툴을 개발하여 실제 게임 제작에서 사용해본 제작자들은 이것이 효율적인 개발에 도움을 주는 것을 체험하게 되었고 게임 개발에서의 게임 제작 툴의 중요성이 점점 높아졌다. 몇몇 회사들은 개발 기간 단축, 데이터 보호, 각 분야 작업의 부담을 덜어주는 이러한 툴들에 가장 높은 비중을 두어 개발하기도 했다[1][3][4].

게임 개발이 과거의 생각과는 달리 황금 시장을 이루는 하나의 산업으로 생각되고 있는 지금, 예전 보다 더 강력하고 효율적인 툴이 필요함은 두말할 나위도 없을 것이다. 여기에서 효율적인 툴이란, 게임 개발사의 지출과 관련이 깊은 개발 기간을 최대한으로 단축하면서도 보다 많은 소득을 올릴 수 있는 고차원의 게임을 쉽게 개발하도록 도와주는 프로그램을 말한다. 이러한 툴을 가짐으로써 게임 제작자들은 마치 양질의 제품을 공장에서 찍어내듯이 수준 높은 새로운 게임을 빨리 소비자들에게 공급할 수 있다. 여기에서는 게임 제작 툴의 발달사를 알아보고 효율적인 게임 제작 툴이 가져야 할 요건을 다루어 보려 한다. 마지막으로 효율적인 게임 제작 툴인 통합 게임 제작 툴의 제작과 그 활용을 소개하고자 한다.

2. 게임 제작 툴의 발달

게임 툴은 게임제작이 본격화되며 그 필요성이 부각되었다. 만화 영화 제작과 같이 단순 반복작업이 많은 게임 제작에 있어서 작업 양을 줄여주고 작업을 단순화시켜주는 툴의 필요

*정회원

성은 일찍이 알려졌으나 초창기에는 게임 제작 툴의 기능과 형태를 정의하기 힘들었다. 이러한 연유로 게임 제작 툴은 게임 제작 회사별로 자신들의 용도를 만족시키기 위해 제작되어 사용되었다. 게임 제작이 활성화되어감에 따라 게임 제작 툴의 요건들이 정리되었다. 여기에서는 게임 제작 툴들이 필요성에 의해 나타난 것을 기준으로 게임 제작 툴들의 발달을 1~4 세대로 구분하여 보았다. 세대가 지남에 따라 게임 제작 툴은 지속적으로 새로운 기능이 추가되었고 새로운 형태의 툴로 발전되었다. 각 세대별 툴들의 특징과 발달 과정을 살펴봄으로써 게임 산업화에 필요한 툴의 이상적인 모형과 발달되어야 할 방향을 알아보기로 하겠다.

2.1 1세대 게임 제작 툴

과거에는 게임 제작 툴이라고 하면 주로 그래픽 분야만을 지원하는 드로잉(Drawing) 프로그램을 말했다. 기존에 나와 있는 드로잉 프로그램이 자신의 회사에서 필요로 하는 기능을 지원하지 않으면 게임 제작 회사는 자신이 원하는 기능을 넣어 새로운 드로잉 프로그램을 만들었다. 이 드로잉 프로그램의 도움으로 그래픽 디자이너들은 강력한 기능의 편리한 툴을 사용하여 같은 시간 동안에 주인공 그림을 한 장이라도 더 그릴 수 있었고, 또한 더 화려한 효과를 넣을 수도 있었다. 이것의 대표적인 예가 “일렉트로닉 아트”의 “디렉스 페인트”이다. 게임 제작을 위해 제작된 이 프로그램은 그 편리한 기능 덕분에 게임만이 아닌 다른 분야에서도 사용되는 드로잉 툴이 되었고, 이를 개발한 “일렉트로닉 아트”도 게임 제작 툴로 만든 이 드로잉 프로그램을 게임 제작 툴이 아닌 상용 프로그램으로 세상에 내놓았다.

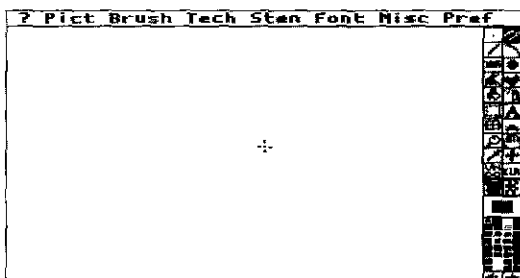


그림 1 디렉스 페인트의 외관

2.2 2세대 게임 제작 툴

게임은 만화 영화와 비슷해서 각 인물들의 하나의 움직임을 위해 여러 개의 동작 그림이 필요하다. 이 동작 그림들의 연결 관계가 그 인물의 움직임의 자연스러움을 결정하게 되는 것이다. 그래서 게임의 꽃인 주인공들과 악당들의 자연스러운 움직임을 위해 애니메이션 기법이 추가된다. 게임 제작에 있어서 인물들의 애니메이션은 필수 불가결하며 이를 매번 프로그래밍 하는 것은 매우 번거로운 일이다. 이런 이유로 1세대에 그저 그림만 그려 주던 게임 제작툴에 애니메이션 효과를 자동으로 제어해주는 “애니메이터”라는 게임 제작 기능이 추가된다. “애니메이터”의 주된 역할은 각각의 동작이 다음 동작과 자연스럽게 연결이 되는지를 확인하고, 부자연스럽다면 이를 수정하는 것이다. 수정을 위해 “애니메이터”는 드로잉 툴의 기능을 함께 포함하고 있다. 이러한 종류의 게임 제작툴의 예로는 “일렉트로닉 아트”의 “디렉스 페인트 애니메이션”이 있다. 이름에서 알 수 있듯이 “디렉스 페인트”에 “애니메이션” 기능이 추가된 것이다.

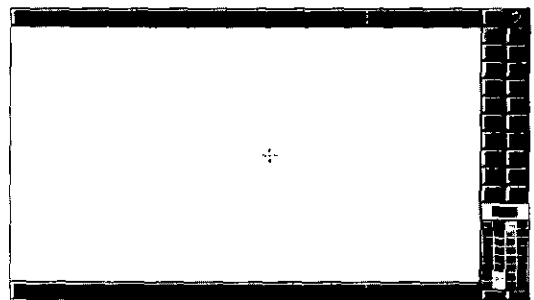


그림 2 디렉스 페인트 애니메이션의 외관

2.3 3세대 게임 제작 툴

1세대와 2세대 게임 제작 툴은 주로 그래픽 처리에 중점을 두어 단순 반복 작업인 그래픽 작업만을 지원하여 주었다. 그러나 이를 통해 만들어진 그림 데이터들은 프로그래머가 일일이 프로그래밍에서 사용할 수 있도록 재수정하여야만 했다. 예를 들어 “스프라이트(sprite)” 처리를 살펴보자. “스프라이트”란 게임에 나오

는 인물에 대한 그림 중 주위의 필요 없는 부분을 제거한 것이다. 즉 하나의 화면에 어떤 인물을 그리면, 그림이 그려지지 않은 주위의 공백을 제거한 실제로 그 인물이 그려진 부분만을 말한다.

스프라이트 처리를 하지 않으면 등장 인물의 형태가 원형이든 다각형이든 간에 사각형의 그림판이 이동하게 된다. 따라서 스프라이트란 어떤 인물을 어떤 배경 위에 출력할 때, 인물의 실제 부분이 아닌 다른 부분이 출력되어 배경을 손상시키는 것을 막는 작업을 의미한다. 이러한 작업을 하지 않으면 배경 위에 출력된 인물의 주위에는 공백의 사각형이 그려져서 그 인물이 그 배경 위에 있는 건는 것처럼 보이지 않는다.

이 작업은 이전까지는 그림을 넘겨받은 프로그래머가 했다. 그러나 이 작업 역시 단순한 반복의 작업이며 자동화하기 쉬운 기능이었다. 따라서 프로그래머들은 이 작업을 2세대 게임 제작 툴인 “애니메이터”와 결합시켰다. 이 툴을 사용하여 그래픽 디자이너가 그림을 그려 저장하면 그 저장된 자료는 프로그래머가 프로그래밍에서 바로 읽어들이 사용할 수 있다. 또한 이 툴에는 “애니메이터”의 기능을 보강하여 각 인물의 세밀한 위치 조정까지도 넣어 “애니메이터”보다 더 자연스러운 움직임 구현할 수 있다. 이러한 스프라이트 기능이 내장된 툴을 “스프라이트 에디터”라고 하며, 각 개발사의 특성에 맞추어 제작되었기 때문에 상용화된 프로그램은 없다.

2.4 4세대 게임 제작 툴

3세대까지의 게임 제작 툴은 게임에서 사용되는 순수 데이터들을 만드는데 사용되었다. 하지만 과거와는 달리 게임 시나리오(줄거리)의 구성이 복잡해지고 게임의 스케일이 커짐에 따라 각 스테이지(판)를 구성하는데 있어서 여러 가지 어려움을 느끼게 된다. 먼저 디렉터와 프로그래머, 그리고 시나리오 작가와의 의사소통을 하는데 있어서, 시나리오를 완벽하게 표현할 수 없어 느끼게 되는 어려움이 존재하고 게임 배경을 저장하는데 많은 용량이 필요하다는 문제점을 느끼게 된다. 이러한 문제점

은 특히 롤플레이팅과 같은 장르의 게임에서 두드러진다. 롤플레이팅은 길러그와 같은 아케이드와는 달리 시나리오가 강조되고 스케일 또한 크기 때문이다.

이러한 점은 해결책으로 각 스테이지(판)를 쉽게 구성할 수 있는 툴과 몇몇 배경 요소들로 큰 스케일의 배경을 만드는(마치 블록 짓기와 유사한) 툴을 요구하게 된다. 이러한 기능이 포함된 툴을 맵툴이라고 하며, 주로 큰 스케일의 배경을 만드는 기능만 있는 것도 있지만 각 스테이지 구성을 할 수 있도록 하는 기능이 포함된 것이 대부분이다.

이러한 툴을 사용하여 디렉터나 시나리오 작가가 자신이 생각하고 있는 구성을 만들어 놓으면, 프로그래머는 그 구성을 보고 프로그램을 작성할 수 있게 된다. 이렇게 함으로써 맵툴은 디렉터, 시나리오 작가 그리고 프로그래머간의 확실한 의사소통을 이루게 한다. 따라서 맵툴을 이용하면 게임 프로그램이 디렉터 또는 시나리오 작가의 원래 의도와 가까운 구성을 가지게 된다. 이 또한 각 개발사의 특성에 맞추어 제작되었기 때문에 상용화된 프로그램은 없다.

하지만, 이와 유사한 개념을 도입한 “쯔꾸르”라는 프로그램이 있다. 이 프로그램은 이러한 기능을 조금 더 발전시킨 형태로서, 내장된 그림을 배치시키고 이벤트가 적절하게 발생하도록 만들어 줌으로서 하나의 게임을 만들어 준다. 하지만, 이를 사용하여 만들어진 게임은 자체적으로 실행하지 못하고 “쯔꾸르”를 통해서만 실행이 되므로 상용화하기 힘들고, 그 실행 속도 또한 느리기 때문에 실제 산업에는 사용하지 못할 듯 싶다. 그러나 이것은 다음 세대의 툴을 위한 좋은 아이디어를 제공한다.

이렇게 해서 게임 제작툴의 발달을 1세대에서 4세대까지로 나누어 살펴왔다. 각 세대의 특징을 다시 한 번 정리하여 보면 다음과 같다.

- 1세대 : 드로잉 도구
- 2세대 : 애니메이션 도구
- 3세대 : 스프라이트 처리 도구
- 4세대 : 스테이지 구성 도구

3. 통합 게임 제작 툴(Integrated Game Development Tool)의 요소

위에서 “�꾸르”라는 프로그램을 소개했다. 그리고 이 프로그램은 좋은 아이디어를 제공한다고 밝혔다. 통합 게임 제작 툴(Integrated Game Development Tool, IGDT)은 쯤꾸르에서 비추어진 아이디어를 좀더 발전시킨 형태이다. 통합 게임 제작 툴(IGDT)이 가져야 할 요소를 알아보기로 하자.

4세대까지의 게임 제작 툴들은 서로 독립적인 도구의 집합 형태의 프로그램이었다. 따라서 하나의 툴에서 완성된 데이터가 다른 툴에서 수정되기 위해서는, 완성된 데이터를 저장하고 그 툴을 종료한 후 다른 툴을 실행시켜 데이터를 다시 읽어들이어 수정해야 한다. 이러한 점 또한 효율성을 낮추는 요인 중의 하나이다. 따라서 이러한 점을 보완하기 위해 통합 게임 제작 툴은 위의 툴들을 하나의 프로그램 안에 두어 중간 과정에서 수정되는 데이터를 임시로 저장하고 다시 불러들이는 것을 줄이고 프로그램을 종료하고 재실행하는 것을 막아, 개발 기간에 포함되는 불필요한 시간을 줄여주는 역할을 해야 할 것이다. 즉, 원하는 데이터를 완성하기 전에는 프로그램을 빠져나가지 않아도 되고 가져오기와 저장(Load, Save)을 하지 않아도 되는 툴이어야 할 것이다.

하지만 이것만으로는 기존 것들과 별다른 차이가 없다. 통합 게임 제작 툴의 정말로 중요한 새로운 기능은 바로 실행 코드화할 수 있는 기능이다. 마치 비주얼 툴과 같이 쉽게 그리고 전문적인 프로그래밍 지식이 없더라도 게임의 실행 파일을 만들 수 있도록 하자는 것이다. 통합 게임 제작 툴에서는 게임에 등장하는 인물의 그림과 그들의 지능 정도를 쉽게 마우스를 드래그하는 것만으로 정할 수 있는 기능이 포함되어야 할 것이다. 또한 통합 게임 제작 툴에서는 각종 이벤트 발생의 적절한 시점을 결정만 하면 마우스 하나만으로 게임에 반영시킬 수 있게 만들어 줘야 한다. 이러한 기능을 가지는 통합 게임 제작 툴이 있으면 단순히 빠른 코드만을 강조(물론 빠른 코드가 중요하지만)하는 게임이 아닌, 게임 하는 사람이 경험

해 보고 싶어하던 것을 경험할 수 있는 구성이 탄탄한 게임을 보다 정밀하고 세련되게 제작할 수 있을 것이다. 물론 통합 게임 제작 툴의 기본 철학인 “완성하기 전에는 종료하지 않아도 된다.”는 점을 지키기 위해 제작된 게임은 바로 실행해볼 수 있도록 해야 할 것이다.

이러한 툴을 통해 얻을 수 있는 이점은 우리의 목표였던 개발 기간 단축과 더욱 세련되고 고차원적인 게임의 개발이다. 부가적으로 데이터를 만들기 위해 여기저기 만들어 두었던 임시 데이터 파일들을 줄일 수 있으므로 데이터를 깔끔하고 정리되게 관리할 수도 있을 것이다. 여기에 데이터 매니저(Data Manager, DM) 기능이 추가된다면 아주 효율적인 데이터 관리를 할 수 있을 것이다.

4. 통합 게임 제작 툴의 구현

통합 게임 제작 툴은 여러 툴들을 서브 프로그램으로 내장한 프로그램이고, 각 서브 프로그램에서 만든 데이터는 파일을 거치지 않고도 서로 가져다 사용할 수 있도록 해야 한다. 또한 마우스를 적극 이용하여 손쉽게 게임을 개발할 수 있도록 해야 하며, 이를 실행하면서 게임에서 생길 수 있는 버그를 잡아낼 수 있도록 해야 한다.

위의 점들을 모두 반영할 수 있는 플랫폼으로는 윈도우즈 95가 적당할 것이다. 윈도우즈 95는 여러 프로그램을 하나의 프로그램에 내장할 수 있는 MDI, 어플리케이션 간의 데이터 전송을 위한 OLE, 다중 처리를 위한 멀티 쓰레드, 마우스를 적극 지원하고 있다. 더군다나 마이크로소프트에서 야심작으로 내놓은 게임 엔진인 Game SDK를 지원하므로 윈도우즈 게임과 DOS 게임 모두를 개발할 수 있는 멀티 플랫폼을 구현할 수도 있고, 3D, 네트워크를 손쉽게 구현할 수 있을 것이다. 또한 개발 시에 LAN 등의 네트워크 환경을 적극 활용한다면 대작 게임을 체계적으로 개발하는 도움이 될 것이다.

가장 중요한 기능은 비주얼 프로그래밍 툴과 이를 사용하여 작성한 것을 실행하는 기능이다. 이것을 구현하는 데는 인터프리터를 이용하

는 것이 좋을 듯싶다. 비주얼 프로그래밍 툴이 직접 실행 파일을 만들려면 컴파일러 수준의 툴을 만들어야 하기 때문이다. 비주얼 프로그래밍 툴은 인터프리터에게 건네줄 명령어들을 간단하게 만들어내고, 인터프리터는 건너온 명령어들을 실시간으로 해석하여 빠르게 실행한다. 이 과정에 멀티 쓰레드를 적극 활용한다면 인터프리터를 거쳐 실행하더라도 실행 파일이 직접 실행되는 것과 비슷한 속도를 낼 수도 있을 것이다.

5. 결 론

제5세대 게임 제작 툴로 구분될 수 있는 통합 게임 제작 툴의 개발은 보다 빠르고 보다 쉽게 게임을 개발할 수 있는 밑거름이 될 것이다. 그러나 통합 게임 제작 툴의 특성상 여러 장르의 게임 개발을 모두 지원할 수 있는 범용 게임 제작 툴은 기대하기 어렵고 게임 장르별 특징을 지원하는 통합 게임 제작 툴을 개발하는 것이 현실적이다. 또한 게임 제작을 목표로 하는 기업이나 게임 관련 연구자에게 모두 통합 게임 제작 툴은 편리한 도구가 될 것이므로 자신의 특성에 알맞은 통합 게임 제작 툴을 먼저 개발하여 게임을 제작한다면 좋을 것이다. 현재 통합 게임 제작 툴의 개발이 본교 게임 제작 동아리에서 활발히 이루어지고 있으며 연말 경에 소개될 예정이다. 보다 효율적이며 효과적인 게임 제작 툴의 개발은 앞으로 게임 산업의 중요성을 감안할 때 매우 중요한 일이며 흥미로운 연구 분야가 될 것이다.

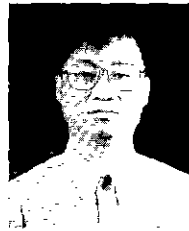
참고문헌

[1] 정영덕, SF-2 제작자와 함께하는 게임 만

들기, 연암 출판사, 1994, 8.

[2] Clayton Walnum, 윈도우 95용 Game SDK 전략 가이드, 정보문화사, 1995.
 [3] Dave Roberts, 게임 프로그래밍 따라하기, 비앤씨, 1995.
 [4] LaMonthe Rateliff, Seminor&Tyler, 게임 설계의 안쪽, 한글과컴퓨터 Press, 1993. 10.
 [5] Lamothe Rateliff, 3D 게임 프로그래밍, 성안당, 1995.
 [6] Michael J. Norton, Spells of fury building Windows 95 games using Directx 2, Waite group press, 1996.

박 용 범



1985 서강대학교 전자계산학과 학사
 1987 NY Polytechnic Univ., Computer Science, M.S.
 1991 NY Polytechnic Univ., Computer Science, Ph. D.
 1992 현대전자 산업주식회사 산전 연구소, 선임연구원
 1993~현재 단국대학교 전자계산학과, 조교수

권 경 희



1976 고려대학교 물리학과 학사
 1979~1984 한국산연 연구원, 연구원
 1986 Old Dominion Univ., Computer Science, M.S.
 1991 Louisiana State Univ., Computer Science, Ph. D.
 1993~현재 단국대학교 전자계산학과, 조교수