

교육적 관점에서 본 Unity 3D 엔진

윤석현^o

^o청강문화산업대학교 게임콘텐츠스쿨

e-mail : shyoon@ck.ac.kr^o

A Study On Unity 3D Engine in the Educational Perspective

Seok-Hyun Yoon^o

^oSchool of Game Contents, Chungkang College of Cultural Industries

● Abstract ●

게임 개발사의 경영진이나 개발자들을 통해 유니티 엔진을 다룰 수 있는 프로그래머에 대한 수요가 무척 많다는 것을 확인할 수 있다. 유니티의 사용 환경으로 과거 PC 온라인 게임을 중심으로 게임 개발이 이루어졌을 때에 비해 개발사의 수도 많아졌고 한 개발사에서 동시에 만드는 게임의 수도 크게 증가하고 있다. 본 논문에서는 Unity 3D 엔진을 심층 분석하여 게임개발 교육 현장에서의 유용성을 검토하였다. 본 연구는 게임 앱 설계 관련 프로젝트의 수행이나 교육용 게임 개발의 학습을 위한 유니티의 현주소를 조명하는 데 있다.

키워드: 유니티 3D 엔진(Unity 3D), 스크립트 언어(Script Language)

I. Introduction

Unity 3D는 다른 도구를 사용하지 않아도 게임을 개발할 수 있다. 이는 익숙한 사용자의 입장에서는 대단히 편리한 환경이라고 할 수도 있지만 한편으로는 복잡한 메뉴를 익혀야 작업을 시작할 수 있다는 점에서 진입 장벽으로 여겨질 수도 있다. 그럼에도 불구하고 일단 이들 메뉴에 익숙해지고 나면, 그 다음에는 오히려 사용하기가 쉬워진다는 점이 Unity의 장점이기도 하다.

결론적으로, 처음에는 어려워 보이는 Unity 3D의 통합 개발 환경도 조금만 익숙해지면 큰 문제가 되지 않기 때문에, Unity 3D는 현업에서 상업용 게임을 개발할 때 뿐 아니라 학교에서 학생들을 상대로 교육을 할 때에도 유용하게 활용할 수 있다고 판단하게 된다.

II. Analysis of Unity 3D

1. Integrated Develop/Environment

유니티 3D는 엔진과 레벨 에디터, 스크립트 편집을 위한 에디터가 모두 통합되어 있다. 유니티 3D의 인터페이스 중에서 가장 복잡해 보이는 곳은 인스펙터 뷰(Inspector View)인데, 이곳은 컴포넌트를 붙였다 떼었다 하면서 관리하는 곳이기 때문에 실제로는 그렇게 복잡한 것은 아니다. 대부분의 게임 오브젝트는 Transform이라는 기본 컴포넌트 외에 몇 가지 추가 컴포넌트 정도만 가지고 있기 때문에 인스펙터 뷰에서 드러나는 항목들의 개수가 그렇게 많지는 않다.

인스펙터 뷰가 복잡해지는 것은 개발자가 새로운 기능을 계속 추가하기 때문에 발생하는 현상이다. 하지만 이 경우는 대개 개발자가 해당 컴포넌트의 기능과 속성을 숙지하고 있는 상황이라, 우리가 걱정하는 것처럼 인스펙터 뷰의 복잡성이 문제가 되지는 않는다.

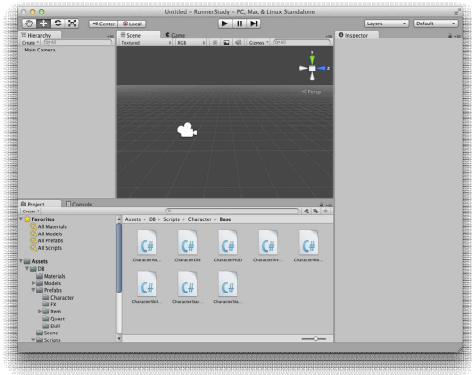


Fig. 1. Integrated Development Environment

2. Testing Support

유니티 3D로 게임을 개발할 때 편리한 점 중 하나는 테스트를 위해 컴파일할 필요가 없다는 것이다. 게임을 만드는 과정에서 언제든지 'PLAY' 버튼을 누르기만 하면 즉시 게임이 플레이되기 때문이다. 그리고 이렇게 시작된 게임은 'PAUSE' 버튼을 이용하여 언제든지 멈출 수 있고, 프레임 단위로 실행시킬 수도 있다. 따라서 버그를 찾아내는 데 대단히 유용하다.

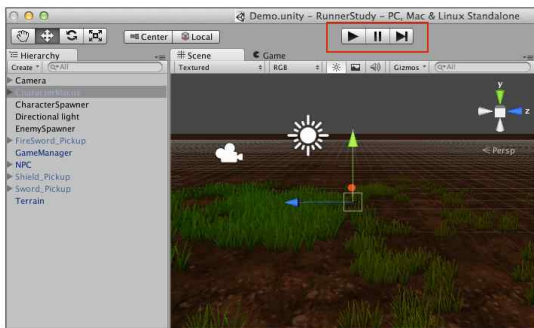


Fig. 2. Testing Support

게임 뷰에서 성능 최적화를 위해 필요한 각종 정보를 실시간으로 입수할 수 있다는 점도 장점이다. 게임 최적화를 위해 가장 중요한 드로우콜(Draw Call) 정보를 확인할 수 있는 것은 물론이고, 텍스처 개수, 비디오 메모리 사용량 등이 실시간으로 표시된다. 또한 게임 테스트 시 유용하게 사용할 수 있는 가이드 라인 표시, 아이콘 표시 기능도 개발자가 원하는 대로 활용할 수 있다.

이렇게 테스트를 위해 컴파일을 하지 않아도 된다는 점은 게임 개발의 생산성을 크게 높여 준다. 기존에는 게임을 코딩하고 테스트하기 위해 반드시 컴파일 단계를 거쳐야 했고, 이 단계는 상당한 시간을 소모했다. 하지만 게임 개발 도중 간단히 테스트할 수 있는 기능을 제공함으로써, 유니티는 게임 개발 과정에서 불필요한 시간과 노력의 낭비를 하지 않도록 도와 주고 있다. 따라서 유니티로 게임을 개발할 경우, 개발 생산성을 높일 수 있을 것이라고 판단된다.

3. Component System

유니티 3D는 컴포넌트 방식의 게임개발 엔진이다. 게임 오브젝트에 고유의 기능과 속성을 가진 다양한 컴포넌트를 추가하는 방식으로 기능을 얼마든지 추가할 수 있다는 뜻이다. 모든 기능이 컴포넌트에 의해 관리되기 때문에 개발자들은 한 번에 하나의 컴포넌트에만 집중하면 된다. 예를 들어 게임 오브젝트에 중력을 적용하고 싶다면 리지드바디(Rigidbody) 컴포넌트를 덧붙이면 되고 사운드를 플레이 하고 싶다면 오디오 소스(Audio Source) 컴포넌트를 붙이면 된다.

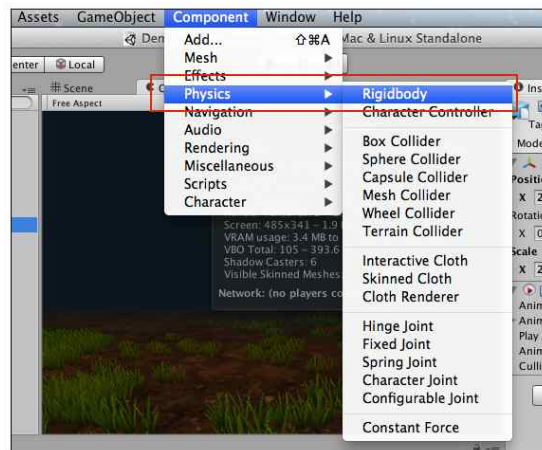


Fig. 3. Component System

4. Script

유니티 3D는 C#과 자바스크립트, 그리고 Boo의 세 가지 스크립트 언어를 제공한다. 대부분 C#이나 자바스크립트를 사용한다. C#으로 스크립트를 짜게 되면 클래스를 자유롭게 사용할 수 있고, 보다 고급 수준의 프로그래밍이 가능하기 때문에 장기적인 관점에서는 C#을 선택하는 것이 옳은 판단으로 보인다. 하지만 간단한 게임을 빠르게 개발하고자 한다면, 자바스크립트를 사용할 수도 있다. C#을 선택했을 때의 장점은 마이크로소프트사에서 제공하는 비주얼 스튜디오 에디터를 사용할 수 있기 때문이다. 이러한 장점 때문이라도 향후 유니티 교육 시, C#을 기준으로 진행하는 것이 옳다고 판단된다.

5. Collider Component

유니티 3D는 그래픽 칩셋 제조사인 엔비디아(Nvidia)사에서 공개한 피직스(PhysX)라는 물리 엔진을 탑재하고 있다. 게임 오브젝트에 리지드바디(Rigidbody) 컴포넌트를 붙이면, 그 때부터 해당 게임 오브젝트는 물리적인 시뮬레이션을 시작하게 된다. 중력의 영향을 받아 아래로 떨어지고, 다른 게임 오브젝트와 충돌하면 튕겨 나가게 되는 것이다. 이처럼 간단히 컴포넌트 하나를 붙이고, 필요한 속성 값을 조절하는 것만으로 우리는 물리 시뮬레이션 기반의 게임을 쉽게 만들 수 있다.

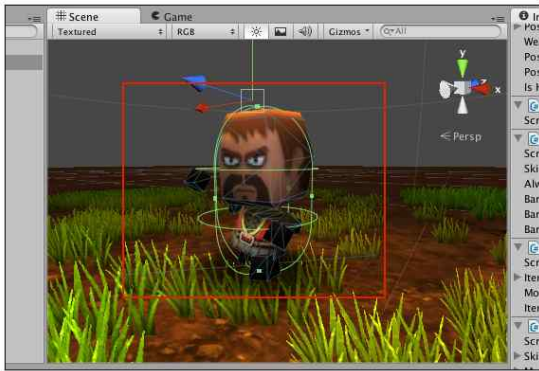


Fig. 4. Collider Component

물리 엔진을 실제 게임 플레이에 활용하기 위해서는 충돌체가 필요하다. 게임을 개발하다 보면 다양한 충돌 상황을 감지하고 그에 맞는 연출을 해야 한다. 유니티 엔진을 사용할 경우, 충돌처리를 위해 별도의 코드를 작성할 필요가 없으며, 엔진에서 미리 제공하는 충돌체 컴포넌트를 그냥 붙이기만 하면 된다.

6. Mecanim Animation

유니티가 버전업 되면서 새롭게 들어간 매카닉(Mecanim) 애니메이션 시스템은, 캐릭터 애니메이션을 쉽게 만들고 공유할 수 있게 함으로써 개발의 생산성을 크게 높인 중요한 기술이다. 과거에는 모델링 작업을 하고, 본 세팅을 한 뒤 해당 모델에만 사용할 수 있는 애니메이션을 일일이 만들어야 했지만 다른 모델링 데이터에서 사용했던 애니메이션을 뽑아내어 어떤 모델에도 적용할 수 있는 획기적인 기능을 제공하고 있다. 매카닉 애니메이션 시스템은 사람의 형태를 갖춘 대상에 대해서만 적용할 수 있다. 이 조건만 충족된다면, 캐릭터에 적용된 본의 개수가 달라도 대부분 애니메이션을 공유할 수 있다는 점이 특징이다. 따라서 다른 사람이 만든 애니메이션을 구입하여 현재 개발 중인 게임의 캐릭터에 연결하여 사용하는 일도 얼마든지 쉽게 할 수 있다. 이렇게 되면 게임 개발에 드는 시간과 비용을 크게 줄일 수 있고, 개발 효율성도 자연스럽게 높아지게 된다.

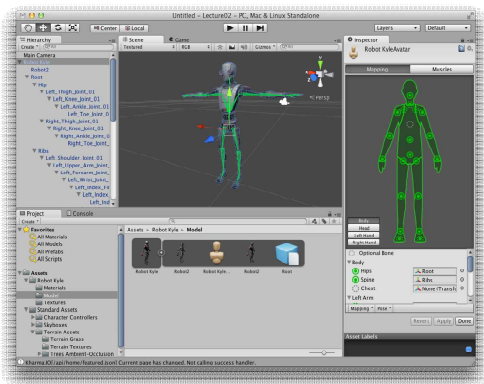


Fig. 5. Mecanim Animation

매카닉 애니메이션 시스템의 또 다른 특징은 블렌딩 기능이다. 서로 다른 애니메이션이 연결되는 과정에서 두 개의 동작을 자연스럽게 섞어 준다. 여기에 레이어 기능까지 추가하여 다양한 애니메이션 상태를 동시에 관리하는 것이 가능하다. 이 기능을 활용하면 게임 내에서 캐릭터의 움직임을 더욱 자연스럽게 표현할 수 있다.

7. Expansiveness using Plug-In

유니티를 이용하여 게임을 개발해 본 결과, 가장 편하다고 느꼈던 부분은 확장성이 뛰어나다는 점이다. 유니티에서 제공하지 않는 기능을 일반 개발자들이 만들어서 공개하거나 유료로 판매하고 있기 때문이다.



Fig. 6. Expansiveness using Plug-In

이렇게 많은 개발자들이 다양한 기능을 계속 추가하고 있기 때문에, 게임을 만들기 위해 필요한 많은 기술을 직접 개발하지 않아도 손쉽게 활용할 수 있다. 현재 유니티 개발자 커뮤니티에서 가장 인기 있는 플러그인은 NGUI, Play Maker, 2D Toolkit을 꼽을 것이다. 이 세 가지 플러그인은 유니티 엔진의 불편함을 해소하고, 그 활용 범위를 크게 넓혔다는 평가를 받고 있다. NGUI 플러그인을 사용하면 다양한 인터페이스 구성 요소들을 마우스 클릭만으로 쉽게 만들고 배치할 수 있다. Play Maker는 상태 머신을 만들고 관리할 수 있도록 하는 플러그인이다. 이 플러그인을 사용하면 유니티 엔진에서 만든 모든 게임 오브젝트에 고유의 유한 상태 머신을 덧붙일 수 있다.

2D Toolkit은 유니티 엔진으로 2D 게임을 쉽게 만들 수 있게 도와주는 플러그인이다. 유니티 엔진은 본래 3D 게임을 만들기 위해 개발된 것이기 때문에 스프라이트를 생성하고 관리하는 기능이 기본 제공되지는 않았지만 2D Toolkit의 등장으로, 스프라이트를 만들고 관리하는 작업이 간단해졌다.

8. Terrain Editor

유니티 엔진은 지형(Terrain)을 생성하고 편집할 수 있는 기능을 제공하고 있다. 지형(Terrain)은 특수한 형태의 게임 오브젝트인데, 크기를 비꾸고, 모양을 변형하는 일이 무척 쉽기 때문에 게임 프로그래밍을 모르는 기획자들도 금방 배워서 사용할 수 있다.

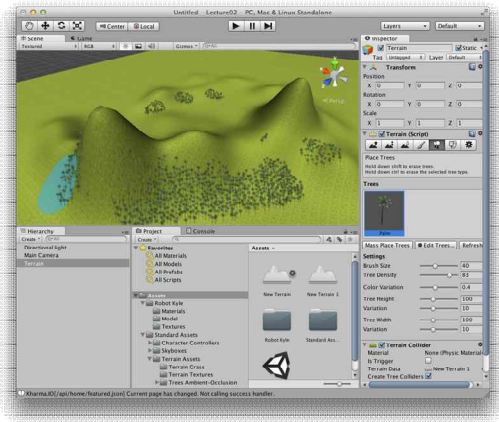


Fig. 7. Terrain Editor

유니티에서 만든 지형은 텍스처로 뽑아낸 뒤, 포토샵과 같은 그래픽 프로그램을 통해 더 세심하게 수정할 수 있다. 이러한 하이트맵 (heightmap) 개념에 익숙해지면, 복잡한 지형을 만들 때 걸리는 시간을 크게 줄일 수 있다. 전체적인 지형의 골격을 그래픽 에디터에서 편집한 뒤, 이를 텍스처로 뽑은 뒤 유니티에서 불러오면 되기 때문이다. 이처럼 성능 좋은 지형 에디터가 내장되어 있다는 점은 유니티의 활용성을 높여준다. RPG 나 전략 게임을 만들 때 특히 편리하다. 이 경우, 기획자(레벨 디자이너)와 프로그래머가 역할을 나눠서 작업하고 같은 장면을 보면서 수정할 수도 있다. 따라서 협업이라는 측면에서도 이 기능은 상당히 유용하다고 하겠다.

9. Multi-platform Compatible

유니티 3D를 사용해 본 결과, 이 엔진의 최고 강점은 멀티 플랫폼 대응 능력이 현존하는 어떤 엔진보다도 뛰어나다는 데 있음을 깨달을 수 있었다. 유니티로 개발한 게임은 별다른 추가 작업 없이도 PC와 맥, 리눅스, 웹브라우저, iOS, 안드로이드 용으로 빌드할 수 있다.

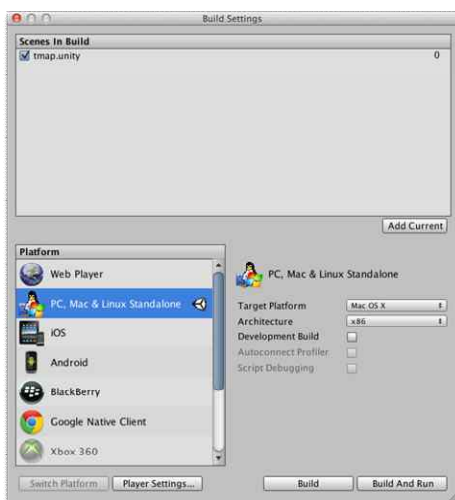


Fig. 8. Multi-platform Compatible

지난 해 말부터 한국 게임업체들은 카카오톡을 통해 게임을 출시하기 위해 많은 노력을 기울이고 있는데, 카카오톡은 안드로이드와 아이폰용 게임 동시 출시를 권장(공식적으로 의무화 했다)가 문제가 되어 표면적으로는 취소했음)하고 있다. 이 때문에 카카오톡에 들어가기를 원하는 많은 게임 개발자들이 현재 유니티 엔진으로 게임을 개발하고 있는 실정이다. 멀티 플랫폼 대응 능력에 관한 한 현 단계에서는 유니티를 선택하는 것이 최선이라고 판단된다.

III. Conclusions

게임 시장의 중심이 모바일 게임으로 이동하면서, 적은 수의 개발자들이 모여 만든 게임도 성공할 수 있다는 것이 증명되고 있다. 이로 인해 많은 개발자들이 기성 회사에 들어가기보다는 독립해서 스타트업 회사를 차리는 길을 택하고 있다. 문제는 이런 회사들 중 상당수가 프로그래머를 구하지 못하고 있다는 점이다. 운 좋게 시작 단계에서 프로그래머를 합류시킨 회사도, 게임 프로젝트 진행 도중에 프로그래머가 퇴사해 버리는 일을 자주 당한다. 이러한 업계 상황을 볼 때, 대학에서 유니티 엔진 프로그래머를 양성하는 것은 아주 시급적 절하다고 보인다. 유니티 엔진 프로그래머에 대한 수요는 과거 어느 때보다 높고, 이를 충족시킬 수 있는 기성 개발자들의 수는 부족한 상황이기 때문이다. 따라서 재학 중 충분한 유니티 개발 경험을 쌓도록 지도한다면, 졸업 후 취업 가능성도 과거에 비해 훨씬 높아질 것이다. 이 때문에 우리 대학에서도 유니티 엔진 교육에 좀 더 많은 관심을 갖고 학생들을 지도해야 할 것이라고 생각한다.

References

- [1] www.unitydev.co.kr
- [2] <http://www.unity3dstudy.com/>
- [3] <http://cafe.naver.com/unityhub>
- [4] <http://www.unitylearning.co.kr/>