



## 레벨 디자인의 날: 질문과 답변 (Level Design in a Day: Your Questions, Answered)

작성자: 코레이 세이퍼트(Coray Seifert)

작성일: 2013년 3월 19일

지난 5년 동안 레벨 디자인의 날(Level Design in a Day) 멤버들은 레벨 디자인의 모든 것에 대해 논하기 위해 [게임 개발자 컨퍼런스\(Game Developers Conference\)](#)<sup>1</sup>의 신성한 홀에 모였다. 올해는 가마수트라에서 게임 개발 커뮤니티와 자유롭게 질문과 답변을 주고 받음으로써 평소 세션에 참가하는 백여 명보다 훨씬 많은 사람들과 소통할 수 있는 훌륭한 기회를 제공해주었다.

그리하여 우리는 3월 26일 화요일에 종일 진행된 올해 [AAA Level Design in a Day Bootcamp](#)<sup>2</sup> 참가자 명단에서 존경받는 레벨 디자인 전문가들을 골라내어 패널을 구성했다. 그들은 다음과 같은 토론 형태로 가마수트라 커뮤니티의 질문에 답변하기로 했다.

**닐 알폰소(Neil Alphonso):** 스플래시 대미지(Splash Damage) 수석 디자이너

**짐 브라운(Jim Brown):** 에픽 게임스(Epic Games) 수석 레벨 디자이너

**조엘 버지스(Joel Burgess):** 베데스다 게임 스튜디오(Bethesda Game Studios) 선임 디자이너

**스티브 게이너(Steve Gaynor):** 풀브라이트 컴퍼니(The Fullbright Company) 공동 창립자

---

<sup>1</sup> 참조링크: <http://www.gdconf.com/>

<sup>2</sup> 참조링크: <http://schedule2013.gdconf.com/session-id/822397>

**세스 마리넬로(Seth Marinello):** EA / 비세랄 게임스(Visceral Games) 레벨 디자이너

**(편집) 코레이 세이퍼트:** 슬링고(Slingo) 부사장, 상품 개발

질문의 대다수는 특히 레벨 디자인의 기술에 초점을 맞추고 있는데, 그 중에는 제작에 대한 폭넓은 관심사와 툴 개발, 엔진의 한계에 대한 탐구와 문제 상황들도 있었다.

나는 항상 레벨 디자인이 실제로 적용된 게임 디자인이라고 말해왔다. 레벨 디자인은 매우 전문화된 기능인 동시에 기술과 메카닉, 대부분은 무형인 재미의 교차점이다. 따라서 독자가 어떤 게임 개발 규칙을 추구하건 간에, 여기서 제공하는 게임 개발자 컨퍼런스 튜토리얼에서 몇 가지 훌륭한 교훈을 찾을 수 있을 것이다.

*첫 번째 질문은 블룸필드 컬리지(Bloomfield College)에서 게임 디자인을 공부하는 로저 로사(Roger Rosa)의 질문이다.*

**1. 레벨의 첫 단계가 완성된 후 레벨 디자이너들이 발견하는 일반적인 실수들이나 중요한 것들은 무엇인가? 레벨 디자인에 나타나는 간과하기 쉬운 결함들은 무엇인가?**

**짐 브라운:** 내가 주로 경계하는 두 가지는 부주의와 레벨 디자인에 대한 형편없는 전제들이다. 스크립트 작업, 커버, 충돌, 그리고 일반적인 레벨 디자인에서 발견되는 버그의 대부분은 누군가가 자만에 빠지거나 디자인의 '지루한' 부분들을 밀어붙이기 때문에 발생한다. 만약 상세한 것들에 주의를 기울이고 레벨의 모든 측면들을 중요하게 다룬다면 디자이너는 결국 크게 (더 빠르고 분명하고 쉽게) 발전할 것이다.

둘째, 레벨 디자이너들은 플레이어가 그 레벨을 만든 본인과 동일한 방식으로 진행해갈 것이라고 전제하고 레벨을 만들 때가 많다. 본인이 그 레벨을 500 번 플레이했다고 해서 엔드 유저가 그렇게 하리라고 볼 수는 없다. 엔드유저는 잘못된 순간에 뒤로 돌아설 테고, 방아쇠를 순서에 어긋나게 당겨서 일을 그르치고, 상상할 수 있는 모든 방식으로 레벨을 망칠 것이기 때문이다. 첫 단계의 지도에 있는 그야말로 '황금길'은 시스템이 압박을 받으면 급격히 무너지는 경향이 있다. 그것

외에도 우리는 종종 '일이 되게 만들어서' 첫 단계의 지도가 가도록 해야 하고, 그 다음에는 실행과 메모리, 페이스 조절, 어려움에서 수많은 최적화 작업을 해야 한다.

**스티브 게이너:** 나에게 첫 단계는 레이아웃과 흐름이고, 두 번째 단계는 조명과 가시성이다. 공간들의 모양과 크기, 연관성을 아는 것은 좋은 첫걸음이다. 그러나 이렇게 하자마자 가능한 한 빨리 디자이너는 플레이어가 하는 것처럼 레벨을 플레이하고, '내가 이 공간에 들어올 때 어디로 가야 할지 어떻게 알 것인가? 적이 어디에서 나타날지 어떻게 알 것인가? 만약 내가 길을 잃는다면 나 자신을 어떻게 바로잡을 것인가?'에 대해 생각해야 한다.

이 사안들의 두 가지 가장 큰 측면은 시선과 조명이다. 디자이너는 각 레벨에서 플레이어가 각각의 지점에서 볼 수 있는 것과 볼 수 없는 것을 결정해야 한다. 예를 들어 플레이어가 어떤 공간에 들어가 멀리 있는 벽에 두 개의 문을 볼 수 있다면, 무엇이 더 중요한가? 플레이어가 첫 번째 문으로 들어가야 하는가? 아마도 시선을 방해하는 것이 있어서 플레이어가 처음에 하나의 문만 보고, 첫 번째 문에 도착할 때까지 두 번째 문은 보지 못한다면 항상 첫 번째 문으로 들어갈 수도 있다. 내가 입구와 출구, 중요한 사물들을 볼 수 있는가?

조명이 너무 단조로우면 우선 순위가 없어진다. 중요한 것들을 강조하기 위해 명암을 부여해서 플레이어가 한눈에 지세를 확인할 수 있도록 할 수 있는 방법들 들여다보라. 일단 흐름을 배치하고 공간들의 주요 콘셉트들에 대한 플레이어의 시각적인 이해가 어떻게 될 것인지에 대해 좋은 아이디어가 생기면, 디자이너는 각각의 공간에 부수적인 것들을 설치하는 작은 실무적인 측면들로 넘어갈 수 있다.

**세스 마리넬로:** 일단 내가 레벨의 화이트박스 레이아웃을 완성하고 나서 가장 먼저 하는 일들 중 하나는 우리의 가시성 전략을 계획하기 위해 공간의 크기와 시선을 재검토하는 것이다. 우리가 <데드 스페이스(Dead Space)>를 위해 만들어진 환경들은 디테일이 매우 뛰어나기 때문에, 초기 단계에서 공간이 실행되기 위해 어떻게 분할될지를 관리하는 것이 매우 중요하다. 발생할 수 있는 최악의 일 중 하나는 그래픽 처리 장치(GPU)에 너무 복잡한 공간을 강제로 구현하려 몇 달 동안 노력한 후에 공간을 반으로 자르게 되는 것이다.

간과하게 되는 문제에 대해서는, 개발 초기에는 페이스 조절을 읽어내기 어려울 수 있다. 대화와 각본이 짜인 순간들이 없다면 레벨이 공허하게 느껴질 수 있기에 그

대응으로 전투를 추가하게 된다. 그 결과, 온라인에서 콘텐츠의 나머지가 추가되고 나면 페이스 조절의 문제들을 초래하게 된다. 이 점을 인식하고 이 사안들을 처리하기 위한 시간을 일정에 넣는 것이 중요하다.

두 번째 질문은 트위터리안 [@Skizomeuh](https://twitter.com/Skizomeuh)<sup>3</sup>의 질문이다.

**2. 최근 (레벨 디자인 측면에서) 허브 위주의 게임들(hub-oriented games)이 적은 이유는 무엇인가? 나는 <메트로이드 프라임(Metroid Prime)>이나 <헥센(Hexen)> 같은 게임을 염두에 두고 있다.**

**닐 알폰소:** 간단한 답변은 허브 기반의 레벨 디자인이 오픈 월드(open world) 모델에 의해 본질적으로 잠식당했다는 것이다. 스트리밍 기술의 발전과 향상된 아트 제작 루트는 '레벨 디자인'에 '레벨'을 처음 집어넣었던 많은 제한들이 더 매끄러운 경험들을 허용하면서 사라져버렸다는 것을 의미했다. 완벽한 사례는 락스테디(Rocksteady)의 허브 기반 모델인 <아캄 어사일럼(Arkham Asylum)>에서 스트리밍과 오픈 월드 모델인 <아캄 씨티(Arkham City)>로의 발전이다. 허브 기반 레벨 디자인의 많은 원리들은 여전히 적용되고 있지만, 그것으로 돌아갈 정도는 아니다.

---

<sup>3</sup> 참조링크: <https://twitter.com/Skizomeuh>



**스티브 게이너:** 이 질문에 대한 답변은 모든 축에서 다양하게 찾을 수 있다. 즉 (시선 거리와 레벨 스트리밍 기술 등에 기초한) 더 열리고 흐름이 자유로운 공간을 허용하는 게 기술적으로 어려울 수 있다. 디자인 관점에서 볼 때 변수들도 많은데, 디자이너가 '만약 플레이어가 북쪽이 아니라 동쪽에서 이 공간으로 들어온다면 어떻게 될 것인가? 만약 플레이어가 다음 구역을 클리어한 후 돌아온다면 무슨 일이 벌어질 것인가? 공간이 복도가 아니라 열린 중심일 때 어떻게 플레이어를 다음 목표로 이끌 것인가?' 등을 고려해야 하기 때문이다.

허브 기반 레벨 디자인의 장점은 분명하다. 바로 훨씬 더 플레이어 지향적인 탐험, 더 '사실적인 느낌의' 세계, 플레이어가 어떤 공간으로 다시 올 때 더 많은 공간적인 배경을 설치하는 대신 그곳의 상태를 바꿀 수 있을 때까지 콘텐츠를 재사용하는 것의 이점들이 그것이다. 그러나 그것을 이끌어내기 위해서는 몇 가지 특정한 투자가 필요하다.

**조엘 버지스:** 주된 원인은 허브를 잘 이끌어내기가 어렵기 때문이 아닐까 생각한다. 허브에 다시 가는 것은 빨리 식상해질 수 있다. 디자이너는 상태의 변화들을 구현하는 데 엄청난 시간을 들이거나, 허브가 발전하고 플레이어의 행동에 반응하도록 할 수도 있다. 그 작업이 디자이너가 동일한 레이아웃을 재사용함으로써 절감한 것 이상일 수도 있다. 우리는 이것을 <스카이림(Skyrim)>의

어둠의 형제단 성소(Dark Brotherhood Sanctuary)와 <폴아웃 3(Fallout 3)>의 마더십제타(Mothership Zeta)에서 실행했다.

내 생각에 알려지지 않은 현세대 콘솔의 허브 기반 게임은 <스플린터 셀: 더블 에이전트(Splinter Cell: Double Agent)>이다. 이 게임은 또한 NPC 캐릭터를 위한 설득력 있는 공간이 되는 문제와 게임플레이에 어울리는 허브를 만드는 곤란한 문제를 해결했다.

다음은 일본 도쿄에서 온 [@DCharlieJP](https://twitter.com/DCharlieJP)<sup>4</sup>의 질문이다.

### 3. 레벨 디자이너들은 게임 엔진의 한계에 대해 얼마나 알고 있는가? 이것이 어떻게 디자인 과정에 요소로 작동하거나 소통되는가?

**스티브 게이너:** 잘 알고 있다. 게임 엔진의 기술적인 제약들은 레벨 디자이너로서 할 수 있는 모든 것을 규정한다.

그것이 어떻게 요소로 작동하는지는 대부분 기술이 어떤 상태에 있는지에 달려 있다. 즉 만약 디자이너가 안정적으로 확립된 엔진으로 작업하고 있다면 시작 단계부터 엔진의 한계가 훨씬 더 분명하고 통제가 잘될 것이다. 만약 기술이 게임 디자인 과정 중에도 여전히 개선되는 중이라면 프로그래밍과 디자인 사이의 대화가 보다 유동적이겠지만, 디자이너가 무슨 제한이 있는지 정확하게 알지 못한다면 그들의 대화는 더 불확실하고 실망스러울 수 있다.

그러나 어떤 수준에서는, 이 경우 게임 엔진이 안정될 때 어떤 제약들이 있는지를 알아내기 위해, 기술의 한계를 극복하는 데 도움을 주고 디자이너가 바라는 것은 무엇인지뿐만 아니라 무엇이 가능한지를 밝히는 데 도움을 주는 것 역시 레벨 디자이너로서의 업무의 일부이다.

**짐 브라운:** 만약 레벨 디자이너들이 자신이 가진 게임 엔진의 능력에 대해 충분히 알지 못한다면 프로젝트가 매우 불리하게 된다. 모든 프로젝트 종반의 마지막 세련화 작업은 항상 가장 어려운 일이지만(항상 예상되기도 한다) 범위에서 완전히

---

<sup>4</sup> 참조링크: <https://twitter.com/DCharlieJP>

벗어난 어떤 것을 만드는 (또는 주요 재디자인을 불러오는) 이해 부족은 용납될 수 없으며 소비적이다. 이것은 예산과 일정, 경력을 망친다.

디자이너는 자신의 팀과 게임 엔진이 생산할 수 있는 틀 안에서 디자인해야 하며, 프로토타입을 만들 때에도 그 목표들을 염두에 두고 있어야 한다. 그리고 물론 팀 전체가 프로젝트 목표를 동의하고 있다는 것을 확실히 해야 한다.

예를 들어 <기어스(Gears)>에서 우리는 언리얼 엔진 3(Unreal Engine 3) 로 디자인을 막 시작하고 있었는데, 그래서 우리가 향상된 셰이더(shader)와 하이폴리(high-poly) 캐릭터들을 원했다는 것을 처음부터 알았다. 우리는 그런 엔진 특성들을 강조하기 위해 3 인칭 시점의 카메라와 '근접' 전투 거리에 대해 하나의 그룹으로서 동의했다. 그것은 확실히 디자인에 전반적으로 영향을 주었다. 그리고 화면상 적들의 숫자와 건물 크기에 영향을 주었기 때문에 고려되어야 했으며, 그것은 폭넓은 방식으로 스크립트 작업에 직면한다.

**닐 알폰소:** 레벨 디자이너들은 자신이 사용하는 게임 엔진의 내부 작동에 가능한 한 친숙해져야 하는데, 기술적인 변화의 속도는 그렇게 하기 어렵게 만든다. 결국 이것은 팀 전체가 나누어 져야 하는 책임이다. 즉 기술 수석들은 레벨과 자산 창조를 위한 지침을 제공해야 하고, 레벨 디자이너들은 이것을 게임의 환경적인 비주얼의 충실성 목표와 조화할 수 있는 레이아웃을 제공해야 하며, 아티스트들은 그 안에서 가능한 한 훌륭한 품질을 추구하고 프레임레이트 목표들을 달성해야 한다.

자동화된 과정으로 문제가 있는 영역을 바꾸기 어려워지기 전에 확인할 수 있게 되면서, 틀은 어쨌든 오늘날 개발에서 이것을 어느 정도 쉽게 만들었다. 기계적으로 위험한 모든 것은 제작되기 전에 프로토타입에서 확인되어야 하는데, 만약 그 위험이 게임에서 널리 쓰이지 않는다면 그렇게 작은 부분에 쓰이는 개발 자원의 변화들이 현저하게 줄어들기 때문이다.

*다음은 게임 산업의 베테랑이자 카밤(Kabam)의 총감독 마이크 셀러즈(Mike Sellers)의 질문이다.*

**4. 통과 측정 기준에 대해:** 플레이어가 그 레벨을 얼마나 좋아하는지, 재미를 잃거나 싫증을 느끼지는 않는지 등은 어떻게 아는가? 이것을 알기 위한 몇 가지 좋은 해결책들이 있지만 많은 사람들에게 (전문가들에게조차) 알려져 있지 않다.

**조엘 버지스:** 가능한 한 일찍 그리고 자주, 사람들을 그 레벨 앞에 앉혀놓고 플레이하는 것을 지켜보라. 레벨이 다듬어지거나 출시자와 프로듀서, 그 밖의 누군가가 게임테스트 세션을 정리해주기를 기다리지 말라. 누군가를 붙잡아 가능한 한 적은 구성이나 안내를 제공하라. 플레이하는 동안 그들이 목소리를 내게 하라. 그리고 나서는 조용히 하라. 방해도 하지 말고, 도와주지도 말고, 바로잡아 주지도 말라. 꼭 필요한 것이 아니라면 직접적인 질문은 무시하라.

플레이어에게서 얻은 여과되지 않은 피드백은 언제나 구원의 빛이 될 것이며, 이미 참여하고 있는 내부 논쟁에서 이길 수 있도록 디자이너를 도와줄 때가 많다.

**짐 브라운:** 가장 간단한 방법은 지켜보고 주의를 기울이는 것이다. 명백하지만 가장 많이 간과되는 방법이다. 디자이너들이 자신의 작업에 지나친 애착을 갖는 것은 드문 일이 아니다. 만약 지나치게 작업에 참여하거나 투자하면 디자이너는 큰 그림을 놓치는 경향이 있다. 나는 몇 년 전 레벨 디자인의 날 발표에서 디자이너의 일이란 '플레이어의 대변자가 되는 것'이라고 말했다. 즉, 디자이너는 자신이 좋아하는 것을 만들거나 자신의 게임을 경험하는 데 있어 플레이어의 역할이 무엇인지 놓쳐서는 안 된다. 디자이너는 자신이 아니라 플레이어를 위한 레벨을 만든다.

한편 유용성 테스트, 포커스 그룹, 히트 맵(heat maps), 스탯 트래킹(stat tracking), 그 밖의 몇 가지 분석 툴은 믿을 수 없을 만큼 유용하며, 언제든지 이용해야 한다. 몇 가지 기본적인 심리학에 대한 연구에 시간을 들이는 것도 충분한 가치가 있다. 인간의 두뇌는 제멋대로여서 항상 인간이 가정한 방식대로 작동하지 않는다. 자신이 만든 레벨을 다른 사람들이 플레이하는 것을 지켜보고 열린 마음을 가져라. 평범한 게이머의 머릿속에 들어가보면 더 좋은 디자이너가 될 수 있을 것이다.

**닐 알폰소:** 비용이 가장 적게 들어가는 방법은 그저 누군가가 플레이하는 것을 지켜보고 부지런히 기록하는 것이다. 많은 스튜디오들은 이런 테스트에서 좀 더 유용한 정보를 캐내기 위해 지금도 생물학적인 자료를 사용한다. 폭넓은 측정 기준을 찾아내기 위한 툴은 지난 몇 년 동안 현저하게 향상되었고, 엄청난 숫자의



사용자 샘플이 주어졌을 때 훨씬 더 분석적인 정보를 제공할 수 있다. 밸브(Valve)가 <하프라이프 2(Half-Life 2)>와 <팀 포트리스 2(Team Fortress 2)>에 가한 스팀(Steam)의 측정 기준에 기초한 변화들은 충분히 실행할 수 있는 정보가 있다면 실망스러운 지점들(또는 때려치우고 싶은 순간)이 현저하게 줄어들 수 있다는 것을 보여주었다.

다음은 [@Jeremy\\_LaMont<sup>5</sup>](#)의 질문이다.

##### 5. 플레이어가 '이리저리 돌아다니는' 대신 주의를 집중하거나 이곳으로 가기를 원할 때 디자이너는 어떻게 자신의 접근법을 바꿀 것인가?

**스티브 게이너:** 나는 확실히 '이리저리 돌아다니는' 시간을 가능한 한 많이 허용하는 편이다. 그러나 디자이너가 (튜토리얼 작업이나 다른 그 어떤 것을 위해) 정말로 플레이어를 구체적인 한 지점으로 이끌어야 한다면 중심을 만드는 것이 전부다.

이는 내가 올해 게임 개발자 컨퍼런스의 레벨 디자인의 날 튜토리얼에 실릴 '레벨 디자인에서 스토리텔링을 위한 내러티브 기술'에서 충분히 논한 내용이다. 몇 가지 훌륭한 방법들이 있다. 중요한 대상들을 강조하기 위해 스포트라이트와 실루엣을 사용하고, 불필요한 모든 상호적인 대상들을 배경에서 제거하고, 플레이어가 한 장면의 중요한 부분으로 걸어가도록 경로를 정리하는 것 등이 그것이다.

기본적으로 디자이너는 중요한 것이 전면에 부각되고 분명하게 보이기를 원하고, 그래서 플레이어는 디자이너에 의해 '강요받는' 게 아니라 기꺼이 참여할 것이다.

**조엘 버지스:** 적절한 음악을 선택하고, 플레이어를 집중하게 하는 요소들을 레이아웃에 포함시키거나 플레이어의 정신을 산만하게 하는 요소들을 최소화하는 등, 긴급한 신호들과 섬세하게 소통할 수 있는 방법들은 많다. 때로는 그것으로 충분하지 않다.

---

<sup>5</sup> 참조링크: [https://twitter.com/Jeremy\\_LaMont](https://twitter.com/Jeremy_LaMont)

레벨 디자이너라면 누구나 스크린 몇 피트 옆에 훌륭한 스크립트 장면이 있는데도 플레이어가 단순 조명을 탐색하는 것을 지켜보는 불편함을 알 거이다.

이런 상황에서 처음으로 해야 하는 일은 어쨌든 무엇이든 실제로 *해야* 하는지를 결정하는 것이다. 타이머, UI 프롬프트, 컷씬(cutscenes)과 그 밖의 장치들이 주의를 끄는 것을 도와줄 수 있지만, 가이드가 필요한 플레이어와 단순히 좀 더 멋진 조명기구에 신경을 쓰는 플레이어 사이의 차이를 알아야 한다. 플레이어로서 재미를 잃고 혼란에 빠지는 것은 짜증스러울 수 있지만, 엄격한 레벨 디자인은 언제나 짜증 나는 일이다.

**닐 알폰소:** 이와 관련해 나의 주된 도구는 밀도인데, 밀도는 많은 형태를 취할 수 있다. 그것은 대상들의 밀도일 수도 있고, 운동의 밀도 또는 상호작용의 밀도일 수도 있으며, 이름만 붙이면 된다. 나는 플레이어에게 그들이 '중요한' 위치에 있다는 사실을 섬세하게 말해줄 좋은 방법을 찾아냈다. 그러나 이 방법은 플레이어를 이끄는 명백하게 간접적인 방법을 유지하기 위해 쓰인다. 즉 그것이 주어진 레벨에서 어떻게 다루어지는지보다, 플레이어를 이끄는 일에서 디자이너가 얼마나 엄격해질 수 있는지가 게임이나 전체 게임의 창조적인 방향으로 향할 수 있게 한다. 이것이 본질적인 사건들이 종종 시네마틱스(cinematics)중이나 UI 와 함께 나타나는 이유다.

만약 디자이너가 자신이 결정적인 정보로 간주하는 것을 놓칠 기회를 플레이어에게 준다면, 문제는 많은 플레이어들이 그것을 정말로 놓치게 된다는 것이다. 이는 플레이어가 부주의해서가 아니라, 플레이어가 게임을 플레이하는 동안 어떤 산만함을 가질 수도 있는지 디자이너가 전혀 알지 못하기 때문이다.

*다음은 풀 세일(Full Sail) 대학 졸업생 [@MrDonaldYoung](https://twitter.com/MrDonaldYoung)<sup>6</sup>의 질문이다.*

**6. 플레이어가 황금길(golden path)을 벗어나게 하기 위한 상황들은 얼마나 자주 만들어야 하는가? 그리고 그 상황들을 만들기 위해 추가적인 자원을 투자할 가치가 있는가?**

---

<sup>6</sup> 참조링크: <https://twitter.com/MrDonaldYoung>

**스티브 게이너:** 물론 가치 있다. 게임의 핵심은 상호작용이고, 상호작용이란 두 명의 플레이어가 정확히 동일한 플레이 경험을 하게 되리라는 것을 의미하지 않는다. 디자이너가 자신의 게임에서 두 플레이어의 경험 사이에 변화를 더 많이 추가할수록, 그들이 느끼게 될 개인적인 유대에 더 많은 변화가 생긴다. " 이곳으로 가기로 했고, 이 또 다른 공간을 탐험하기로 했다. 나는 다른 사람이 찾아내지 못한 것을 찾아냈다."

가능한 한 위험한 경로는 줄이고 선택적인 공간은 늘이면서, 깊이 파고들어 생각해 보며 스스로 차지해야 할 것들을 플레이어에게 훨씬 더 많이 제공하라. 만약 '플레이어가 한 번도 본 적 없을 콘텐츠를 우리가 얼마나 많이 만들고 있는지 보라!'라는 관점에서 위험하지 않은 경로 공간의 생산 비용에 대해 생각해 본다면, 디자이너는 스스로 모든 것을 의무적인 것으로 만들고 모든 플레이어의 경험을 동일한 것으로 만드는 일에 대해 쉽게 말할 수 있고, 따라서 누구도 아무것도 '놓치지' 않는다. 그러나 만약 선택적인 공간들이 제공하는 스스로 방향을 정한다는 느낌, 투자하고 탐험하고 발견한다는 느낌에 내재된, 보이지 않는 가치에 대해 생각해본다면, 전반적인 경험은 디자이너가 스프레드시트에서 쉽게 측정할 수 있는 것보다 훨씬 더 향상된다.

**조엘 버지스:** 나는 개인적으로 가능한 한 언제든지 중요하지 않은 콘텐츠를 포함하는 것을 선호한다. 이것은 탐험하는 플레이어들에게 보상을 줄 뿐만 아니라, 게임 세계가 플레이어와 그의 이야기에 인위적으로 초점이 맞추어져 있다는 느낌이 덜 들게 하는 데 도움이 된다. 나에게 다행스러운 점은, 약 90 퍼센트의 베데스다 스튜디오의 게임들이 황금길에서 벗어나 있어서 우리가 중요하지 않은 콘텐츠에 자원을 쓰는 일이 있다는 사실이다. 하지만 그것은 우리의 게임이 주는 느낌의 일부분이다. 상황은 달라질 수도 있다.



**세스 마리넬로:** 기본적인 답변은 가능한 한 자주 만들라는 것이다. 플레이어들이 독특한 경험을 하고 특별한 무엇을 찾았다는 느낌을 얻을 수 있는 기회가 많을수록 게임은 그들에게 더 중요해질 것이다.

<데드 스페이스>의 레벨을 설계하는 동안 나는 '보물 주머니'와 '베타 공간(beta rooms)'이라는 선택적인 콘텐츠를 추가하려 한다. 전자는 단순히 탐험에 대한 보상이다. 예를 들어 만약 내가 긴 복도에 있고 알파 흐름(alpha flow)이 플레이어를 복도 중간까지만 보내준다면, 그 끝에는 수집물이라 해도 재미있는 무언가가 있어야 한다. 공간의 끝까지 가보는 것에 대해 보상해줌으로써 디자이너는 막다른 끝을 발견으로 바꿀 수 있다.

베타 공간은 말 그대로 독특하고 레벨의 알파 경로에서 분리된 공간이다. 이 공간이 세계를 더 건설하게 하고 사람이 살고 있다고 느껴지도록 만들려 하는데, 이것이 내가 SF 적인 레벨에 쿼터와 욕실, 세탁소같이 사람이 사는 공간들을 만들어 넣는 이유다.

다음은 비디오게임 저널리스트이자 번역가 [@andymonza](https://twitter.com/andymonza)<sup>7</sup>의 질문이다.

---

<sup>7</sup> 참조링크: <https://twitter.com/andymonza>

## 7. 리플레이 가치와 (보통 대량의 스크립트 작업에서 비롯한) 영화적인 시퀀스를 동일한 경험에서 공존하게 하는 것이 정말 가능한가?

**짐 브라운:** 물론이다. 그러나 그것은 디자이너들이 자신의 통제권을 약간 포기해야 하며(이게 반드시 나쁜 것은 아니다) 경험을 신선한 것으로 유지하기 위해 신뢰할 수 있는 시스템(유동적인 스크립트 작업, 강력한 인공지능, 영리한 세계 건설 등)을 적재적소에서 사용해야 한다는 것을 의미한다. <기어스 오브 워: 저지먼트(Gears of War: Judgment)>에서는 적들이 무작위로 나타나게 하고 맞닥뜨릴 때마다 나타나는 위치를 바꾸기 위해 스마트 스폰 시스템(Smart Spawn System)을 사용했고, <레프트 4 데드(Left4Dead)>에서는 페이스 조절을 통제하기 위해 디렉터(Director)를 사용했고, <스카이림>에는 세계에서 마주치게 되는 반복을 피하는 일련의 가능성들이 있다. 이것들은 몇 가지 사례에 지나지 않는다. 이 게임들은 다른 게임들만큼 자주는 아니지만, 여전히 영화적인 콘텐츠와 스크립트가 있는 시퀀스를 사용한다.

개인적인 의견으로는, 이것은 플레이어에게 통제권을 돌려주기 때문에 믿을 수 없을 만큼 대단한 것이며, 플레이어가 게임 스토리를 좀 더 독특하게 자신의 것으로 만들 수 있게 해준다. <워킹 데드>에서조차 높은 리플레이 가치와 함께 영화적인 콘텐츠가 많이 쓰이고 있는데, 왜냐하면 그것들이 지켜져야 할 하나의 경로에 플레이어를 묶어놓지 않기 때문이다.

**닐 알폰소:** 시네마틱스를 생략할 수 있다면 가능하다.

주요 이슈는 시네마틱스가 미디어의 수동적인 형태에서 많은 것을 차용하기 때문에 유동적이지 않다는 점이다. 영화를 작동하게 하는 메카닉과 게임을 작동하게 하는 메카닉은 근본적으로 다르고, 영화의 메카닉을 기본 레벨에 적용하려 노력하는 것은 종종 슬픈 결과로 이어진다. 리플레이 가치가 기계적인 깊이와 다양성에서 가장 자주 나오기 때문에, 정말 이상한 결합이 될 수 있다.

그래서 그것이 가능한가? 그렇다. 하지만 전통적인 트리플 A 라는 의미에서 이것은 싸우는 것을 정당화하기 위한 힘든 싸움이다. 그러나 때때로 전통은 깨지기 위해 만들어진다.

**세스 마리넬로:** 영화적인 콘텐츠는 대부분의 내러티브 위주 게임들에서 여전히 중요한 부분이지만, 본질적으로 유동적인 게임플레이 해결책의 개념으로 간주된다. 몇몇 스튜디오들이 매개라는 의미를 시도하고 유지하기 위해 확장하는 영화적인 순간들을 만드는 일에 투자했는데, 이것은 비용이 많이 들고 항상 성공한다는 보장이 없다. 디자이너이자 이야기 작가로서 나는 이 에너지가 좀 더 플레이어 위주로 하는 방식으로 동일한 정보를 전달하기 위한 방법을 찾고, 영화적인 콘텐츠를 게임플레이 순간들 사이의 연결고리로 사용하는 것이 언제 불가능한지를 알아내는 데 집중되는 것이 더 좋다고 생각한다. 가능한 한 언제든지 우리는 스크립트가 있는 순간들이 주인공의 통제에서 벗어난 어떤 사건에 반응하게 하려고 노력한다. 그런 방식으로 우리는 플레이어의 노력 없이 결정을 내리는 캐릭터를 만들지 않는다.

*키타로 게임스(Kyttaro Games)의 [@gnomeslair](#)<sup>8</sup>의 질문이다.*

## 8. 매우 다양한 게임플레이 결과를 위해 비슷한 요소들을 어떻게 재사용하는가?

**조엘 버지스:** <스카이림>과 <폴아웃 3> 같은 대작 게임에서는 우리가 모든 요소들을 우리 마음대로 효과적으로 사용(그리고 재사용)할 수 있다는 것이 매우 중요하다. 사실 이것은 게임 개발자 컨퍼런스 레벨 디자인의 날 세션에서 내가 다루게 될 주제의 큰 부분이다.

한 가지 좋은 방법은 요소들이 어떻게 사용되어야 하는지에 대한 모든 선입견을 제거하려고 노력하는 것이다. 특정한 배경 유형을 특정한 마주침이나 게임플레이 유형과 강하게 결합하려는 유혹을 물리쳐라. 이 요소들을 더 많이 짜맞출 수 있게 될수록 디자이너가 발견하게 될 잠재적인 다양성은 더 많아진다.

이 기대를 내적으로 가짐으로써 디자이너는 또한 자기 자신과 팀이 다양한 메카닉과 아트 자산 등을 어떻게 구현할 것인지에 대해 더 자주 생각해보도록 격려한다. 이것은 디자이너의 특성 설정이 (원한다면) 전반적으로 더 확고해지고 버그에 강해지리라는 것을 의미한다.

---

<sup>8</sup> 참조링크: <https://twitter.com/gnomeslair>

**세스 마리넬로:** 게임플레이 관점에서 보면, 플레이어가 이해하게 될 패턴들을 만든 다음 그것에 다양한 장식을 입히는 것은 재미있는 경험을 전달하는 데 핵심이다. 디자이너는 플레이어가 마스터할 수 있는 임무들을 만든 다음 그것들이 확장된 도전을 제공하도록 이끌어야 한다. <포털(Portal)>은 이런 디자인의 교과서적인 사례다. 다른 사람들이 언급했듯이 이것은 올해 레벨 디자인의 날 세션에서 더 깊이 다루게 될 주제다.

**짐 브라운:** 이것은 기술이 향상되면서 확실히 쉬워졌다. 더 높은 해상도의 텍스처와 더 좋은 소재들은 우리가 모델들을 너무 비슷해 보이지 않게 하면서 다양한 방식으로 비교하고 바꾸고 재사용할 수 있다는 것을 의미한다. 더 높은 밀도의 모델들과 향상된 표현은 우리가 모델들의 다양한 영역에 더 많은 디테일을 추가한 다음 그것들이 나타나는 방식을 다양화하기 위해 그것들을 다양하게 조명할 수 있다는 것을 의미한다.

그러나 궁극적으로 나는 좋은 결과를 얻을 수 있는 최선의 방법은 실제 건축과 심리학에 대해 이해하는 것이라고 생각한다. 즉 만약 어떤 것이 '실제'인 것처럼 보이거나 '평범해' 보인다면 사람들은 무의식적으로 그것을 생각보다 훨씬 잘 받아들일 것이다. 자연에서 일어나는 특정한 양(그리고 스타일)의 반복이 있고, 두뇌가 너무 많은 여과를 거치지 않고 받아들이게 될 모양과 구조에는 일반적인 모습이 있다. 레벨 디자이너에게 레벨을 제한된 자원과 결합하는 것은 재미있는 퍼즐, 또는 그 자체로 게임이 된다.

*다음은 프로게이머이자 게임 제작자 칼 샤프(Kal Shah)의 질문이다.*

## **9. 레벨 디자이너의 일을 쉽게 해주고 작업 과정을 전반적으로 향상하기 위해 제작자는 무엇을 할 수 있는가?**

**스티브 게이너:** 제작의 가장 큰 이점은 아무도 자신이 지금 당장 할 수 있는 가장 가치 있는 작업을 하는 데 방해받지 않는다는 것을 보장한다는 점이다. 디자이너를 방해하는 것들은 다음과 같다. 디자이너가 게임플레이에 집어넣어야 하는 공간을 만들지 않는 것, 디자이너의 레벨이 기반을 두게 될 아트 자산을 갖고 있지 않는 것, 디자이너의 레벨을 플레이할 수 있는 것으로 만들기 위해 필요한 메카닉을 적재적소에 갖고 있지 않는 것. 그러므로 디자인과 제작, 그 밖의 부서들 사이의

열린 소통이 " 샷건 전투의 첫 단계를 구현해야 하는데 아직 적들이 활성화되지 않았어"라든지 "추락한 헬리콥터를 둘러싼 게임플레이를 만들어야 하는데 헬리콥터 크기를 모르겠어"라는 식으로 이루어질 수 있게 하는 것은 다른 부서들이 자신의 작업에 우선순위를 매기는 것을 도와줄 것이다.

그러나 단순히 다른 사람들에게 작업을 주는 것 외에도, 레벨 디자이너들이 자신을 방해하는 것을 제거하기 위한 방법을 수월하게 하면 제작에 더 유용할 수 있다. 예를 들어 환경 아티스트에게 마야(Maya) 라이선스와 간단한 설명서를 주어서 레벨 디자이너들이 기다리는 동안 임시 메쉬(mesh) 모델을 만들 수 있게 하거나, 스크립트 액션을 프로그래밍해서 레벨 디자이너들이 완성된 코드를 기다리는 대신 스크립트 작업을 통해 새로운 기능의 프로토타입을 만들 수 있게 하는 것이다. 디자이너가 다른 부서들과 잘 소통함과 동시에 좀 더 스스로 만족하게 도와주면 생산성을 향상시키고 방해 요소들을 줄여줄 것이다.

**세스 마리넬로:** 제작자가 레벨 디자인 과정을 지원할 수 있는 방법은 크게 두 가지가 있다. 첫째는 그룹들 사이의 인터페이스에 대한 것이다. 레벨이 화이트 박스(white box)에서 최후로 가면서 많은 콘텐츠가 통합되어야 하고 구성 요소 각각의 진행을 추적하는 것은 시간이 오래 걸릴 수 있다. 만약 그 탐색 작업을 할 수 있고 작업 진행이 주요 자산에 의해 이루어지고 있다는 것을 보장할 수 있는 제작자가 있다면 디자이너는 게임플레이와 퍼포먼스 스크립트 작업을 반복하기가 자유로워진다. 두 번째 방법은 디자인을 위한 외부의 도움이다. 디자인에 너무 몰두해서 레벨이 제공하는 경험이 최종 사용자에게 어떻게 보일지를 간과하기가 쉬운데, 제작자는 디자이너의 작업이 집중 테스트에 실패하기 전에 이슈들을 놓치지 않게 도와줄 수 있다.

**닐 알폰소:** 레벨은 많은 개발 작업의 최종 목적지다. 레벨이 "고무가 길을 떠나는 곳"이라는 말이 자주 쓰인다. 그러므로 제작자가 레벨 디자인 과정을 돕기 위해 할 수 있는 가장 중요한 일은 레벨 디자이너의 작업을 구성하는 요소들을 적시에 전달하는 것을 보장하는 것이다. 또한 가능하면 언제든지 임시적인 상품을 제공하는 것도 중요한데, 왜냐하면 이것이 레벨 디자이너가 발전하는 콘텐츠에 레벨을 더 빨리 적용하는 것을 도와주기 때문이다. 이는 특히 아트 자산에서 교묘해질 수 있는데, 왜냐하면 아티스트들이 '완성되지 않은' 것을 제출하지 않는 것으로 악명 높을 수 있기 때문이다. 파이프라인이 아트 콘텐츠로서 통합의 많은



국면들을 포함한다는 것을 보장하면 결국 잘 플레이되는 레벨을 방해하는 예기치 못한 골치거리들을 매우 효과적으로 줄여줄 수 있다.

*마지막 질문은 유비소프트(Ubisoft)의 레벨 디자이너 마일스 커윈(Myles Kerwin)이 [레벨 디자인의 날 페이스북 그룹](#)<sup>9</sup>을 통해 보내왔다.*

**10. 지난 10 년 동안 레벨 디자인이 어떻게 발전해왔으며, 앞으로 어떻게 변할 것이라고 생각하는지 알고 싶다.**

**짐 브라운:** 전통적인 의미에서의 레벨 디자인은 위험에 빠진 기능이라고 생각한다. 레벨 디자인이라는 개념은 온라인 게임의 출현과 함께 탄생했다. 사람들은 자신이 처음부터 끝까지 작업했던 자급자족적인 레벨을 만들 수 있었다. 우리는 우리만의 텍스처를 만들어냈고, 우리만의 프로그래밍, 스크립트 작업, 디자인, 조명 작업, 경로 만들기 등 모든 것을 해냈다. 더 최근에는 게임 회사들이 조명 전문가, 기술 아티스트, 스크립트 작성자, 게임플레이 디자이너, 유용성 전문가 등 수많은 구체적인 인재들을 구분했다.

그렇게 해서 레벨 디자이너는 퍼즐의 한 조각 역할을 매우 잘하는 미시적인 전문가가 되었다. 나아가 경계가 모호해지면서 '레벨'을 정의 내리기가 점점 더 어려워질 것이다. 현재 관련된 시스템들이 많아져서 디자이너는 그것들 모두가 어떻게 작동하는지 이해해야 한다. 나는 우리가 다방면에 유능한 전문가로 돌아갈 것이며, 실제로 우리 기술들을 레벨, 창조물, 무기, 전투, 비주얼, 스크립트, 퍼포먼스, 유용성 등 단순히 '레벨'이 아니라 '경험'을 제작하는 것과 관련된 모든 일반적인 게임 디자인으로 확장할 것이라고 장담한다.

**세스 마리넬로:** 지난 10 년 동안 레벨 디자인의 작업 흐름에서 발생한 가장 큰 근본적인 변화는 브러시에서 정적인 메시로 이동해오고 있다. <퀘이크 1(Quake 1)>에서 초기 소스(Source) 엔진의 게임에 이르는 동안 레벨 디자이너는 줄곧 환경 아티스트이기도 했다. 그때부터 우리는 많은 작업을 에디터에서 마야 같은 3D 모델링 프로그램들로 옮겼다. 이것은 우리가 만들 수 있는 게임의 비주얼 품질을 크게 향상시켰으며, 동시에 디자인 과정에서 레벨 디자이너의 역할을 철저하게

---

<sup>9</sup> 참조링크: <https://www.facebook.com/groups/308354239212854/>

변화하게 했다. 이제 우리는 제작자일 뿐만 아니라 통합하는 사람이며 레벨의 전망을 지원하는 전체 팀과 협력한다. 나는 지난 몇 년 동안 좀 더 개방적인 환경 덕에 성공한 게임들을 많이 보았다. 앞으로 우리가 레벨 디자인이 스크립트를 따르는 것을 넘어 실험을 가능케 하는 것에 초점을 맞출 수 있기를 바란다.

**조엘 버지스:** 내가 처음 레벨 디자인에 관심을 가지게 되었을 때 레벨 디자인은 혼자서 하는 작업이었다. 초기의 매퍼(mapper)들은 레이아웃에서 조명, 스크립트에 이르기까지 레벨의 모든 측면들을 만들어냈다. 바로 몇 년 후 내가 게임 산업에 입문했을 때 그것은 이미 변하고 있었다. PS2 와 Xbox 같이 당시 새로 등장한 콘솔들은 더 높은 비주얼 성능을 요구했고 개발 툴은 전보다 사용하기 어렵고 복잡해졌다. 레벨 디자인은 종종 더 전문화된 역할을 담당하는 두세 명의 인원이 필요한, 더 광범위한 과정이 되었다.

이것은 폭넓은 전문화에 관심이 없는 사람들에게는 암담한 미래처럼 보일 수도 있다. 나는 스크립트 작업이나 레이아웃에만 초점 맞추는 것을 선호하는 디자이너들을 보았는데, 개인적으로는 게임 개발의 모든 측면들을 다 해보는 것을 좋아한다. 그리고 본질적으로 다른 요소들을 훌륭한 게임플레이로 총동원하는 데 성공한 균형 잡힌 레벨 디자이너들을 역사적으로 발견했다. 전문화는 이런 레벨 디자이너를 양성하는 것에 어느 정도 반하는 것이다.

그럼에도 불구하고 우리가 지금 게임과 레벨 디자인에서 고무적인 정점에 있다고 생각한다. 성능의 상위의 끝이 계속해서 높아지고 있지만, 모든 유형과 규모의 게임들을 위한 공간은 어느 때보다 많아졌다. 이것은 레벨 디자이너들에게 좋은 소식인데, 왜냐하면 디자이너가 가지고 있는 기술과 관심의 독특한 조합이 무엇이든 간에, 그로 인해 완벽한 레벨 디자이너가 되는 게임이 있기 때문이다.