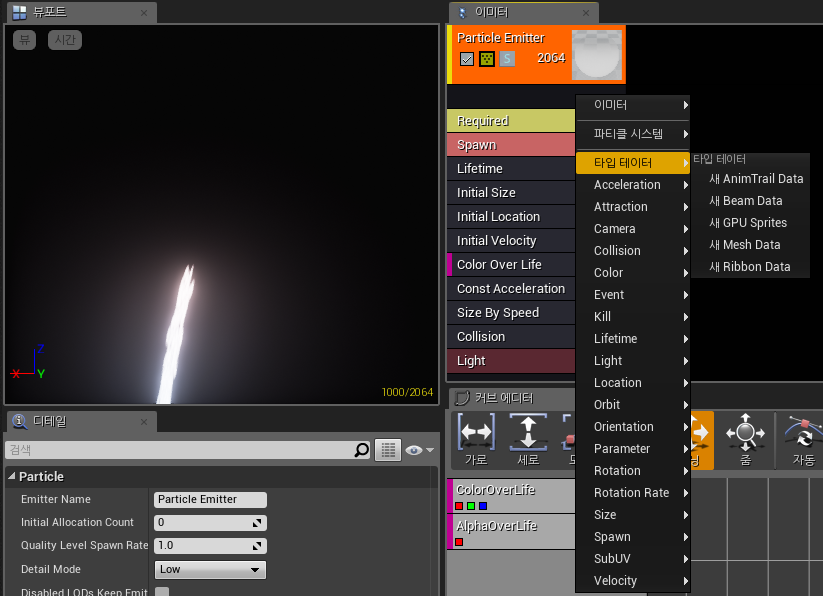
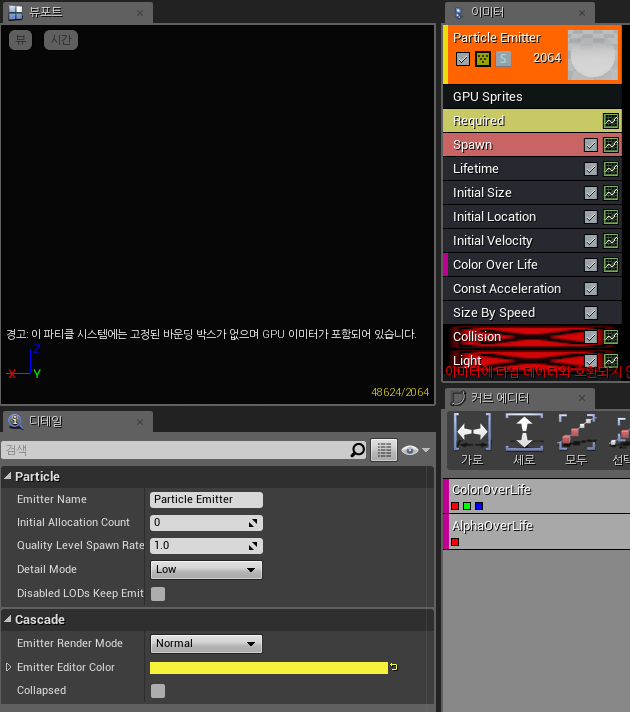
이번 챕터부터는 언리얼에서 제공하는 예제프로젝트를 사용하는 것이 편리합니다.

이번에는 GPU\_Sprite를 만들어 보도록 하겠습니다. GPU\_Sprite는 강력한 연산장치인 그래픽카드를 이용하는 파티클입니다. 작업의 편의를 위하여 저번에 만든 Sprite이미터를 복사하여 사용하도록 하겠습니다.

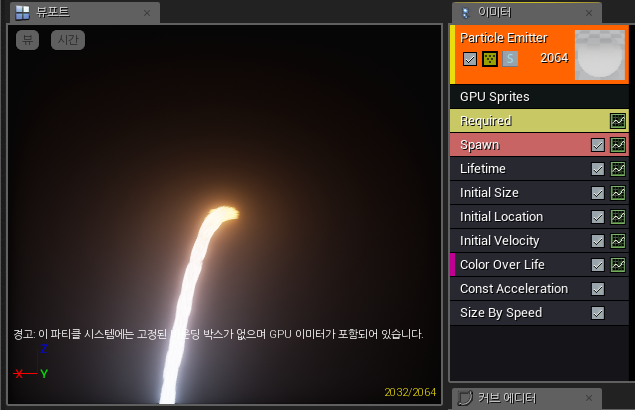
이미터의 빈 부분에 오른쪽 클릭을 하여 메뉴중 타입데터의 새 GPU Sprite를 선택합니다.



선택하면 다음과 같이 화면이 나옵니다.



붉은 박스를 지워주지 않으면 뷰포트에 아무것도 보이지 않습니다. GPU파티클시스템에서는 케스케이드의 모든 모듈이 동작하지 않습니다. 파티클 수의 제한이 사라지는 대신 기능 제한이 생깁니다. 붉은 박스를 지워주도록 합니다.

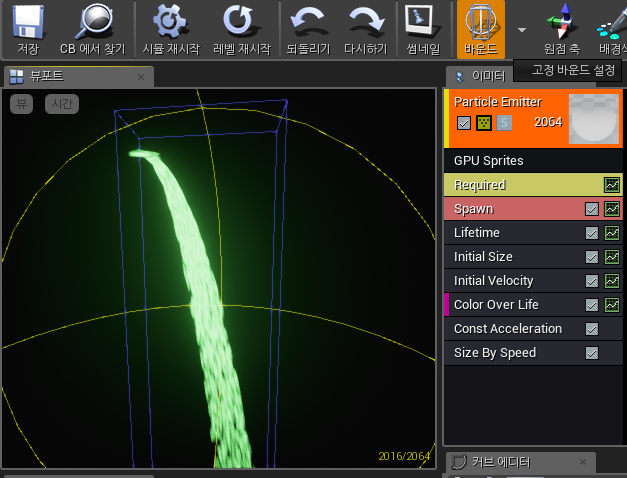


위의 화면과 같이 잘 돌아가는 모습을 볼 수 있습니다.

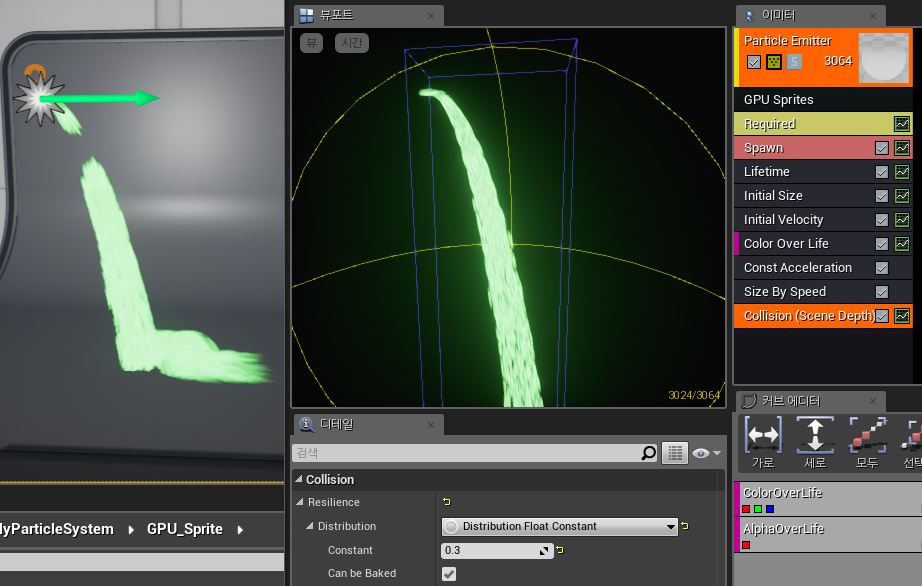
기존의 색이 너무 밝으므로 색을 바꿔보도록 하겠습니다. 이번에는 색을 변하지 않게 해보겠습니다. Distribution에서 Distribution Vector Constant로 바꿔 줍니다. 앞서의 것에서 커브가 빠진 겁니다. 색깔은 밝은 녹색으로 하겠습니다. 아래의 화면과 같이 설정해주시면 됩니다.



한 점에서 파티클이 생성되게 Initial Location을 없애줍니다. 그런데 월드에 파티클을 생성하고 회전시키면서 보면 파티클이 갑자기 안보이는 것을 볼 수 있습니다. 해결방법은 고정범위지정을 해주면 됩니다. 밑의 화면의 고정 바운드 설정을 누르면 해결됩니다. 월드 뷰에서 계속해서 보이지 않는현상이 발생한다면 캐스케이드에서 레벨 재시작을 눌러주면 됩니다. 수동으로 바운드 설정을 하고자 한다면 이미터가 없는 빈공간을 클릭하면 파티클 시스템의 디테일이 나오는데, 스크롤을 내려서 Bounds 의 속성값들을 직접 변경하여 주면 됩니다.



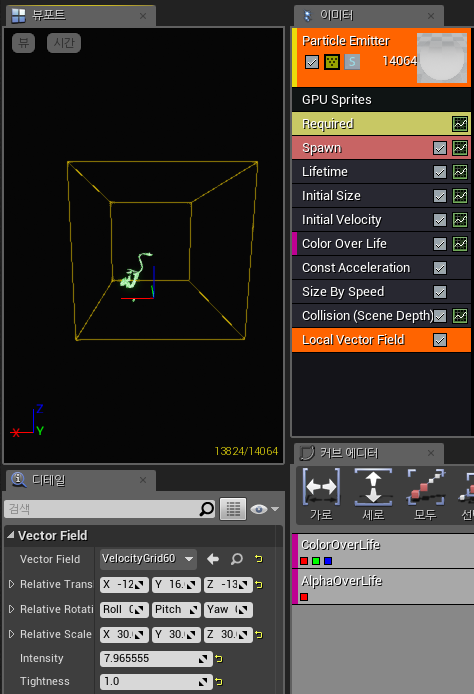
다음으로 콜리전을 다시 설정하겠습니다. Collision(Scene Depth)를 추가하면 충돌이 다시 생깁니다.



위의 화면은 탄성계수를 0.3으로 준 것입니다. 파티클이 살짝 튀기고서 밀리는 모습을 볼 수 있습니다. 바닥의 마찰력이 0으로 되어 있어서 파티클의 생존시간이 길다면 옆으로 쭉 미끄러지는 모습을 볼 수 있습니다.

다음으로 GPU파티클의 강력한 기능인 Vector Field를 사용해 보겠습니다. 사실 동영상 강의에 나오는 벡터필드는 소용돌이치면서 퍼지는 모양이지만 제공되는 예제 프로젝트를 사용해도 영상의 벡터필드는 제공되지 않습니다. 별 수 없이 예제 프로젝트에 제공되는 벡터 필드를 사용했을 때에 아래 화면과 같은 벡터필드가 만들어 집니다.

Relative Scale은 벡터필드의 크기, Intensity는 벡터필드의 강도, Tightness는 벡터필드의 영향력입니다. 아래 화면의 벡터필드는 알 수 없는 모양으로 움직이긴 하지만 벡터필드가 동작함을 알 수 있습니다. 만일 파티클이 벡터 필드의 범위 밖으로 벗어날 경우 그대로 떨어지게 됩니다.



벡터필드를 회전 시킬 수도 있습니다. Vector Field Rotation Rate를 만들고 디테일에서 Rotation Rate의 값을 변경해주면 설정한 값대로 회전하게 됩니다. 다음과 같은 화면을 볼 수 있습니다. 앞서 말했듯이 벡터 필드를 벗어나면 그대로 떨어지게 되므로 씬에 파티클을 잔뜩 뿌리는 화려한 모습을 볼 수 있습니다. 벡터필드의 강도를 살짝 낮춰주면 더 많이 흩뿌리는 것을 볼 수 있습니다.

