**FPS(Frame Per Seconds)**

: 비디오나 필름의 화면이 바뀌는 속도를 나타내는 단위. 1초당 렌더링 하는 장면 수

**고정 프레임(Fixed Timestep)**

: 3D 렌더링을 하다보면 계산량이 많은 부분에서는 FPS가 낮게 나오고 계산량이 높은 부분에서는 fps가 높게 나온다. 이 문제를 해결하기 위한 방법으로 계산량에 관계없이 fps를 고정시키는 방법

**Semi-fixed Time step (Fixed Timestep, variable Rendering)**

: 물리, 기타 연산은 고정 프레임마다 하고 렌더링은 가변적으로 실행. ‘spiral of death’라는 문제가 있음.

참고 사이트: <http://rapapa.net/?p=381>, <http://gafferongames.com/game-physics/fix-your-timestep/>

시간 측정 방법

1. <ctime> 내 함수 이용.
   1. 최대 정밀도 1초, 한프레임당 수십ms 가 걸리는 게임에는 적절하지 않음.
2. timeGetTime(), GetTickCount() 사용
3. QueryPerformanceFrequency(), QueryPerformanceCounter() 사용
   1. 정밀도가 CPU주기, 3GHz CPU는 초당 30억 번 시간 체크(0.333ns).
   2. QueryPerformanceCounter ()
      1. 64bit 카운트 레지스터를 읽어옴
   3. QueryPerformanceFrequency ()
      1. CPU에서 초당 카운터를 몇번 증가시키는지 리턴.
   4. 멀티코어 프로세서의 경우 코어마다 정밀 타이머가 독립적이라서 그 값이 점점 벌어질 가능 성이 있다.

참고 <http://egloos.zum.com/NeoTrinity/v/943773>, 게임엔진 아키텍처

**Timer Class**

: 시간 측정 클래스

1. QueryPerformanceFrequency(), QueryPerformanceCount() 사용
2. ms 시간 단위 계산
3. TimeScale로 시간 흐름 속도 제어

