게임프로그래밍

Animation Tips

박종승

Dept. of CSE, Incheon Nat. Univ.
jong@inu.ac.kr
http://ecl.inu.ac.kr

Animation.h

04.SimplePathAnimationSample

- Animation 클래스
  - (참고)순수가상함수(몸체가 없는 함수; 함수 선언부에서 "=0"로 표시함)인 ComputeValue()를 포함하여 인스턴스화가 불가능한 추상 클래스임

```
template <typename T>
class Animation {
protected:
    virtual T ComputeValue(T time) = 0; // 애니메이션 진행 시간 time 에서의 애니메이션 현재 위치
protected:
    T m_Duration;                       // 애니메이션 지속 시간; 초 단위;
    T m_Start;                          // 이동 경로 상에서의 애니메이션 시작 위치
    T m_End;                            // 이동 경로 상에서의 애니메이션 끝 위치
};
```

  - 함수들(public)

```
public:
    Animation(T start, T end, T duration) : m_Start(start),m_End(end),m_Duration(duration){}
    void SetStart(T start){ m_Start = start; }
    T GetStart(){ return m_Start; }
    void SetEnd(T end){ m_End = end; }
    T GetEnd(){ return m_End; }
    void SetDuration(T duration){ m_Duration = max(0, duration); }
    T GetDuration(){ return m_Duration; }
    T GetValue(T time){ time = min(max(time, 0), m_Duration); return ComputeValue(time); }
```

Animation.h'

- Animation을 상속한 클래스 XXXAnimation

```
template <typename T>
class XXXAnimation : public Animation<T> {
public:
    XXXAnimation(T start=0, T end=0, T duration=0) : Animation(start, end, duration){}
protected:
    virtual T ComputeValue(T time) { /*BODY*/ }
};
```

- LinearAnimation: Linearly Interpolate Between Start and End

```
virtual T ComputeValue(T time) { return m_Start + ((m_End - m_Start) * (time / m_Duration)); }
```

- EaseInExponentialAnimation

```
T ComputeValue(T time) {return m_Start + (m_End - m_Start) * pow(2, 10 * (time/m_Duration - 1)); }
```

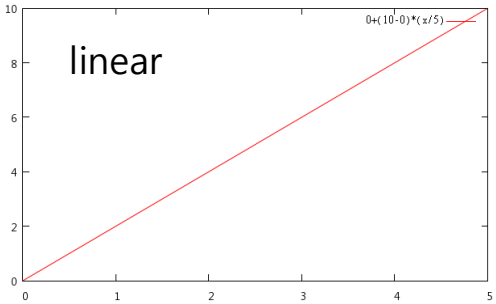
- EaseOutExponentialAnimation

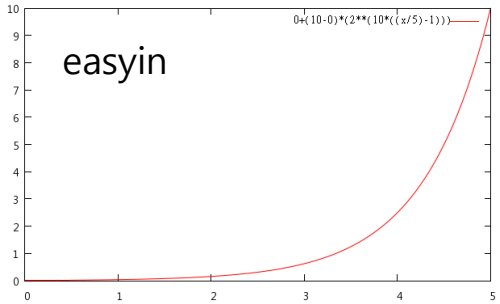
```
T ComputeValue(T time) { return m_Start + (m_End - m_Start) * (-pow(2, -10 * time/m_Duration) + 1); }
```

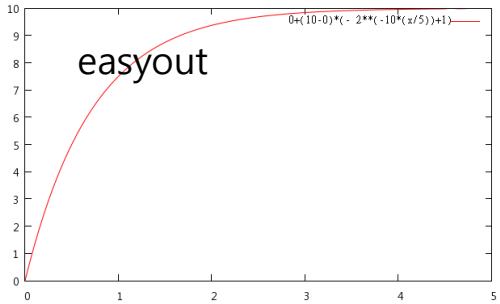
Animation.h”

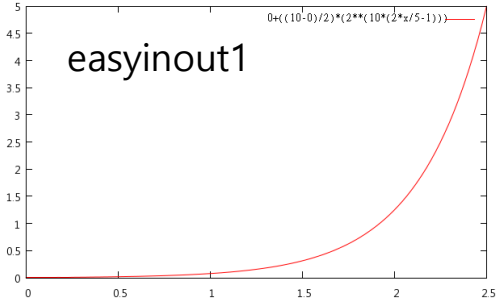
- EaseInOutExponentialAnimation

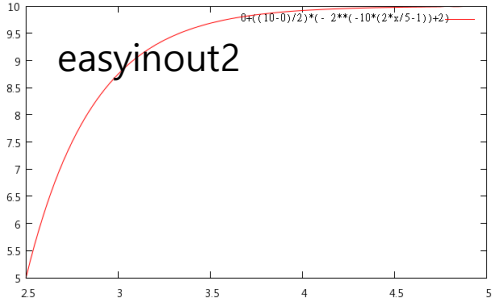
```
T ComputeValue(T time) {
        //compute the current time relative to the midpoint
        time = time / (m_Duration / 2);
        //if we haven't reached the midpoint, we want to do the ease-in portion
        if (time < 1) { return m_Start + (m_End - m_Start)/2 * pow(2, 10 * (time - 1)); }
        //otherwise, do the ease-out portion
        return m_Start + (m_End - m_Start)/2 * (-pow(2, -10 * --time) + 2);
    }
```











Animation.h 사용하기

- Animation.h 사용하기
  - 준비하기
    - #include "Animation.h"
  - 변수 선언
    - EaseInOutExponentialAnimation<float> m_Animation;
  - 초기화

```
m_Animation.SetStart(0);         //start at beginning of path
m_Animation.SetEnd(length);      //length at end of path
m_Animation.SetDuration(5.0f);  //seconds
```

  - OnRender()에서 현재 시간에서의 애니메이션 위치 얻기

```
static float float_time = 0.0f;
float length = m_Animation.GetValue(float_time);
m_pPathGeometry->ComputePointAtLength( length, NULL, &point, &tangent);
```

  - OnRender()에서 애니메이션 후에 현재 시간 갱신하기

```
if (float_time >= m_Animation.GetDuration())
            // When we reach the end of the animation, loop back to the beginning.
            float_time = 0.0f;
else  float_time += elapsedTime;
```

FPS

- DWM(Desktop Window Manager)(link) 사용하여 경과시간 구하기
  - 준비하기
    - 헤더파일: #include <Dwmapi.h>
    - 라이버러리 파일: Dwmapi.lib
  - 변수 선언
    - DWM_TIMING_INFO m_DwmTimingInfo;
  - 초기화
    - DWM가 실행 중일 때에, composition refresh rate 얻기

```
ZeroMemory(&m_DwmTimingInfo, sizeof(m_DwmTimingInfo));
m_DwmTimingInfo.cbSize = sizeof(m_DwmTimingInfo);
hr=DwmGetCompositionTimingInfo(NULL, &m_DwmTimingInfo)
```

    - DWM가 사용되지 않는 경우 GDI로부터 refresh rate 얻기(대부분 60Hz)

```
if (FAILED( hr )) {
  HDC hdc = GetDC(m_hwnd);
  m_DwmTimingInfo.rateCompose.uiDenominator = 1;
  m_DwmTimingInfo.rateCompose.uiNumerator = GetDeviceCaps(hdc, VREFRESH);
  ReleaseDC(m_hwnd, hdc);
}
```

FPS'

- DWM 사용하여 경과시간 구하기'
  - Animation Rendering 함수 OnRender()
    - 반복해서 실행되어야 함
    - WndProc()에서 WM_PAINT 메시지 함수로 OnRender()를 호출하는 경우
      - 호출한 후에, ValidateRect()를 호출하지 말아야 함
      - 호출한 후에, DefWindowProc()도 호출되지 않도록 해야 함
  - OnRender()

```
static float elapsedTime = 0.0;
elapsedTime += static_cast<float>(m_DwmTimingInfo.rateCompose.uiDenominator) /
                    static_cast<float>(m_DwmTimingInfo.rateCompose.uiNumerator);
```