

Window Programming

Visual C++ MFC Programming

Lecture 08

김예진

Dept. of Game Software

Notices

- 03/20: HW 1 (Due: 03/26) → Avg: 8.9
- 04/05: HW 2 (Due: 04/13)
- 04/19: Midterm
 - 5문제, ~75 min., 강의록 1~8

Plan

- MFC와 Timer
 - 연습 1: 공 움직이기
 - 연습 2: 공 튕기기

MFC와 Timer

주기적인 작업을 지정하기



Image와 Timer

- Still Image
 - One image



Still Life [Cézanne, Paul]

- Animation
 - Lots of images: Image sequence



Image와 Timer

- 30 → 60 → 120 frames per seconds (fps)



SetTimer 함수

```
void SetTimer(int id, int time, void *fp) ;
```

- 매 설정된 시간마다 WM_TIMER 메시지 발생

```
예) SetTimer(0, 100, NULL);
```

– id

- timer의 id (예: 0, 1, 2, ...)
- timer가 여러 개 있을 경우 구분하기 위해 사용

– time:

- 알람을 울릴 주기 (=millisec)
- 1000 = 1 초

- 반드시 윈도우가 만들어 진 후 설정한다.

– 주로 WM_CREATE의 핸들러인 OnCreate 내부에 설정

WM_TIMER 핸들러

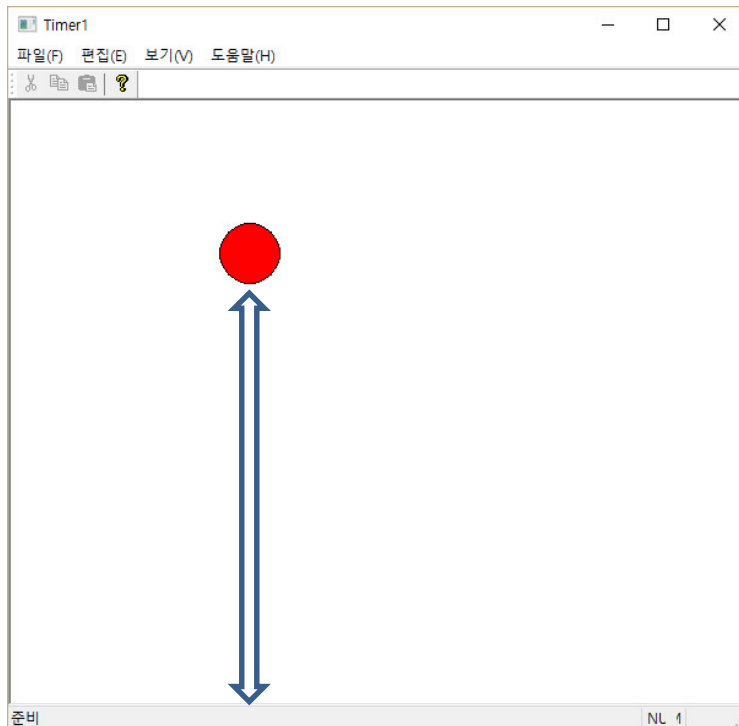
```
afx_msg void OnTimer(int nIDEvent)
```

- nIDEvent
 - 현재 WM_TIMER를 발생시킨 타이머의 id
- 코딩 예)

```
void CChildView::OnTimer(int nIDEvent)
{
    if (nIDEvent == 0)
    {
        // 주기마다 해 줘야 할 일
    }
}
```


연습 1: 공 움직이기

- 공이 상하로 움직이는 장면 만들기



1. 위치 저장 변수 선언

```
CPoint m_pt;
```

2. 위치에 원그리기

(OnPaint)

```
dc.Ellipse(m_pt.x, ...);
```

3. WM_CREATE 핸들러 추가

(OnCreate)

4. OnCreate 함수에 Timer 세팅

```
SetTimer(0, 30, NULL);
```

5. WM_TIMER 핸들러 추가

(OnTimer)

```
if (nIDEvent == 0)
```

.....

운동학 (Dynamics)

- 시간에 따라 ~~그림~~이 변한다 = Dynamics

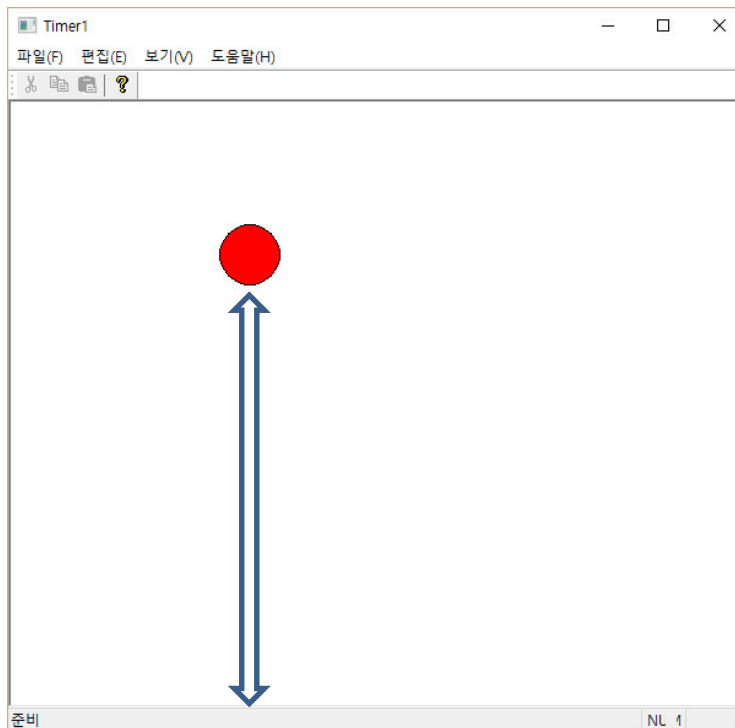
값(숫자)

- 도형의 값(values/properties)?
 - 색
 - 모양
 - 위치

운동학 (Dynamics)

- 운동학(Dynamics): 도형의 위치를 의미하는 변수의 값을 시간에 따라 변화 시켜 주는 것

- Ex)



속도 일정
➔ 재미 없다!

보다 복잡한 움직임은?

운동학 (Dynamics)

- 물체의 운동을 기술하기 위해 필요한 값들
 - 위치 (position) : 보통 p 로 표현
 $p(t)$: t 초 때 위치
 - 속도 (velocity) : 위치의 시간에 따른 변화 (dp/dt)
 $v(t) = p(t+1) - p(t)$
 - 가속도 (acceleration): 속도의 시간에 따른 변화 (dv/dt)
 $a(t) = v(t+1) - v(t)$

시간에 따른 위치가 주어지면 속도, 가속도를 구할 수 있다.
그 반대는?

운동학 (Dynamics)

- 물체의 운동을 기술하기 위해 필요한 값들
 - 가속도 (acceleration): 보통 a 로 표현
 $a(t)$: t 초 때 가속도
 - 속도 (velocity) : t 초 후 속도차이는 가속도 만큼
 $v(t+1) = v(t) + a(t)*1\text{초}$
 - 위치 (position) : t 초 후 위치 차이는 속도 만큼
 $p(t+1) = p(t) + v(t)*1\text{초}$

운동학 (Dynamics)

- Newton의 운동 방정식(Equation of Motion):

$$\mathbf{f} = m\mathbf{a}$$

- 힘이 주어지면 가속도를 계산할 수 있다.
 - 가속도가 주어지면 Δt 초 후의 속도가 계산 가능
 - 속도가 주어지면 Δt 초 후의 위치가 계산 가능
-
- Ex)
 - 자유낙하
 - 힘 = 중력가속도 g ($= -9.8\text{m/sec}^2$) * 무게
 - 스프링
 - 힘 = 기준 위치와의 차이 $f = k x$ (k: 스프링 상수)

운동학 (Dynamics)

- 운동학을 프로그래밍

1. 위치/속도/가속도를 저장할 변수를 만든다.
(m_p , m_v , m_a)

2. 운동학 (Dynamics)정해진 시간마다 다음의 일을 반복

- A. 주어진 상태에서의 힘 계산: (Ex) 중력 or 스프링 힘

- B. 가속도 값 갱신: $a = f/m$

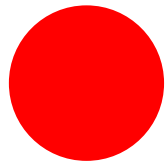
- C. 속도 값 갱신: $v = v + a * dt$

- D. 위치 값 갱신: $p = p + v * dt$

- E. 변경된 위치에 그림 그리기

연습 2: 공 튕기기

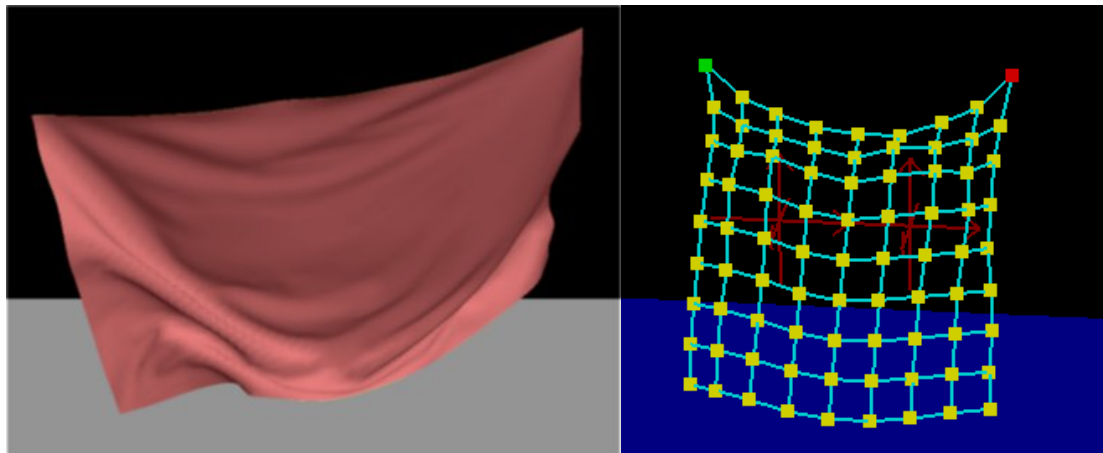
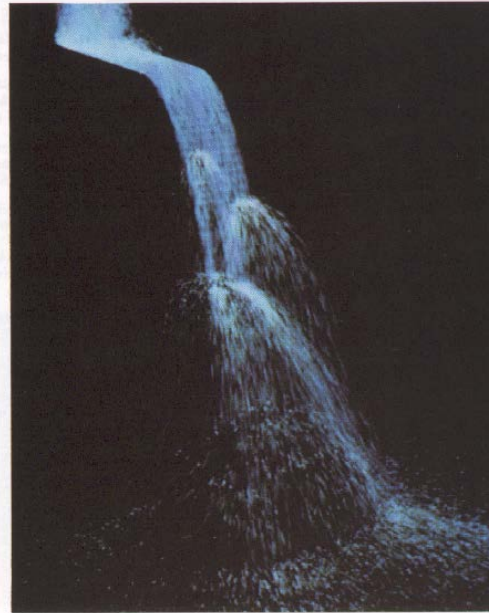
- 공 튕기기



아래와 같은 순서대로 각자 코딩 해 보자

1. 정해진 위치에서 공이 자유 낙하
→ $a = g$
2. 마우스로 클릭하면 공의 위치를 다시 세팅
3. 밑에 벽이 있어 공이 다시 튕겨 올라 간다
벽에 닿는 순간 다음과 같이 값을 변경
→ $p(t+dt) = \text{벽과 닿은 위치}$
→ $v(t+dt) = -e * v(t)$ (e : 반발계수, 보통 0.8)
4. 마우스로 공을 클릭하면 그 순간만 가속도 증가
(= 드리볼)
→ $a(t) = g + f$ (f : 임의의 값)
→ 이 후 다시 $a = g$ 로 회귀 해야 함

더 다양한 예제를 찾아보자



Q & A