

CSE3013 (컴퓨터공학 설계 및 실험 I)

WIN-3 예비 보고서

서강대학교 컴퓨터공학과 박수현 (20181634)

서강대학교 컴퓨터공학과

1 목적

- SDI에 관하여 교재에서 설명한 MFC의 AFX 클래스를 이해한다.
- CDC 객체를 사용하여 그리기를 수행하는 방법을 이해한다.
- 물 흐르는 경로를 찾는 알고리즘 및 자료구조를 고안한다.

2 문제

2.1 AFX 클래스

- `class CAboutDlg : public CDialogEx`: 기본적으로 작성되는 프로그램 정보 다이얼로그이다. Waterfall.cpp에 저장된다.
- `class CMainFrame : public CFrameWndEx`: 메인 프레임 윈도우이다. MainFrm.h, MainFrm.cpp에 저장된다.
- `class CWaterfallApp : public CWinAppEx`: CWinApp을 상속해 선언하고 구현된다. Waterfall.h, Waterfall.cpp에 저장된다.
- `class CWaterfallView : public CView`: CView를 상속해 뷰 윈도우를 선언하고 구현한다. WaterfallView.h, WaterfallView.cpp에 저장된다.
- `class CWaterfallDoc : public CDocument`: CDocument를 상속해 도큐먼트를 구현한다. WaterfallDoc.h, WaterfallDoc.cpp에 저장된다.

2.2 CDC 객체

CDC(Class of Device Context) 객체는 DC(Device Context)에 그리는 객체이다. DC는 임의의 출력 하드웨어를 소프트웨어 인터페이스로 재구성한 것이며, 여기서는 HTML의 `<canvas>` 태그와 비슷하게 간단히 그리짐이 일어나는 곳이라고 이해할 수 있겠다.

다음과 같은 과정을 통해 CDC로 DC에 그릴 수 있다.

- 그릴 도구(CGdiObject)를 선택한다.
- 도구의 속성을 지정한다.
- SelectObject로 CGdiObject를 CDC에 등록한다.
- CDC의 함수를 이용해 DC에 그리기를 수행한다.
- CGdiObject가 더 이상 필요하지 않을 경우 메모리에서 제거한다.

CGdiObject에는 다음과 같은 항목들이 있다.

이름	기능
CPen	임의의 선분 그리기
CBrush	면의 내부 채우기
CFont	글리프 그리기
CBitmap	비트맵 그리기
CRgn	임의의 도형 그리기

또한 CDC에는 다음과 같은 멤버 함수들이 있다.

함수	기능
MoveTo(int x, int y)	도구 이동
LineTo(int x, int y)	도구의 현재 위치에서 (x, y)까지 선분 그리고 도구 이동
TextOut(int x, int y, LPCTSTR lpszString, int nCount)	글리프 그리기
Rectangle(int x1, int y1, int x2, int y2)	좌상단 (x1, y1), 우하단 (x2, y2)인 직사각형 그리기
Ellipse(int x1, int y1, int x2, int y2)	좌상단 (x1, y1), 우하단 (x2, y2)에 내접하는 타원 그리기

2.3 물 흐르는 경로 계산 알고리즘

자료구조 시작점, 선분, 물 흐르는 경로를 저장해야 한다.

- 점들은 2차원 평면상에 존재하므로, 실수의 2-tuple을 이용해 만들 수 있겠다. `std::pair`를 상속하는 클래스를 만들거나, 단순히 `x, y` 두 개의 필드가 있는 **struct**로 구현할 수 있을 것이다.
- 선분은 시작 점과 끝 점으로 나타낼 수 있다. 따라서 위에서 만든 점 자료구조의 2-tuple으로 나타낼 수 있다. 점 자료구조의 구현 방법과 유사하게 구현할 수 있을 것이다.

- 점들의 집합과 선분의 집합을 나타내는 자료구조로는 가변 배열인 `std::vector`를 사용하는 것이 좋겠다. 물이 흐르는 경로도 점들의 순차 집합으로 생각할 수 있는데, 이도 `std::vector`로 나타낼 수 있다.

알고리즘 선분이 n 개, 물구멍이 m 개 있다고 하자. 또한 선분들의 집합 S_s , 물구멍들의 집합 S_w 가 있다고 하자.

- 1 S_s 에 선분 $y = 0$ 추가
- 2 가변 크기 배열들을 원소로 갖는 크기가 n 인 배열 *list* 선언
- 3 $0 \leq i < n$ 인 모든 정수 i 에 대해 반복:
 - a 점 $P(P_x, P_y) = (S_w)_i$ 선언
 - b *list_i*에 P 추가
 - c $P_y > 0$ 일 경우 반복:
 - i 선분 $h: x = P_x$, 점 $M(M_x, M_y = 0)$, 선분 $l: L_1 \rightarrow L_2$ 선언
 - ii $k \in S_s$ 에 대해 반복:
 - (1) h 과 k 의 교점 $Q(Q_x, Q_y)$ 계산
 - (2) $Q_y < M_y$ 일 경우, 다음 반복 단계로 진행
 - (3) $Q_y > P_y$ 일 경우, 다음 반복 단계로 진행
 - (4) $M \leftarrow Q, l \leftarrow k$
 - iii *list_i*에 M 추가
 - iv $l = (y = 0)$ 일 경우, 반복 종료
 - v *list_i*에 L_1, L_2 중 y 좌표가 작은 점 L_m 추가
 - vi $P \leftarrow L_m$
- 4 *list* 반환

m 개의 물구멍에 대해 최악의 경우 n 개의 선분에 대해 다른 모든 선분들과의 교점을 찾는 작업을 진행하므로, 시간 복잡도는 $\mathcal{O}(mn^2)$ 이다.