B577030 homework1 구현 방법 및 어려웠던 점

1.1 mouseButtonDown(R,L) 함수와 mouseMove 함수를 통해 마우스 위치 좌표 값을 얻고 그 좌표값을 각 도형 클래스의 draw 함수에 넘겨줘 적절하게 클라이언트에 그려냄.

1.2 mouseMove 함수가 호출될 때 마다 invalidate를 실행시켜 WM\_PAINT 메시지를 통해 onPaint 함수를 호출해 마우스가 움직일 때 마다 클라이언트에 도형을 새로 그림.

1.3 단색은 CBrush를 통해 구현하고 비트맵은 비트맵 전용 DC를 하나 선언한 뒤 그곳에 비트맵을 그려넣고 stretchBit를 통해 구현.

1.4 가상 DC를 선언하여 원은 가상 DC에 직접 Eclipse 함수를 통해 그려내고 사각형은 가상 DC에 위에서 말한 stretchBit를 통한 방법으로 그려냄. 그려낸 후 가상 DC의 화면을 bltBit 함수를 통해 현재 클라이언트에 복사함.

2.1 도형이 그려질 때 마다 CShape 클래스 포인터 배열에 도형의 정보를 저장하고 onPaint가 호출될 때 마다 각 자식 클래스의 draw 함수로 도형을 가상 DC에 다시 그려냄. onPaint 함수 마지막에 bltBit로 가상 DC를 한번 더 복사해서 클릭을 통해 도형이 그려지지 않아도 창 크기를 조절을 통해서 도형이 지워지지 않음.

2.2 CShape 더블 포인터를 통해 CRectangle, CCircle의 정보를 저장하여서 순서가 유지됨.

3.1 CShape 함수에 가상으로 draw 함수를 선언해주고 자식 클래스인 CRectangle, CCircle 클래스에서 재정의 하여 각 용도에 맞게 사용함.

3.2 CShape를 더블 포인터로 선언하여 new를 통해 동적 할당해주고 \*CShape는 NULL로 비워서 자식 클래스가 진입할 수 있게 처리. \*CShape 배열을 통해서만 도형의 순서를 저장.

3.3 free 함수를 이용해 소멸할 때 사용한 동적 메모리를 모두 해제.

4.1 각 클래스의 선언은 헤더 파일에서 하고 정의는 소스 파일에서 진행.

5.1 MMB 클릭으로 좌표를 받아 정점을 찍고 그 자리에 원을 그려줌.

5.2 더블 클릭으로 onPaint에 진입 후 CRgnn 클래스의 draw 함수로 다각형 완성 후. 도형을 저장.

어려웠던 점

더블 버퍼링 적용과 도형 재출력 문제

클릭으로 인한 메모리 누수 관리 문제

윈도우 크기 조절 시 클라이언트 내부에 잔상이 남는 문제 (더블 버퍼링 문제)