

고급프로그래밍

2019203102

유지성

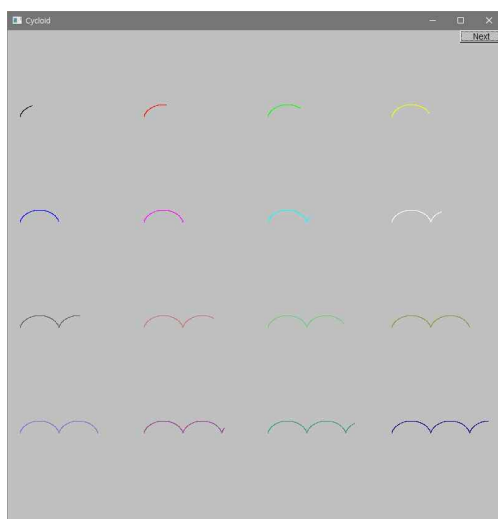
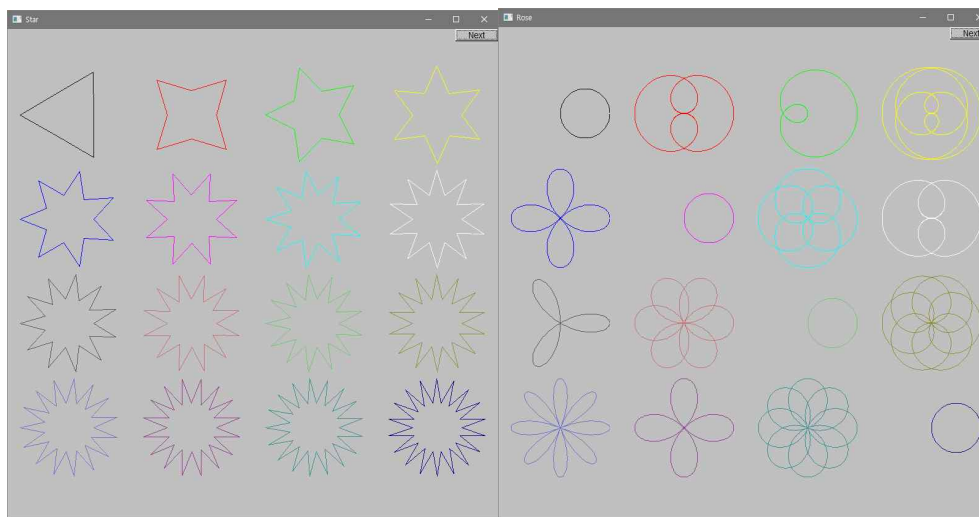
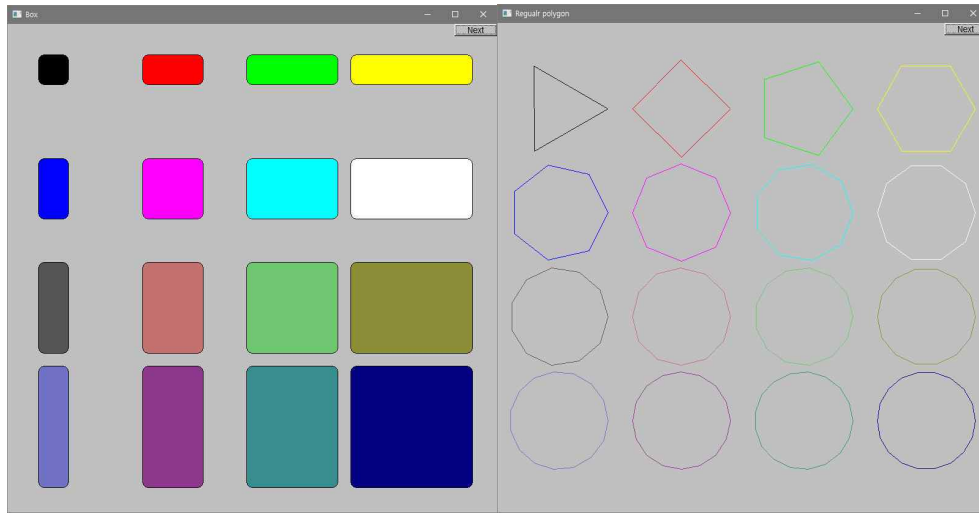
2020.06.05.

소프트웨어학부

요약

Box	O
Regular_polygon	O
Star	O
Rose	O
(Optional) Cycloid	O

결과



요구 조건

Basic

1. Box

Graph.h

Shape 클래스를 상속받았다. 생성자 함수를 이용해 왼쪽 위 꼭짓점, 넓이, 높이, 모서리 반지름을 순서대로 입력받게 하였다. getter 함수로 height, width, radius를 만들었다. private 영역에 넓이, 높이, 반지름에 해당하는 w, h, r 변수를 선언하였다.

Graph.CPP

Box의 draw_lines 함수를 재정의하였다.

선

: fl_line으로 box의 상하좌우 직선을 그렸다. 각 모서리에서 반지름의 길이 만큼을 빼서 둥근 모서리를 만들 공간을 남겨놨다. 둥근 모서리는 fl_arc로 표현하였다.

내부 색

: fl_rectf로 box의 모서리를 제외한 부분을 표현하고, fl_arc로 모서리를 채웠다. fl_rectf로 직사각형 두 개를 만들어 십자 모형으로 겹치고, fl_pie로 만든 중심각이 90도인 부채꼴 모형을 십자 모형의 왼쪽 위, 오른쪽 위, 왼쪽 아래, 오른쪽 아래에 붙여 box 모형을 완성했다.

2. Regular_polygon

Graph.h

Shape 클래스를 상속받았다. 생성자 함수를 이용해 중심점, 꼭짓점의 개수, 외접원의 반지름 길이를 순서대로 입력받게 하였다. getter 함수로 number, radius를 만들었다. private 영역에 꼭짓점의 개수, 외접원의 반지름에 해당하는 n, r 변수를 선언하였다. private 영역에 정다각형의 꼭짓점을 배열에 저장하는 savepoints 함수를 정의하였다.

Graph.cpp

Regular_polygon의 draw_lines 함수를 재정의하였다.

선

: fl_line을 이용해 savepoints 함수를 사용해 배열에 저장해놓은 점들을 선으로 이어줬다.

내부 색

: fl_begin_polygon과 fl_end_polygon을 이용해 savepoints 함수를 사용해서 배열에 저장해놓은 점들을 다각형의 꼭짓점으로 지정해줬다.

3. Star

Graph.h

Shape 클래스를 상속받았다. 생성자 함수를 이용해 중심점, 꼭짓점의 개수, 내접원의 반지름, 외접원의 반지름을 순서대로 입력받게 하였다. getter 함수로 number, radius1(내접원), radius2(외접원)을 만들었다. private 영역에 꼭짓점의 개수, 내접원의 반지름, 외접원의 반지름에 해당하는 n, r1, r2 변수를 선언하였다. private 영역에 별의 꼭짓점을 배열에 저장하는 savepoints 함수를 정의하였다.

Graph.cpp

Star의 draw_lines 함수를 재정의하였다.

선

: fl_line을 이용해 savepoints 함수를 사용해 배열에 저장해놓은 점들을 선으로 이어줬다.

내부 색

: fl_begin_polygon과 fl_end_polygon을 이용해 savepoints 함수를 사용해서 배열에 저장해놓은 점들을 별의 꼭짓점으로 지정해줬다.

Advanced

Rose

Graph.h

Shape 클래스를 상속받았다. 생성자 함수를 이용해 중심점, 분자, 분모, 반지름을 순서대로 입력받게 하였다. getter 함수로 numerator, denominator, radius를 만들었다. private 영역에 분자, 분모, 반지름에 해당하는 n, d, r 변수를 선언하였다.

Graph.cpp

Rose의 draw_lines 함수를 재정의하였다.

선

: fl_vertex로 도형의 선들을 이루고 있는 점들의 좌표를 지정해준 다음에 fl_begin_line과 fl_end_line으로 하나의 선으로 만들어 주었다.

내부 색

: fl_vertex로 도형의 선들을 이루고 있는 점들의 좌표를 지정해준 다음에 fl_begin_complex_polygon과 fl_end_complex_polygon으로 하나의 도형으로 만들어 주었다.

Optional

Cycloid

Graph.h

Shape 클래스를 상속받았다. 생성자 함수를 이용해 시작점, 반지름, 회전 각도를 순서대로 입력받게 하였다. getter 함수로 radius, degree를 만들었다. private 영역에 반지름, 각도에 해당하는 r, d 변수를 선언하였다.

Graph.cpp

Cycloid의 draw_lines 함수를 재정의하였다.

fi_vertax로 도형의 선들을 이루고 있는 점들의 좌표를 지정해준 다음에 fl_begin_line과 fl_end_line으로 하나의 선으로 만들어 주었다.

Cycloid는 면이 있는 도형이 아니므로 내부 색을 채우는 알고리즘은 구현하지 않았다.

main_shapes.cpp

Vector_ref<Shape>로 각 도형을 저장하는 배열을 만들었다. for 문을 이용해 도형들을 만들고 attach로 창에 붙였다. 버튼을 누르면 도형들이 detach되고 다른 유형의 도형들이 나오게 된다.

Comment

처음 다뤄보는 것이라 익숙하지 않다는 게 가장 큰 어려움이었다. 함수들이 어떤 것들이 있는지 잘 모르고 그것들을 어떻게 사용하는지 잘 몰라서 일일이 찾아가며 하는 것이 가장 걸림돌이었다. 적응하고 난 다음에는 비교적 쉽게 코드를 짤 수 있었다.

아쉬운 점은, 아직도 모르는 명령어들이 많아 코드를 더 효율적으로 짤 수 없었다는 것이다.

또한, 선들이 매끄럽지가 않다는 것도 아쉽다. 이는 좌푯값이 정수로만 표현되어서 그런 것으로 보인다. 그래서 삼각함수를 이용한 계산이나 나누기 연산자를 사용한 계산처럼 정수가 아닌 값들이 결과로 나오는 경우 정수로 변환하며 생기는 오차 때문에 정확한 표현을 할 수가 없다. 이로 인해 특히 점들을 하나하나 지정해줘서 그린 곡선들은 더욱더 매끄럽지가 않았다