

🏠 / C++프로그래밍과실습 (CB3500572-062)

/ intermediate - 과제 042 - function - drawing shapes (SFML 은 선택)

개요

제출

편집

코딩 결과

intermediate - 과제 042 - function - drawing shapes (SFML 은 선택)

제출 마감일: 2023-03-31 23:59

업로드 가능한 파일 수: 4

제출 방식: 개인

주의

과제 042 은 keylog 분석 및 jplag 를 이용한 표절 검사가 상시 수행됩니다.

- 키로그 파일(p42.csv)에 소스 코드 이름이 Shape.cpp, main.cpp 등으로 제출 파일명과 일치하지 않으면 로그 점수가 계산되지 않습니다.
- 키로그 파일(p42.csv)에 키보드 입력 정보가 없는 경우 로그 점수가 계산되지 않습니다. (로그가 IdeState 나 Action 타입 밖에 없는 경우가 종종 있습니다.)
- 계속 잘 안될 때에는 초심으로 돌아가 activitytracker 를 clear 하고, 소스 파일명을 제출 파일명과 일치 시킨 후 직접 키보드로 코딩하는 것을 추천 드립니다.

목적

C++ 함수 및 STL algorithm 을 연습해 봅니다.

문제

다음 명령을 수행하는 프로그램을 구현하시오.

- RECT: 사각형을 생성합니다. 왼쪽 위 좌표와 오른쪽 아래 좌표가 주어집니다.
 - CIRCLE: 원을 생성합니다. 중심 좌표와 반지름이 주어집니다.
 - MOVE: 도형의 중심을 x, y 만큼 이동합니다.
 - SORT: 도형의 넓이를 기준으로 오름차순 정렬합니다. (std::sort 사용)
 - MAX: 원점으로부터 중심이 가장 먼 도형을 찾습니다. (std::max_element 사용)
 - REMOVE: 해당 ID의 도형을 삭제합니다. (std::remove_if 사용)
 - ALIGN: 넓이 순서로 도형을 정렬 후 $y=x$ 직선 위에 도형의 중심을 위치 시킵니다. 이때 서로 오버랩 되지 않고 최대한 붙입니다. 도형은 음수 영역에 위치할 수 없습니다. (1사분면에만 위치함)
 - PRINT: 도형 목록을 출력합니다.
-
- (선택) DRAW: 도형 목록을 화면에 표시합니다. (가점 30%, SFML 라이브러리 사용)

<참고>

std::vector 혹은 std::map 중 하나를 선택하여 사용하세요.

```
// shape_main.cpp
```

```
#include "shape.h"
#include <iostream>
#include <SFML/Graphics.hpp>

int main() {
    //sf::RenderWindow window(sf::VideoMode(SCREEN_WIDTH, SCREEN_HEIGHT), "Drawing Shapes");
    //sf::View view(sf::FloatRect(0, SCREEN_HEIGHT, SCREEN_WIDTH, -SCREEN_HEIGHT));
    //window.setView(view);

    bool isQuit = false;
    //while(window.isOpen() && !isQuit) {
    //    sf::Event event;
    //    while (window.pollEvent(event)) {
    //        if (event.type == sf::Event::Closed)
    //            window.close();
    //    }
    while(!isQuit) {
        std::string command;
        std::cin >> command;
        if (command == "RECT") {

            // ... implement your code

        } else if (command == "DRAW") {
```

```

// window.clear();
// draw_shapes(window);
// window.display();
} else if (command == "QUIT") {
    isQuit = true;
} else {
    std::cout << "Invalid command." << std::endl;
}
}
return 0;
}

```

// shape.h

```

const int SCREEN_WIDTH = 640;
const int SCREEN_HEIGHT = 640;

```

```

enum class ShapeType {
    RECT,
    CIRCLE
};

```

```

struct Point {
    int x;
    int y;
};

```

```

struct Shape {
    int id;
    ShapeType type;
    Point pos1;
    Point pos2;
    Point center;
    int radius;
    //std::unique_ptr<sf::RectangleShape> rect;
    //std::unique_ptr<sf::CircleShape> circle;
};

```

```

void create_rectangle(Point& pos1, Point& pos2);
void create_circle(Point& center, int radius);
void move_shape(int id, int x, int y);
void sort_shapes();
int find_furthest_shape();
void remove_shape_by_id(int id);
void align_shapes();
void print_shapes();
//void draw_shapes(sf::RenderWindow& window);

```

//hint - 추가 과제를 진행하는 분들은 자신에 맞게 수정해서 사용하세요.

```

void draw_shapes(sf::RenderWindow& window) {

```

```

for (auto& shape : shape_list) {
    if (shape.type == ShapeType::RECT) {
        shape.rect->setSize(sf::Vector2f(shape.pos2.x - shape.pos1.x, shape.pos2.y - shape.pos1.y));
        shape.rect->setPosition(shape.pos1.x, shape.pos1.y);
        shape.rect->setFillColor(sf::Color::White);
        window.draw(*shape.rect);
    } else {
        shape.circle->setRadius(shape.radius);
        shape.circle->setPosition(shape.center.x - shape.radius, shape.center.y - shape.radius);
        shape.circle->setFillColor(sf::Color::White);
        window.draw(*shape.circle);
    }
    std::shared_ptr<sf::CircleShape> point = std::make_shared<sf::CircleShape>();
    point->setRadius(5);
    point->setPosition(shape.center.x-5, shape.center.y-5);
    point->setFillColor(sf::Color::Black);
    window.draw(*point);
}
}

```

//shape.cpp

```
#include "shape.h"
```

```
std::vector<Shape> shape_list;
```

```
int next_id = 0;
```

```
// implement your code
```

입력

```

RECT 0 20 20 600
CIRCLE 400 440 10
CIRCLE 500 500 20
SORT
RECT 50 100 150 200
CIRCLE 100 100 20
CIRCLE 60 60 30
SORT
ALIGN
PRINT
QUIT

```

출력

ID: 1, Type: CIRCLE, Center: (10, 10), Radius: 10, Area: 314.159
ID: 2, Type: CIRCLE, Center: (40, 40), Radius: 20, Area: 1256.64
ID: 4, Type: CIRCLE, Center: (80, 80), Radius: 20, Area: 1256.64
ID: 5, Type: CIRCLE, Center: (130, 130), Radius: 30, Area: 2827.43
ID: 3, Type: RECT, Position: (210, 210), Width: 100, Height: 100, Area: 10000
ID: 0, Type: RECT, Position: (290, 290), Width: 20, Height: 580, Area: 11600

제출파일

shape.h
shape.cpp
shape_main.cpp
p42.csv

(선택) 추가과제

- ▶ PLATO 서버에는 SFML 라이브러리가 없습니다.
- ▶ 추가 과제를 수행한 분들은 "지난과제 제출하기" 게시판에 프로젝트 폴더를 전체 압축하여 제출해 주세요!
- ▶ 제출하신 코드가 실행이 되어야 채점이 가능합니다.
- ▶ 테스트는 QUIT 이전에 DRAW 하여 아래 이미지와 유사하게 출력되는지 확인해 보세요.



