

Graphics

11&12 - Đồ hoạ với SDL

Nội dung

- Thư viện SDL
 - Cài đặt, khởi tạo, sử dụng, giải phóng
- Xây dựng API vẽ
 - Lớp Painter
- Vẽ hình bằng bút vẽ
 - Đường thẳng, hình vuông, tam giác ...
 - Phối hợp tạo thành *các hình tuyệt đẹp*
 - Vẽ ảnh JPG, PNG
- Vẽ hình fractal
 - Kỹ thuật đệ quy

Đồ họa với SDL

- <https://www.libsdl.org/>
- Hướng dẫn: <http://wiki.libsdl.org/FrontPage>
- SDL có thể phát triển trò chơi chuyên nghiệp
- SDL dễ dàng kết nối với CodeBlocks
- SDL chạy trên nhiều nền tảng (Windows, Linux, Android, iOS ...)

Lựa chọn thư viện

- Phát triển phần mềm trên thực tế
 - Thường cần thư viện (bên thứ 3 - *third party library*) ngoài tính năng của ngôn ngữ và thư viện chuẩn của C++
 - Lựa chọn thư viện cho dự án: ***cần thiết và quan trọng***
- Trong khóa học này, dùng SDL bởi
 - Chỉ dùng các tính năng đồ họa đơn giản
 - Đa nền tảng (cross-platform) - dễ cho sinh viên
- Để phát triển game thực thụ
 - Thường dùng các Game Engine
 - https://en.wikipedia.org/wiki/List_of_game_engines

Các tính năng của SDL

<https://wiki.libsdl.org/Introduction>

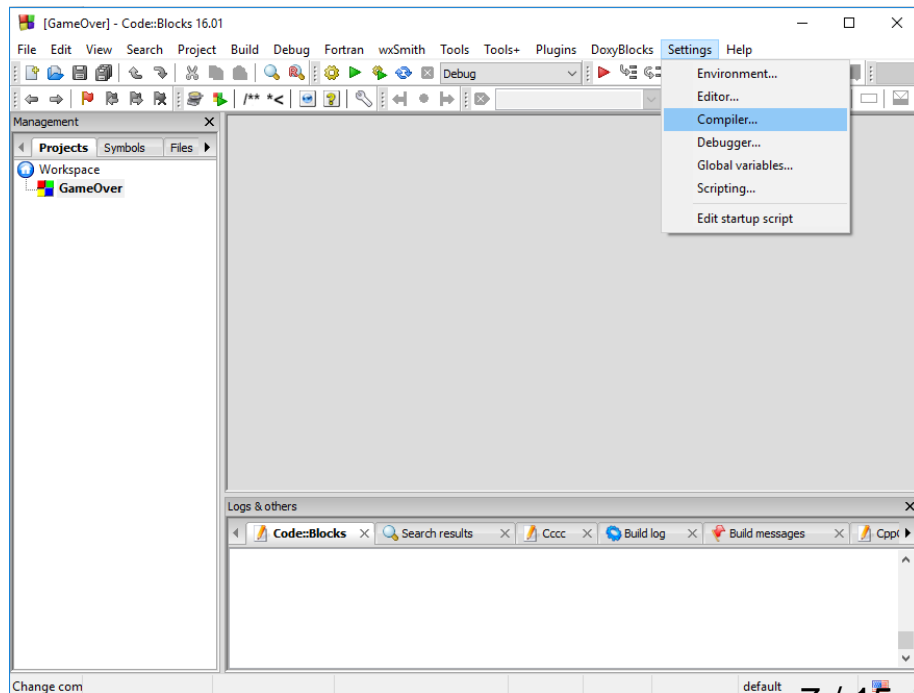
- Video (2D, 3D)
- Input events
- Force Feedback
- Audio
- File I/O
- Shared objects
- Threads
- CPU Detection
- Timer
- Endian independence
- Power Management

Cài đặt SDL với CodeBlocks-MinGW

- Tải về <https://www.libsdl.org/release/SDL2-devel-2.0.5-mingw.tar.gz>
- Giải nén vào một thư mục nào đó, trong đó có 2 thư mục
 - Bản 32bit: i686-w64-mingw32
 - Bản 64bit: x86_64-w64-mingw32
- Ở đây ta dùng bản 32 bit (vì CodeBlock đang dùng mingw32), trong thư mục này có 4 thư mục **bin**, **include**, **lib**, **share**
- Thư mục **bin** chứa SDL2.dll (liên kết khi chạy, **copy file này vào thư mục mã nguồn project**)
- Thư mục **include** chứa các file .h (như stdio.h) khai báo các hàm của SDL
- Thư mục **lib** chứa các thư viện (mã đối tượng) để liên kết chương trình

Cấu hình CodeBlocks

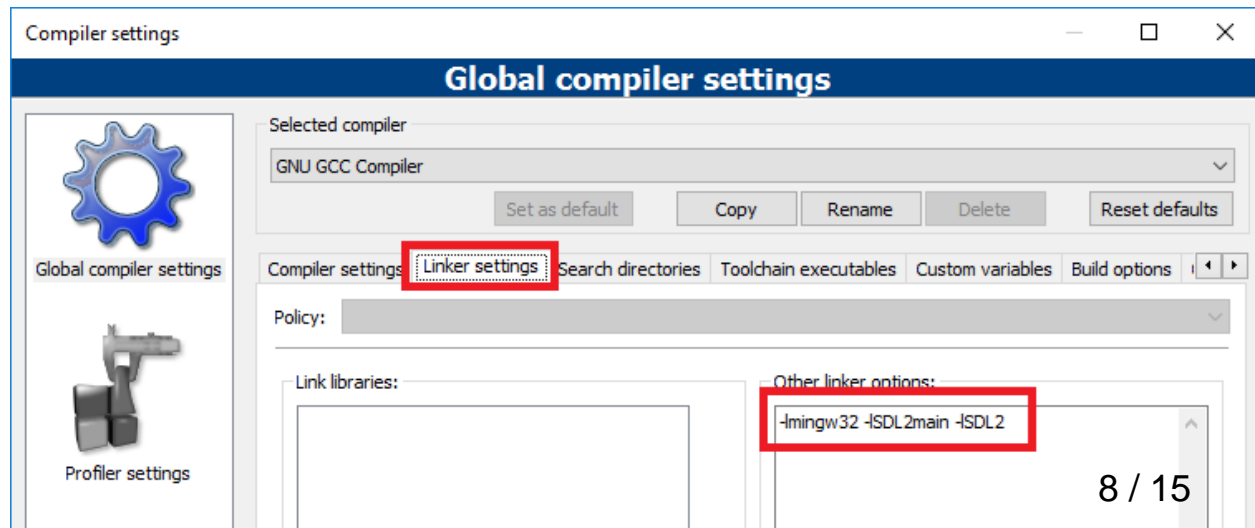
Settings / Compiler ...



Cấu hình CodeBlocks

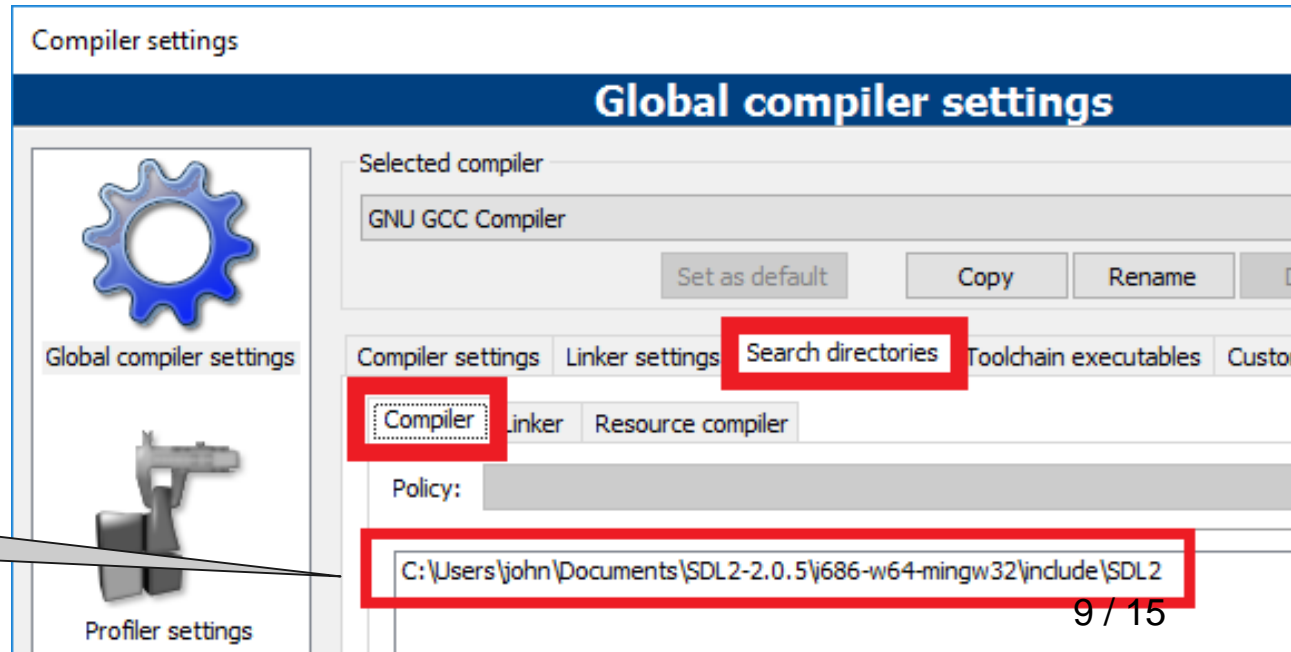
Liên kết thư viện:

- Linker settings: **-lmingw32 -lSDL2main -lSDL2**



Cấu hình CodeBlocks

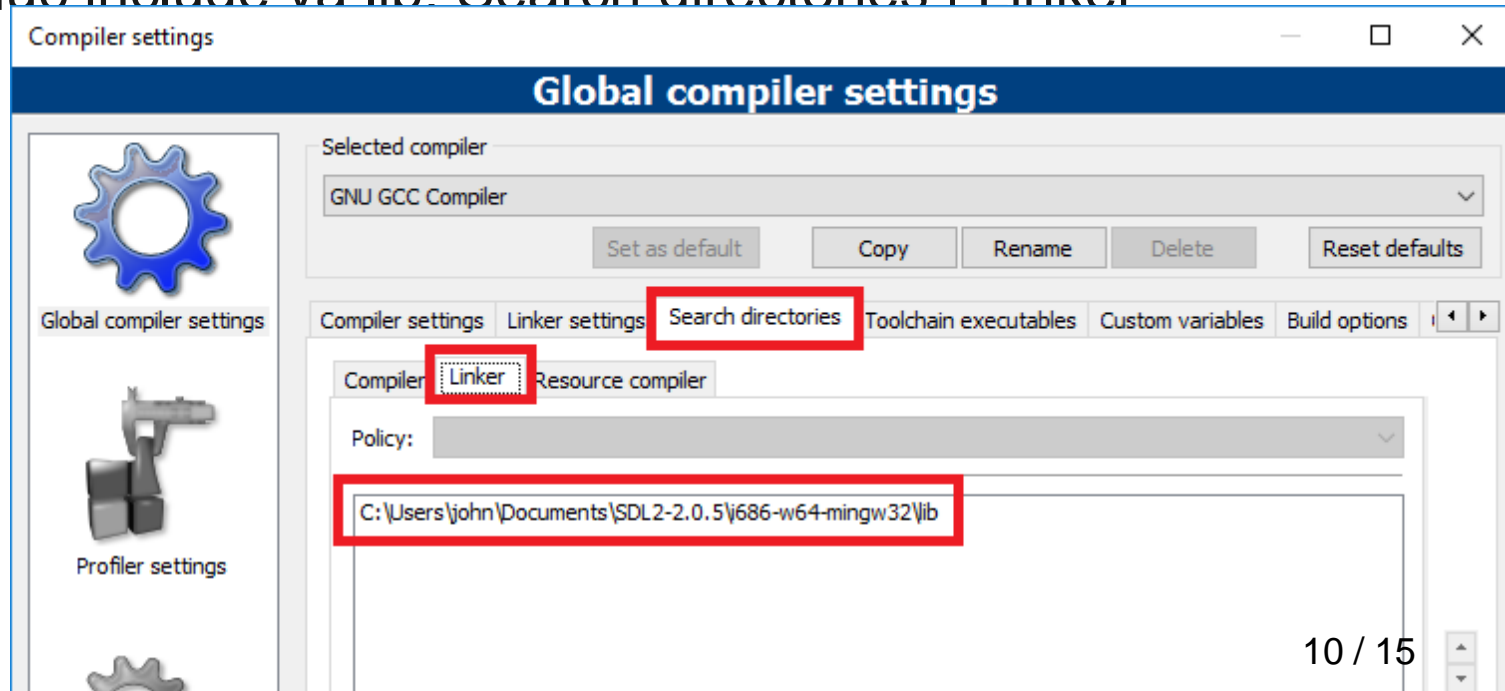
Vị trí thư mục include và lib: Search directories | Compilers



Đường dẫn đến thư
mục chứa SDL

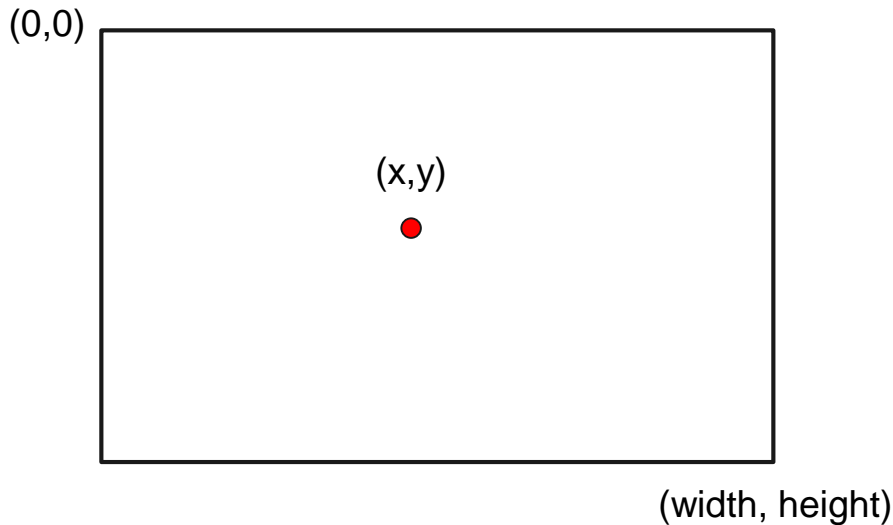
Cấu hình CodeBlocks

Vị trí thư mục include và lib: Search directories | Linker



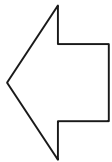
Sử dụng SDL

- Khởi tạo
 - `SDL_Init()`
 - Mở cửa sổ để vẽ
 - Lấy bút vẽ của cửa sổ
- Vẽ hình
- Giải phóng SDL
 - Giải phóng bút vẽ, cửa sổ
 - `SDL_Quit()`



Khởi tạo SDL

```
#include <iostream>
#include <SDL.h>
```



sử dụng thư
viện SDL2

```
using namespace std;
```

```
int main(int argc, char* argv[])
{
    return 0;
}
```

Báo lỗi SDL

```
void logSDLError(std::ostream& os,  
                const std::string &msg, bool fatal = false);  
  
void logSDLError(std::ostream& os,  
                const std::string &msg, bool fatal)  
{  
    os << msg << " Error: " << SDL_GetError() << std::endl;  
    if (fatal) {  
        SDL_Quit();  
        exit(1);  
    }  
}
```

Khởi tạo SDL

```
const int SCREEN_WIDTH = 800;  
const int SCREEN_HEIGHT = 600;  
const string WINDOW_TITLE = "An Implementation of Code.org Painter";
```

```
void initSDL(SDL_Window* &window, SDL_Renderer* &renderer);
```



Đại diện cho cửa sổ vẽ



Đại diện cho bút vẽ

```
void initSDL(SDL_Window* &window, SDL_Renderer* &renderer)
```

```
{
```

```
    if (SDL_Init(SDL_INIT_EVERYTHING) != 0)
        logSDLError(std::cout, "SDL_Init", true);
```

Mở cửa sổ vẽ
theo kích thước
đã chọn

```
    window = SDL_CreateWindow(WINDOW_TITLE.c_str(), SDL_WINDOWPOS_CENTERED,
        SDL_WINDOWPOS_CENTERED, SCREEN_WIDTH, SCREEN_HEIGHT, SDL_WINDOW_SHOWN);
    //window = SDL_CreateWindow(WINDOW_TITLE.c_str(), SDL_WINDOWPOS_CENTERED,
        SDL_WINDOWPOS_CENTERED, SCREEN_WIDTH, SCREEN_HEIGHT, SDL_WINDOW_FULLSCREEN_DESKTOP);
```



```
    if (window == nullptr)
        logSDLError(std::cout, "CreateWindow", true);
```

```
    renderer = SDL_CreateRenderer(window, -1, SDL_RENDERER_ACCELERATED |
        SDL_RENDERER_PRESENTVSYNC);
```



Lấy bút vẽ

```
    //SDL_Renderer *renderer = SDL_CreateSoftwareRenderer(SDL_GetWindowSurface(window));
    if (renderer == nullptr)
        logSDLError(std::cout, "CreateRenderer", true);
```

```
    SDL_SetHint(SDL_HINT_RENDER_SCALE_QUALITY, "linear");
    SDL_RenderSetLogicalSize(renderer, SCREEN_WIDTH, SCREEN_HEIGHT);
```

```
}
```

Giải phóng SDL

```
void quitSDL(SDL_Window* window, SDL_Renderer* renderer);
```

```
void quitSDL(SDL_Window* window, SDL_Renderer* renderer)
{
    SDL_DestroyRenderer(renderer);
    SDL_DestroyWindow(window);
    SDL_Quit();
}
```

Giải phóng
bộ nhớ quản
lý cửa sổ và
bút vẽ

Đợi 1 phím để thoát

```
void waitUntilKeyPressed();
```

```
void waitUntilKeyPressed()
```

```
{
```

```
    SDL_Event e;
```

```
    while (true) {
```

```
        if ( SDL_WaitEvent(&e) != 0 &&  
            (e.type == SDL_KEYDOWN || e.type == SDL_QUIT) )
```

```
            return;
```

```
        SDL_Delay(100);
```

```
    }
```

```
}
```



Xác định sự
kiện bàn phím

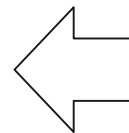
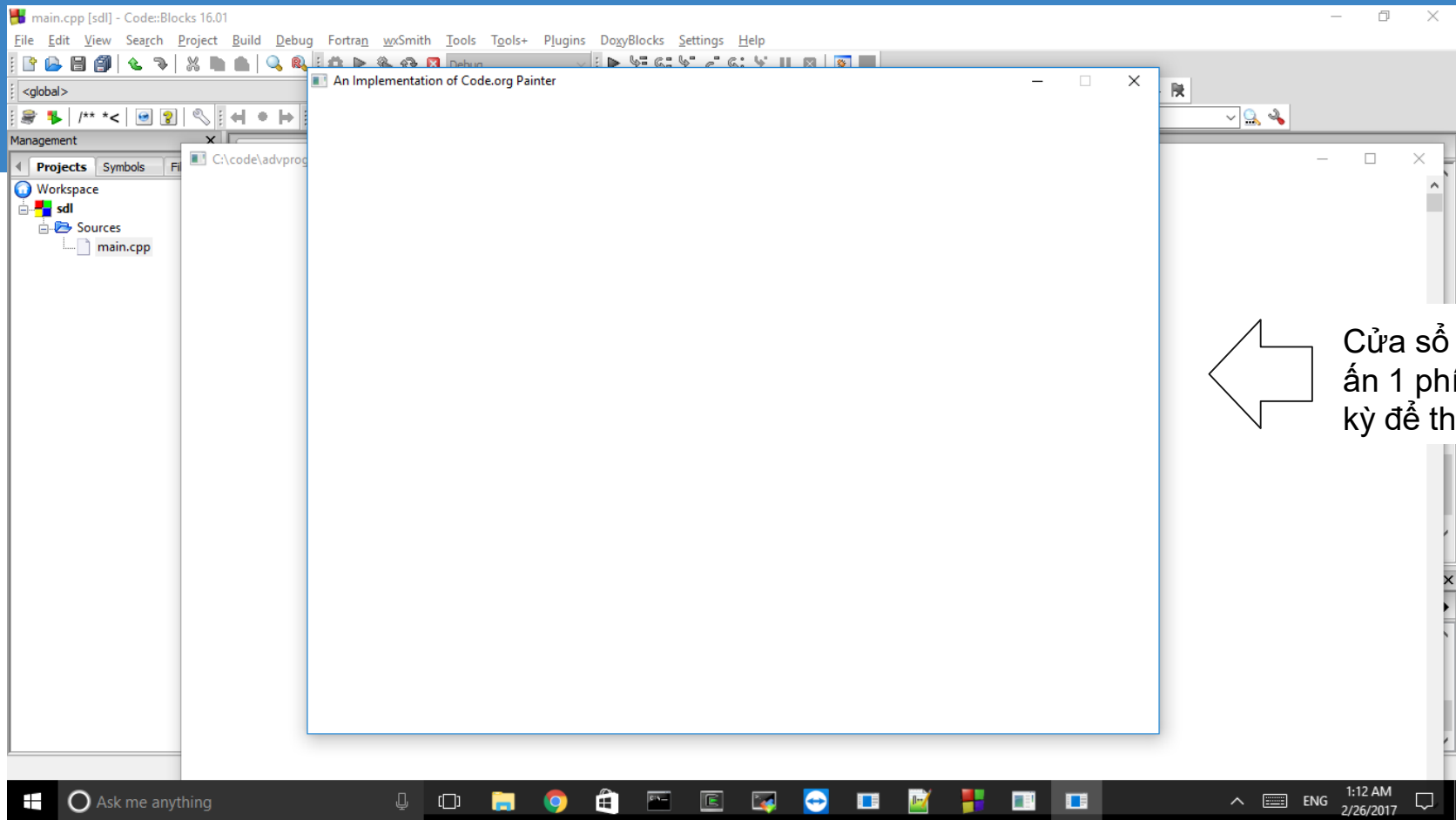
Hàm main()

```
int main(int argc, char* argv[])
{
    SDL_Window* window;
    SDL_Renderer* renderer;
    initSDL(window, renderer);

    // Your drawing code here
    // use SDL_RenderPresent(renderer) to show it

    waitUntilKeyPressed();
    quitSDL(window, renderer);
    return 0;
}
```

<https://github.com/tqlong/advprogram/raw/6f01f8f6f96afe0aa9e107d65dcde17802f1e1e3/lec10-sdl/main.cpp>



Cửa sổ trắng,
ấn 1 phím bất
kỳ để thoát

Nội dung

- Thư viện SDL:
 - Cài đặt, khởi tạo, sử dụng, giải phóng
- **Xây dựng API vẽ**
 - **Lớp Painter**
- **Vẽ hình bằng bút vẽ**
 - Đường thẳng, hình vuông, tam giác ...
 - Phối hợp tạo thành *các hình tuyệt đẹp*
 - Vẽ ảnh JPG, PNG
- **Vẽ hình fractal**
 - Kỹ thuật đệ quy

Vẽ hình với SDL

- SDL đã cung cấp hàm
 - Vẽ điểm, đoạn thẳng, hình chữ nhật ...
 - Với các hình khối phức tạp hoặc ảnh
 - Dùng thư viện mở rộng SDL_image
 - Dùng SDL với OpenGL (3D)
- Ta cũng có thể xây dựng thư viện riêng
 - Dựa vào các hàm vẽ cơ bản của SDL
 - Đơn giản hóa các thao tác vẽ
 - Lệnh SDL khá phức tạp bởi có nhiều tham số
 - *Vẽ hình theo phong cách của riêng chúng ta*

Lớp Painter

- API vẽ (*Application Program Interface*)
- Cách vẽ
 - Bắt đầu tại điểm giữa màn hình, hướng sang phải (0 độ), màu vẽ trắng, màu nền xanh
 - Các chức năng vẽ cơ bản
 - Thay màu bút vẽ, tô nền bằng màu mới
 - Tiến lên phía trước một quãng đường
 - Quay phải, quay trái theo góc quay (xoay giấy)
 - Nhảy về phía trước một quãng đường (nhấc bút)

Lớp Painter

```
class Painter
{
    float x;
    float y;
    float angle;
    int width;
    int height;
    SDL_Color color;
    SDL_Renderer* renderer;
```

public:

```
    Painter(SDL_Window* window,
            SDL_Renderer* renderer);
    void setPosition(float x, float y);
    float getX() const { return x; }
    float getY() const { return y; }

    void setAngle(float angle);
    float getAngle() const { return angle; }

    int getWidth() const { return width; }
    int getHeight() const { return height; }

    void setColor(SDL_Color color);
    SDL_Color getColor() const { return color; }
    void clearWithBgColor(SDL_Color color);

    SDL_Renderer* getRenderer() const
        { return renderer; }
};
```

Một số màu hay dùng

```
const SDL_Color CYAN_COLOR = {0, 255, 255};
const SDL_Color BLUE_COLOR = {0, 0, 255};
const SDL_Color ORANGE_COLOR = {255, 165, 0};
const SDL_Color YELLOW_COLOR = {255, 255, 0};
const SDL_Color LIME_COLOR = {0, 255, 0};
const SDL_Color PURPLE_COLOR = {128, 0, 128};
const SDL_Color RED_COLOR = {255, 0, 0};
const SDL_Color WHITE_COLOR = {255, 255, 255};
const SDL_Color BLACK_COLOR = {0, 0, 0};
const SDL_Color GREEN_COLOR = {0, 128, 0};

const SDL_Color DEFAULT_COLOR = BLACK_COLOR;
```

Nên đặt đoạn
mã này ở đâu ?

<http://stackoverflow.com/questions/2387083/where-should-you-put-global-constants-in-a-c-program>

Lớp Painter: Hàm khởi tạo

```
Painter::Painter(SDL_Window* window, SDL_Renderer* renderer_)
    : renderer(renderer_)
{
    SDL_RenderGetLogicalSize(renderer, &width, &height);
    if (width == 0 && height == 0)
        SDL_GetWindowSize(window, &width, &height);

    setPosition(width/2, height/2);
    setAngle(0);
    setColor(WHITE_COLOR);
    clearWithBgColor(BLUE_COLOR);
}
```

Lấy kích
thước cửa sổ

Khởi tạo tọa độ,
màu và hướng ban
đầu của bút vẽ, tô
nền bằng màu xanh

```
void Painter::setPosition(float x, float y)
{
    this->x = x;
    this->y = y;
}
```

```
void Painter::setAngle(float angle)
{
    this->angle = angle
        - floor(angle/360)*360;
}
```

Các phương thức
thay đổi vị trí, màu
sắc, hướng của bút
vẽ và tô màu nền

```
void Painter::setColor(SDL_Color color)
{
    this->color = color;
    SDL_SetRenderDrawColor(
        renderer, color.r, color.g, color.b, 0);
}
```

```
void Painter::clearWithBgColor(SDL_Color bgColor)
{
    SDL_Color curColor = color;
    setColor(bgColor);
    SDL_RenderClear(renderer);
    setColor(curColor);
}
```

Đi tới vẽ đoạn thẳng

```
public:
```

```
...
```

```
// basic drawing functions
```

```
void moveForward(float length);
```

```
void jumpForward(float length);
```

Di chuyển bút vẽ
theo hướng sẵn có
và vẽ đoạn thẳng

```
void Painter::moveForward(float length)
{
```

```
    float prevX = x, prevY = y;
```

```
    jumpForward(length);
```

```
    SDL_RenderDrawLine(renderer,
        (int)prevX, (int)prevY,
        (int)x, (int)y);
```

```
}
```

```
void Painter::jumpForward(float length)
{
```

```
    float rad = (angle / 180) * M_PI;
```

```
    x += cos(rad) * length;
```

```
    y -= sin(rad) * length;
```

```
}
```

Đi lùi, nhảy lùi

```
public:
```

```
...
```

```
    void moveBackward(float length)
```

```
    {
```

```
        moveForward(-length);
```

```
    }
```

```
    void jumpBackward(float length)
```

```
    {
```

```
        jumpForward(-length);
```

```
    }
```

Quay trái, quay phải

```
public:
...
    void turnLeft(float angle)
    {
        setAngle(this->angle+angle);
    }
    void turnRight(float angle)
    {
        turnLeft(-angle);
    }
```

Đi dạo

```
float generateRandomNumber()
{
    return (float) rand() / RAND_MAX;
}

void randomWalk(Painter& painter)
{
    const int STEPS = 10;
    const float MAX_LENGTH = 100;
    for (int i = 0; i < STEPS; i++) {
        float length = generateRandomNumber() * MAX_LENGTH;
        painter.moveForward(length);
        float angle = generateRandomNumber() * 360;
        painter.turnLeft(angle);
    }
}
```

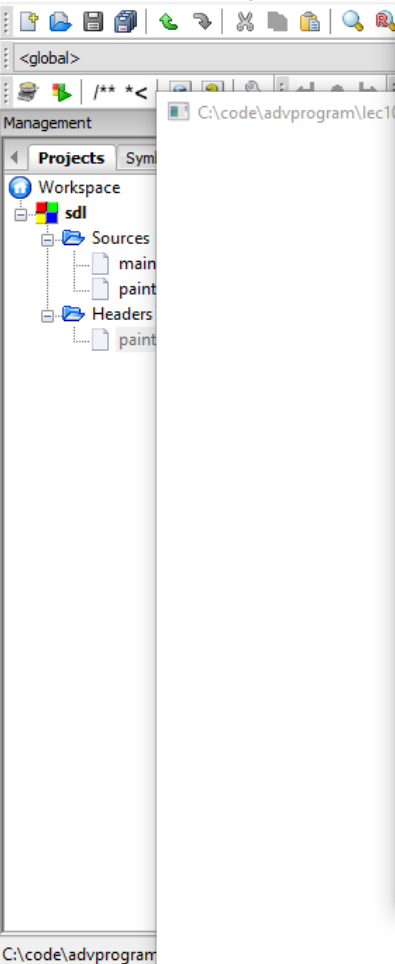
```
int main(int argc, char* argv[])
{
    srand(time(0));
    ...
    initSDL(window, renderer);

    Painter painter(window, renderer);
    randomWalk(painter);

    SDL_RenderPresent(renderer);
    ...
}
```



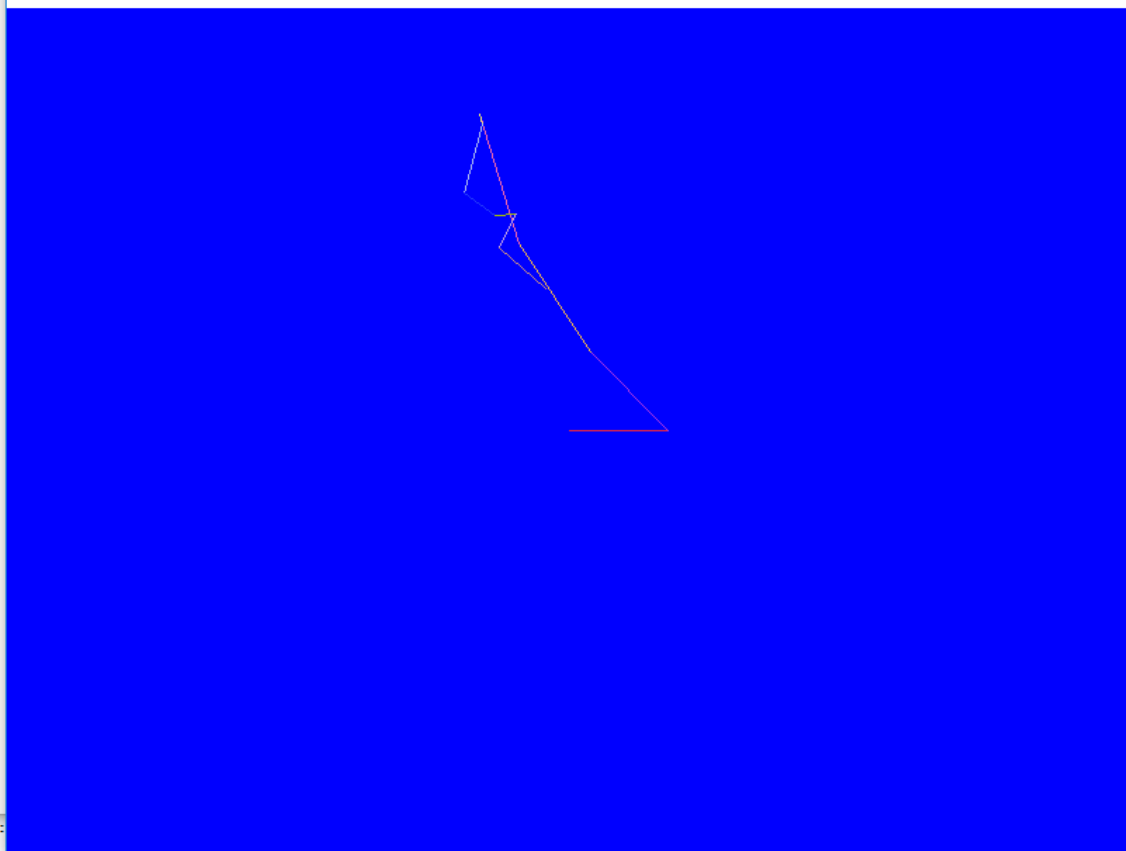
Chọn độ dài bước và
hướng đi ngẫu nhiên



Chọn màu ngẫu nhiên

```
void Painter::setRandomColor()
{
    Uint8 r = rand() % 256;
    Uint8 g = rand() % 256;
    Uint8 b = rand() % 256;
    SDL_Color color = { r, g, b };
    setColor(color);
}
```

```
void randomWalk(Painter& painter)
{
    ...
    for (int i = 0; i < STEPS; i++) {
        painter.setRandomColor();
    }
    ...
}
```

Executing:

C:\code\advprogram\lec10-sdl\.

Nội dung

- Thư viện SDL
 - Cài đặt, khởi tạo, sử dụng, giải phóng
- Xây dựng API vẽ
 - Lớp Painter
- **Vẽ hình bằng bút vẽ**
 - Đường thẳng, hình vuông, tam giác ...
 - **Phối hợp tạo thành *các hình tuyệt đẹp***
 - **Vẽ ảnh JPG, PNG**
- Vẽ hình fractal
 - Kỹ thuật đệ quy

Vẽ các hình quen thuộc

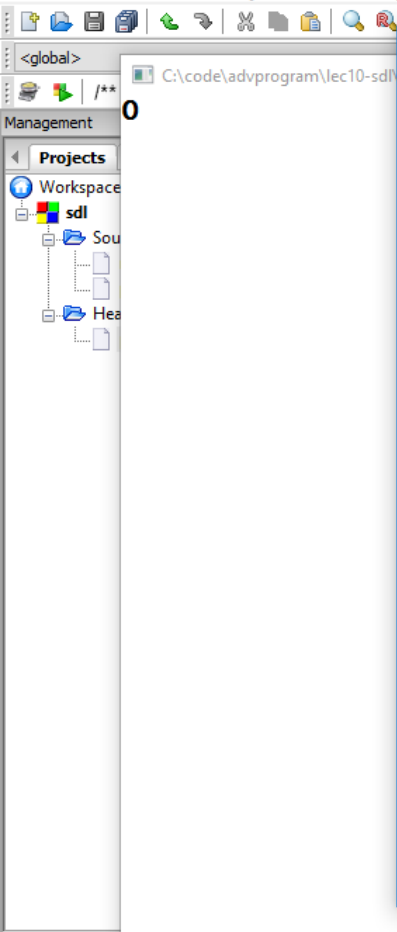
- Lấy tham số từ dòng lệnh
 - 0: hình vuông
 - 1: hình tam giác
 - 2: tô kín tam giác
 - 3: hình bát giác
 - 4: sao năm cánh
 - 5: sao sáu cánh
 - 6: nhím 8 gai
 - 7: sáu hình vuông
 - 8: hình tròn
 - 9: vòng tròn các hình tròn
 - 10: nhiều hình vuông
 - 11: nhiều đường kẻ
 - 12: hình thoi
 - 13: nhiều hình tròn lồng nhau
 - 14: bông tuyết tám cánh
 - 15: đi dạo (ngẫu nhiên)

Hình vuông

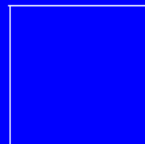
```
int figNumber = argc > 1 ? atoi(argv[1]) : 0;
switch (figNumber)
{
    /* Square */
    case 0:
        painter.setColor(WHITE_COLOR);
        for (int i = 0; i < 4; ++i)
        {
            painter.moveForward(100);
            painter.turnRight(90);
        }
        break;
```



Quay 90 độ 4 lần và
đi tới



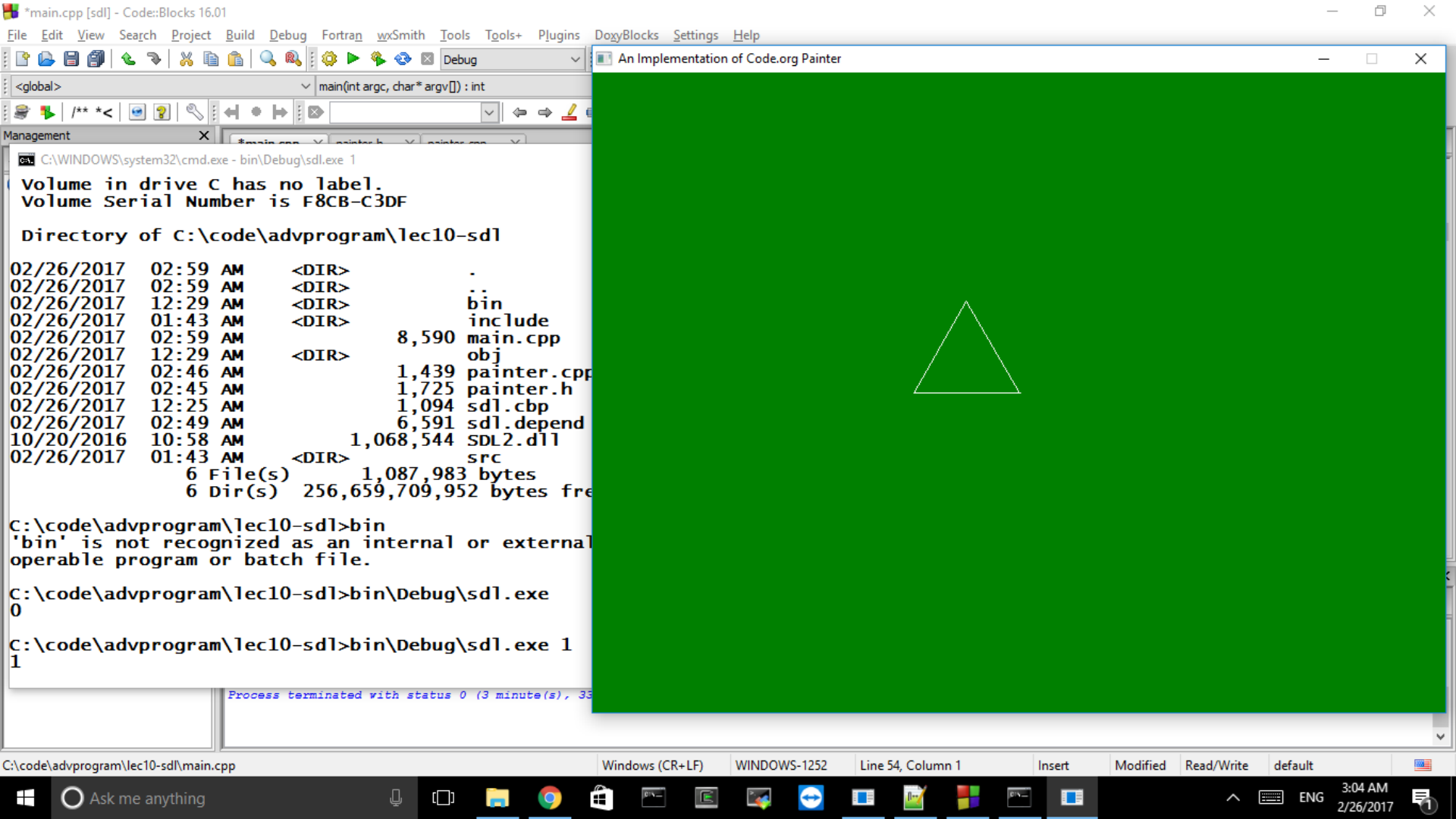
An Implementation of Code.org Painter



Hình tam giác

```
...  
    /* Triangle */  
    case 1:  
        painter.setColor(WHITE_COLOR);  
        painter.clearWithBgColor(GREEN_COLOR);  
        for (int i = 0; i < 3; ++i)  
        {  
            painter.turnLeft(120);  
            painter.moveForward(100);  
        }  
        break;
```

← Quay 120 độ 3 lần
và đi tới



Tô kín tam giác

```
...
/* Filled Triangle */
case 2:
{
    int curX = painter.getX();
    int curY = painter.getY();
    painter.setColor(WHITE_COLOR);
    painter.turnLeft(60);
    int size = 150;
```

Vẽ các tam giác có
kích thước nhỏ dần

```
for (int i = 0; i < size; ++i)
{
    for (int j = 0; j < 3; ++j)
    {
        painter.turnLeft(120);
        painter.moveForward(size - i);
    }
    painter.setPosition(curX, curY);
    painter.jumpBackward(i+1);
}
painter.setPosition(curX, curY);
break;
} // case 2
```




C:\WINDOWS\system32\cmd.exe - bin\Debug\sd1.exe 2

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 2

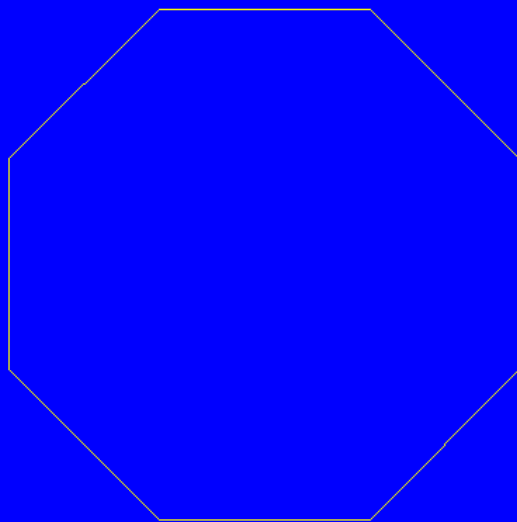
Process terminated with status 0 (0 minute(s), 0
0 error(s), 0 warning(s) (0 minute(s), 0 second(s)

Hình bát giác

```
...  
    /* Octagon */  
    case 3:  
        painter.setPosition(350, 500);  
        painter.setColor(YELLOW_COLOR);  
        for (int i = 0; i < 8; ++i)  
        {  
            painter.moveForward(150);  
            painter.turnLeft(45);  
        }  
        break;
```



Quay 45 độ 8 lần và
đi tới



Executing:

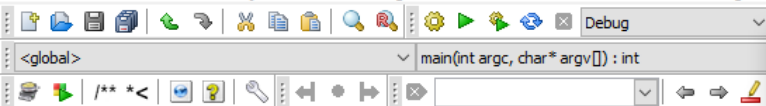
C:\code\advprogram\lec10-sdl\.

Sao năm cánh

```
...  
    /* Star of fives */  
    case 4:  
        painter.setPosition(350, 200);  
        painter.setColor(YELLOW_COLOR);  
        for (int i = 0; i < 5; ++i)  
        {  
            painter.moveForward(200);  
            painter.turnRight(144);  
        }  
        break;
```



Quay 144 độ 5 lần
và đi tới



Management X

C:\WINDOWS\system32\cmd.exe - bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 2

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

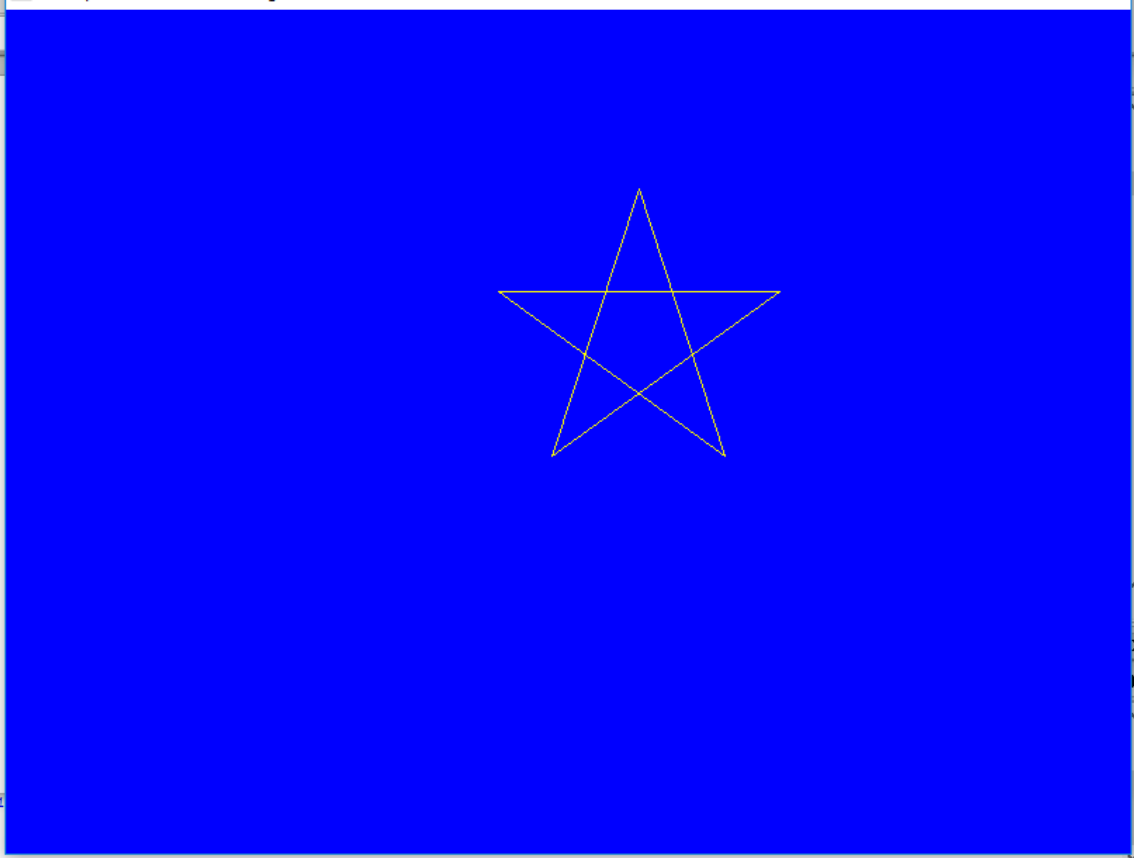
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

An Implementation of Code.org Painter



Sao sáu cánh

```
/* Star of David */  
case 5:  
    painter.setPosition(350, 400);  
    painter.setColor(YELLOW_COLOR);  
    painter.turnLeft(60);  
    for (int i = 0; i < 3; ++i)  
    {  
        painter.moveForward(150);  
        painter.turnLeft(120);  
    }
```

Vẽ 2 tam giác đều

```
painter.turnLeft(30);  
painter.jumpForward(  
    150 * 2 / 1.73205080757));  
// sqrt(3) = 1.73205080757  
painter.turnLeft(150);  
for (int i = 0; i < 3; ++i)  
{  
    painter.moveForward(150);  
    painter.turnLeft(120);  
}  
break;
```

<global>
main(int argc, char* argv[]) : int

Management

C:\WINDOWS\system32\cmd.exe - bin\Debug\sd1.exe 5

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 2

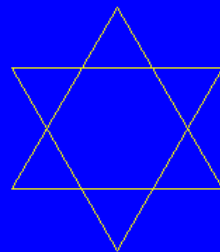
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 4

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 5

Process terminated with status 0 (0 minute(s), 1

An Implementation of Code.org Painter



Nhóm 8 gai

```
/* Eight lines crossing at center*/
```

```
case 6:
```

```
    painter.setColor(WHITE_COLOR);
```

```
    for (int i = 0; i < 8; ++i)
```

```
    {
```

```
        painter.moveForward(100);
```

```
        painter.moveBackward(100);
```

```
        painter.turnLeft(45);
```

```
    }
```

```
    break;
```

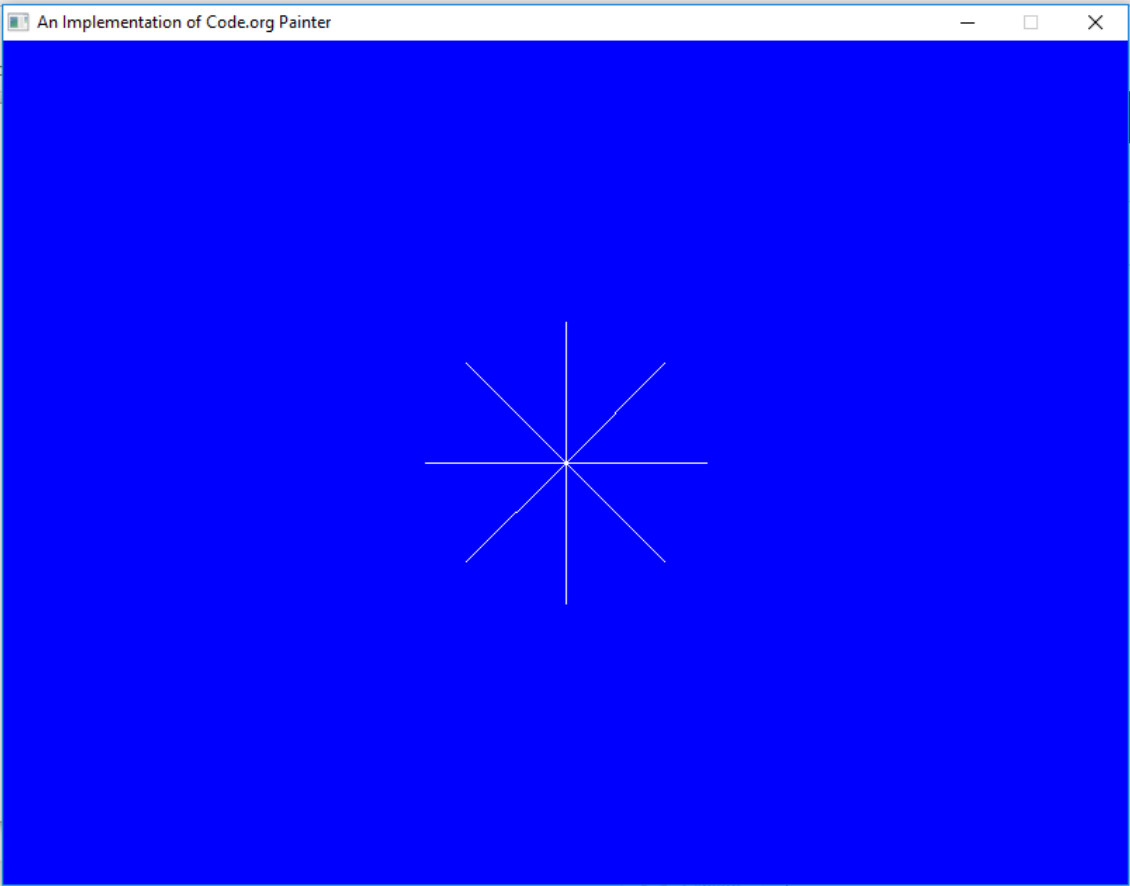


Đi tới đi lui 8 lần,
mỗi lần quay 45 độ

Lec10-SDL

File Edit View Insert Slide Format Arrange Tools Table Help All c

```
C:\WINDOWS\system32\cmd.exe - bin\Debug\SDL.exe 6  
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 2  
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 4  
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 4  
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 5  
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 5  
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 5  
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 5  
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 6
```



Sáu hình vuông

```
/* Six squares */  
case 7:  
    for (int i = 0; i < 6; ++i)  
    {  
        for (int j = 0; j < 4; ++j)  
        {  
            painter.moveForward(100);  
            painter.turnRight(90);  
        }  
        painter.turnLeft(60);  
    }  
    break;
```



Vẽ một hình vuông

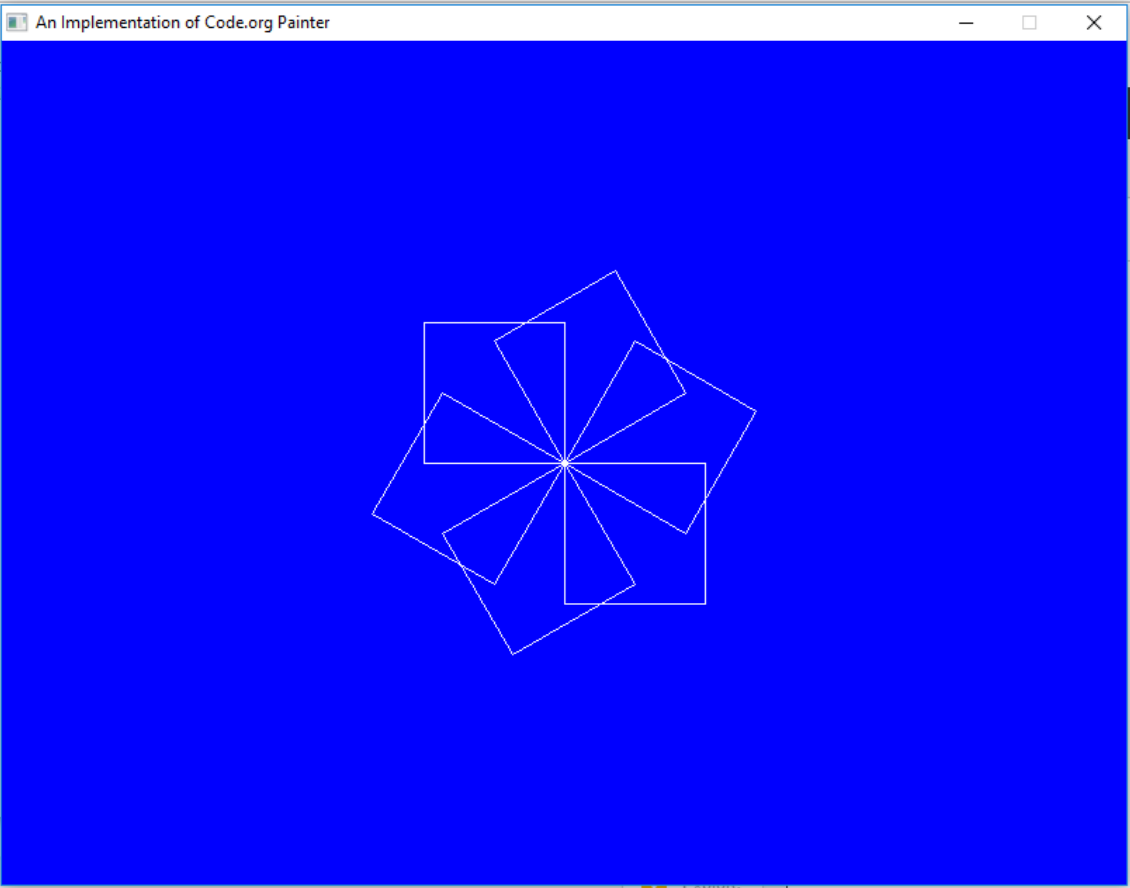


Quay 60 độ, vẽ tiếp
hình vuông tất cả 6
lần

Lec10-SDL

File Edit View Insert Slide Format Arrange Tools Table Help All

```
C:\WINDOWS\system32\cmd.exe - bin\Debug\SDL.exe 7  
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 2  
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 4  
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 4  
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 5  
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 5  
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 5  
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 5  
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 6  
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 7
```



Hình tròn

- Giải thuật vẽ hình tròn
 - Thuật toán điểm giữa
 - https://en.wikipedia.org/wiki/Midpoint_circle_algorithm

Hình tròn

```
void Painter::createCircle(float radius)
{
    double rad = (angle / 180) * M_PI;
    int centerX = x + cos(rad) * radius;
    int centerY = y - sin(rad) * radius;

    int dx = radius;
    int dy = 0;
    int err = 0;
```

Hình

```
void Painter::c
{
    double rad
    int centerX
    int centerY

    int dx = ra
    int dy = 0;
    int err = 0
```

```
while (dx >= dy) {
    SDL_RenderDrawPoint(renderer, centerX + dx, centerY + dy);
    SDL_RenderDrawPoint(renderer, centerX + dy, centerY + dx);
    SDL_RenderDrawPoint(renderer, centerX - dy, centerY + dx);
    SDL_RenderDrawPoint(renderer, centerX - dx, centerY + dy);
    SDL_RenderDrawPoint(renderer, centerX - dx, centerY - dy);
    SDL_RenderDrawPoint(renderer, centerX - dy, centerY - dx);
    SDL_RenderDrawPoint(renderer, centerX + dy, centerY - dx);
    SDL_RenderDrawPoint(renderer, centerX + dx, centerY - dy);

    if (err <= 0) {
        dy += 1;
        err += 2*dy + 1;
    }
    if (err > 0) {
        dx -= 1;
        err -= 2*dx + 1;
    }
} // while
} // createCircle()
```

- Từ 1 điểm suy ra 7 điểm khác bằng tính đối xứng của hình tròn
- Tính điểm kế tiếp (tăng *dy* hoặc giảm *dx* 1 điểm ảnh) bằng cách kiểm tra *err*

Chi tiết thuật toán

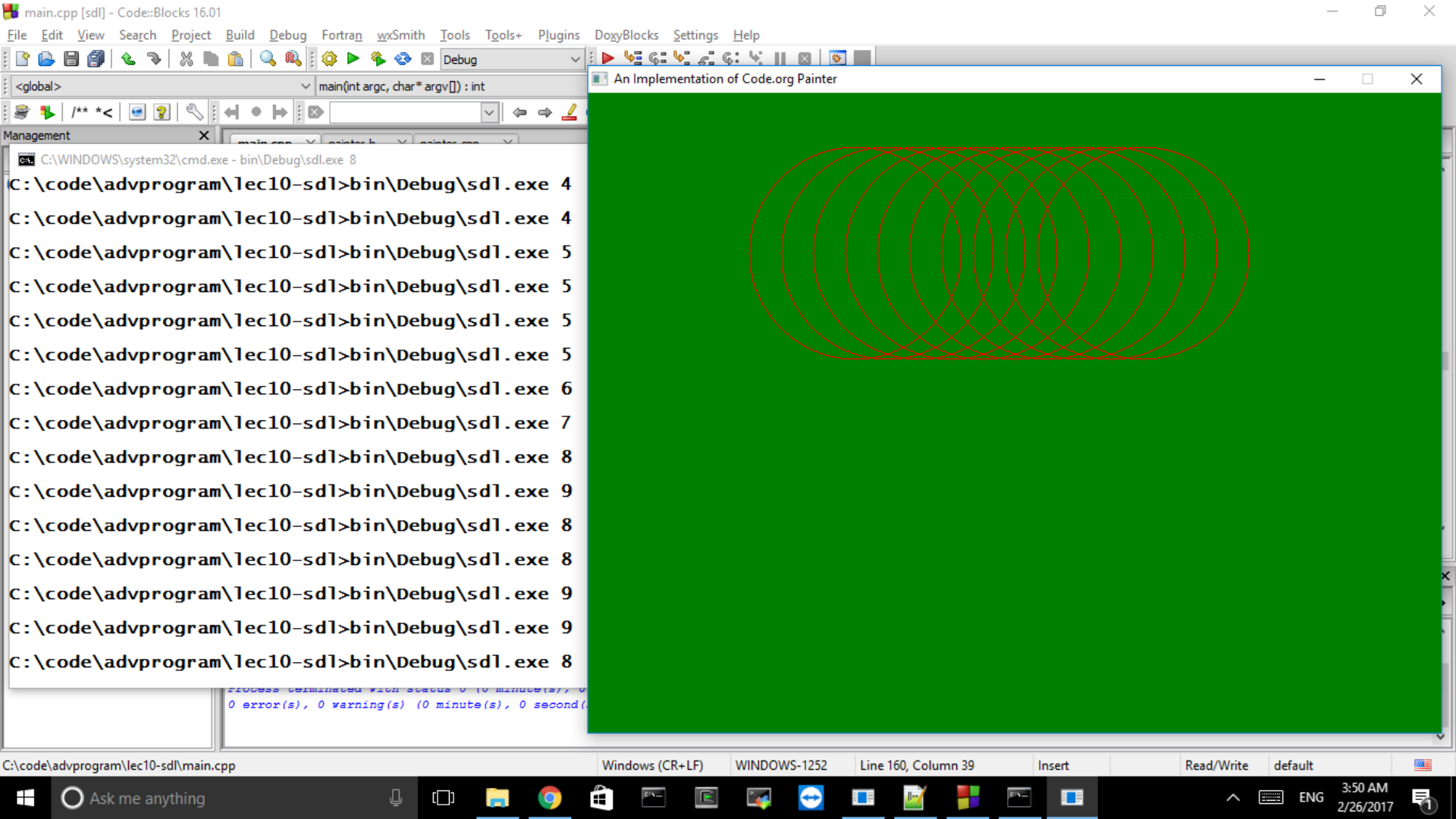
https://en.wikipedia.org/wiki/Midpoint_circle_algorithm

Hình tròn

```
/* Circles in line */  
case 8:  
    painter.clearWithBgColor(BLACK_COLOR);  
    painter.setColor(RED_COLOR);  
    painter.setPosition(150, 150);  
    for (int i = 0; i < 10; ++i)  
    {  
        painter.createCircle(100);  
        painter.jumpForward(30);  
    }  
    break;
```



Vẽ một hình tròn,
nhấc bút lên, đi tới
30 điểm ảnh



Vòng tròn các hình tròn

```
/* Circles in circle */
```

```
case 9:
```

```
    painter.setPosition(350, 150);
```

```
    painter.clearWithBgColor(BLACK_COLOR);
```

```
    for (int i = 0; i < 20; ++i)
```

```
    {
```

```
        painter.setRandomColor();
```

```
        painter.createCircle(100);
```

```
        painter.jumpForward(1);
```

```
        painter.createCircle(100);
```

```
        painter.jumpForward(50);
```

```
        painter.turnRight(18);
```

```
    }
```

```
    break;
```

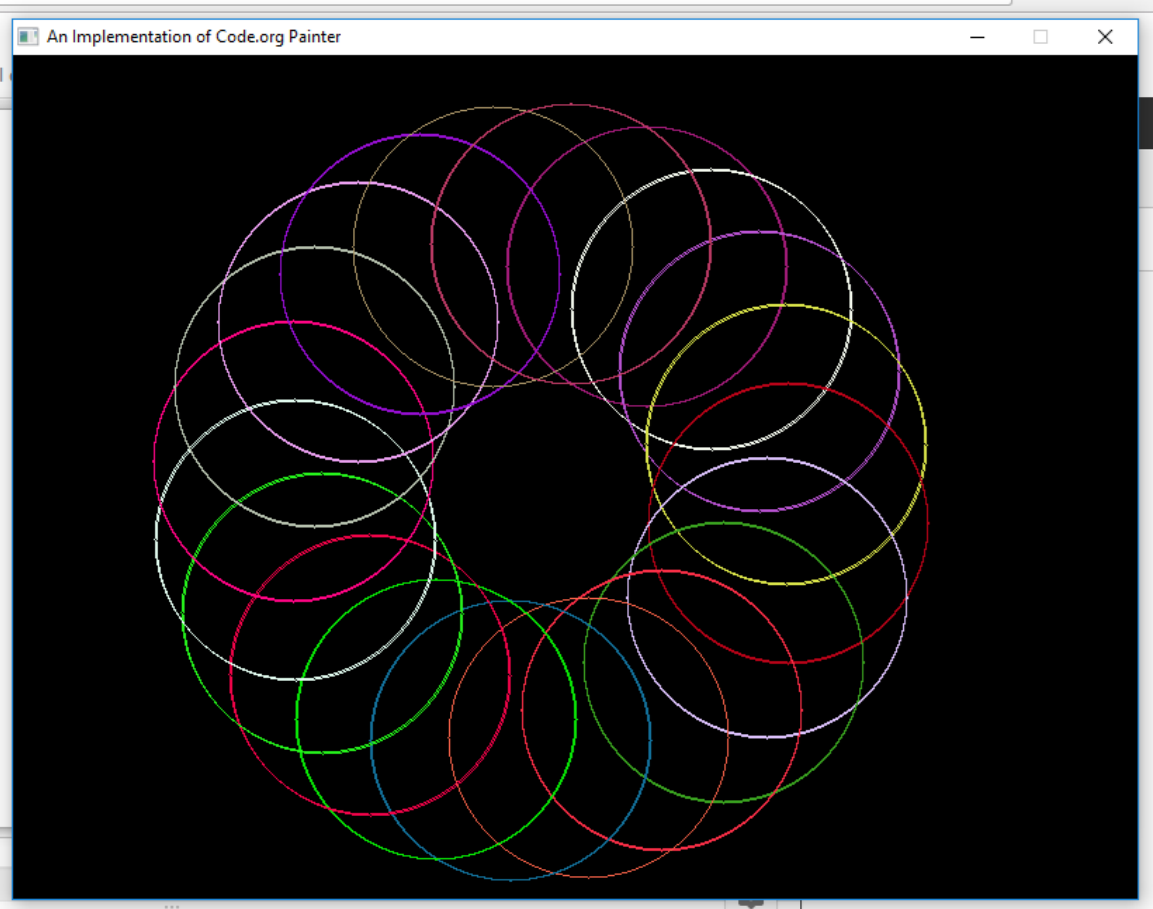


Vẽ 2 hình tròn liền
nhau để tạo cảm
giác cạnh dày hơn



Đi tới 50 điểm ảnh,
xoay hướng 18 độ
18 x 20 lần = 360 độ

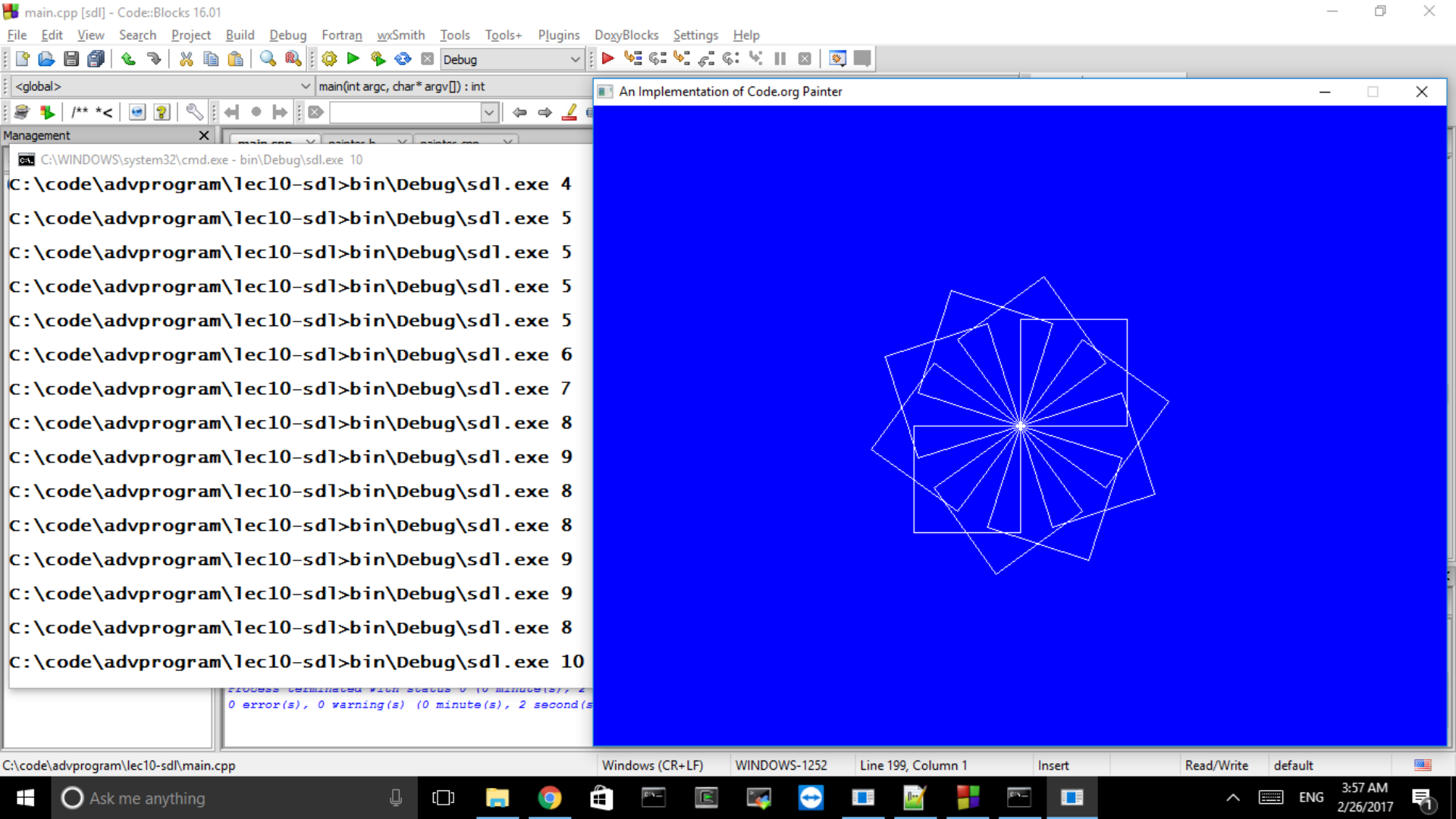
```
C:\WINDOWS\system32\cmd.exe - bin\Debug\sdl.exe 9
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 2
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 4
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 4
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 5
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 5
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 5
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 5
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 6
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 7
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 8
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 9
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 8
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 8
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 9
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 9
```



Nhiều hình vuông

```
void Painter::createSquare(float size)
{
    for (int i = 0; i < 4; ++i) {
        moveForward(size);
        turnLeft(90);
    }
}
```

```
/*    Ten squares in circle */
case 10:
    painter.setColor(WHITE_COLOR);
    for (int i = 0; i < 10; ++i)
    {
        //painter.randomColor();
        painter.createSquare(100);
        painter.turnRight(36);
    }
    break;
```



Nhím nhiều màu

```
/* Multitude lines in cicles */  
case 11:  
    for (int i = 0; i < 90; ++i)  
    {  
        painter.setRandomColor();  
        painter.moveForward(150);  
        painter.jumpBackward(150);  
        painter.turnRight(4);  
    }  
    break;
```

```
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 5
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 5
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 5
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 5
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 6
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 7
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 8
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 9
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 8
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 8
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 9
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 9
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 8
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 10
C:\code\advprogram\lec10-sdl>bin\Debug\sdl.exe 11
```

```
PROCESS TERMINATED WITH STATUS 0 (0 minute(s), 2 second(s))
0 error(s), 0 warning(s) (0 minute(s), 2 second(s))
```

Windows (CR+LF)

Line 212, Column 1

Read/Write



Hình thoi

```
void Painter::createParallelogram(  
    float size)  
{  
    for (int i = 0; i < 2; ++i) {  
        moveForward(size);  
        turnLeft(60);  
        moveForward(size);  
        turnLeft(120);  
    }  
}
```

```
/* Pattern of Ten parallelograms */  
case 12:  
    painter.setColor(WHITE_COLOR);  
    for (int i = 0; i < 10; ++i)  
    {  
        painter.createParallelogram(100);  
        painter.turnRight(36);  
    }  
    break;
```

main.cpp [sd1] - Code::Blocks 16.01

FileEditViewSearchProjectBuildDebugFortranwxSmithToolsTools+PluginsDoxyBlocksSettingsHelp

Debug

<global>main(int argc, char* argv[]) : int

Management

C:\WINDOWS\system32\cmd.exe - bin\Debug\sd1.exe 12

C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 5
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 5
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 5
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 6
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 7
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 8
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 9
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 8
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 8
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 9
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 9
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 8
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 10
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 11
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 12

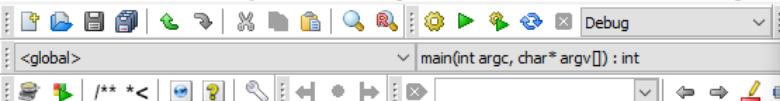
An Implementation of Code.org Painter

Windows (CR+LF)WINDOWS-1252Line 220, Column 1InsertRead/Writedefault

Windows taskbar with icons for Start, Search, File Explorer, Chrome, and others. System tray shows 4:03 AM on 2/26/2017.

Nhiều hình tròn lồng nhau

```
/* Five and five circles */  
/**  
case 13:  
    painter.setColor(WHITE_COLOR);  
    painter.clearWithBgColor(GREEN_COLOR);  
    for (int i = 0; i < 5; ++i)  
    {  
        painter.createCircle(100);  
        painter.createCircle(50);  
        painter.turnRight(72);  
    }  
    break;
```



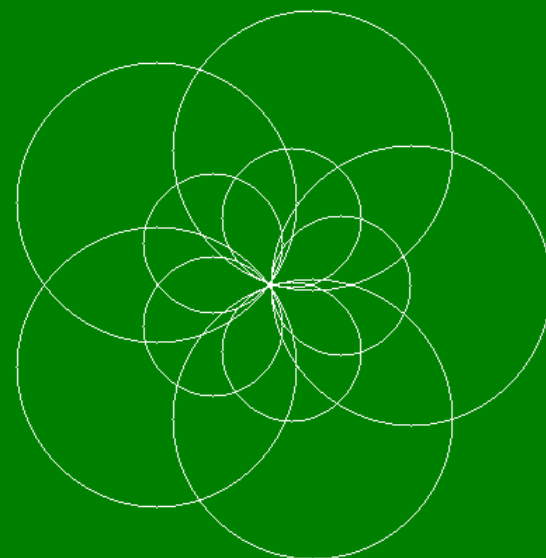
Management X

C:\WINDOWS\system32\cmd.exe - bin\Debug\sd1.exe 13

```
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 5
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 5
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 6
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 7
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 8
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 9
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 8
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 8
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 9
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 9
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 8
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 10
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 11
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 12
C:\code\advprogram\lec10-sd1>bin\Debug\sd1.exe 13
```

```
Process terminated with status 0 (0 minute(s), 1
0 error(s), 0 warning(s) (0 minute(s), 1 second(s)
```

An Implementation of Code.org Painter



Windows (CR+LF)

WINDOWS-1252

Line 227, Column 6

Insert

Read/Write

default



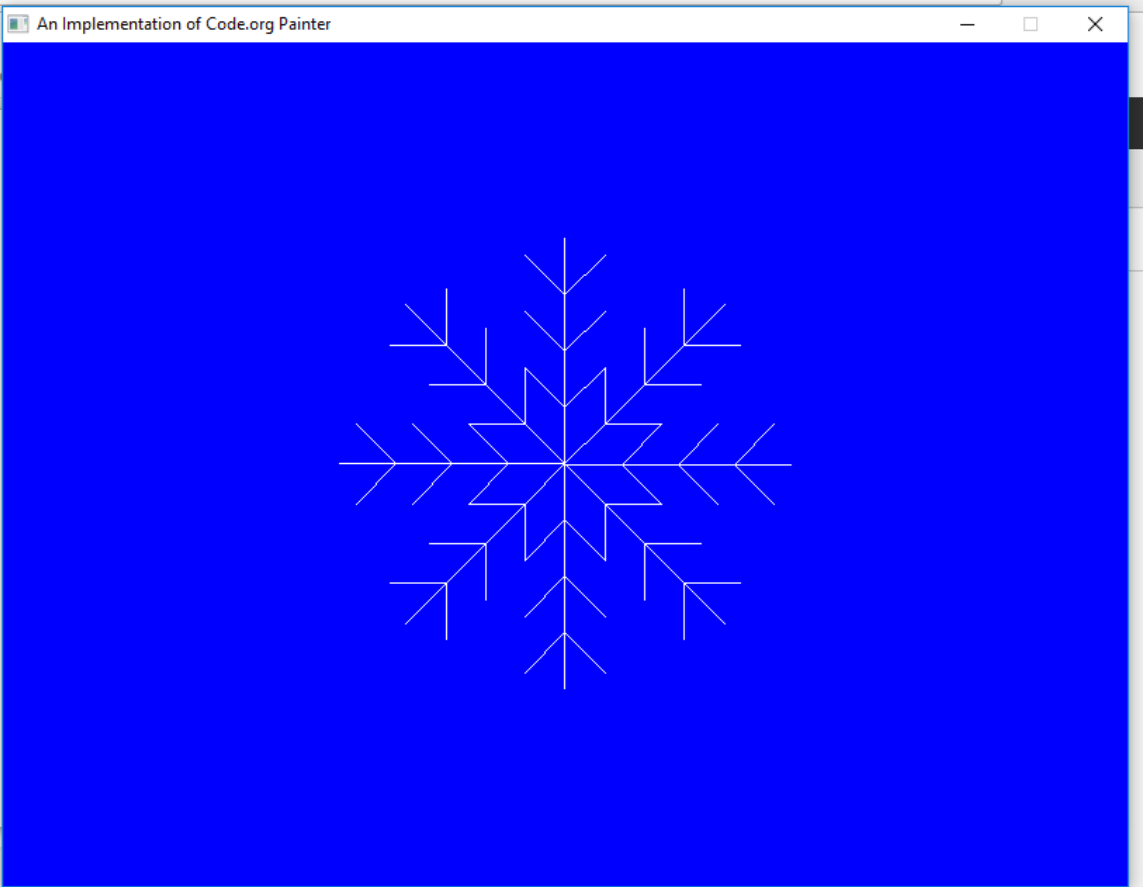
Bông tuyết 8 cánh

```
/* Snow flake*/  
case 14:  
    painter.setColor(WHITE_COLOR);  
    for (int i = 0; i < 8; ++i)  
    {  
        int size = 40;  
        painter.moveForward(size);  
        // code vẽ một cánh tuyết  
        painter.jumpBackward(4*size);  
        painter.turnRight(45);  
    }  
    break;
```

<https://github.com/tqlong/advprogram/archive/3677695699840c851d6e22972eb4ff7353540e00.zip>

```
for (int j = 0; j < 3; ++j)  
{  
    painter.turnLeft(45);  
    painter.moveForward(size);  
    painter.jumpBackward(size);  
    painter.turnRight(90);  
    painter.moveForward(size);  
    painter.jumpBackward(size);  
    painter.turnLeft(45);  
    painter.moveForward(size);  
}
```

```
C:\WINDOWS\system32\cmd.exe - bin\Debug\SDL.exe 14
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 5
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 6
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 7
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 8
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 9
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 8
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 8
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 9
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 9
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 8
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 10
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 11
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 12
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 13
C:\code\advprogram\lec10-sdl>bin\Debug\SDL.exe 14
```



Vẽ ảnh JPG, PNG

- Thêm thư viện **SDL2_image**
 - https://www.libsdl.org/projects/SDL_image/
 - **SDL2_image.dll**, **lib-jpeg9.dll**, **libpng16-16.dll**, **libtiff-5.dll** vào thư mục mã nguồn
 - http://lazyfoo.net/tutorials/SDL/07_texture_loading_and_rendering/index.php
- Đọc ảnh vào **SDL_Surface** (bitmap)
- Chuyển sang **SDL_Texture** (phụ thuộc driver)
- Dùng **SDL_RenderCopy()** vẽ **SDL_Texture**
- Giải phóng **SDL_Texture**

Chuẩn bị SDL_Texture

```
SDL_Texture* Painter::loadTexture( string path )
{
    SDL_Texture* newTexture = NULL;
    SDL_Surface* loadedSurface = IMG_Load( path.c_str() );
    if ( loadedSurface == NULL )
        cout << "Unable to load image " << path << " SDL_image Error: "
              << IMG_GetError() << endl;
    else {
        newTexture = SDL_CreateTextureFromSurface( renderer, loadedSurface );
        if( newTexture == NULL )
            cout << "Unable to create texture from " << path << " SDL Error: "
                  << SDL_GetError() << endl;
        SDL_FreeSurface( loadedSurface );
    }
    return newTexture;
}
```

Vẽ ảnh

```
bool Painter::createImage( SDL_Texture* texture )
{
    if( texture == NULL ) return false;
    SDL_RenderCopy( renderer, texture, NULL, NULL );
    return true;
}
```

Giải phóng Texture

```
SDL_Texture* texture = NULL;
...

SDL_DestroyTexture(texture);
quitSDL();
...
```

```
/* Load image */
case 16:
{
    if (argc <= 2) {
        cout << "Please provide image file path" << endl;
        break;
    }
    texture = painter.loadTexture(argv[2]);
    painter.createImage(texture);
}
break;
}
```

main.cpp [sdl] - Code::Blocks 16.01

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Debug <global> SDL_main(int argc, char* argv[]): int

Management

Projects Symbols Files FSymbols

Workspace

- sdl
 - Sources
 - main.cpp
 - painter.cpp
 - Headers
 - painter.h

main.cpp

```
256 painter.moveForward(size);
257 painter.jump
258 painter.turn
259 painter.move
260
261 painter.jumpBack
262 painter.turnRigh
263
264 break;
265
266
```

painter.h

An Implementation of Code.org Painter

C:\WINDOWS\system32\cmd.exe - bin\Debug\sdl.exe 16 test.jpg

C:\Users\doe\Documents\advprogram\lec10-sdl>bin\Debug\sdl.exe 16 test.jpg
Unable to load image test.jpg SDL_image Error: Failed loading libjpeg-9.dll


^C
C:\Users\doe\Documents\advprogram\lec10-sdl>
C:\Users\doe\Documents\advprogram\lec10-sdl>bin\Debug\sdl.exe 16 test.jpg

277
278 SDL_Texture* tex
279 painter.clearWit
280 painter.createIm
281
282 break;
283
284
285 SDL_RenderPresent(render
286

Logs & others

Code::Blocks Search results

----- Build: Debug in sdl (compiler:
Target is up to date.
Nothing to be done (all items are up-to-date).



GSMarena
www.gsmarena.com

C:\Users\doe\Documents\advprogram\lec10-sdl\main.cpp Windows (CR+LF) WINDOWS-1252 Line 273, Column 10 Insert Read/Write default

1:59 PM
2/26/2017

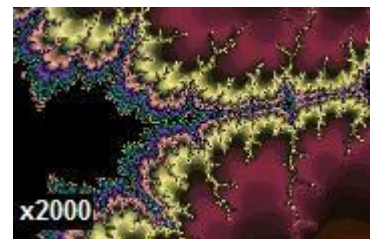
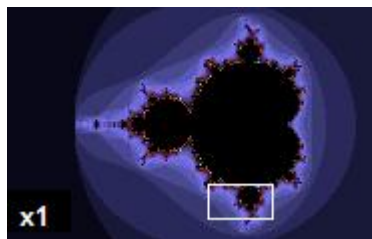
Nội dung

- Thư viện SDL
 - cài đặt, khởi tạo, sử dụng, giải phóng
- Xây dựng API vẽ
 - Lớp Painter
- Vẽ hình bằng bút vẽ
 - Đường thẳng, hình vuông, tam giác ...
 - Phối hợp tạo thành *các hình tuyệt đẹp*
 - Vẽ ảnh JPG, PNG
- **Vẽ hình fractal**
 - **Kỹ thuật đệ quy**

Fractal

<https://en.wikipedia.org/wiki/Fractal>

<https://www.youtube.com/watch?v=0jGaio87u3A>



Tập hợp Mandelbrot

- Hình fractal
 - Càng phóng to, càng *nhiều chi tiết đẹp*
 - Toán học chạm đến nghệ thuật

Tập hợp Mandelbrot

- Với mỗi số phức $c = x_0 + i y_0$ (một điểm trên mặt phẳng), lập dãy số phức $z_n = x_n + i y_n$
 - $z_0 = 0 = 0 + i 0$
 - $z_{n+1} = z_n^2 + c$
- Nếu dãy phân kì thì c không thuộc tập hợp
 - Tô màu với số n mà $|z_n|^2 = x_n^2 + y_n^2 \geq 4$ (phân kì)
 - Mỗi số n ứng với một màu trong bảng màu (*palette*)
- Nếu $n > \text{MAX_ITERATION}$, coi c thuộc tập hợp (*tô màu đen*)

Thuật toán điểm thoát (escape)

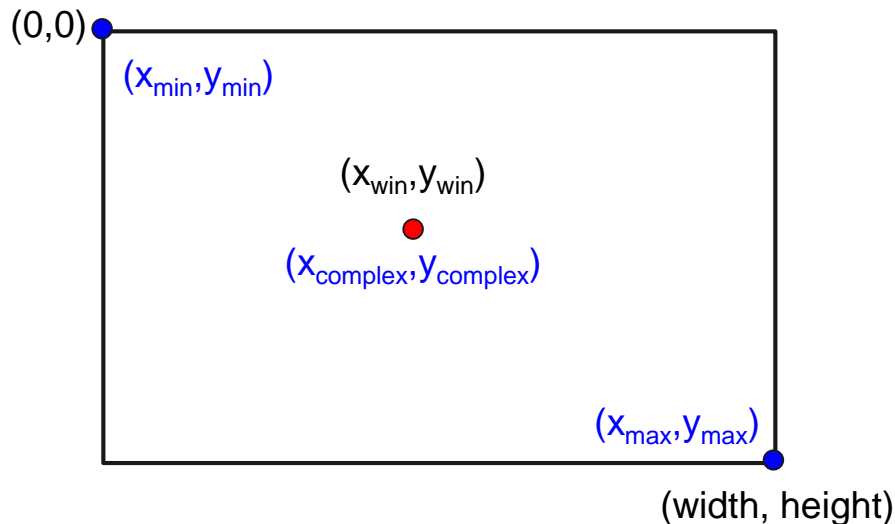
For each pixel (Px, Py) on the screen, do:

```
{  
  x0 = scaled x coordinate of pixel (scaled to lie in the Mandelbrot X scale (-2, 1.5))  
  y0 = scaled y coordinate of pixel (scaled to lie in the Mandelbrot Y scale (-2, 1.5))  
  x = y = 0.0  
  iteration = 0  
  max_iteration = 1000  
  while (x*x + y*y < 2*2 AND iteration < max_iteration) {  
    xtemp = x*x - y*y + x0  
    y = 2*x*y + y0  
    x = xtemp  
    iteration = iteration + 1  
  }  
  color = palette[iteration]  
  plot(Px, Py, color)  
}
```

https://en.wikipedia.org/wiki/Mandelbrot_set#Escape_time_algorithm

Chuyển đổi tọa độ

Tọa độ hiển thị \Leftrightarrow tọa độ trên mặt phẳng phức



Ánh xạ giữa 2 hệ tọa độ

- Tọa độ hiển thị (màn hình)
- Tọa độ mặt phẳng phức

$$x_{complex} = \frac{x_{win}}{width} (x_{max} - x_{min}) + x_{min}$$

$$y_{complex} = \frac{y_{win}}{height} (y_{max} - y_{min}) + y_{min}$$

```
void drawMandelbrot(Painter& painter, float xmin = -2, float ymin = -1.5, float xmax = 2, float ymax = 1.5);  
...  
case 17:  
    drawMandelbrot(painter);  
    break;
```

```

void drawMandelbrot(Painter& painter, float xmin, float ymin, float xmax, float ymax)
{
    int width = painter.getWidth(), height = painter.getHeight();
    const int MAX_ITERATION = 1000;
    for (int px = 0; px < width; px++) {
        for (int py = 0; py < painter.getHeight(); py++) {
            float x0 = (float)px / width * (xmax-xmin) + xmin, x = 0;
            float y0 = (float)py / height * (ymax-ymin) + ymin, y = 0;
            int iteration = 0;
            while (x*x+y*y < 2 && iteration < MAX_ITERATION) {
                float xtemp = x*x-y*y+x0;
                y = 2*x*y+y0;
                x = xtemp;
                iteration++;
            }
            SDL_Color color = iteration < MAX_ITERATION ?
                PALLETTE[iteration % PALLETTE_COUNT] : BLACK_COLOR;
            painter.setColor(color);
            painter.drawPoint(px, py);
        }
    }
}

```

Chuyển đổi
tọa độ từ
(px,py) qua
(x0,y0)

Tính điểm
thoát của
c = (x0,y0)

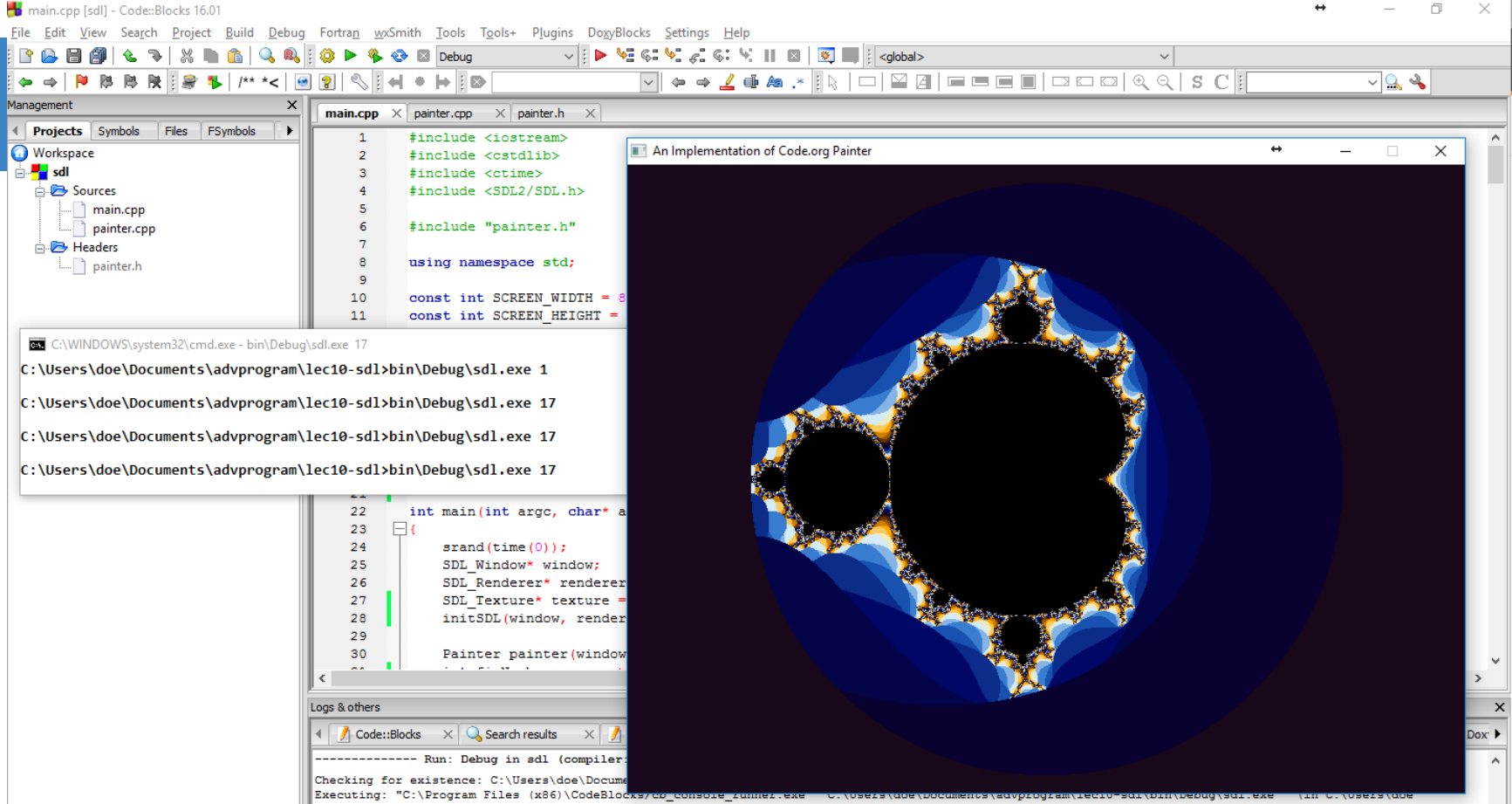
Kiểm tra c
có nằm
trong tập
Mandelbrot

Sử dụng
SDL_RenderDrawPoint(renderer, px, py)
cài đặt hàm này

Bảng màu

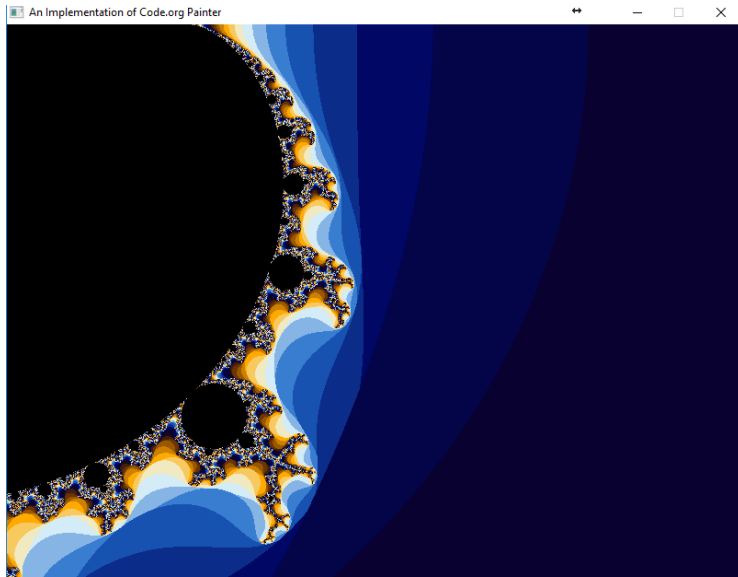
<http://stackoverflow.com/questions/16500656/which-color-gradient-is-used-to-color-mandelbrot-in-wikipedia>

```
const SDL_Color PALLETTE[] = {
    SDL_Color({66, 30, 15}), SDL_Color({25, 7, 26}),
    SDL_Color({9, 1, 47}), SDL_Color({4, 4, 73}),
    SDL_Color({0, 7, 100}), SDL_Color({12, 44, 138}),
    SDL_Color({24, 82, 177}), SDL_Color({57, 125, 209}),
    SDL_Color({134, 181, 229}), SDL_Color({211, 236, 248}),
    SDL_Color({241, 233, 191}), SDL_Color({248, 201, 95}),
    SDL_Color({255, 170, 0}), SDL_Color({204, 128, 0}),
    SDL_Color({153, 87, 0}), SDL_Color({106, 52, 3})
};
const int PALETTE_COUNT = sizeof(PALLETTE) / sizeof(SDL_Color);
```

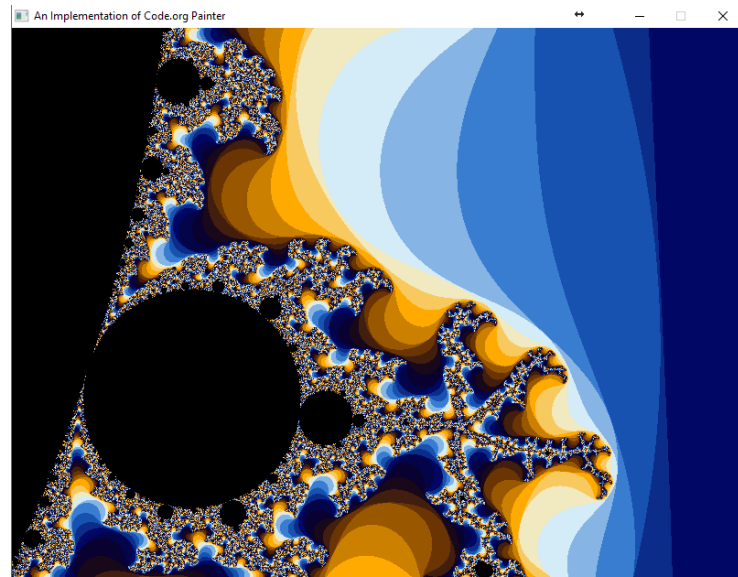


<https://github.com/tqlong/advprogram/archive/9d8e1a0d5aed0f94e1095d89813cf0b2ee99bb21.zip>

Zoom in



$(0, 0, 2*0.5, 1.5*0.5)$



$(2*0.17, 1.5*0.17, 2*0.25, 1.5*0.25)$

Fractal đệ quy

- Khái niệm đệ quy trong toán học / lập trình
 - Một đối tượng (có tham số) được định nghĩa thông qua chính nó với tham số khác (thường nhỏ hơn)
 - Tổng các số tự nhiên từ 1 đến n: $s(n) = n + s(n-1)$
 - Giai thừa: $\text{factorial}(n) = n * \text{factorial}(n-1)$
 - Fibonacci: $\text{fibonacci}(n) = \text{fibonacci}(n-1) + \text{fibonacci}(n-2)$
- Hình fractal đệ quy
 - <http://natureofcode.com/book/chapter-8-fractals/>
 - Self-similarity: ***each part is a “reduced-size copy of the whole.”*** (Mandelbrot)

Tính giai thừa

```
int factorial(int n) {  
    if (n == 1) {  
        return 1;  
    } else {  
        return n * factorial(n-1);  
    }  
}
```

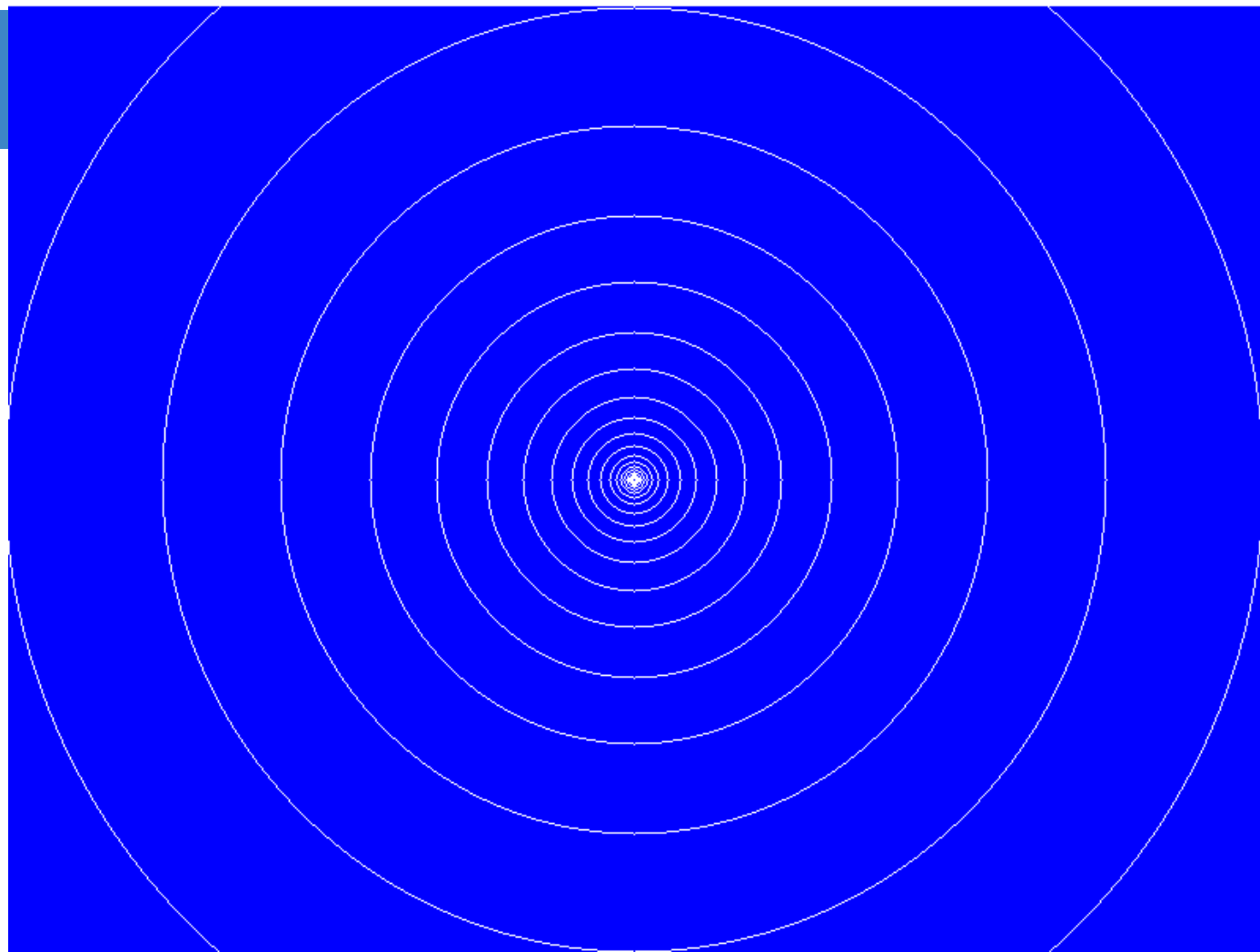


Điều kiện
dừng đệ quy

Vẽ hình tròn lồng nhau

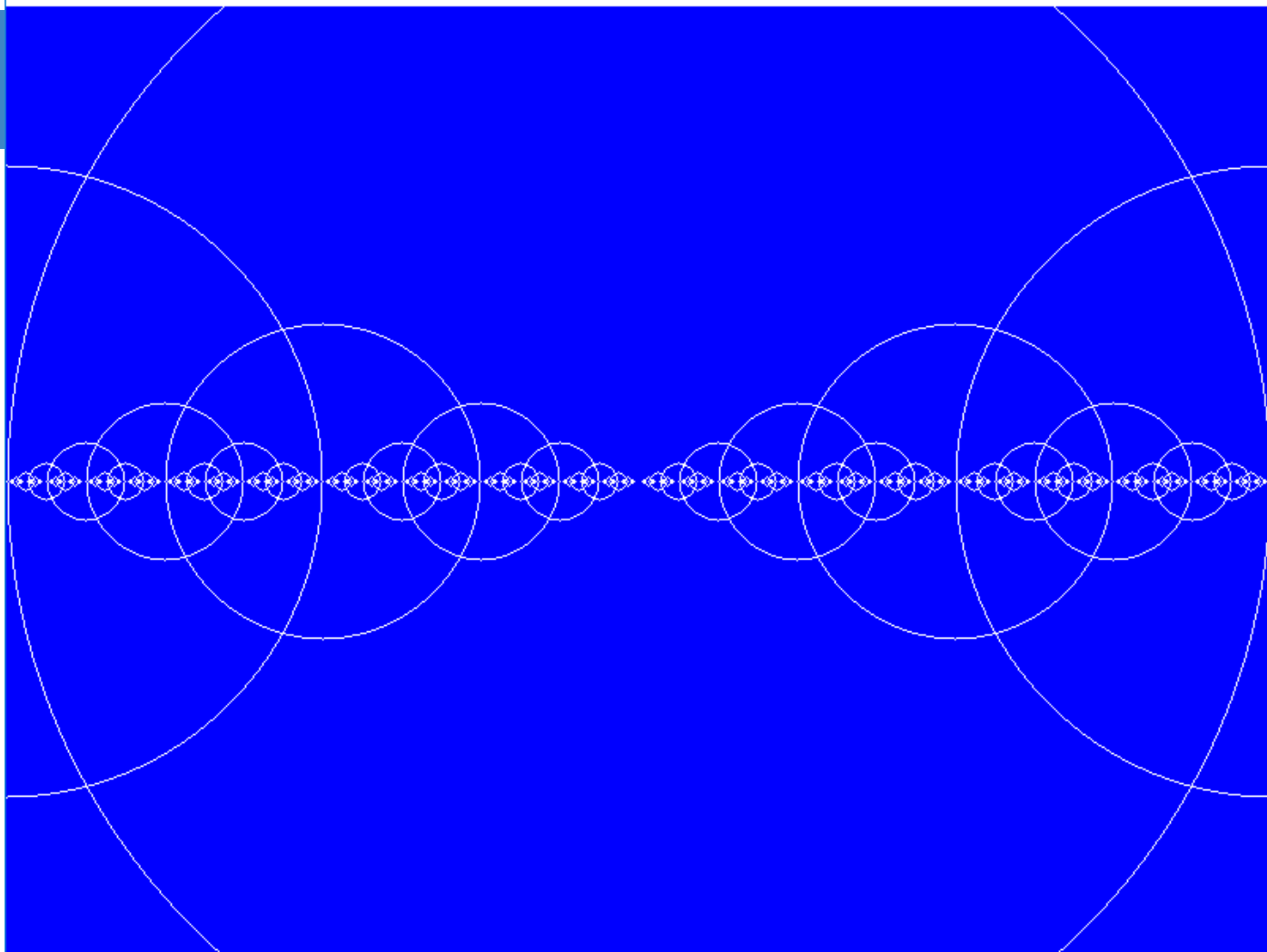
```
void drawRecursiveCircle(Painter& painter, float radius = 400);  
...  
    case 18:  
        painter.jumpBackward(400);  
        drawRecursiveCircle(painter, 400);  
        break;  
...  
void drawRecursiveCircle(Painter& painter, float radius)  
{  
    painter.createCircle(radius);  
    if(radius > 2) {  
        painter.jumpForward(radius*0.25f);  
        drawRecursiveCircle(painter, radius*0.75f);  
        painter.jumpBackward(radius*0.25f);  
    }  
}
```

Điều kiện đệ quy (bán kính lớn hơn 2)



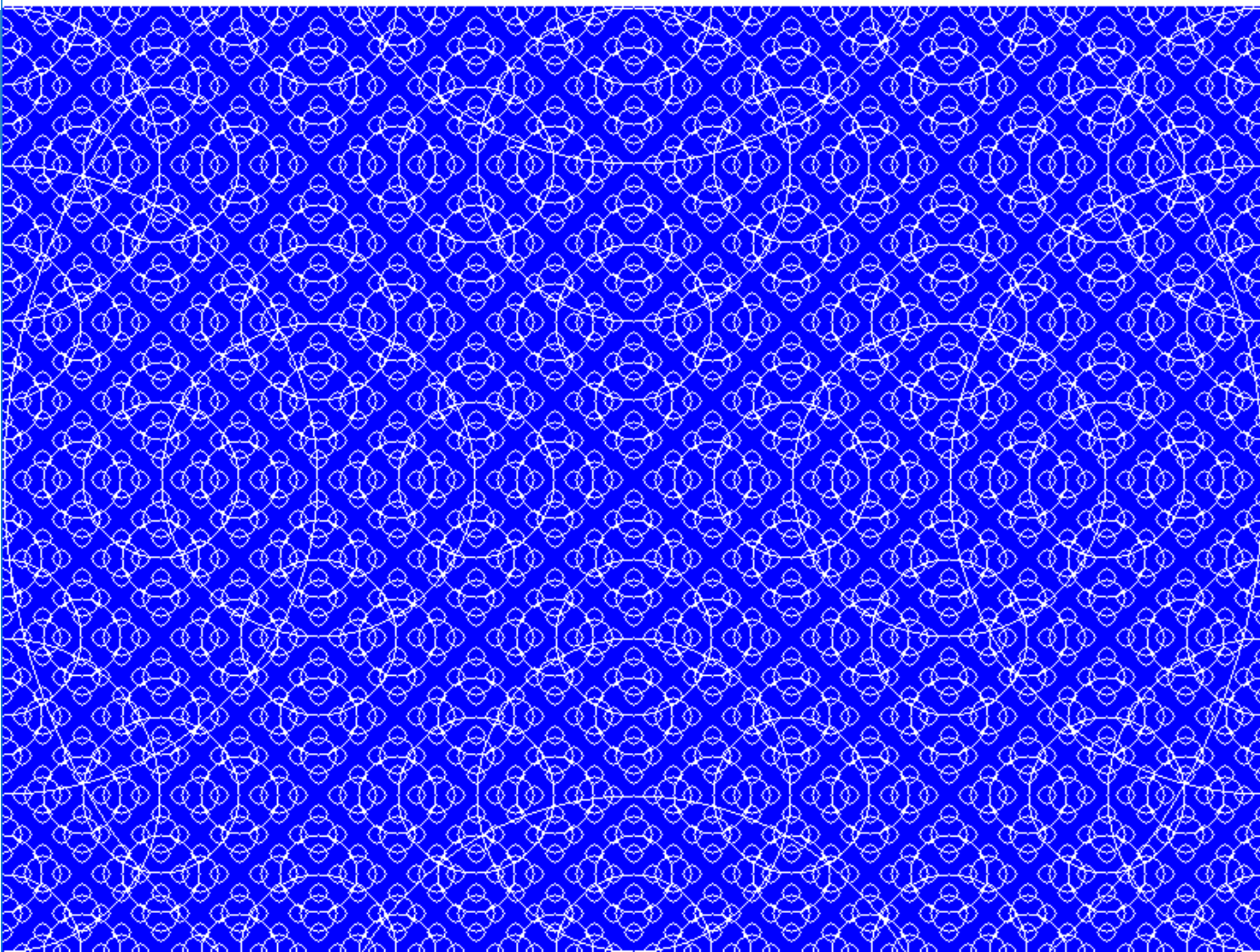
Vẽ 2 hình tròn bên trái, phải

```
void drawRecursiveCircle2(Painter& painter, float radius)
{
    painter.createCircle(radius);
    if(radius > 2) {
        painter.jumpBackward(radius / 2);
        drawRecursiveCircle2(painter, radius / 2);
        painter.jumpForward(radius * 2);
        drawRecursiveCircle2(painter, radius / 2);
        painter.jumpBackward(radius*3/2);
    }
}
```



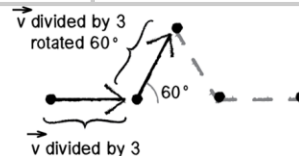
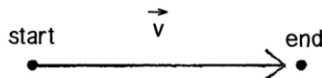
Thêm 2 hình tròn trên, dưới

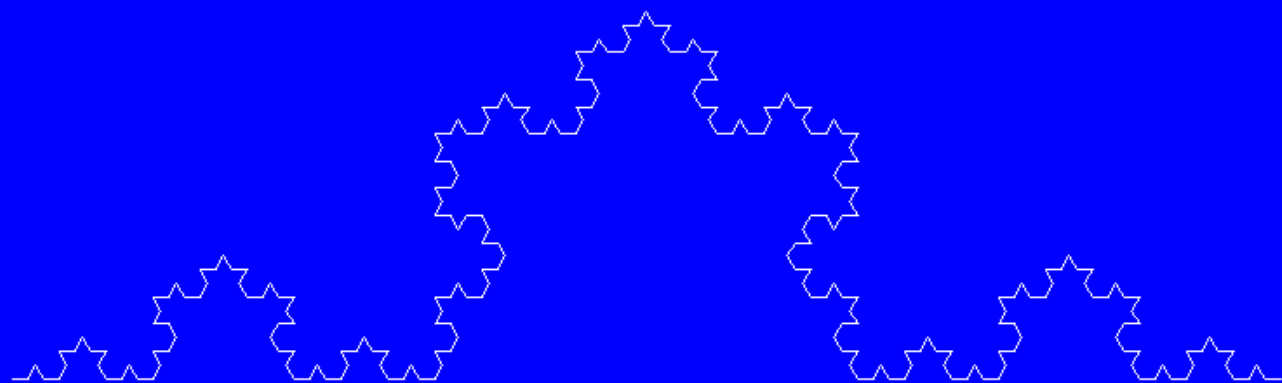
```
void drawRecursiveCircle4(Painter& painter, float radius)
{
    painter.createCircle(radius);
    if(radius > 8) {
        float x = painter.getX(), y = painter.getY();
        painter.setPosition(x-radius/2,y);
        drawRecursiveCircle4(painter, radius / 2);
        painter.setPosition(x+radius/2,y-radius);
        drawRecursiveCircle4(painter, radius / 2);
        painter.setPosition(x+radius*3/2,y);
        drawRecursiveCircle4(painter, radius / 2);
        painter.setPosition(x+radius/2,y+radius);
        drawRecursiveCircle4(painter, radius / 2);
        painter.setPosition(x,y);
    }
}
```

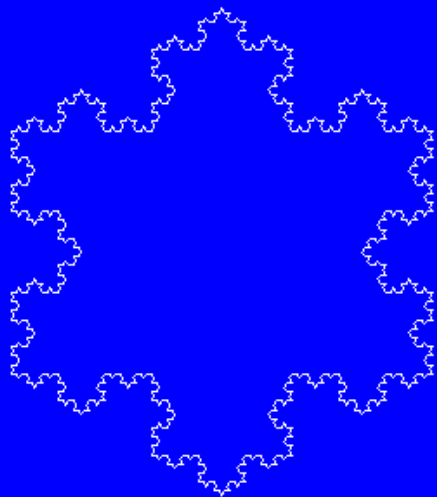



Đường Koch

```
void drawKoch(Painter& painter, float len, int levels)
{
    if (levels == 1) {
        painter.moveForward(len);
    } else {
        drawKoch(painter, len/3, levels-1);
        painter.turnLeft(60);
        drawKoch(painter, len/3, levels-1);
        painter.turnRight(120);
        drawKoch(painter, len/3, levels-1);
        painter.turnLeft(60);
        drawKoch(painter, len/3, levels-1);
    }
}
```







<https://github.com/tqlong/advprogram/archive/bf6e82b1c9465dc78b3c2a9bdcdf2d79b83a584c.zip>

Bài tập

- Viết thêm hàm trong **Painter**
 - Vẽ hình chữ nhật
 - Vẽ bánh răng (trong cơ khí)
- Vẽ tập hợp Julia
 - https://en.wikipedia.org/wiki/Julia_set
- Tìm kiếm và vẽ các loại fractal đệ quy
- Tìm hiểu cách ghi ảnh thành tệp JPG, PNG
- Thử sức: phát triển phần mềm quản lý ảnh trên máy: xem ảnh, slideshow, thumbnails ...

