前面介绍了使用window system调用来画贪吃蛇的图形，看起来还是不那么方便。

EasyX 是基于 C++ 的图形界库，可以帮助 C/C++ 初学者快速上手图形和游戏编程，哪怕你仅仅会C语言也可以直接使用，非常便捷，利用easyX的绘图函数，轻松绘制矩形、圆形等图案，发挥你的创造力，编写诸如俄罗斯方块、贪吃、五子棋这样的小游戏，甚至可以开发图形相关的算法及研究工作

# 参考文档：

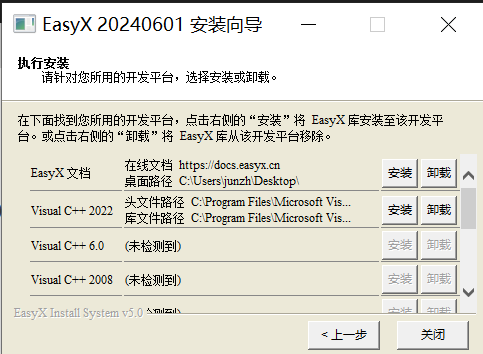
<https://www.dotcpp.com/course/easyx-go/>

[easyX图形编程入门 - C语言图形编程 - C语言网 (dotcpp.com)](https://www.dotcpp.com/course/easyx-go/)

# 下载和安装

<https://easyx.cn>

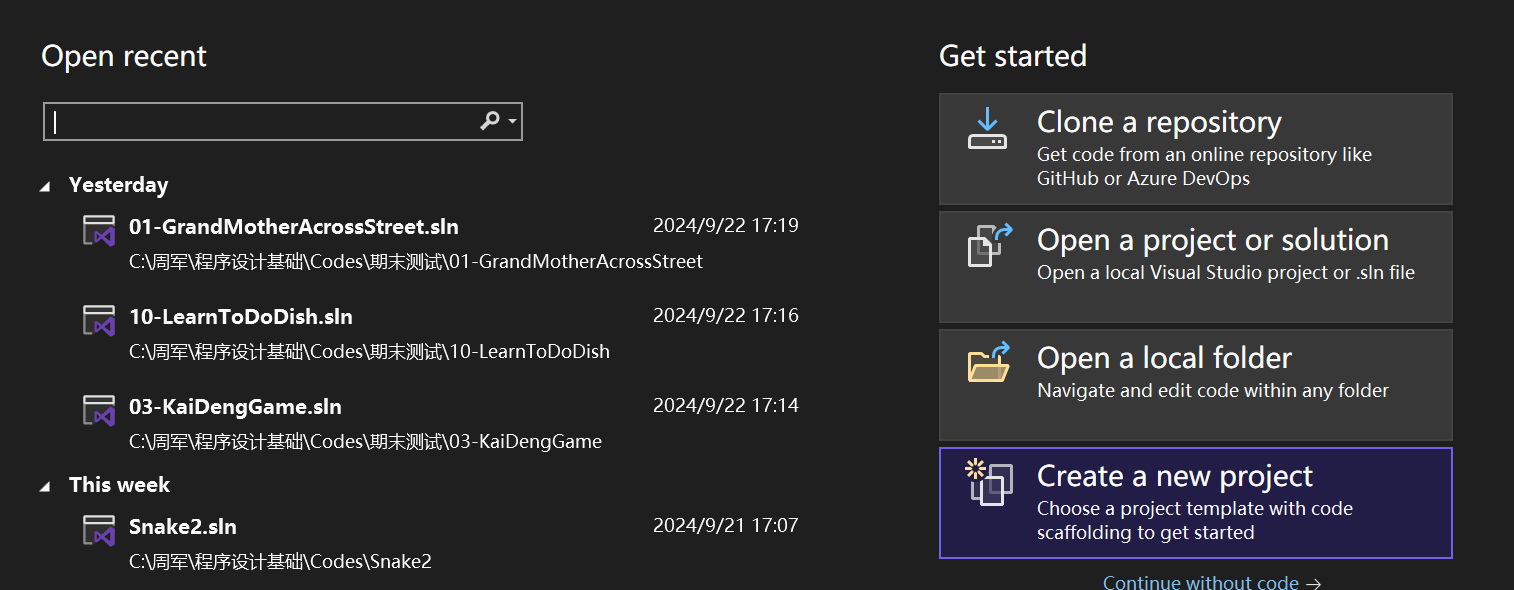


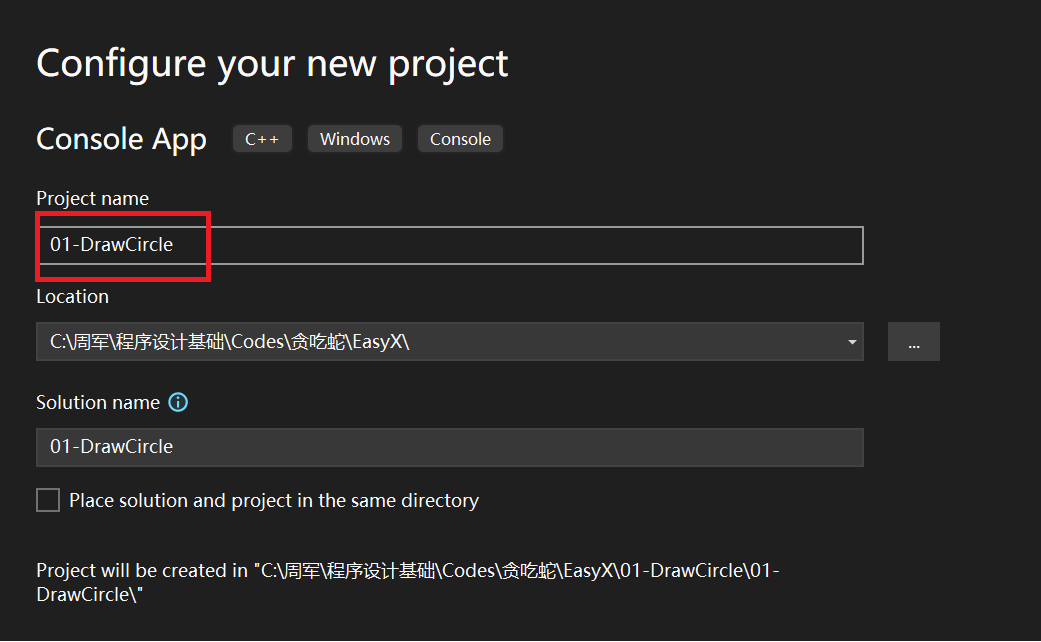


点击“安装”

安装的本质，是将easyX的.h头文件和.lib库文件复制到Visual C++的include和lib文件夹中。

# 第一个例子





代码如下：

#include <iostream>

//EasyX头文件

#include <graphics.h>

#include <conio.h>

int main()

{

initgraph(640, 480); // 绘制窗口，大小640\*480像素

setfillcolor(YELLOW); // 设置填充颜色为黄色

fillcircle(200, 200, 100); // 画圆，圆心（200，200），半径100

// 按任意键继续

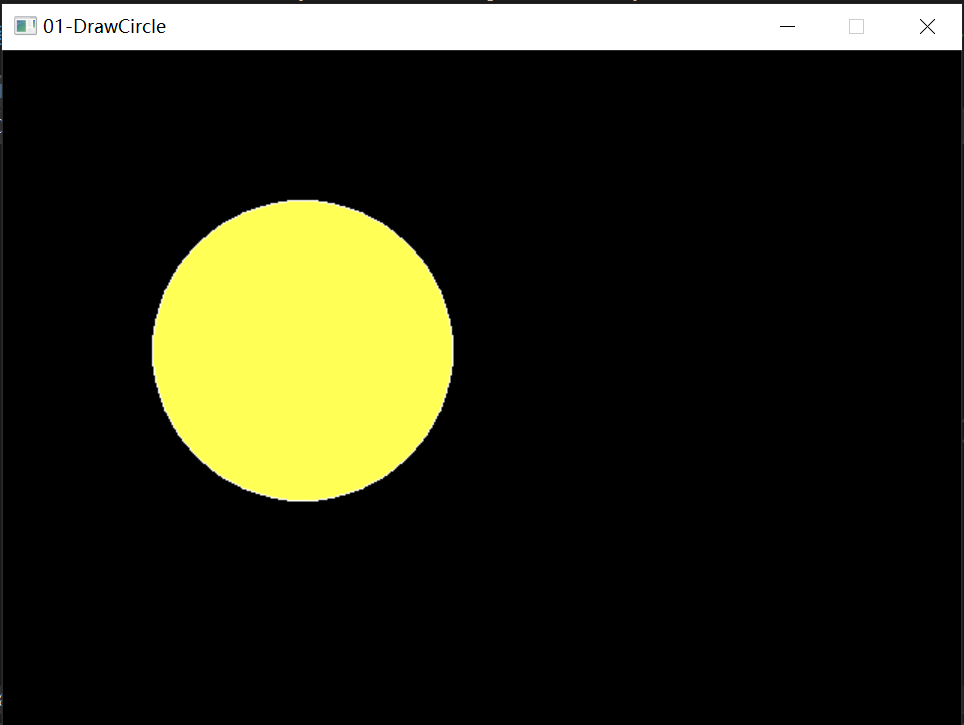
getchar();

closegraph();

return 0;

}

运行结果：

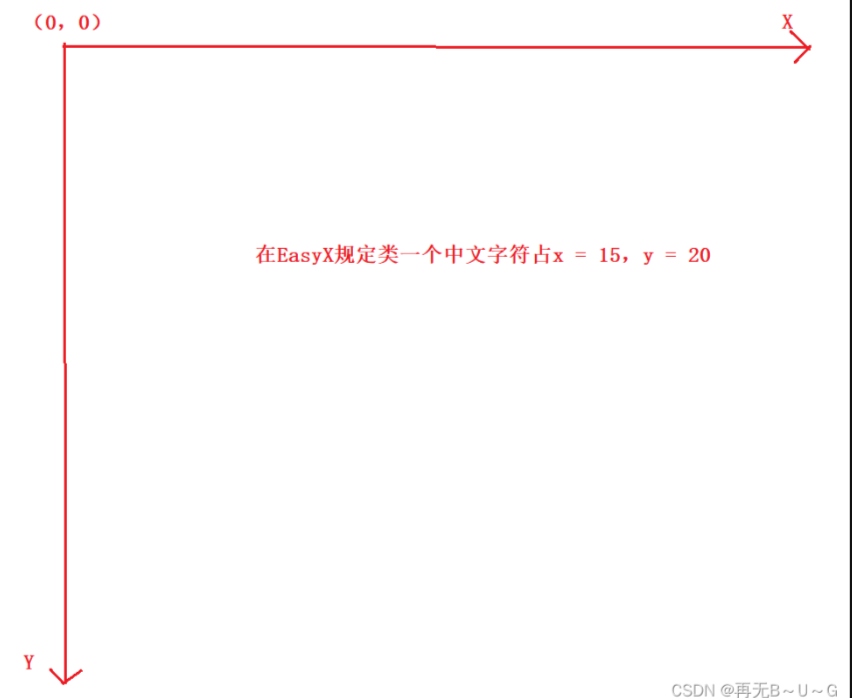


# EasyX的坐标

看到C语言使用easyX绘图的效果后，给大家普及一下一直会用到的几个概念：

## 坐标

  不仅仅画圆，以及画点、线、圆等各种图案都离不开坐标的概念，那么这个坐标是怎么计算的，可能大家已经发现了，那就是以窗口程序的左上角为（0，0）点，x轴和y轴分别向右和向下递增，单位为像素点



另外需要特别注意的是，一个中文字符占x=15,y=20

# 第二个例子：绘制矩形

代码：

#include <iostream>

//EasyX头文件

#include<graphics.h>

#include<conio.h>

int main()

{

// 绘制窗口，大小640\*480像素

initgraph(640, 480);

// 绘制矩形, 左上点坐标（50，50），右下坐标（200，200）

rectangle(50, 50, 200, 200);

// 按任意键继续

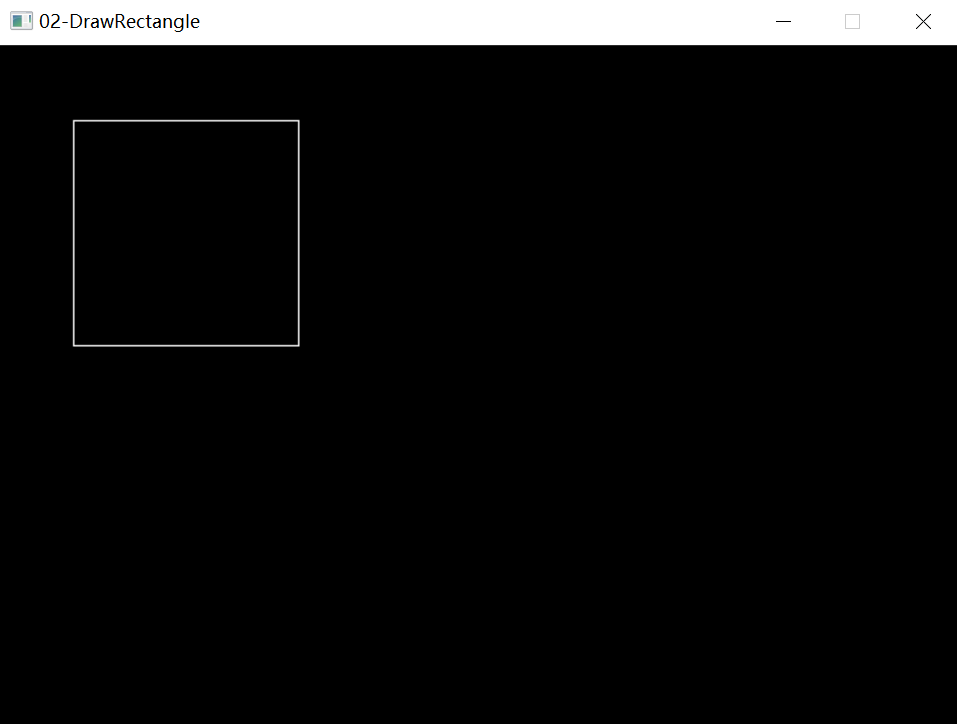
getchar();

closegraph();

return 0;

}

运行结果：



# 第三个例子：加载外部图片并显示

easyX首先通过loadimage加载图片，然后使用putimage输出到窗口上，

涉及到两个函数：

**loadimage**

第一个参数是传入参数，因此需要加地址符，

第二个参数使用的路径是相对路径，所以用../，绝对路径当然也可以。需要注意的是，因为C语言中字符串转义字符的存在，这里需要写两个/

第三个/第四个参数：load图片的宽和高

**putimage**

在指定位置输出即可

// 加载图片

IMAGE img;

loadimage(&img, \_T("..//..//Images//Wall3.jpg"), 15, 20, true);

//向（10,10）位置开始输出此图片

putimage(0, 0, &img);

完整代码如下：

#include <iostream>

// EasyX头文件

#include <graphics.h>

#include <conio.h>

int main()

{

// 创建绘图窗口，大小为 640x480 像素

initgraph(900, 500);

// 加载图片

IMAGE img;

loadimage(&img, \_T("..//..//Images//Wall3.jpg"), 15, 20, true);

//向（10,10）位置开始输出此图片

putimage(0, 0, &img);

//任意键继续

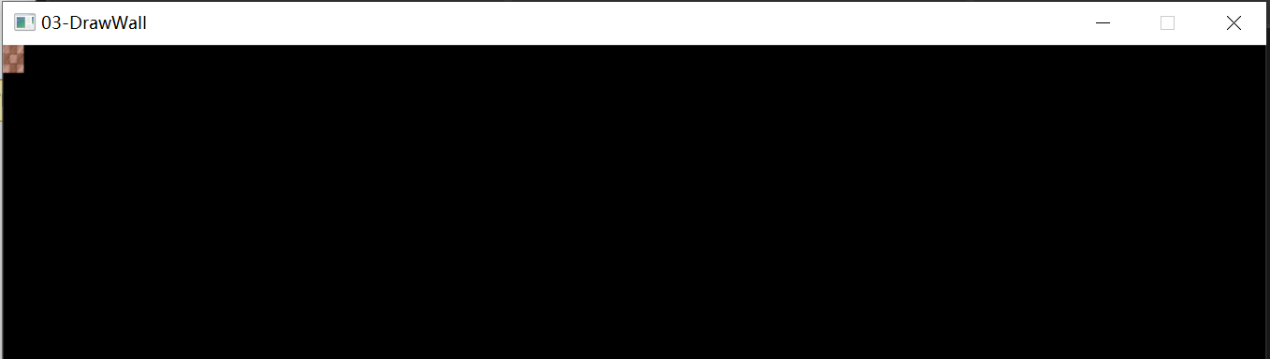
getchar();

closegraph();

return 0;

}

输出效果：



上面只是显示一张图片的例子，我们在编写贪吃蛇游戏的时候，会绘制一圈墙，下面就是程序代码：（注意一个中文字符在EasyX坐标中占x=15,y=20）

#include <iostream>

// EasyX头文件

#include <graphics.h>

#include <conio.h>

int main()

{

// 创建绘图窗口，大小为 900x500 像素

initgraph(900, 500);

// 加载图片

IMAGE img;

loadimage(&img, \_T("..//..//Images//Wall3.jpg"), 15, 20, true);

//上面的围墙

for (int i = 0; i<= 900; i=i+15) {

putimage(i, 0, &img);

}

//下面

for (int i = 0; i <= 900; i = i + 15) {

putimage(i, 500-20, &img);

}

//左

for (int i = 20; i < 500; i = i + 20) {

putimage(0, i, &img);

}

//右

for (int i = 20; i < 500; i = i + 20) {

putimage(900 - 15, i, &img);

}

//任意键继续

getchar();

closegraph();

return 0;

}

显示效果：



# 第四个例子：向窗口指定位置输出文本

C语言使用printf函数输出，而且还只能顺着光标的位置打印，如果想特定位置输出还需要不停的输出换行或者回车调整光标位置。而且颜色只有白色的。而easyX可以很方便的实现这些功能，其函数如下：

**void** outtextxy(**int** x,**int** y,**LPCTSTR** str);

代码例子：

#include <iostream>

//EasyX

#include <graphics.h>

#include<conio.h>

int main()

{

initgraph(640, 480); // 创建绘图窗口，大小为 640x480 像素

outtextxy(100, 100, \_T("www.dotcpp.com"));

getchar(); // 按任意键继续

closegraph(); // 关闭绘图窗口

return 0;

}

以上代码的意思，是向窗口（100，100）位处开始输出“www.dotcpp.com”字符串，默认是白色字体，使用是不是非常方便?

运行效果：



这里我们可以测试一下在EasyX中输出中文，是不是坐标x=15, y=20.。

代码如下：

#include <iostream>

//EasyX

#include <graphics.h>

#include<conio.h>

int main()

{

initgraph(640, 480); // 创建绘图窗口，大小为 640x480 像素

outtextxy(100, 100, \_T("www.dotcpp.com"));

//测试汉字输出

outtextxy(120, 120, \_T("你"));

// 横着连续输出

outtextxy(120+15, 120, \_T("好"));

//竖着输出

outtextxy(120, 120+20, \_T("好"));

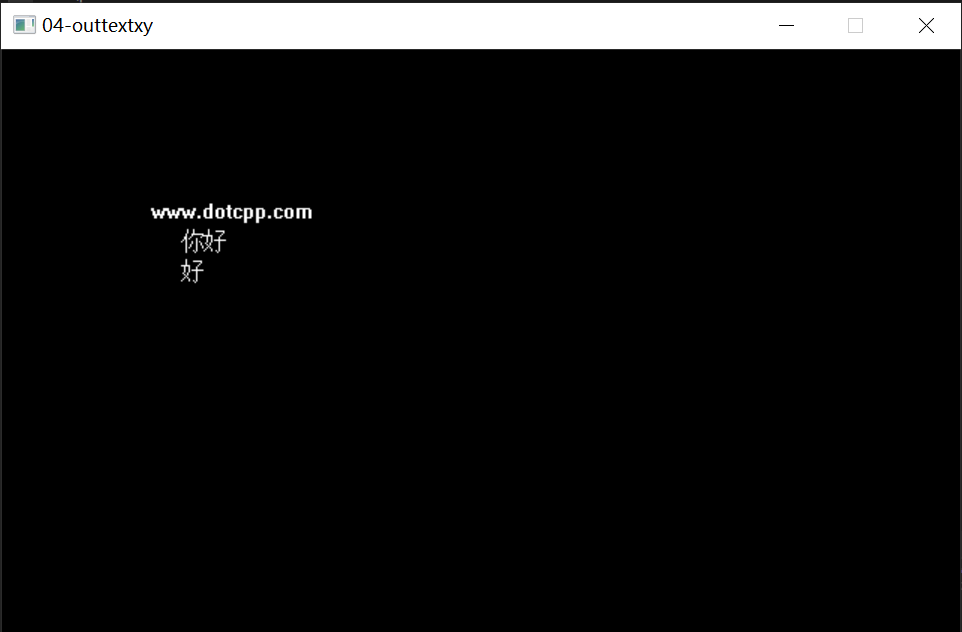
getchar(); // 按任意键继续

closegraph(); // 关闭绘图窗口

return 0;

}

运行效果：



# 如何擦除已经绘制的图案

对于有动态效果的程序，我们为了实现动画效果，会非常频繁的对图案有擦除的需求，对于已经绘制的图案如何擦除呢？这里给大家提供三个思路：

第一个思路，调用cleardevice()函数清空屏幕，函数原型分别为：

|  |  |
| --- | --- |
| 1 | **void** cleardevice(); |