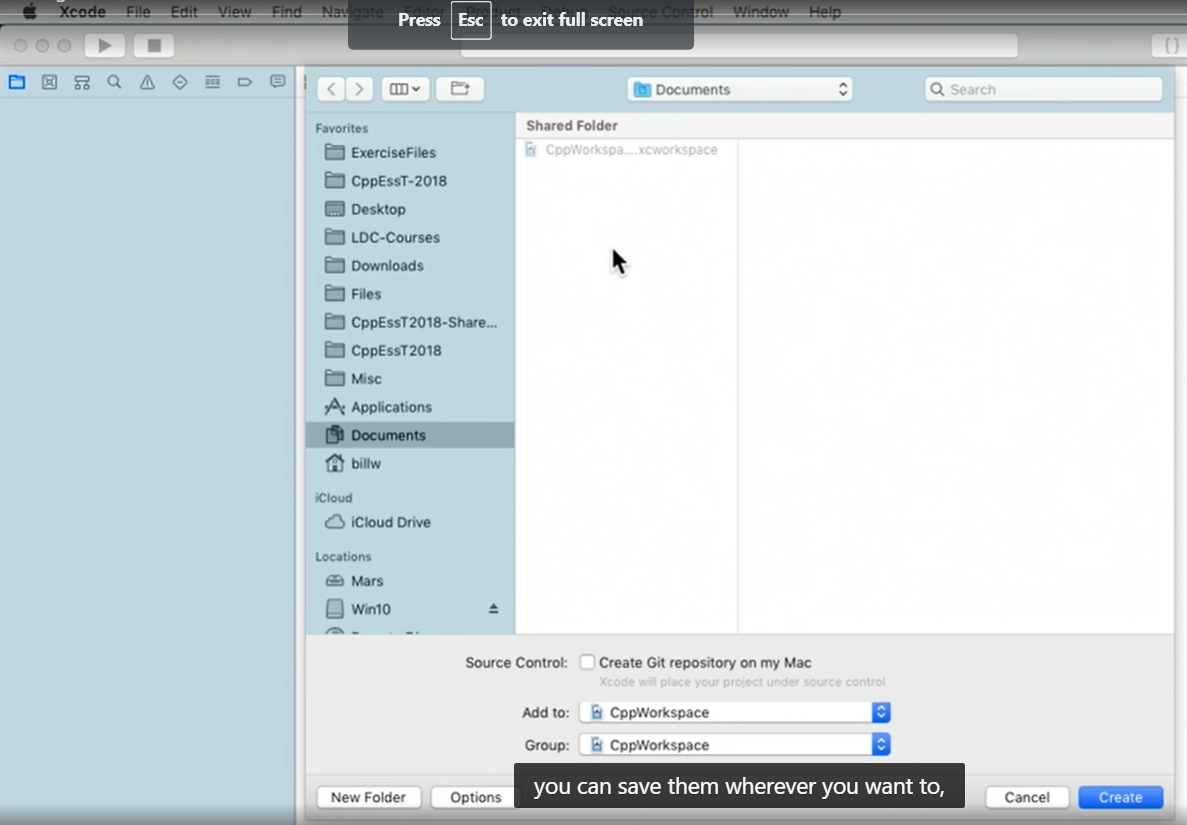
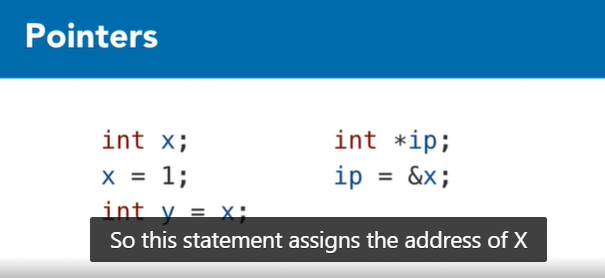


10点25分 2021年7月1日重新振作

苹果操作系统中用Xcode 配置C++

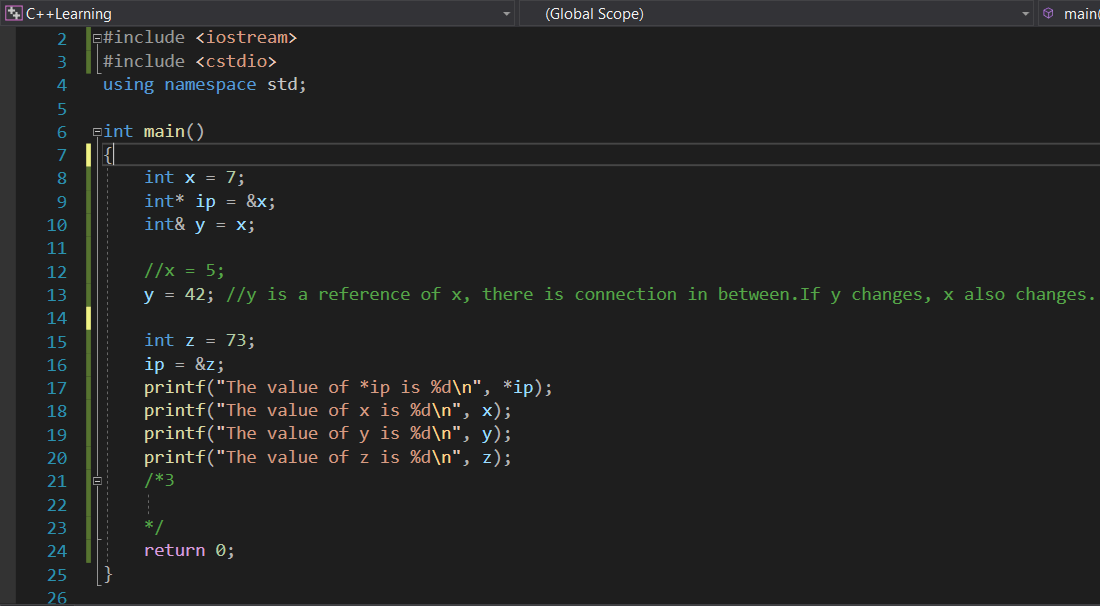


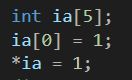


The statement assigns the address of X to integer pointer ip

\*ip means \*(&x) means finding the content of x’s address means x.

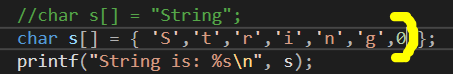
y is a reference of x, there is connection in between. If y changes, x also changes.





Row 2 and Row 3 has the same effect.

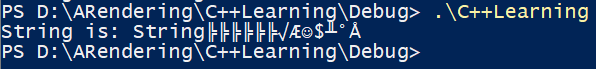
int\* ip = ia;//This assigns the address of the array to the pointer.

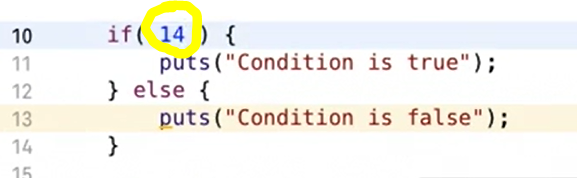


Inside char s[], using ‘x’ instead of “x”.

Moreover, if there is no 0 at the end, the output will become like below (some strange characters will come out at the end):







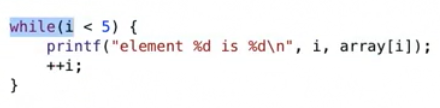
Any non-zero value, inside if, returns “Condition is true”

0 inside if, returns “Condition is false”

Comparing 2 values.



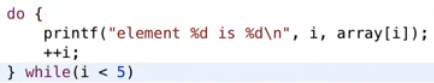
//While and Do



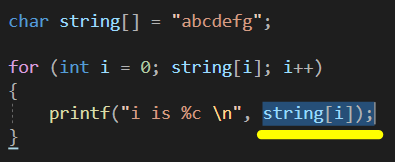
While(aa){xxx;} is the same as

do{xxx;}

while(aa)



String[i] only will print a character.



Stdout output in C++

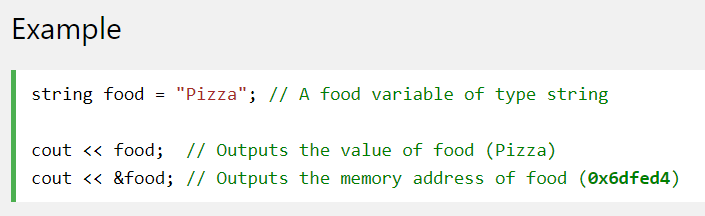


Output:



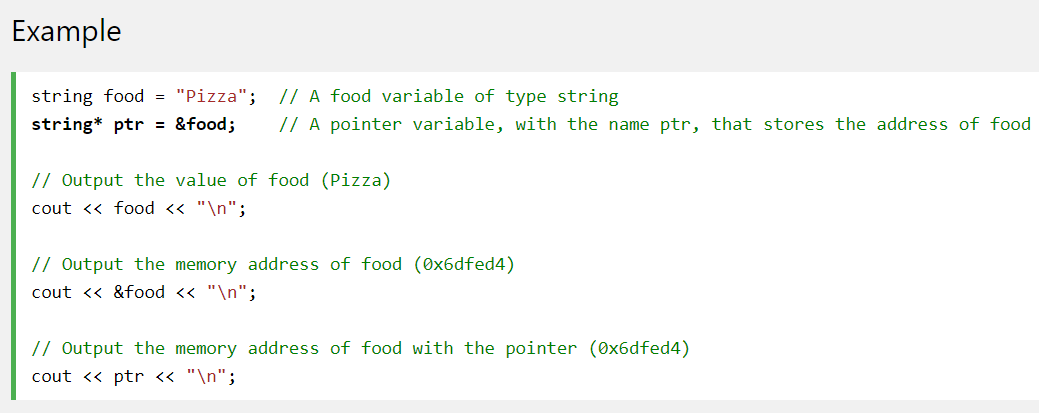
Referenced Here to get pointers in C++

<https://www.w3schools.com/cpp/cpp_pointers.asp>



&x means to get the address of x means a string of strange numbers.

A **pointer** however, is a variable that **stores the memory address as its value**.



string\* ptr=&x; // A pointer with the name ptr, which stores x’s address.

cout<<ptr<<”\n”; // Output the memory address of food with the pointer.

Create a pointer variable with the name ptr, that **points to** a string variable, by using the asterisk sign \* (string\* ptr). Note that the type of the pointer has to match the type of the variable you're working with.

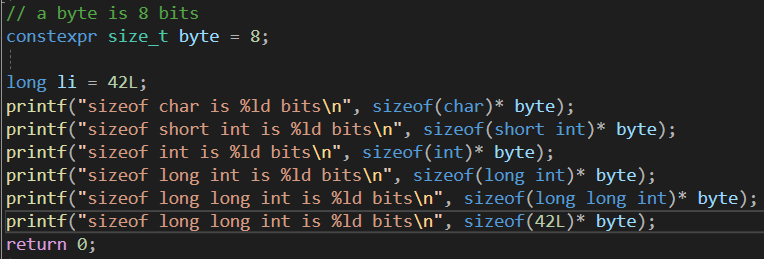
Use the & operator to store the memory address of the variable called food, and assign it to the pointer.

Now, ptr holds the value of food's memory address.

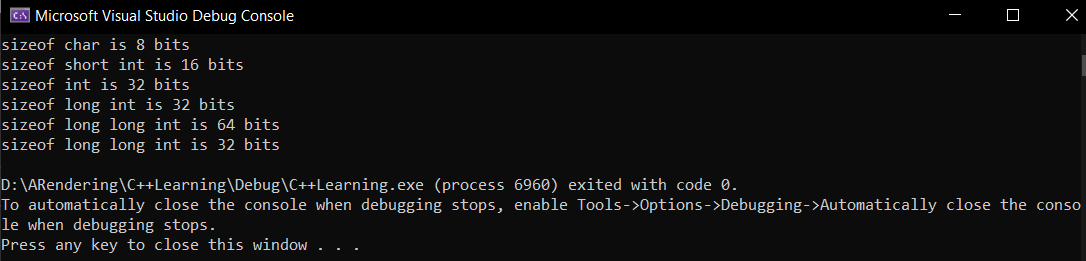
string\* mystring; // Preferred

2021年3月3日

There must be an asterisk\* to point the content, otherwise there will be error on byte.

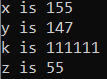


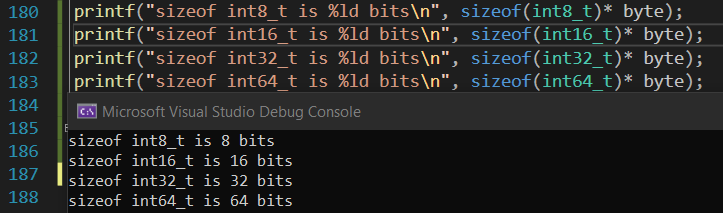
O/p:



int x =111+(a litter as suffix);

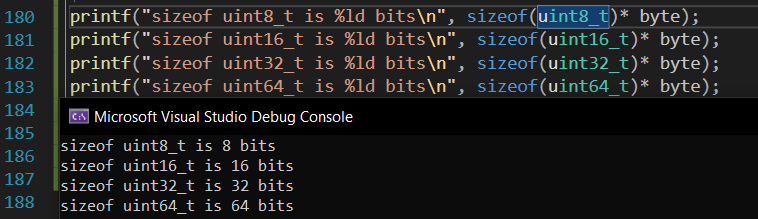
printf(“x is %(1 to 3 litters corresponding to the suffix)”, x);

  Output

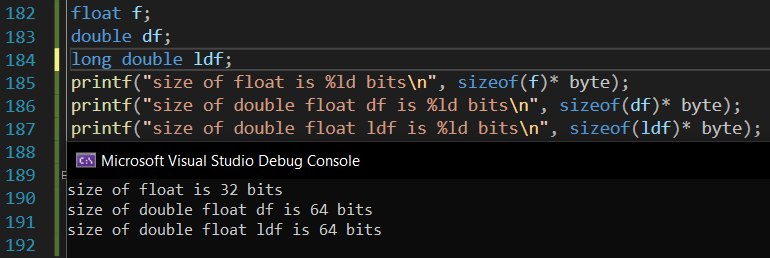


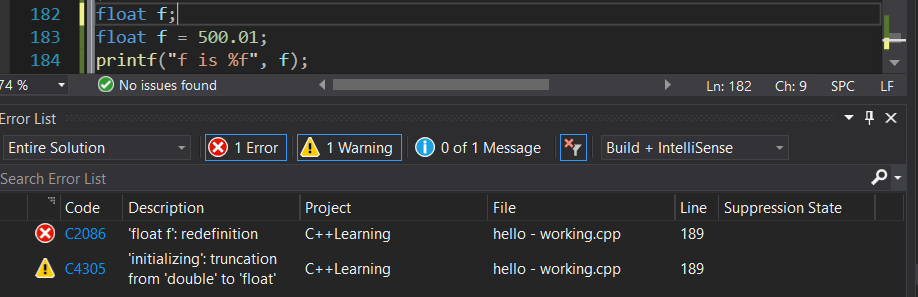
Signed and unsigned, two places to add “u”, output are exactly the same incase of the size.

There must by asterisk\*, otherwise redline will be shown under byte.



Float bit size.

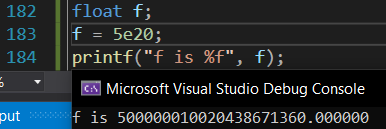




Just remove float in front of f=500.01; the problem will be solved.

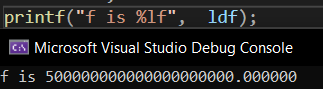


f=5e20;

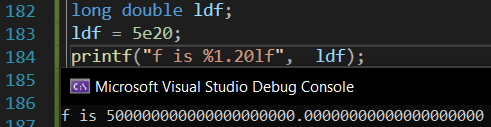


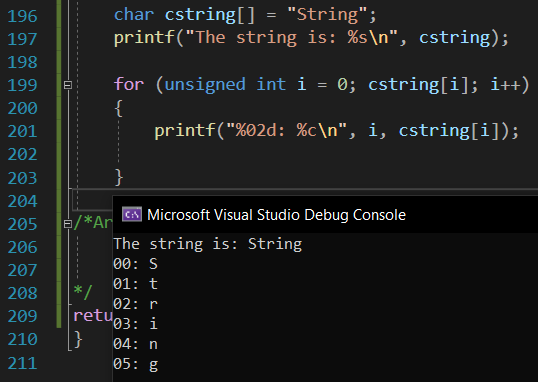
%lf is for a more precious value, where 0.3 not = 0.300000;

%lf made this output.



Where %1.20lf made this. In case of accounting, large integer types are required.





These two work the same. Because the character pointer can be used as array.



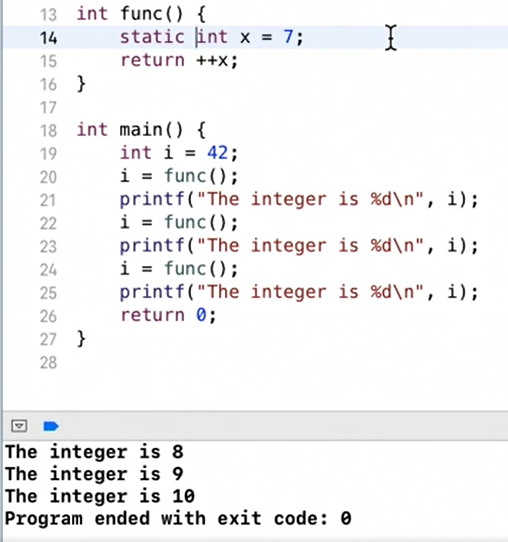
Inside “”, \x40 will be printed as @.

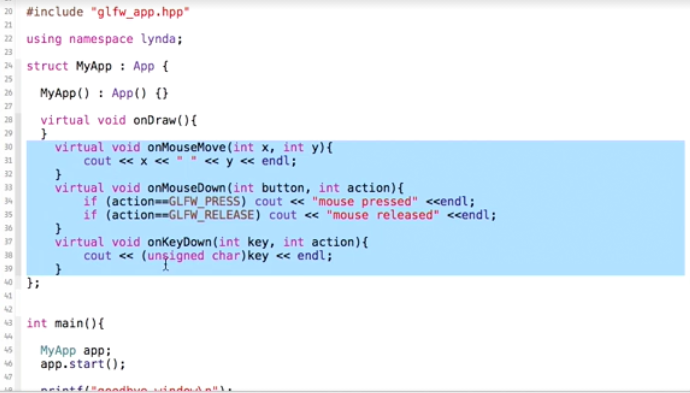
Qualifier 限定词 const static

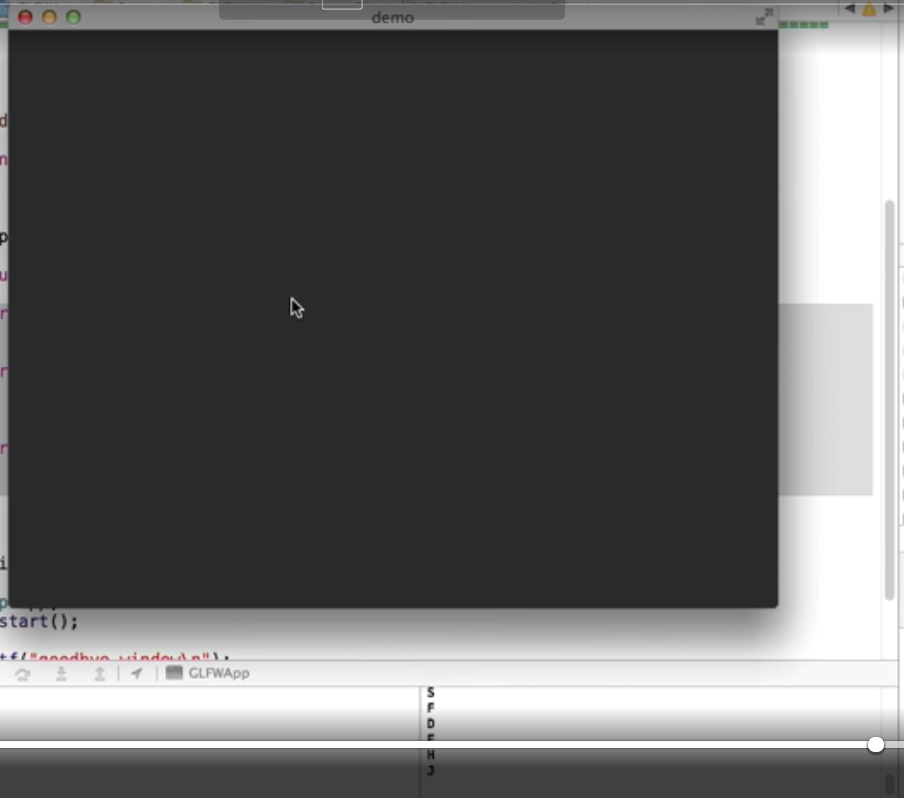




The propose of const qualifier.





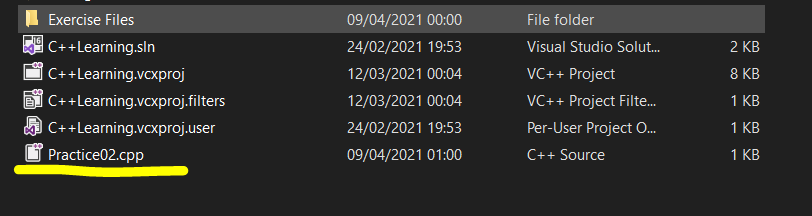


Mouse Printing.

2021年4月14日星期三 12点48分



需要删除内存，不然会报错。

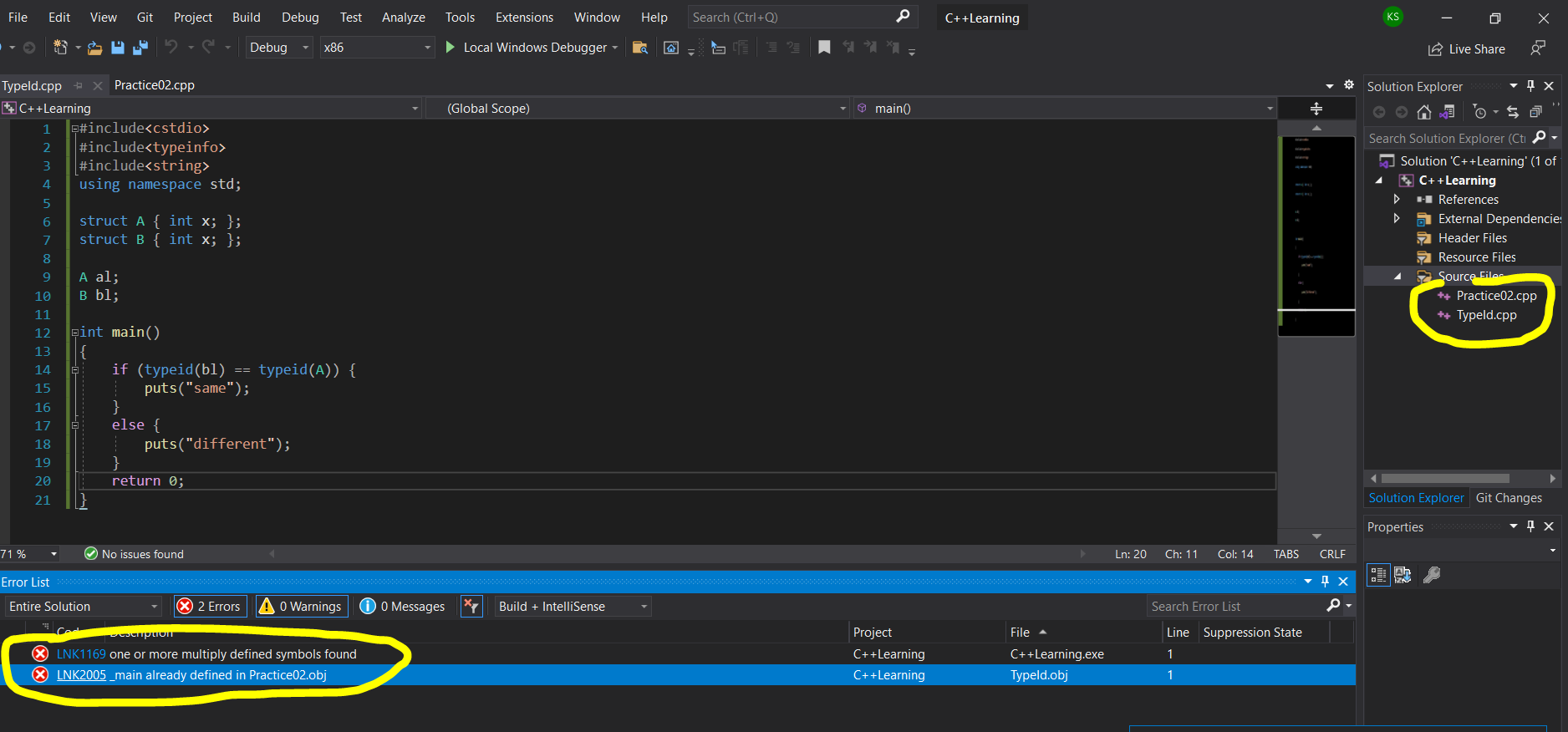


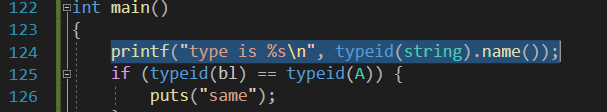
Practice01 is missing, may be inside the archive file of Skybox-17

Try catch is a better way.

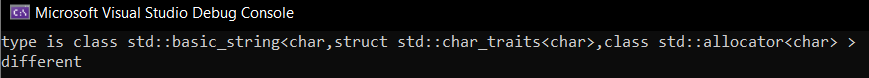
Every object allocated with new must be destroyed with delete, otherwise your program will leak memeory.

一个sln只能有一个主函数 main.cpp





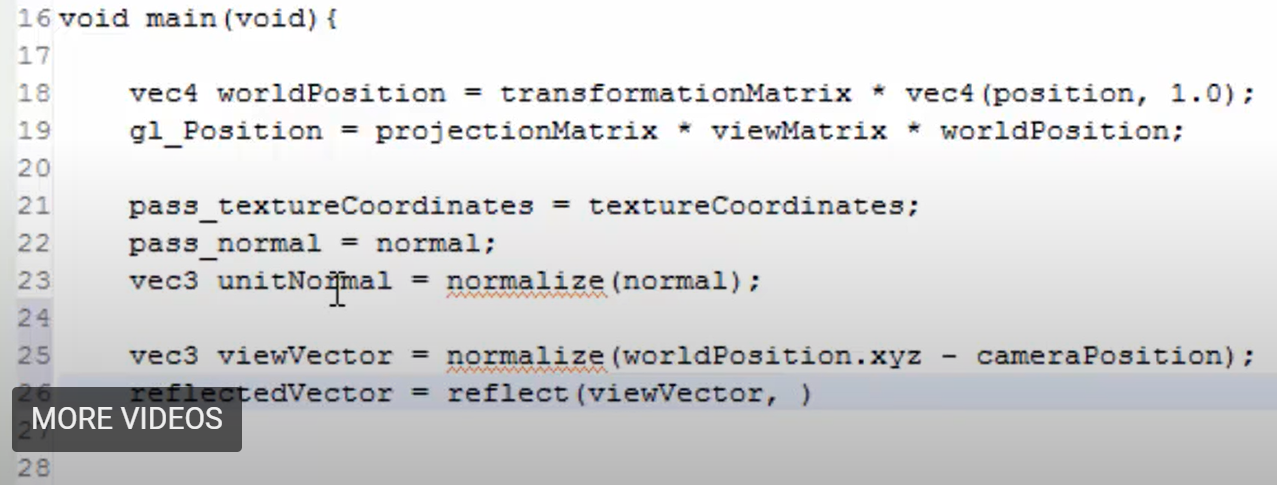
Name function is for getting the name.



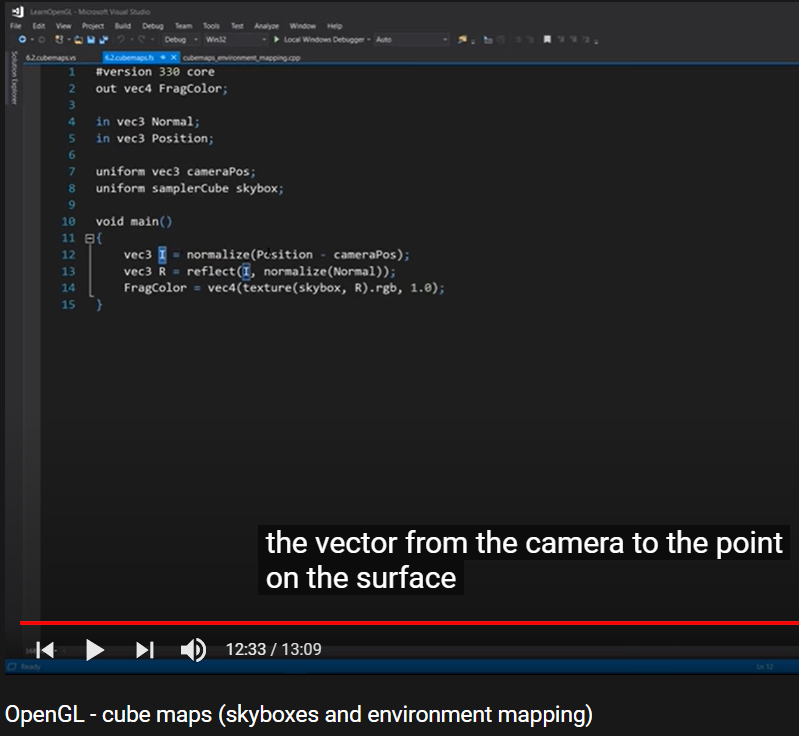
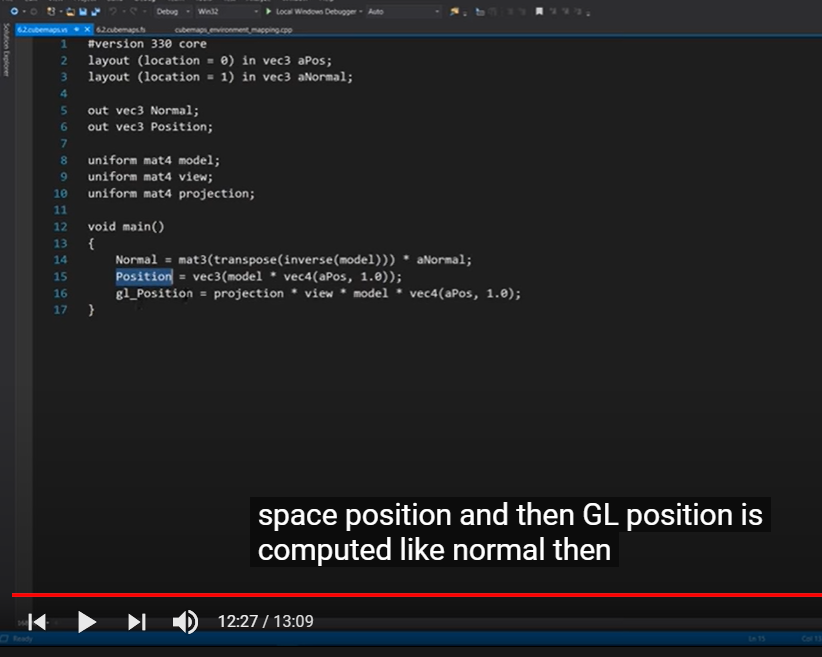
15点50分2021年4月14日星期三

看了这个教程[OpenGL Tutorial 51: Cube Map Reflections - Bing video](https://www.bing.com/videos/search?q=Reflection+opengl&docid=608018879289837307&mid=897080CD223B47B89AD3897080CD223B47B89AD3&view=detail&FORM=VIRE)

是用Java在eclipse 上写的，不过发现或小改shader , texture 中有cube map 要添加。

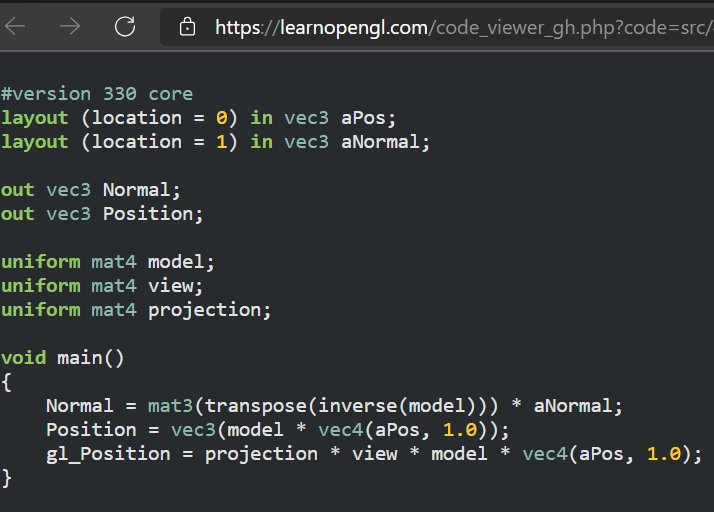


在Skybo.cpp上面改。

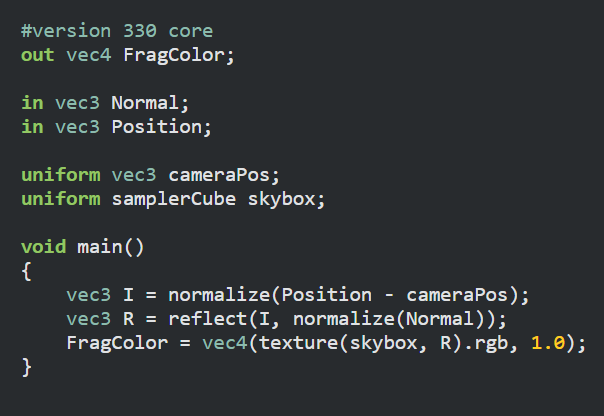


Vs上 Fs下

LearnOpenGL教程中的vs



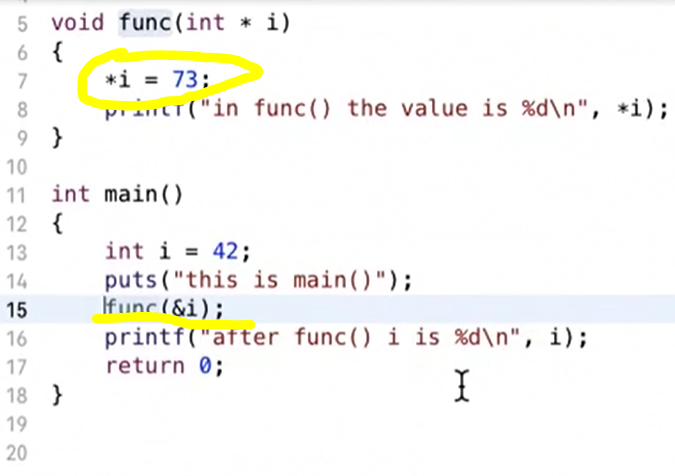
教程中的fs



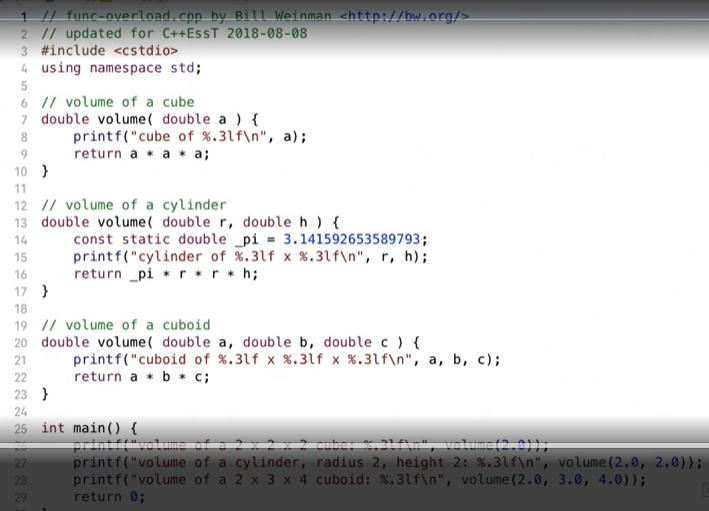
Change the shader to 6.2… no differences but no reflection

还有一个Skybox.cpp待改

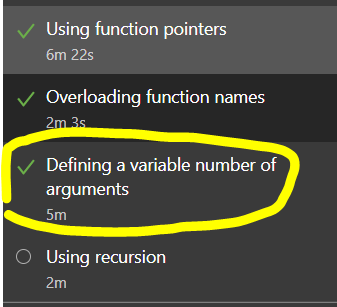
实在不行仅仅保留球体，甚至去掉多余的directional lighting的代码。



This 2 works the same.

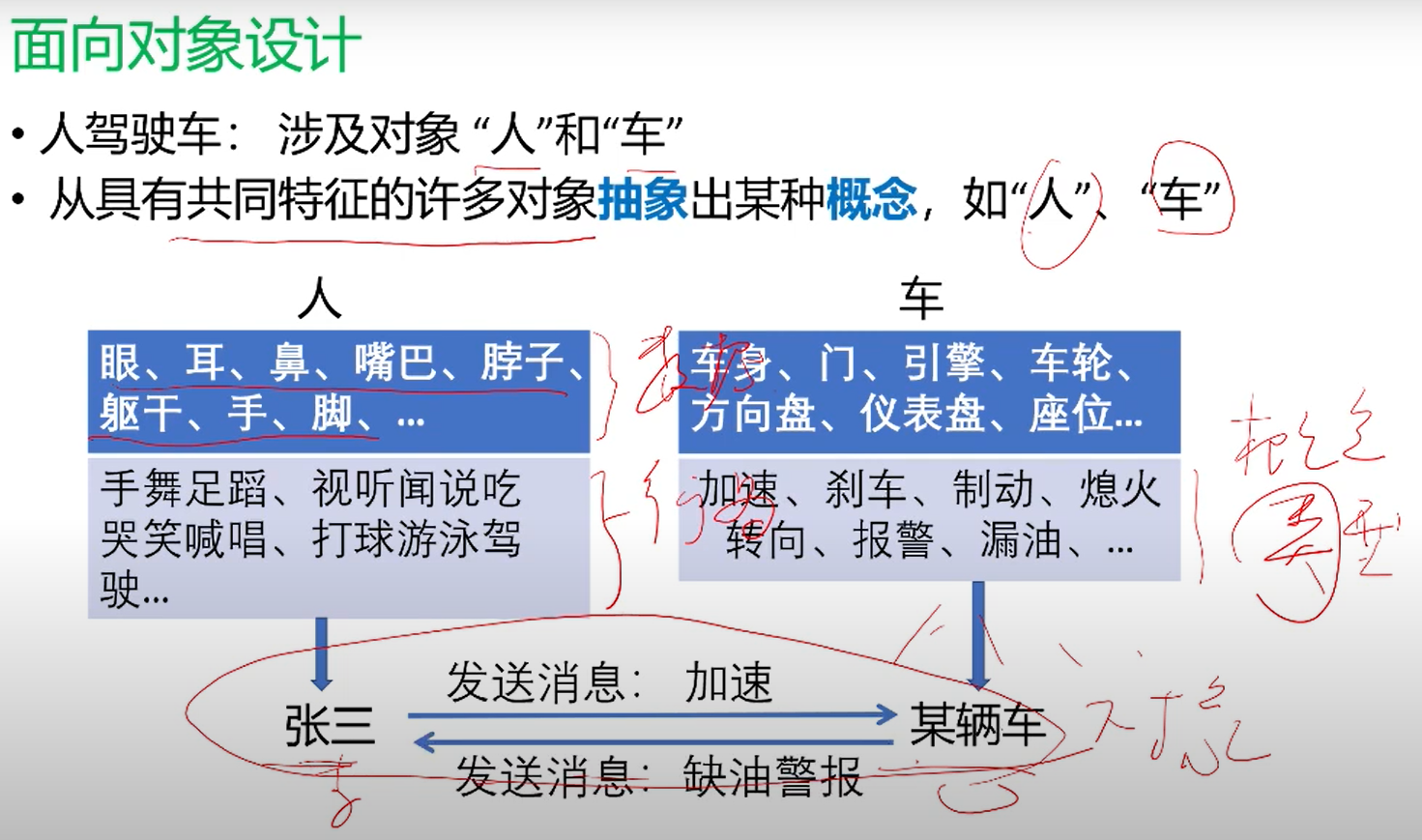


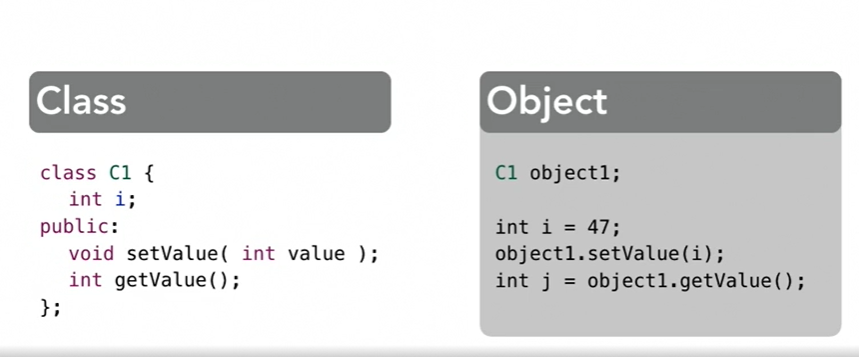
有不同Argument的函数可以有一个相同的名称，C++编译器可自动识别。

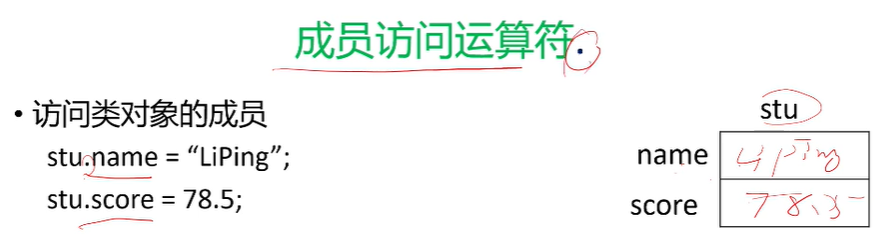


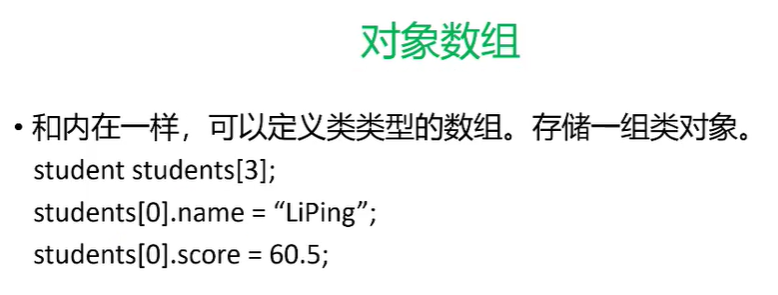
下次从这一章开始看。

09点56分 2021年6月14日









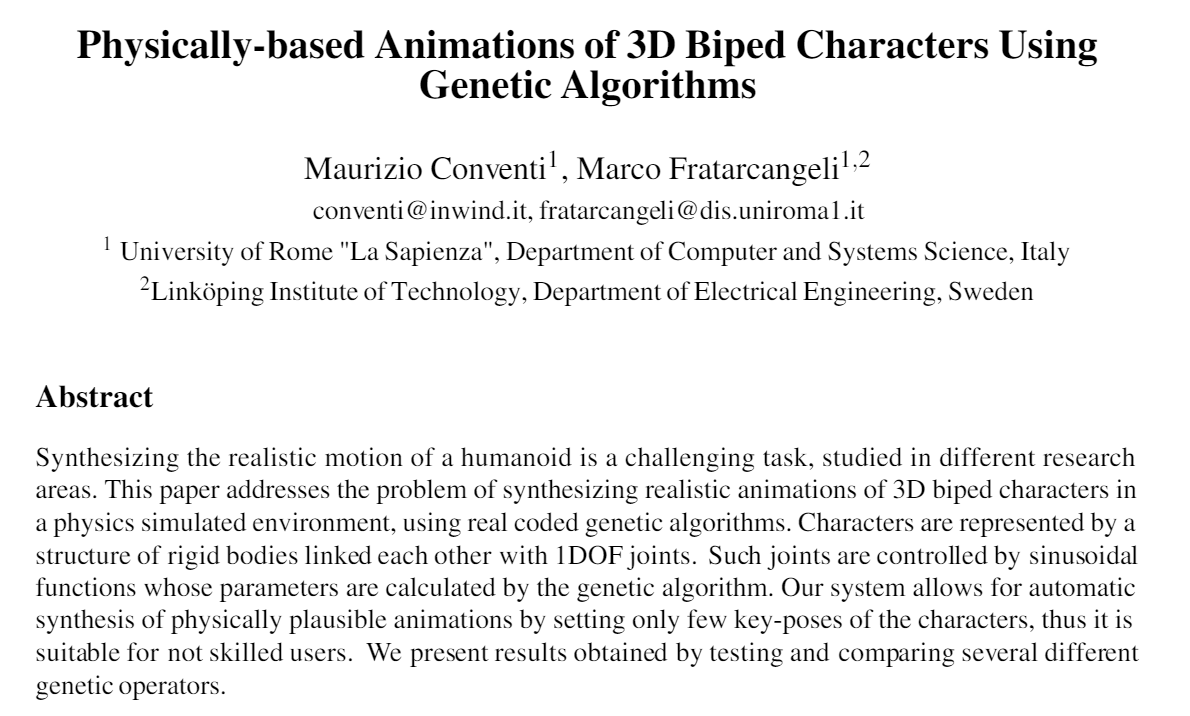
**论文项目先看此**

[How to Import Character Model Into Unity - YouTube](https://www.youtube.com/watch?v=Q8lJpoUwaBA)

10点41分 2021年6月14日

[(PDF) Physically-based Animations of 3D Biped Characters with Genetic Algorithms | Marco Fratarcangeli and Maurizio Conventi - Academia.edu](https://www.academia.edu/14819134/Physically_based_Animations_of_3D_Biped_Characters_with_Genetic_Algorithms)

再看此论文



[Unity for beginners in 2021- importing Mixamo characters #1 - YouTube](https://www.youtube.com/watch?v=0QA2O7juuWQ)

根据此动画，成功让老板人物可以动了，在Unity上。

现在先去做饭吃后，回来见系主任。

09点28分 2021年6月16日星期三



00点03分 2021年6月26日

Kinect not found

16点51分 2021年6月28日星期一 系主任邮件指导

Hi Yuzhou,

Note that there are several different Kinect versions and this is a Kinect v1. You will first need to find and install the drivers from Microsoft.

I’m sure there are several links tutorials for Unity (e.g. <https://assetstore.unity.com/packages/tools/kinect-with-ms-sdk-7747> )

Although Unity will be good for 3D visualization, you may need to look at more low level options (including the raw data from the Kinect) in order to make a useful scientific contribution.

This is possible in Unity but may be easier using a simpler tool to begin with.

For this I would suggest something like Processing ([www.processing.org](http://www.processing.org/)) which uses a very simple (java/c++ like language).

Inside Processing you can install kinect library and examples:  
Menu bar -> Sketch -> Import Library --> Add Library : find Kinect4Win SDK  and install

Then you can go to Menu bar à File à Examples : Under Contribute Libraries find Kinect4WinSDK and try some of the examples

See: <http://www.magicandlove.com/blog/research/kinect-for-processing-library/>

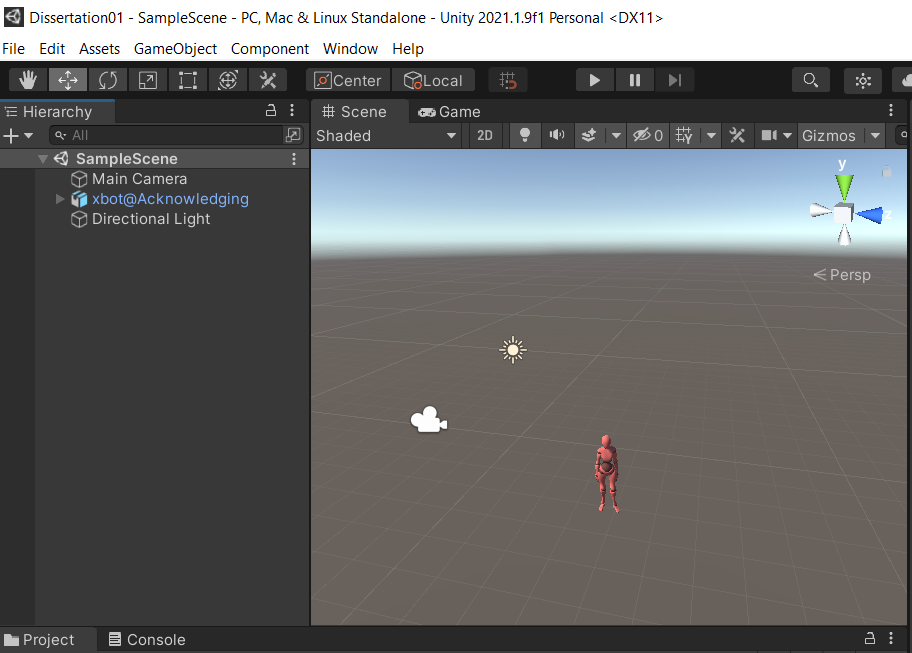
Later on (if needed) we can look at later Kinect Versions.

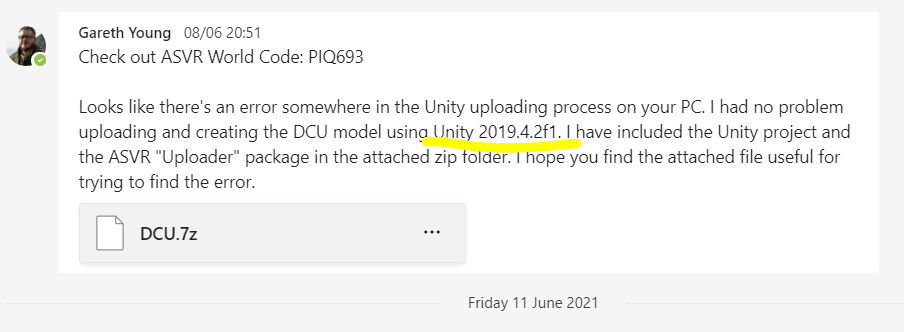
**七月 -10点26分 2021年7月1日**

等会儿加天空盒

胖大叔的人物由于昨晚未知的改动导致完全无法运行。

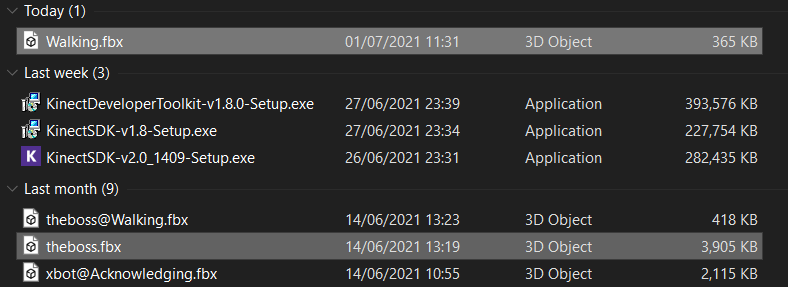
现在继续Dissertation01的项目来做。





助教之前用的是这个版本。

文件名有点问题，暂时跳过这个。



出现unity error后我删除了女性机器人的运动骨骼，就暂未报错了。

添加了天空盒，为了画面协调，删除了plane，后若有必要则加之。

现在尝试用git上传之前做过的Cpp项目到github

新生成的文件会被自动保存在同sln的路径中。



若删除，则会被永久删除，而不是保存在回收站中。

Yes but you do not have to follow the canvas itself precisely.

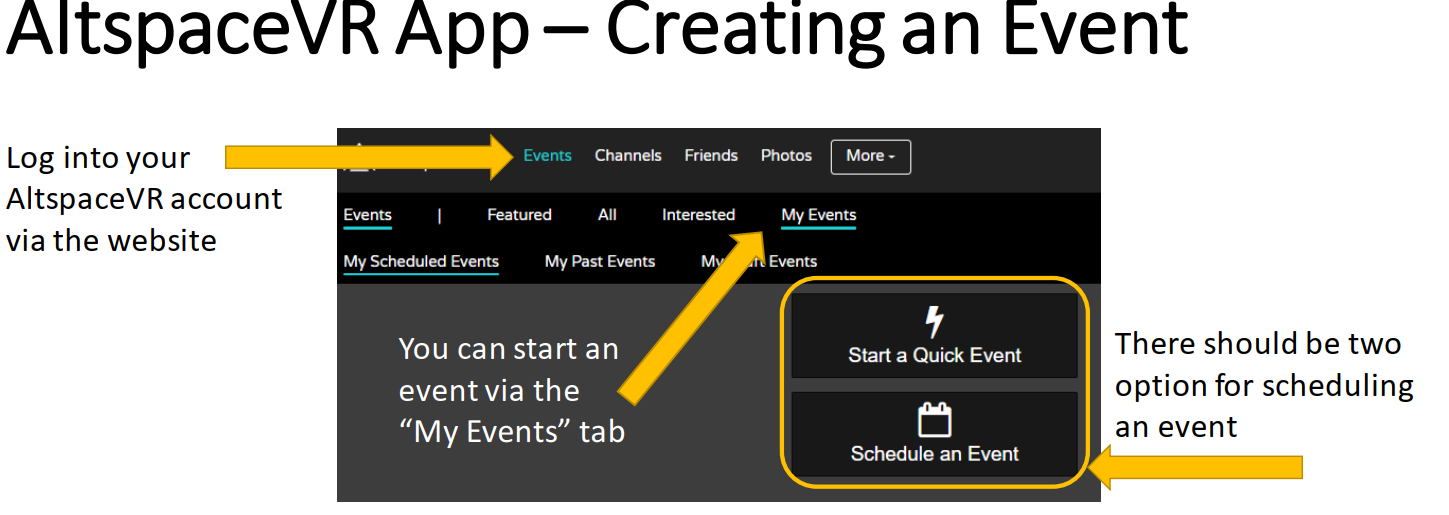
What I was suggesting that the same headings will help in the dissertation, and you should start immediately trying to fill out the points. This will not only constitute most of Chapter 1 but it will help plan the research as a whole.

In particular, I would suggest trying to populate the points on page 1, slightly modified below:

1. Research Question
2. Research Objectives
3. Methodology that will be taken to reach objectives
4. How you expect it will be evaluated
5. Main contributions (how does your work advance state of the art)

If you would like to meet I would be available after 1:30pm today.

Regards,



此处或原先忽略

没有忽略，这是最后几步。

上传成功

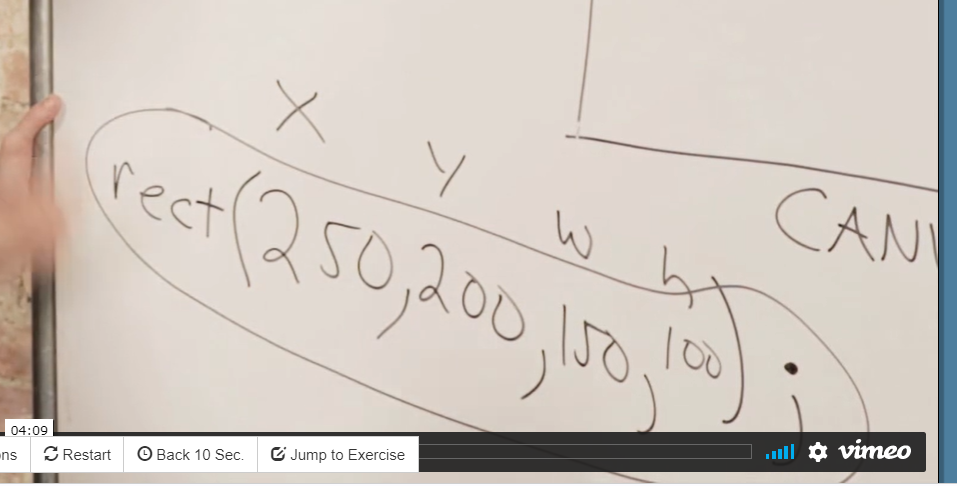
08点42分 2021年7月11日星期日

[Processing Hour of Code | Editor](https://hello.processing.org/editor/)

早起三明

看这个教程

长方形 代码



准备打开IDE 干代码。

18点57分 2021年7月11日星期日

昨天下载了一个文库，回查之。

**左上x坐标 矩形x轴长**

rect(250, 200, 150, 100);

**左上点y坐标 矩形y轴长**

Processing上面的IDE与此不同，到底在哪里写代码啊，或回看



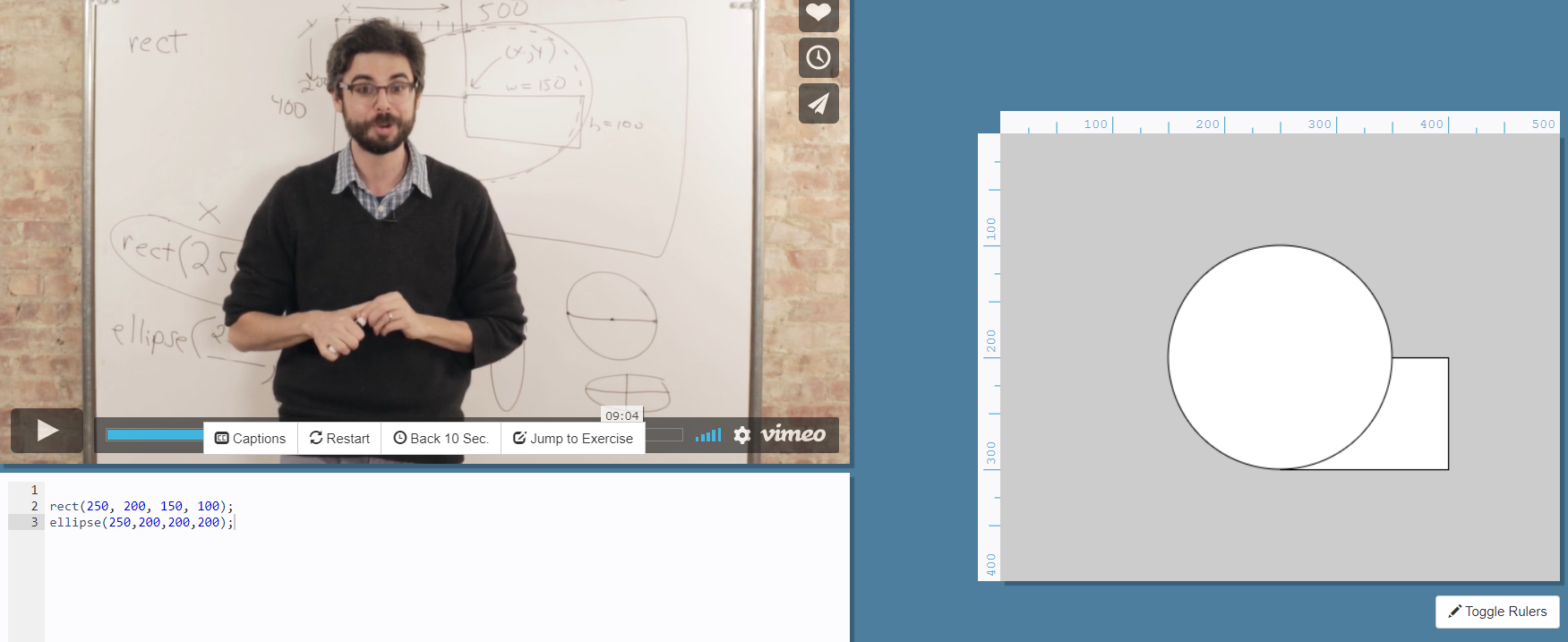
有运行的选项，然而不论如何调参数，输出总是恒定。或搜相关油管Processing教程。

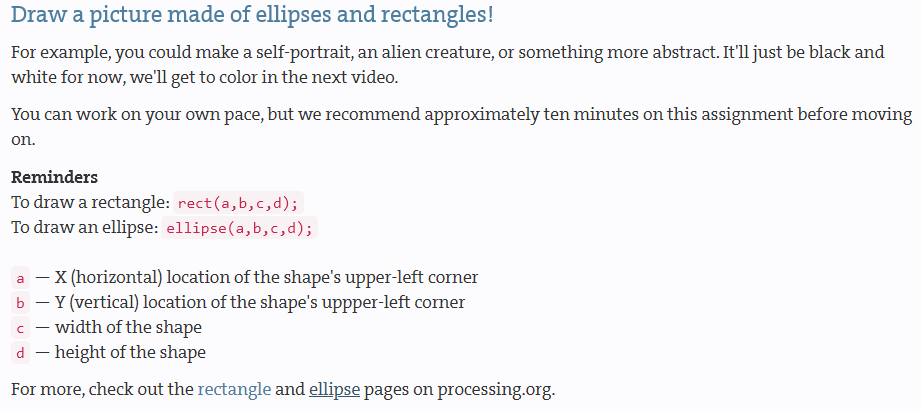


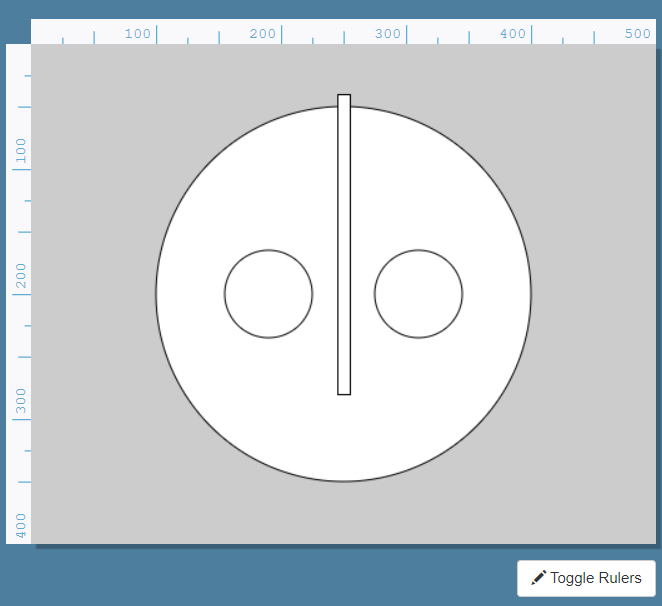
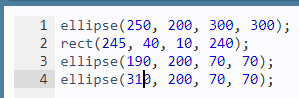
[2.1: How to use Processing - Processing Tutorial - YouTube](https://www.youtube.com/watch?v=5N31KNgOO0g) 教程中亦似java(如上图右上角)

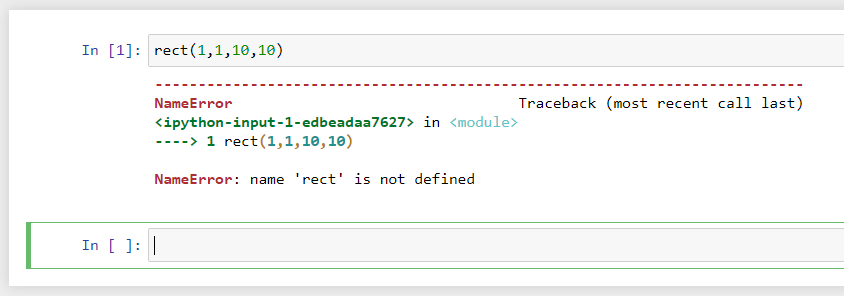
size(640,360) 可以增加输出窗口尺寸。

后出来的图形会覆盖前面的图形。

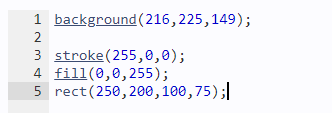


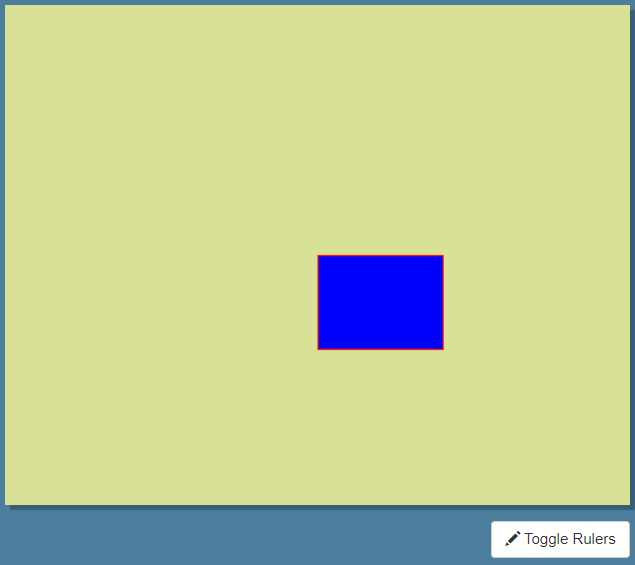


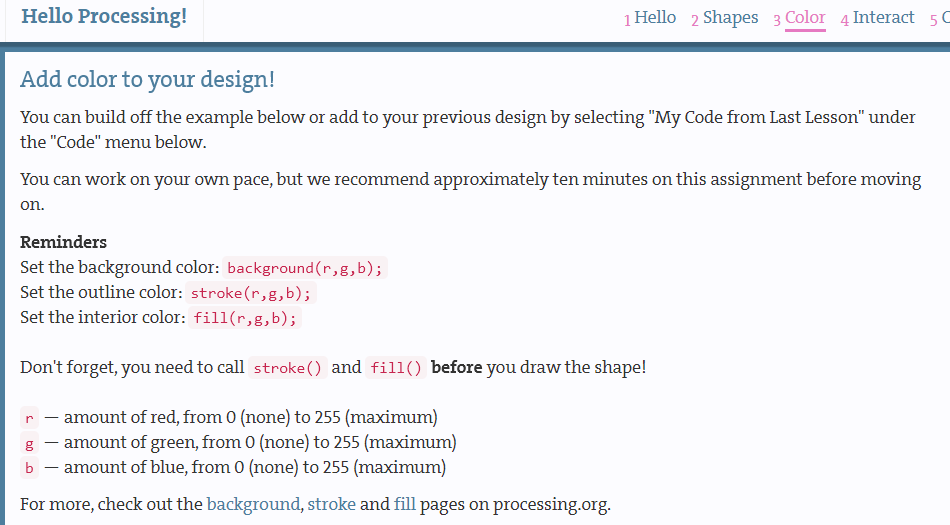
 

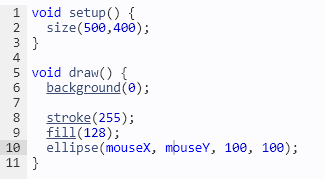


rect暂时无法在Jupyter上运行

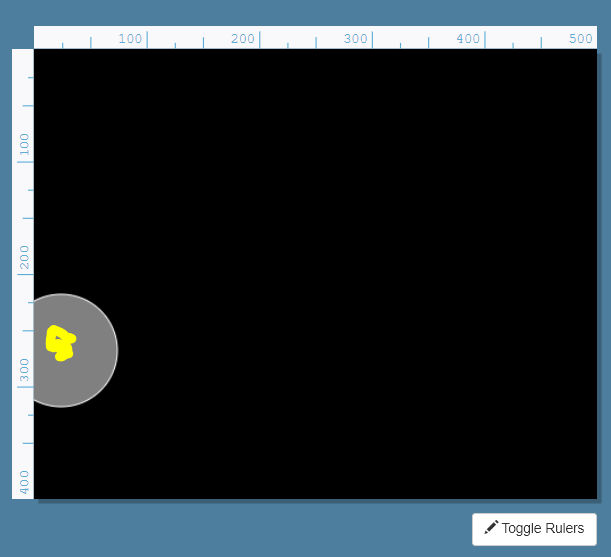


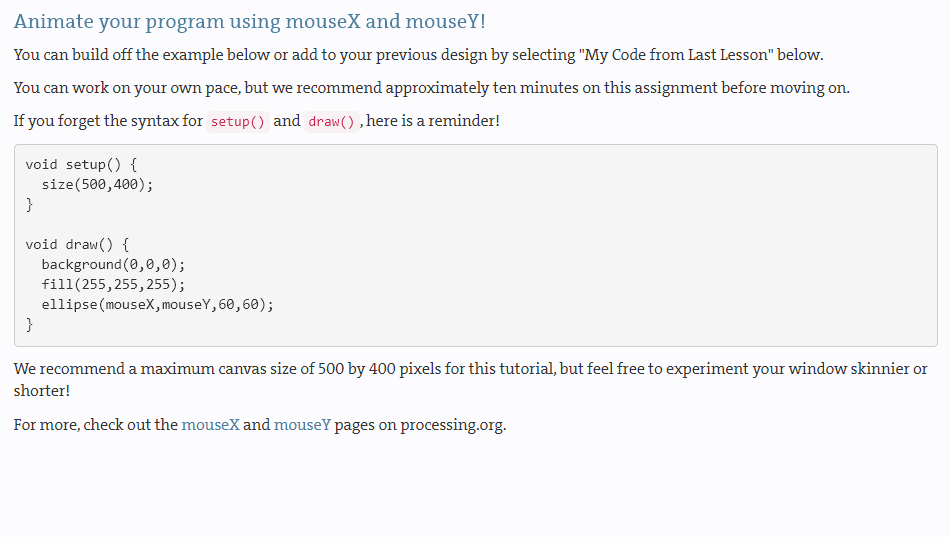


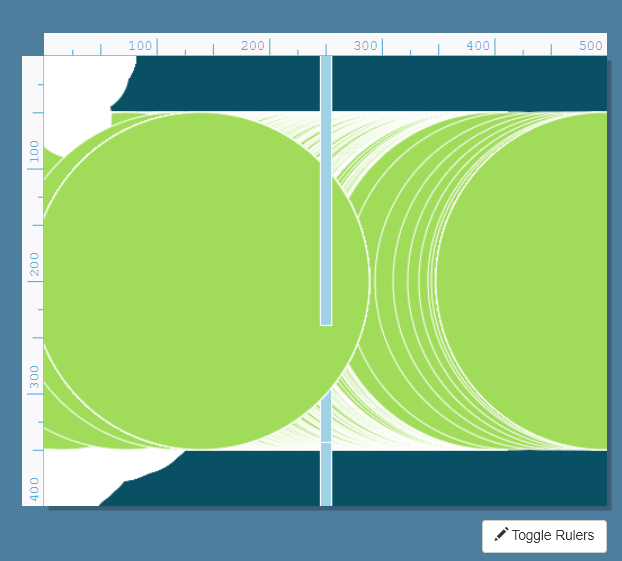


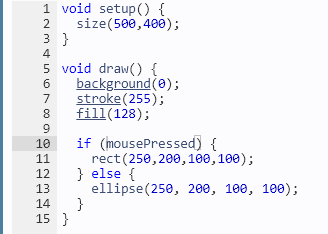


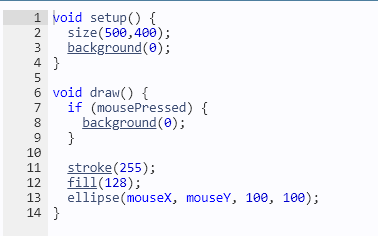
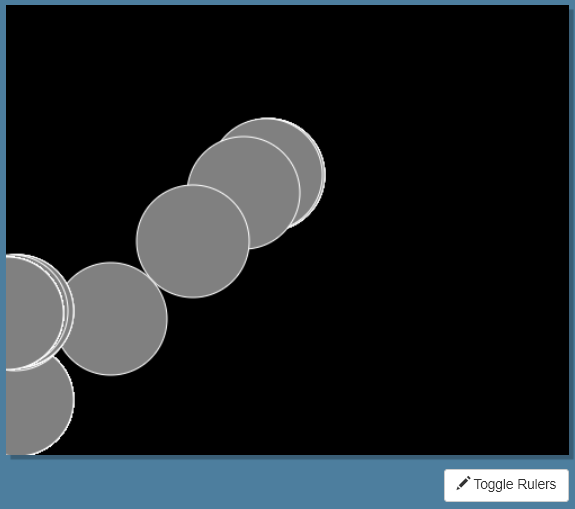
鼠标动，圆动。

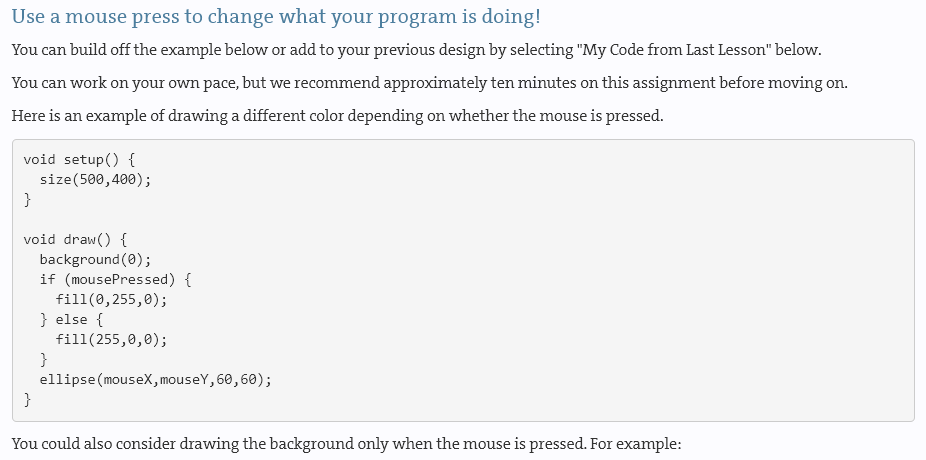


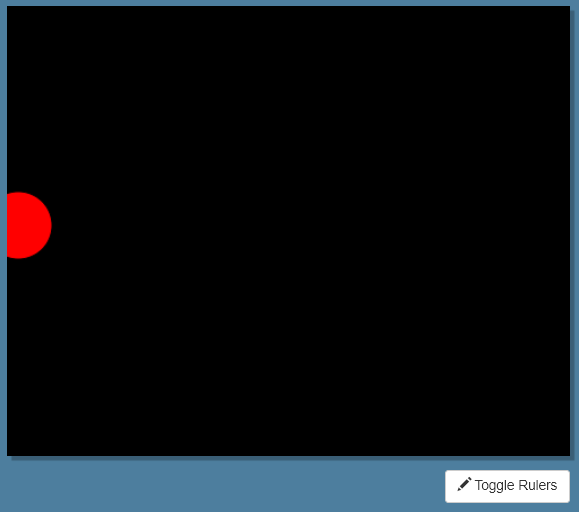








 鼠标松开或按住，红绿切换。

