大家好，我是 Hello,World 组的高鹤箫，我们今天的汇报内容是……
（BSOD）
没错，今天的汇报内容是多级指针。说实话这个主题没什么好讲的，无非就是指针套娃。当我们拿到这个问题的时候我们都很发愁，（） 因为多级指针所涉及的东西很少，没有什么有趣的东西向大家展示。不过经过一些讨论，最终我们决定，通过编写一个物理引擎的方式向大家展示多级指针的实际应用场景。这样一来能让原本无趣的程序代码变得生动活泼起来，也可以展示我们所学的知识在实际编程中的应用。

在此之前，需要先讲明白什么是多级指针。这是一个变量，指着变量的叫指针，指着指针的叫二级指针。如果我们让他在多指几次，多级指针。就是这么简单。多级指针的一种典型使用场景就是作为参数传递给函数，从而实现函数对指针进行操作。接下来我们实际操作一下，大家打开 devc++。

D: > Steve > Documents > C chain.c > main()

```c
#include <stdio.h>
#include <stdlib.h>

int main(){
    int a=5;
    int* ptr=&a;
    int** pptr=&ptr;

    printf("a=%d\n",a);
    printf("ptr=%p,*ptr=%d\n",ptr,*ptr);
    printf("pptr=%p,*pptr=%p,**pptr=%d\n",pptr,*pptr,**pptr);

    return 0;
}
```

（main）跟着我一起写：首先创建一个变量（a）然后创建一个指针（ptr）再创建一个二级指针(pptr)。Print 一下内容，输出指针和解引用后的内容。根据这些内容，我们可以得到这样的结论（sheet）：每一行中的东西都是互相等价的，自下而上形成一连串的引用。（vscode）这个实例中出现了二级指针，更高级的指针可以以此类推进行引用和解引用。不过我个人建议大家在实际进行编码的过程中尽量减少对高级指针的使用，因为这会提高程序的复杂程度，破坏连贯性和一致性。但是如果在实际的开发场景中有对于高级指针的切实需要，我们可以按需对其进行使用。

在开始写我们的物理引擎之前，我们还需要简单认识一个东西：结构体。关于他的具体的细节留给以后的同学去讲，我们现在只要知道，他是一个像数组一样可以存储一系列数据的对象，但是不局限于一种数据类型。具体的调用方法是（）这样：左边是结构体的名称，右边是结构体内的成员，中间用一个点来分隔。

一个物理引擎就是一套可以模拟现实世界物理法则的计算机程序。为此，我们需要两种数据，（）世界和物体。世界会生成一个实体树，而物体会通过指针一层一层的串在实体树上。接下来，让我们开始写吧：

接下来我要找一个找一个同学完成球体碰撞部分代码的编写。有谁想来吗？（许庆基）
（球体是圆形的，可以通过比较球心距离和半径和的大小进行判断）
成功！
最后我们总结一下：多级指针其实就是指向指针的指针，可以用于对变量的间接引用，也可以作为函数的参数传入函数从而操作指针。在实际用例中，多级指针通常用于处理复杂的数据结构或者在函数中修改指针的指向。多级指针的使用可以在某些情况下简化代码逻辑，但也容易造成代码的复杂性增加，需要谨慎使用。

该程序源代码已经上传 GitHub，感兴趣的同学可以克隆到本地进行研究。其中还有不少可以优化的部分：比如屏幕一直闪烁：这是一个屏幕绘制问题，可以通过加入双重缓冲技术得以缓解。
以及物体的所有数据信息全部都是整形，在进行模拟时会比较容易出现精度问题。
还有物体的质量：对物体进行操作的函数全都是在直接操作加速度，这个系统中没有质量，密度，风阻等等一系列的考虑因素。
像这样的问题还有很多，由于时间限制在这里就不进行展开了。以上就是我们作业汇报的所有内容。也希望大家能给我在 github 上的 repo 点点 star。Star 数超过 5 个，我会放出带注释的版本。超过 50 个，我可以位在坐的每个同学随时提供有关此代码的讲解。超过 500，我会放出将上述所有问题全部优化完的版本。心动不如行动，赶快召集你身边的人助力吧！
谢谢大家！