PVZ Cheater汇编小组作业报告

覃诗睿 朱旭 张泽卿

1. 开发环境

操作系统：Windows 10/11

IDE: Visual Studio 2022

内存信息获取工具：Cheat Engine 7.5

目标程序：PlantsVsZombies.exe(95版)

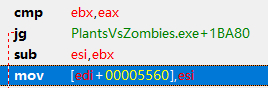
1. 实现原理
2. 进程内存读取的基础知识

首先区分以下概念：PID·分配基址·句柄

* PID是进程标识符，在进程开启时被分配
* 分配基址是进程对应的汇编代码块在进程的内存空间中的起始地址，即代码模块（本项目中为PlantsVsZombies.exe）在操作系统为进程分配的虚拟内存中的偏移量。
* 句柄是操作系统提供给程序引用和操作进程的机制，可以获取进程数据、访问进程内存等等。

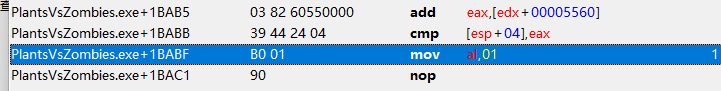
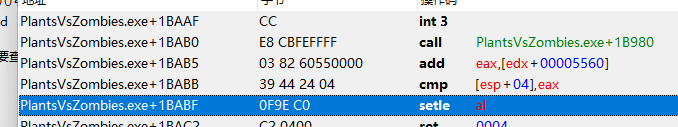
由此，要改写游戏代码，只要找到反汇编代码的分配基址+偏移量。

1. 其它
2. 实现的功能及难点
3. freeSun

首先找到阳光变量存储的地址，然后检查是什么改变了该变量，由此找到使用阳光时的代码位置，根据原理将减法代码删掉，就可以不消耗阳光种植。

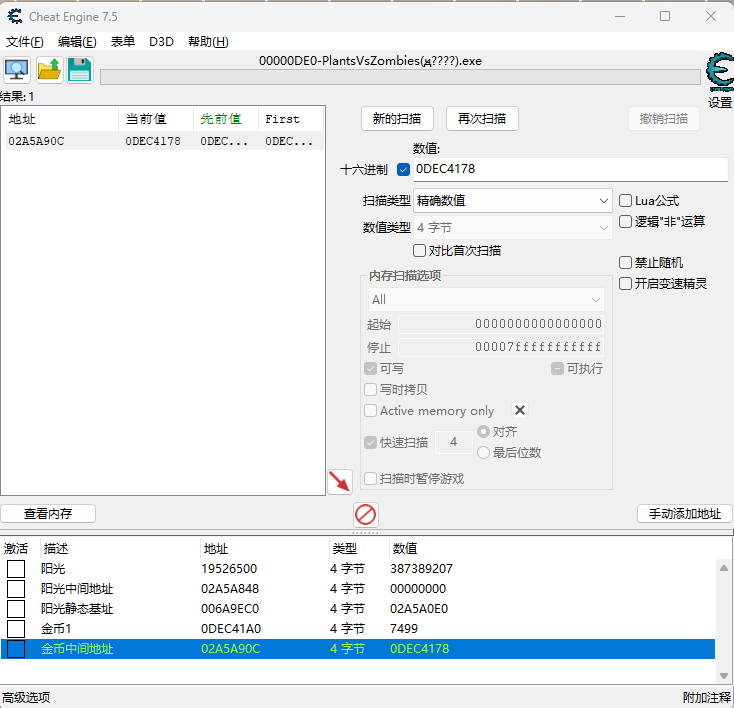
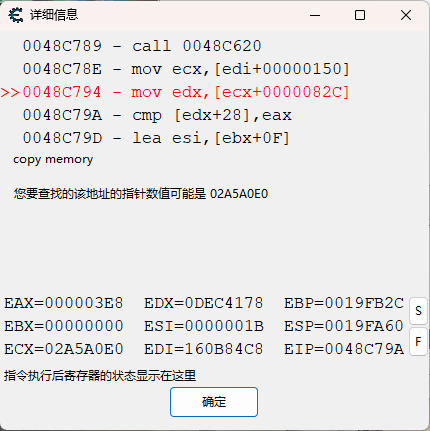
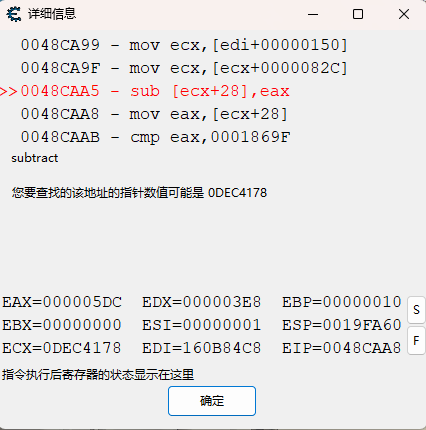
**难点**：仅完成以上操作后如果当前阳光不足，植物处于灰色无法种植。为此先寻找访问阳光变量的地址，发现有两个一直在访问的指令，经检测发现，一个是显示阳光数量的指令，另一个是检测植物阳光是否够用的代码。在检测阳光是否够用处，把比较设定的setle改成直接赋值mov即可。同时要把消耗阳光处的二次确认删掉。

**最终效果：**使用指令后，冷却结束的植物全部亮起，终止时不消耗阳光。再次使用取消该效果。

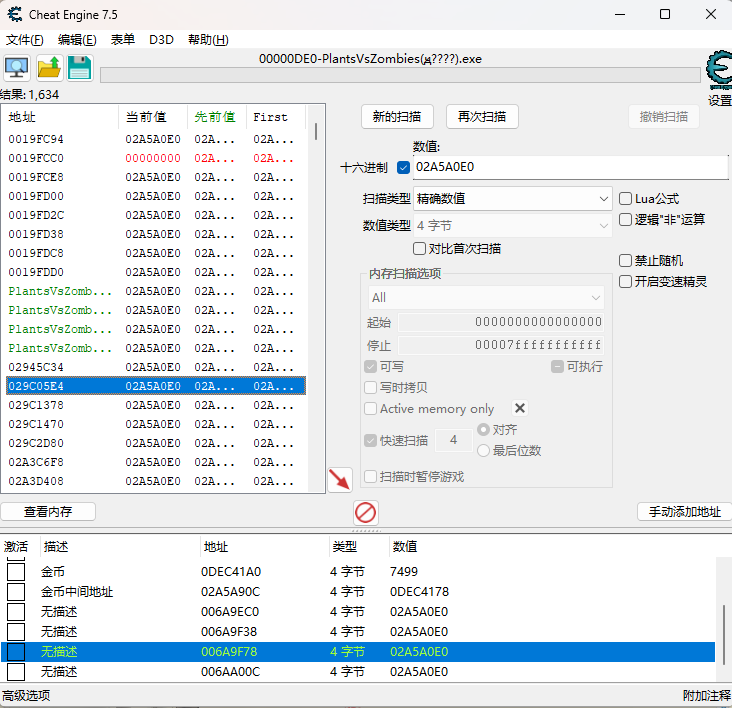


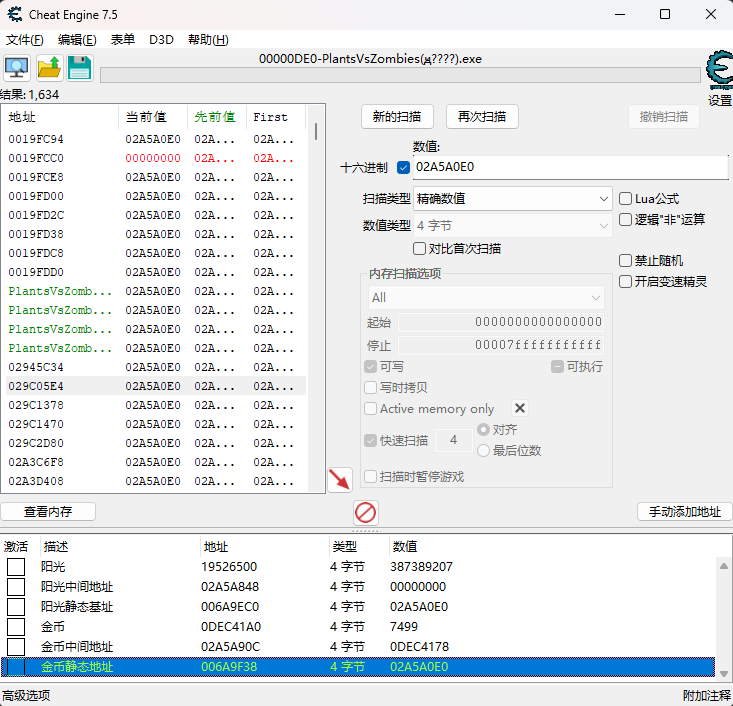
1. stealMoney

首先找到金币变量存储的地址，然后检查是什么改变了该变量，打开该地址的详细信息，在游戏中消耗或者收集金币，来找到改变的汇编语句，记录下偏移量1（如下图为28h）。然后按十六进制的方式查找详细信息显示的可能的地址指针数值0DEC4178，得到金币的中间地址，此时得到的地址并不是金币存储的静态基址，需要对这个金币中间地址进行二次搜索，继续搜索该中间地址0DEC4178，在得到的结果中选择查看是什么访问了这个地址，记录下偏移量2（如下图82Ch）。



继续搜索指针02A5A0E0，得到静态基址，下图左方四个绿色的地址代表某个值的基地址，通过再一次在游戏中修改金币值找到其中对应的金币的地址得到金币基地址6A9F38h。





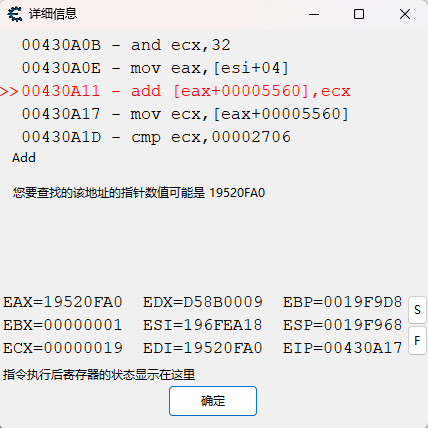
通过金币静态地址+偏移量2（82Ch）+偏移量1(28h)之后得到的地址，就是可以来修改改变金币的地址，在stealMoney.ams中将这个地址存储的值改为想要的值就可以得到修改后的金币值。

由于游戏的金币设定比较特殊（金币数量一定为10的整数倍），我们将修改后的金币值也为该数值的十倍。

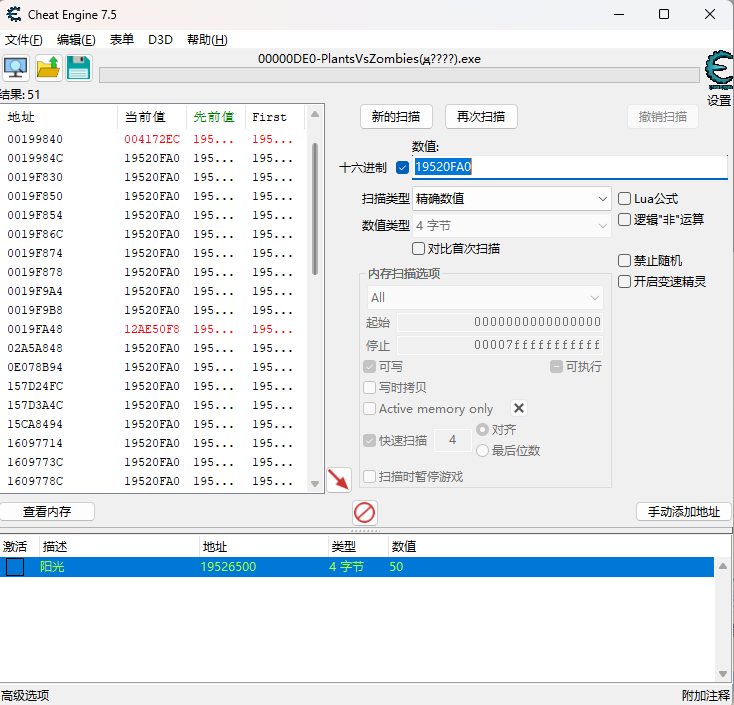
**难点**：《植物大战僵尸》游戏中对金币进行了一个小的加密操作，即金币地址中显示的值并非实际的金币值，而是金币值的十分之一，寻找该值相比其它的多了一环。在寻找金币的静态基址时，由于不清楚进行了几次指针的转换偏移，只能不停的做相同的搜索，查看，修改，再搜索的操作来得到金币的静态基址，这样才能在汇编中完成修改的效果。

**最终效果：**使用指令后，输入想要的金币值，就可以将游戏中的金币值改为输入数值的十倍。

1. stealSun

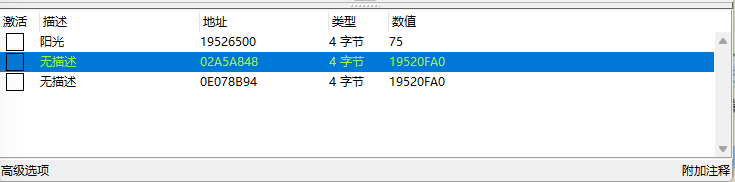
修改阳光值的原理与修改金币值的原理类似。首先找到阳光变量存储的地址，然后检查是什么改变了该变量，打开该地址的详细信息，在游戏中消耗或者收集阳光，来找到改变的汇编语句，记录下偏移量1（如下图为5560h）。然后按十六进制的方式查找详细信息显示的可能的地址指针数值19520FA0。

此时发现可能的地址有51个之多，需要在其中找到阳光的中间地址。发现02A5A848和0E078B94两个地址值相对于其它地址值较为特殊，对这两个地址信息进行查看什么访问了该地址的操作，在游戏中改变阳光值，发现0E078B94无法继续搜索得到静态基址，02A5A848可以得到。

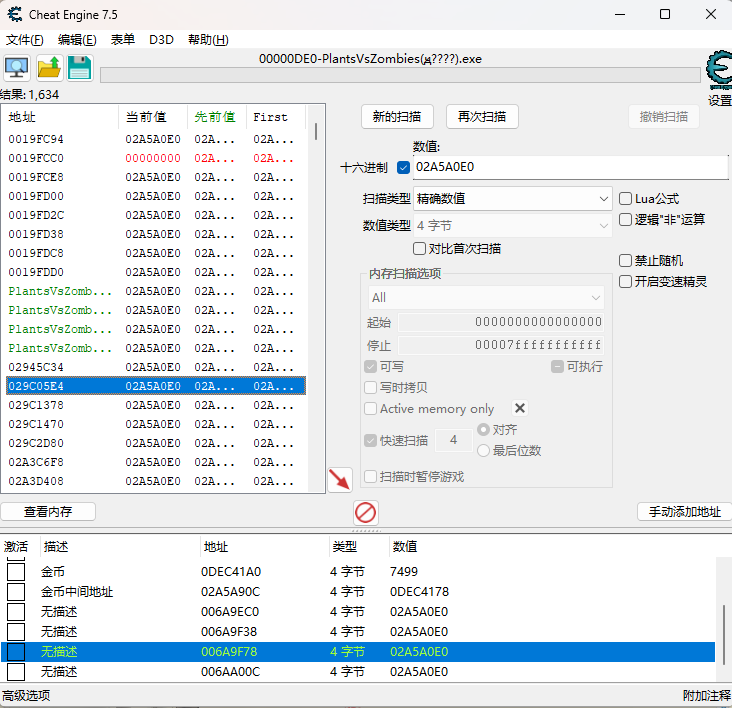


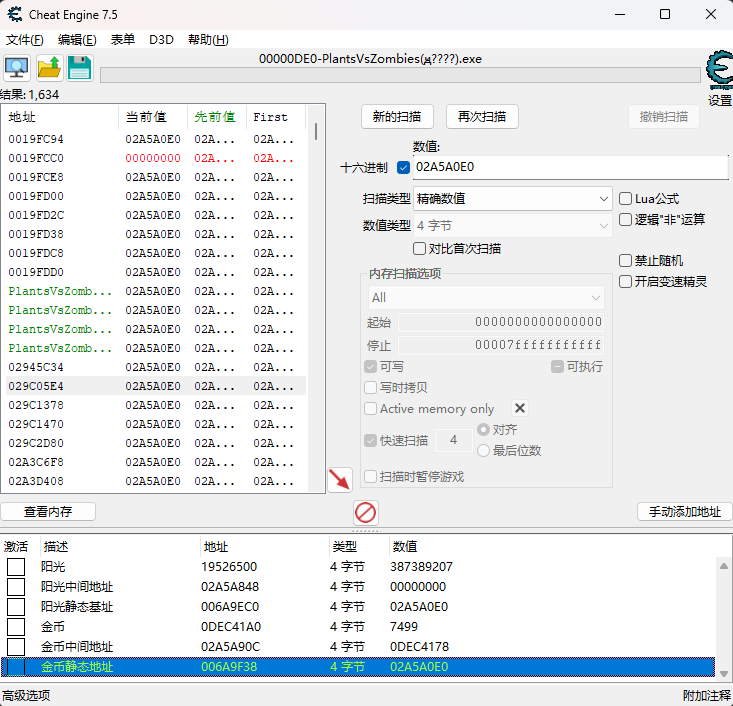
那么阳光的中间地址就是02A5A848，此时得到的地址也不是阳光存储的静态基址，需要对这个阳光中间地址进行二次搜索，继续搜索该中间地址，在得到的结果中选择查看是什么访问了这个地址，记录下偏移量2（如下图768h）。





继续搜索指针02A5A0E0，得到静态基址，下图左方四个绿色的地址代表某个值的基地址，通过再一次在游戏中消耗或收集阳光值找到其中对应的阳光的地址得到阳光基地址6A9EC0h。





通过阳光静态地址+偏移量2（768h）+偏移量1（5560h）之后得到的地址，就是可以来修改改变阳光的地址，在stealSun.ams中将这个地址存储的值改为想要的值就可以得到修改后的阳光值。

**难点**：在寻找阳光的静态基址时，同样由于不清楚进行了几次指针的转换偏移，只能不停的做相同的搜索，查看，修改，再搜索的操作来得到阳光的中间基址。而且与金币的修改不同，在搜索过程中搜索指针出现了51个相符合的地址，对查找正确的中间地址增添了不少麻烦，需要筛选掉不太可能的值然后进行挨个搜索查看的操作，最后得到阳光，这样才能在汇编中完成修改的效果。

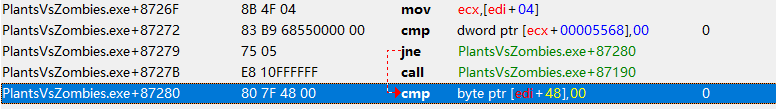
**最终效果：**使用指令后，输入想要的阳光，就可以将游戏中的阳光值改为输入的数值。

1. alwaysOn

首先尝试找到一个植物冷却标志变量，参考阳光影响的亮起，认为应该有个bool值确定植物是否冷却中，尝试后发现，植物冷却结束后有一个值设为1，冷却时设为0。接下来寻找改变该值的地址。找到了放置时清空和冷却结束置1两处代码，把放置后清空改掉即可。

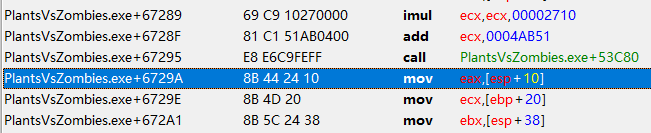
**难点**：以上操作出现问题，当游戏刚开始或刚放下某些植物后，处于冷却中的植物没有办法立刻停止冷却。为此研究置1部分代码，在这部分程序确认植物是否被选中，然后判断是否在冷却中，如果在冷却中就开始计数的更新。我们直接截断这一步骤。

**最终效果：**使用该指令后，所有植物立即取消冷却。再次使用取消。



1. randomBullet

首先找到决定子弹种类的语句，通过改变屏幕中的子弹种类来寻找。最终定位到了反汇编代码。这里esp索引的是植物相关属性，10偏移处存了植物子弹类型。



为了改变植物子弹类型，我们把这个语句改成先清空eax，再设置al的值，以确保机器指令仍是4个byte。

难点：以上方式仅仅实现了改变所有植物的子弹。我们想通过随机的方式，让每个植物在每个瞬间投出不一样的子弹。为此我们在收到指令后开启了一个线程，让它每100毫秒修改一次子弹的种类。这样处理之后，由于植物发射子弹的冷却时间较长，每株植物基本上会在不同时间发射。

最终效果：开启指令后，投射、发射类植物发射随机子弹。再次使用取消。

1. 创新点

全中文环境修改器。实现了比较有趣的功能，不只是作弊，而更加强调游戏趣味的增强。尤其是随机子弹等功能，让游戏有了更多乐趣！

1. 小组分工

框架及一些函数：朱旭

文档及一些函数：覃诗睿

演示及一些函数：张泽卿