汇编语言与逆向技术实验报告

Lab7 CTF（Capture The Flag）夺旗赛

学号：2112060　　姓名：孙璐　　专业：信息安全

一、　实验目的

1、熟悉静态反汇编工具 IDA Freeware；

2、掌握对二进制代码内部逻辑关系的分析；

3、掌握对二进制代码的修改和保存。

二、　实验原理

1. CTF

CTF 是一种流行的信息安全竞赛形式，可意译为“夺旗赛”。其大致流程是，参赛团队之间通过进行攻防对抗、程序分析等形式，率先从主办方给出的比赛环境中得到一串具有一定格式的字符串或其他内容，并将其提交给主办方，从而夺得分数。

CTF 竞赛模式具体分为以下三类：

一、解题模式（Jeopardy）

在解题模式 CTF 赛制中，参赛队伍可以通过互联网或者现场网络参与，这种模式的 CTF 竞赛与 ACM 编程竞赛、信息学奥赛比较类似，以解决网络安全技术挑战题目的分值和时间来排名，通常用于在线选拔赛。题目主要包含**逆向分析**、漏洞挖掘与利用、Web 渗透、密码、取证、隐写、安全编程等类别。

二、攻防模式（Attack-Defense）

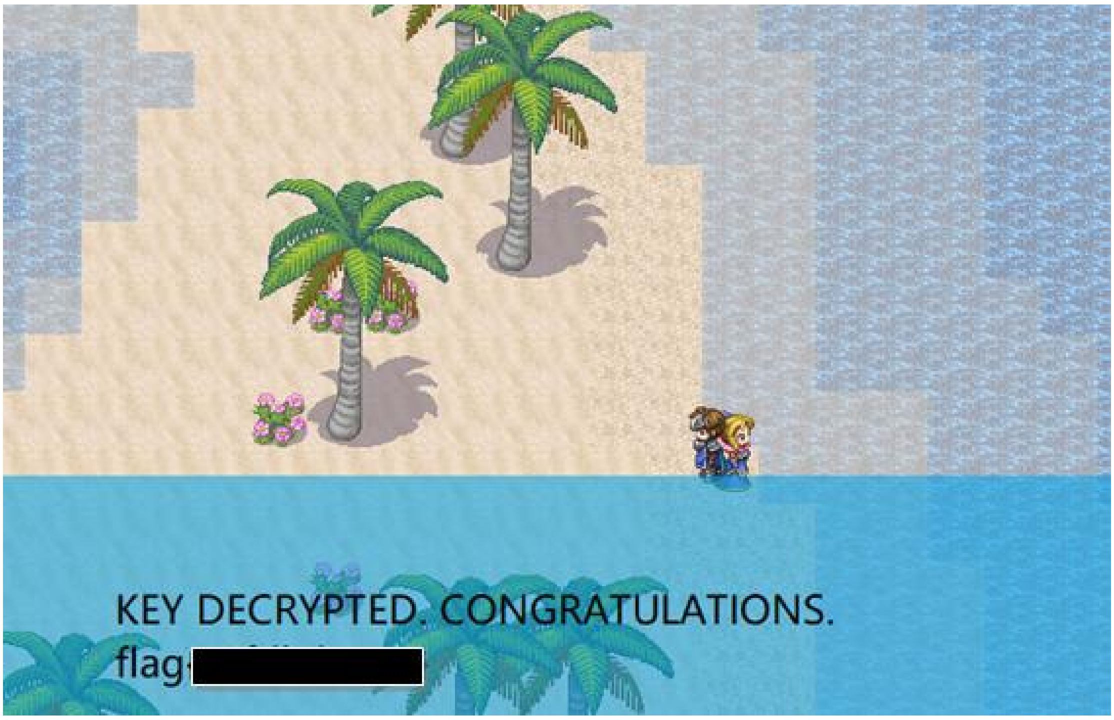
在攻防模式 CTF 赛制中，参赛队伍在网络空间互相进行攻击和防守，挖掘网络服务漏洞并攻击对手服务来得分，修补自身服务漏洞进行防御来避免丢分。

三、混合模式（Mix）

结合了解题模式与攻防模式的 CTF 赛制，比如参赛队伍通过解题可以获取一些初始分数，然后通过攻防对抗进行得分增减的零和游戏，最终以得分高低分出胜负。

2. 解题

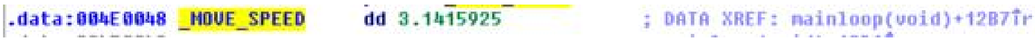
Flag 隐藏在 game.exe 的二进制代码中。通过对 game.exe 的修改，使 game.exe 能够顺利的执行，完成对 Flag 的解密。



KEY DECRYPTED. CONGRATULATIONS.
flag[redacted]

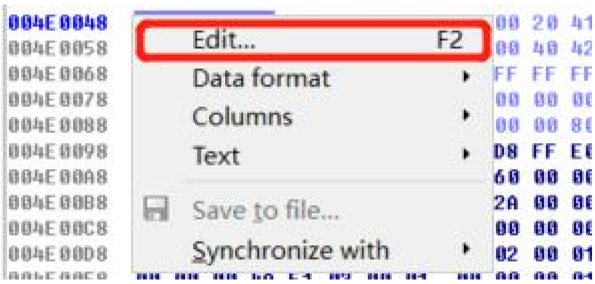
1) 技巧 A：利用 IDA 修改静态资源

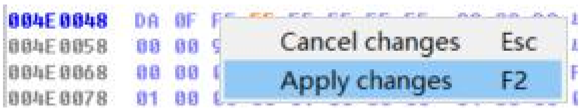
- 第一步，在反汇编代码中（IDA View）找到静态资源。



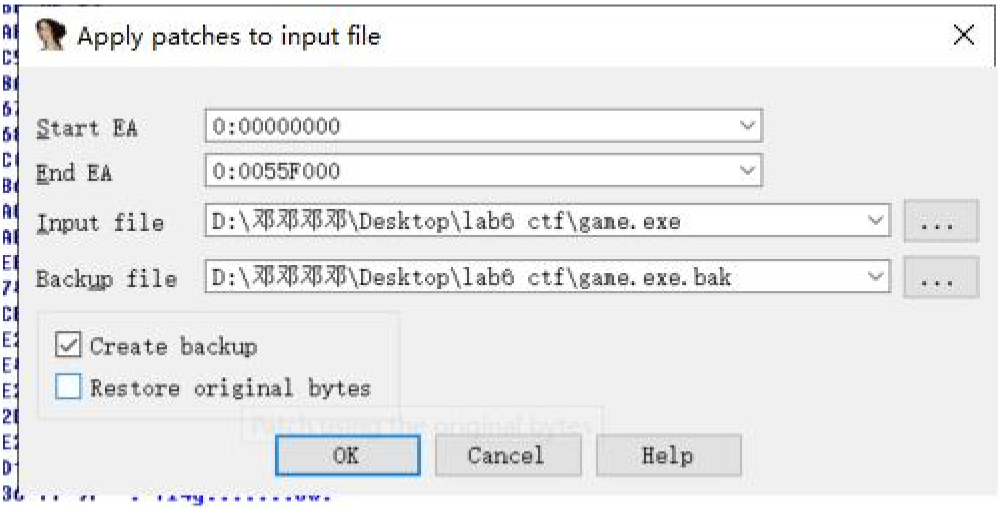
```
.data:004E0048 _MOVE_SPEED     dd 3.1415925          ; DATA XREF: mainloop(void)+12B7↑r
```

- 第二步，在十六进制视图中（Hex View）找到指定区域，右键选择 Edit 对资源进行修改。修改完毕后，右键选择 Apply changes 应用修改。





- 第三步，点击 Edit->Patch program->Apply patches to input file，建议选中创建备份的选项，完成修改。


Apply patches to input file

Start EA 0:00000000

End EA 0:0055F000

Input file D:\邓邓邓邓\Desktop\lab6 ctf\game.exe

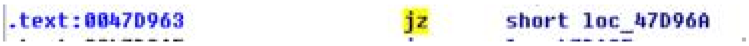
Backup file D:\邓邓邓邓\Desktop\lab6 ctf\game.exe.bak

☑ Create backup

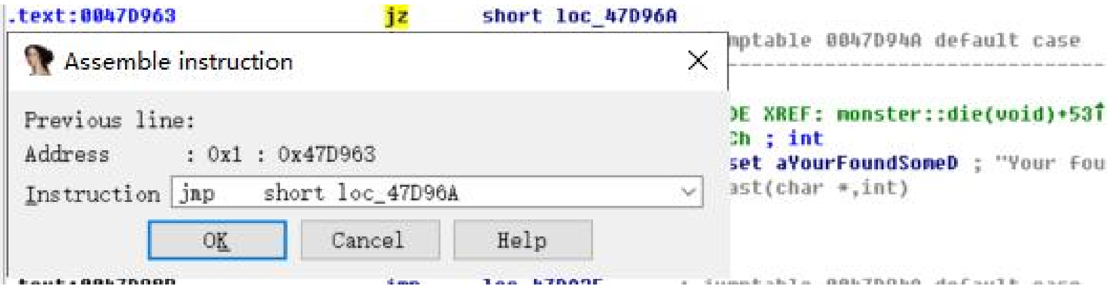
□ Restore original bytes

2) 技巧 B：利用 IDA 修改汇编指令

- 第一步，在反汇编代码中（IDA View）找到需要修改的汇编指令。


```
.text:0047D963                 jz      short loc_47D96A
```

- 第二步，点击 Edit->Patch program->Assemble，输入新的汇编指令。


```
.text:0047D963                 jz      short loc_47D96A
```

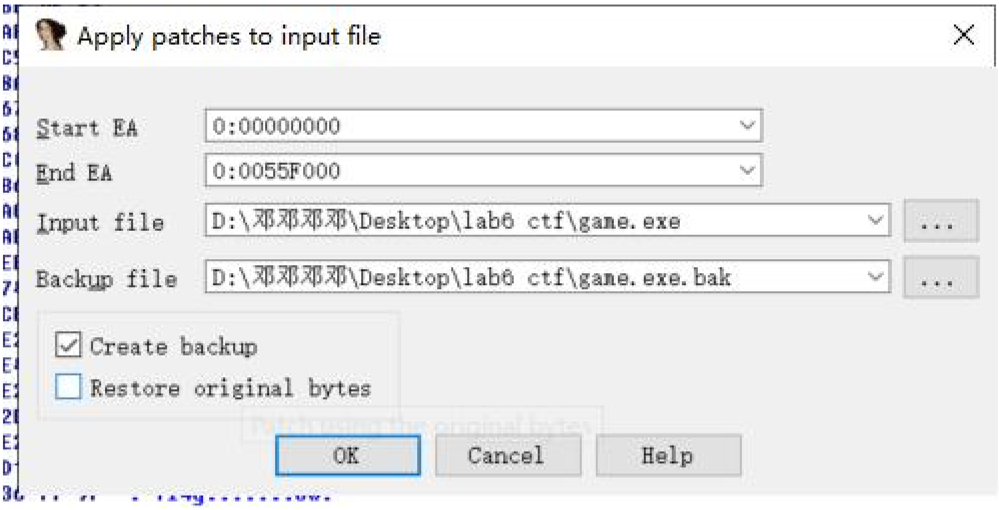
Assemble instruction

Previous line:

Address : 0x1 : 0x47D963

Instruction jmp short loc_47D96A

- 第三步，点击 Edit->Patch program->Apply patches to input file，建议选中创建备份的选项，完成修改。


Apply patches to input file

Start EA 0:00000000

End EA 0:0055F000

Input file D:\邓邓邓邓\Desktop\lab6 ctf\game.exe

Backup file D:\邓邓邓邓\Desktop\lab6 ctf\game.exe.bak

☑ Create backup

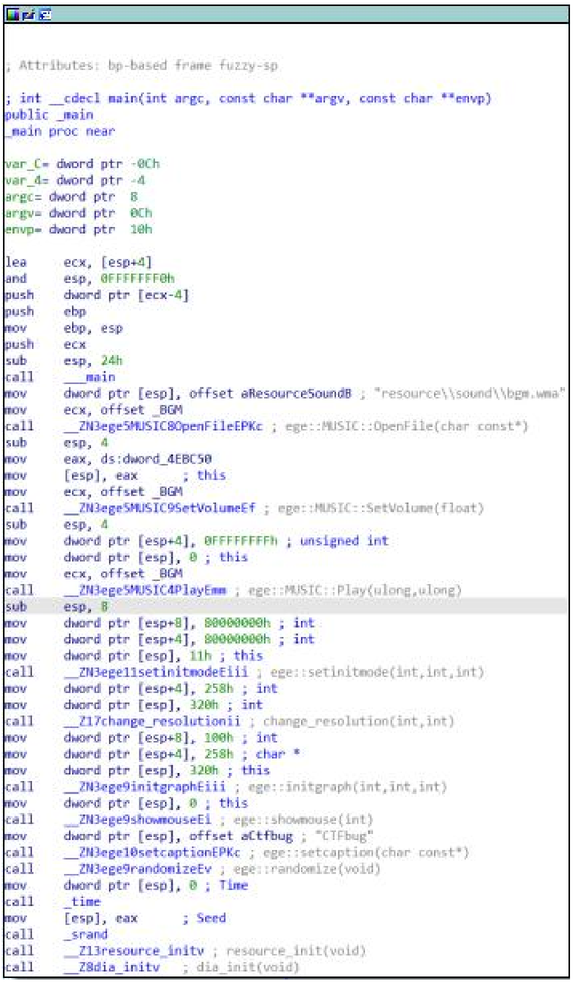
□ Restore original bytes

三、 实验报告

1. 逆向分析 game.exe 二进制代码的主要逻辑结构和重要数据。
2. 修改 game.exe 二进制代码，获得最后的 Flag。实验报告要说明逆向分析、代码修改的具体过程，以及最后获得的 Flag。
3. 实验报告的提交时间：12 月 21 日之前提交。

四、 逆向分析 game.exe 二进制代码的主要逻辑结构和重要数据

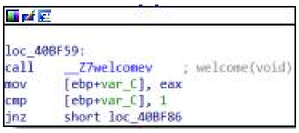
（1）



```
; Attributes: bp-based frame fuzzy-sp

; int __cdecl main(int argc, const char **argv, const char **envp)
public _main
_main proc near

var_C= dword ptr -0Ch
var_4= dword ptr -4
argc= dword ptr  8
argv= dword ptr  0Ch
envp= dword ptr  10h

lea     ecx, [esp+4]
and     esp, 0FFFFFFF0h
push    dword ptr [ecx-4]
push    ebp
mov     ebp, esp
push    ecx
sub     esp, 24h
call    ___main
mov     dword ptr [esp], offset aResourceSoundB ; "resource\\sound\\bgm.wma"
mov     ecx, offset _BGM
call    __ZN3ege5MUSIC8OpenFileEPKc ; ege::MUSIC::OpenFile(char const*)
sub     esp, 4
mov     eax, ds:dword_4EBC50
mov     [esp], eax      ; this
mov     ecx, offset _BGM
call    __ZN3ege5MUSIC9SetVolumeEf ; ege::MUSIC::SetVolume(float)
sub     esp, 4
mov     dword ptr [esp+4], 0FFFFFFFFh ; unsigned int
mov     dword ptr [esp], 0 ; this
mov     ecx, offset _BGM
call    __ZN3ege5MUSIC4PlayEmm ; ege::MUSIC::Play(ulong,ulong)
sub     esp, 8
mov     dword ptr [esp+8], 80000000h ; int
mov     dword ptr [esp+4], 80000000h ; int
mov     dword ptr [esp], 11h ; this
call    __ZN3ege11setinitmodeEiii ; ege::setinitmode(int,int,int)
mov     dword ptr [esp+4], 258h ; int
mov     dword ptr [esp], 320h ; int
call    __Z17change_resolutionii ; change_resolution(int,int)
mov     dword ptr [esp+8], 100h ; int
mov     dword ptr [esp+4], 258h ; char *
mov     dword ptr [esp], 320h ; this
call    __ZN3ege9initgraphEiii ; ege::initgraph(int,int,int)
mov     dword ptr [esp], 0 ; this
call    __ZN3ege9showmouseEi ; ege::showmouse(int)
mov     dword ptr [esp], offset aCtfbug ; "CTFbug"
call    __ZN3ege10setcaptionEPKc ; ege::setcaption(char const*)
call    __ZN3ege9randomizeEv ; ege::randomize(void)
mov     dword ptr [esp], 0 ; Time
call    _time
mov     [esp], eax      ; Seed
call    _srand
call    __Z13resource_initv ; resource_init(void)
call    __Z8dia_initv   ; dia_init(void)
```

背景音乐，背景图片的播放和展示设置。包括如音量设置，打开文件设置，键盘移动设置等。

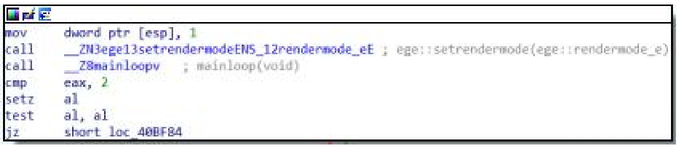
（2）



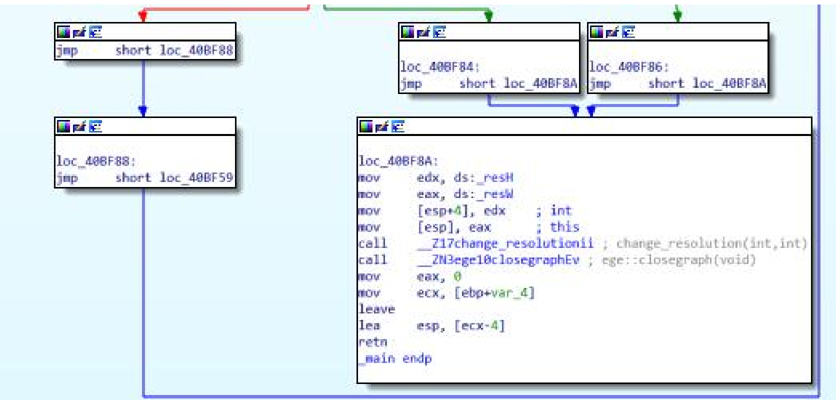
```
loc_40BF59:
call    __Z7welcomev    ; welcome(void)
mov     [ebp+var_C], eax
cmp     [ebp+var_C], 1
jnz     short loc_40BF86
```

进入游戏界面

（3）



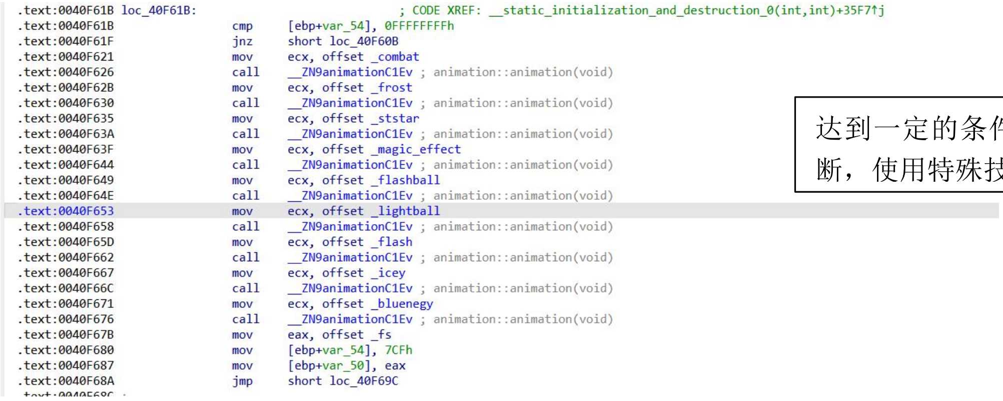
```
mov     dword ptr [esp], 1
call    __ZN3ege13setrendermodeENS_12rendermode_eE ; ege::setrendermode(ege::rendermode_e)
call    __Z8mainloopv   ; mainloop(void)
cmp     eax, 2
setz    al
test    al, al
jz      short loc_40BF84
```



```
jmp     short loc_40BF88

loc_40BF88:
jmp     short loc_40BF59

loc_40BF84:
jmp     short loc_40BF8A

loc_40BF86:
jmp     short loc_40BF8A

loc_40BF8A:
mov     edx, ds:_resH
mov     eax, ds:_resW
mov     [esp+4], edx    ; int
mov     [esp], eax      ; this
call    __Z17change_resolutionii ; change_resolution(int,int)
call    __ZN3ege10closegraphEv ; ege::closegraph(void)
mov     eax, 0
mov     ecx, [ebp+var_4]
leave
lea     esp, [ecx-4]
retn
_main endp
```

根据不同条件判断跳转到相应函数的位置

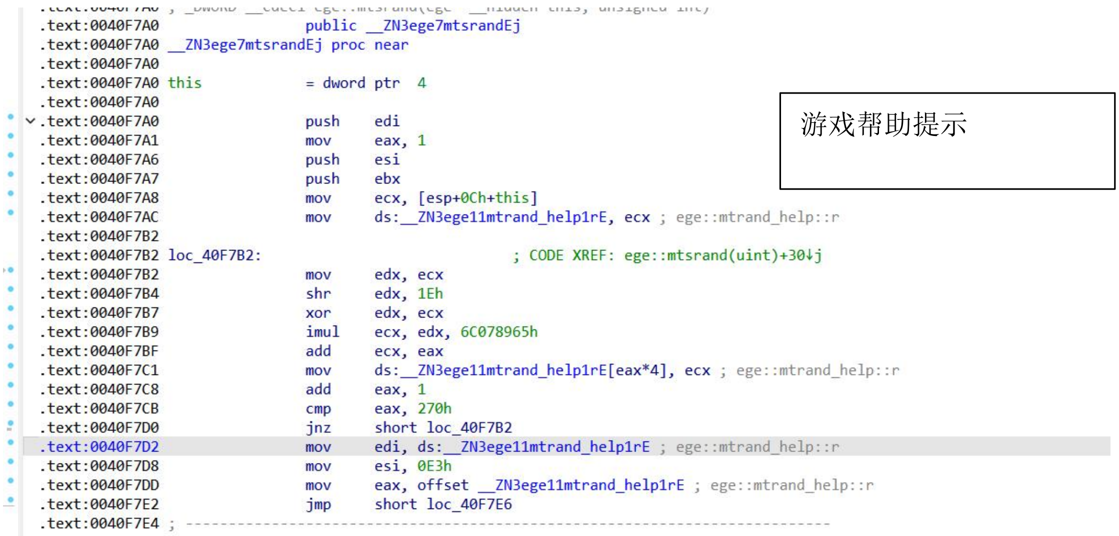
（4）



```
.text:0040F61B loc_40F61B:                             ; CODE XREF: __static_initialization_and_destruction_0(int,int)+35F7↑j
.text:0040F61B                 cmp     [ebp+var_54], 0FFFFFFFFh
.text:0040F61F                 jnz     short loc_40F60B
.text:0040F621                 mov     ecx, offset _combat
.text:0040F626                 call    __ZN9animationC1Ev ; animation::animation(void)
.text:0040F62B                 mov     ecx, offset _frost
.text:0040F630                 call    __ZN9animationC1Ev ; animation::animation(void)
.text:0040F635                 mov     ecx, offset _ststar
.text:0040F63A                 call    __ZN9animationC1Ev ; animation::animation(void)
.text:0040F63F                 mov     ecx, offset _magic_effect
.text:0040F644                 call    __ZN9animationC1Ev ; animation::animation(void)
.text:0040F649                 mov     ecx, offset _flashball
.text:0040F64E                 call    __ZN9animationC1Ev ; animation::animation(void)
.text:0040F653                 mov     ecx, offset _lightball
.text:0040F658                 call    __ZN9animationC1Ev ; animation::animation(void)
.text:0040F65D                 mov     ecx, offset _flash
.text:0040F662                 call    __ZN9animationC1Ev ; animation::animation(void)
.text:0040F667                 mov     ecx, offset _icey
.text:0040F66C                 call    __ZN9animationC1Ev ; animation::animation(void)
.text:0040F671                 mov     ecx, offset _bluenegy
.text:0040F676                 call    __ZN9animationC1Ev ; animation::animation(void)
.text:0040F67B                 mov     eax, offset _fs
.text:0040F680                 mov     [ebp+var_54], 7CFh
.text:0040F687                 mov     [ebp+var_50], eax
.text:0040F68A                 jmp     short loc_40F69C
.text:0040F68C ;
```

达到一定的条件进行判断，使用特殊技能

（5）

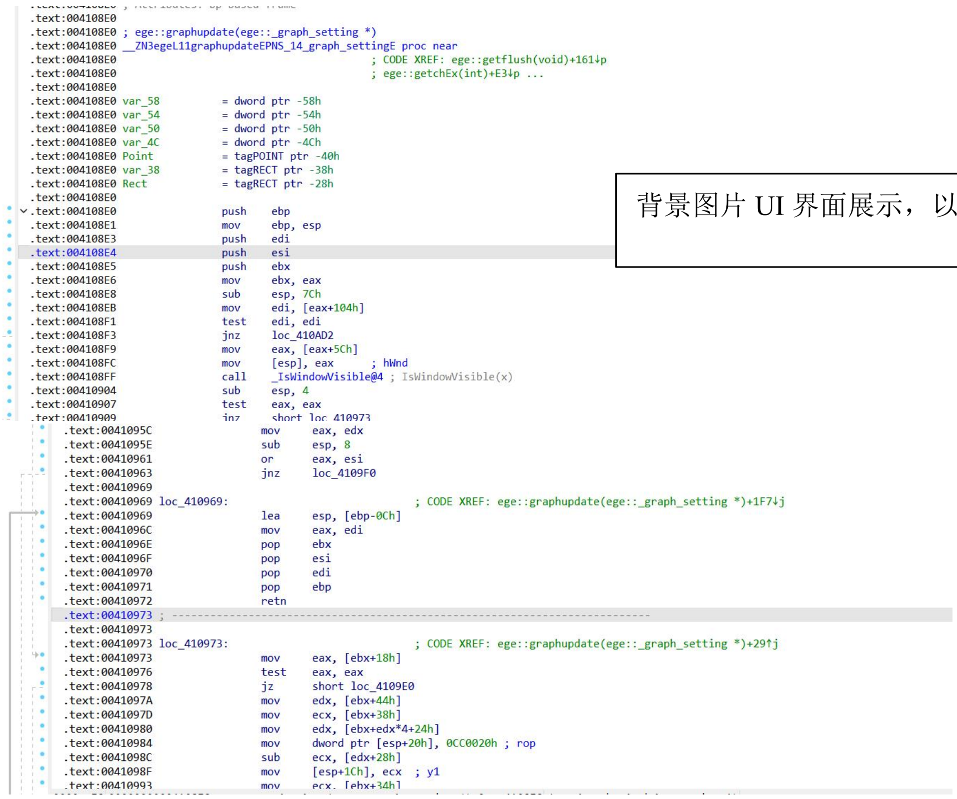


```
.text:0040F7A0 ; _DWORD __cdecl ege::mtsrand(ege *__hidden this, unsigned int)
.text:0040F7A0                 public __ZN3ege7mtsrandEj
.text:0040F7A0 __ZN3ege7mtsrandEj proc near
.text:0040F7A0
.text:0040F7A0 this            = dword ptr  4
.text:0040F7A0
.text:0040F7A0                 push    edi
.text:0040F7A1                 mov     eax, 1
.text:0040F7A6                 push    esi
.text:0040F7A7                 push    ebx
.text:0040F7A8                 mov     ecx, [esp+0Ch+this]
.text:0040F7AC                 mov     ds:__ZN3ege11mtrand_help1rE, ecx ; ege::mtrand_help::r
.text:0040F7B2
.text:0040F7B2 loc_40F7B2:                             ; CODE XREF: ege::mtsrand(uint)+30↓j
.text:0040F7B2                 mov     edx, ecx
.text:0040F7B4                 shr     edx, 1Eh
.text:0040F7B7                 xor     edx, ecx
.text:0040F7B9                 imul    ecx, edx, 6C078965h
.text:0040F7BF                 add     ecx, eax
.text:0040F7C1                 mov     ds:__ZN3ege11mtrand_help1rE[eax*4], ecx ; ege::mtrand_help::r
.text:0040F7C8                 add     eax, 1
.text:0040F7CB                 cmp     eax, 270h
.text:0040F7D0                 jnz     short loc_40F7B2
.text:0040F7D2                 mov     edi, ds:__ZN3ege11mtrand_help1rE ; ege::mtrand_help::r
.text:0040F7D8                 mov     esi, 0E3h
.text:0040F7DD                 mov     eax, offset __ZN3ege11mtrand_help1rE ; ege::mtrand_help::r
.text:0040F7E2                 jmp     short loc_40F7E6
.text:0040F7E4 ; ---------------------------------------------------------------------------
```

游戏帮助提示

（6）

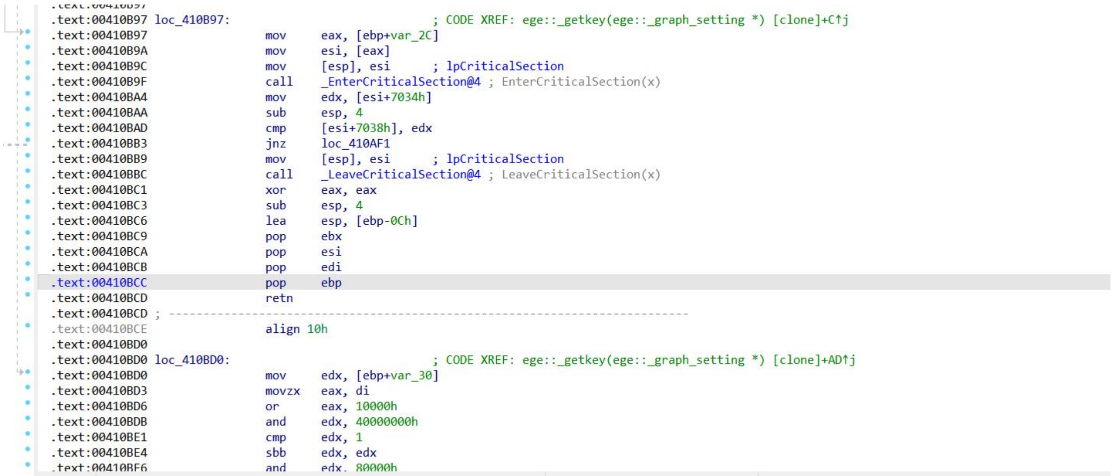
背景图片 UI 界面展示，以及背景切换



```
.text:004108E0 ; Attributes: bp-based frame
.text:004108E0
.text:004108E0 ; ege::graphupdate(ege::_graph_setting *)
.text:004108E0 __ZN3egeL11graphupdateEPNS_14_graph_settingE proc near
.text:004108E0                                         ; CODE XREF: ege::getflush(void)+161↓p
.text:004108E0                                         ; ege::getchEx(int)+E3↓p ...
.text:004108E0
.text:004108E0 var_58          = dword ptr -58h
.text:004108E0 var_54          = dword ptr -54h
.text:004108E0 var_50          = dword ptr -50h
.text:004108E0 var_4C          = dword ptr -4Ch
.text:004108E0 Point           = tagPOINT ptr -40h
.text:004108E0 var_38          = tagRECT ptr -38h
.text:004108E0 Rect            = tagRECT ptr -28h
.text:004108E0
.text:004108E0                 push    ebp
.text:004108E1                 mov     ebp, esp
.text:004108E3                 push    edi
.text:004108E4                 push    esi
.text:004108E5                 push    ebx
.text:004108E6                 mov     ebx, eax
.text:004108E8                 sub     esp, 7Ch
.text:004108EB                 mov     edi, [eax+104h]
.text:004108F1                 test    edi, edi
.text:004108F3                 jnz     loc_410AD2
.text:004108F9                 mov     eax, [eax+5Ch]
.text:004108FC                 mov     [esp], eax      ; hWnd
.text:004108FF                 call    _IsWindowVisible@4 ; IsWindowVisible(x)
.text:00410904                 sub     esp, 4
.text:00410907                 test    eax, eax
.text:00410909                 jnz     short loc_410973
```

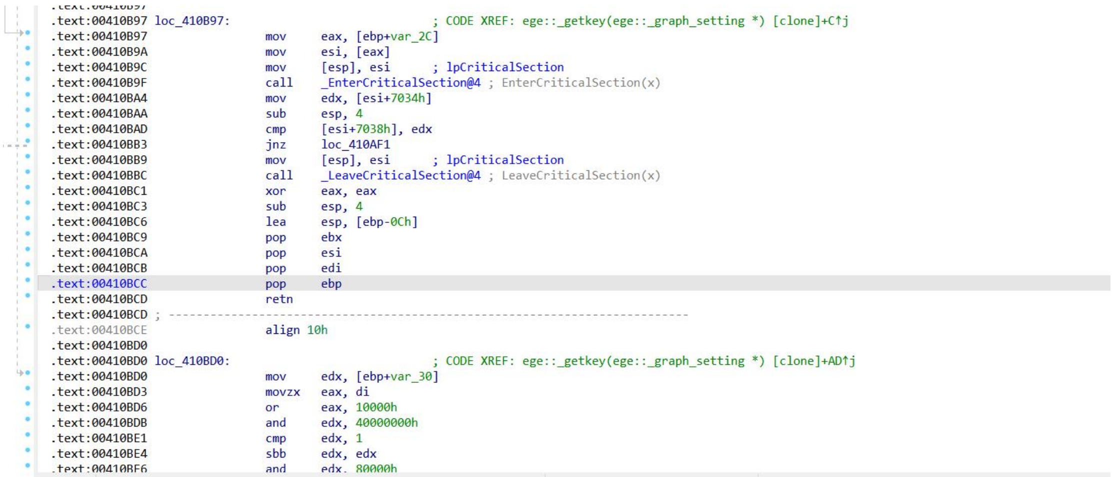
```
.text:0041095C                 mov     eax, edx
.text:0041095E                 sub     esp, 8
.text:00410961                 or      eax, esi
.text:00410963                 jnz     loc_4109F0
.text:00410969
.text:00410969 loc_410969:                             ; CODE XREF: ege::graphupdate(ege::_graph_setting *)+1F7↓j
.text:00410969                 lea     esp, [ebp-0Ch]
.text:0041096C                 mov     eax, edi
.text:0041096E                 pop     ebx
.text:0041096F                 pop     esi
.text:00410970                 pop     edi
.text:00410971                 pop     ebp
.text:00410972                 retn
.text:00410973 ; ---------------------------------------------------------------------------
.text:00410973
.text:00410973 loc_410973:                             ; CODE XREF: ege::graphupdate(ege::_graph_setting *)+29↑j
.text:00410973                 mov     eax, [ebx+18h]
.text:00410976                 test    eax, eax
.text:00410978                 jz      short loc_4109E0
.text:0041097A                 mov     edx, [ebx+44h]
.text:0041097D                 mov     ecx, [ebx+38h]
.text:00410980                 mov     edx, [ebx+edx*4+24h]
.text:00410984                 mov     dword ptr [esp+20h], 0CC0020h ; rop
.text:0041098C                 sub     ecx, [edx+28h]
.text:0041098F                 mov     [esp+1Ch], ecx  ; y1
.text:00410993                 mov     ecx, [ebx+34h]
```

（7）

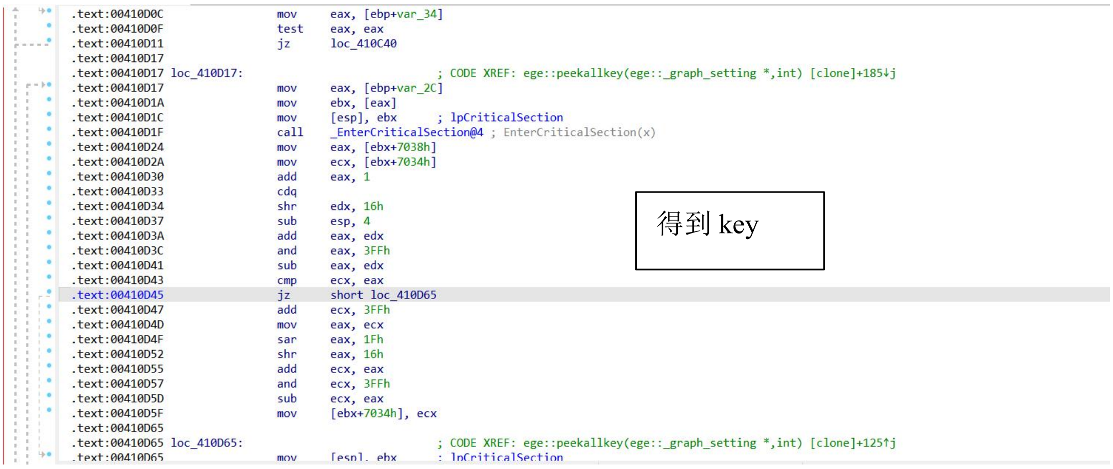


```
.text:00410B97
.text:00410B97 loc_410B97:                             ; CODE XREF: ege::_getkey(ege::_graph_setting *) [clone]+C↑j
.text:00410B97                 mov     eax, [ebp+var_2C]
.text:00410B9A                 mov     esi, [eax]
.text:00410B9C                 mov     [esp], esi      ; lpCriticalSection
.text:00410B9F                 call    _EnterCriticalSection@4 ; EnterCriticalSection(x)
.text:00410BA4                 mov     edx, [esi+7034h]
.text:00410BAA                 sub     esp, 4
.text:00410BAD                 cmp     [esi+7038h], edx
.text:00410BB3                 jnz     loc_410AF1
.text:00410BB9                 mov     [esp], esi      ; lpCriticalSection
.text:00410BBC                 call    _LeaveCriticalSection@4 ; LeaveCriticalSection(x)
.text:00410BC1                 xor     eax, eax
.text:00410BC3                 sub     esp, 4
.text:00410BC6                 lea     esp, [ebp-0Ch]
.text:00410BC9                 pop     ebx
.text:00410BCA                 pop     esi
.text:00410BCB                 pop     edi
.text:00410BCC                 pop     ebp
.text:00410BCD                 retn
.text:00410BCD ; ---------------------------------------------------------------------------
.text:00410BCE                 align 10h
.text:00410BD0
.text:00410BD0 loc_410BD0:                             ; CODE XREF: ege::_getkey(ege::_graph_setting *) [clone]+AD↑j
.text:00410BD0                 mov     edx, [ebp+var_30]
.text:00410BD3                 movzx   eax, di
.text:00410BD6                 or      eax, 10000h
.text:00410BDB                 and     edx, 40000000h
.text:00410BE1                 cmp     edx, 1
.text:00410BE4                 sbb     edx, edx
.text:00410BE6                 and     edx, 80000h
```

（8）



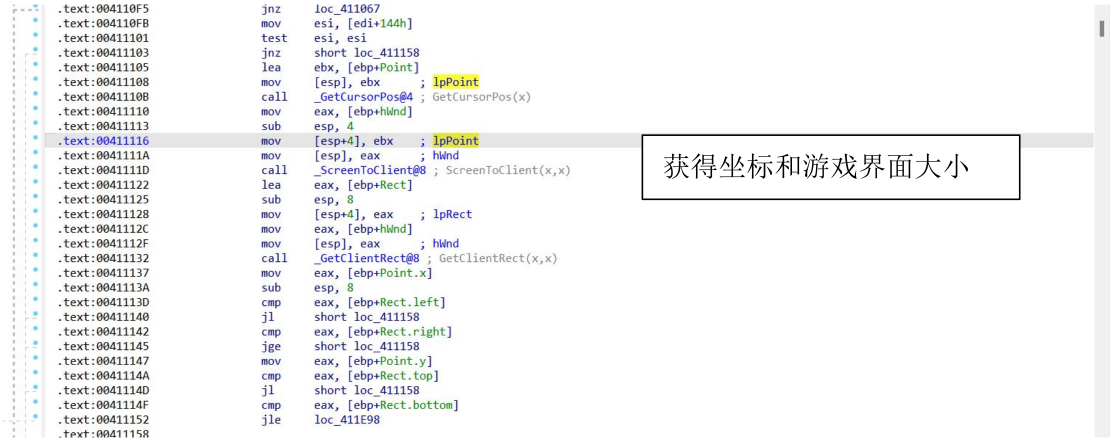
```
.text:00410B97
.text:00410B97 loc_410B97:                             ; CODE XREF: ege::_getkey(ege::_graph_setting *) [clone]+C↑j
.text:00410B97                 mov     eax, [ebp+var_2C]
.text:00410B9A                 mov     esi, [eax]
.text:00410B9C                 mov     [esp], esi      ; lpCriticalSection
.text:00410B9F                 call    _EnterCriticalSection@4 ; EnterCriticalSection(x)
.text:00410BA4                 mov     edx, [esi+7034h]
.text:00410BAA                 sub     esp, 4
.text:00410BAD                 cmp     [esi+7038h], edx
.text:00410BB3                 jnz     loc_410AF1
.text:00410BB9                 mov     [esp], esi      ; lpCriticalSection
.text:00410BBC                 call    _LeaveCriticalSection@4 ; LeaveCriticalSection(x)
.text:00410BC1                 xor     eax, eax
.text:00410BC3                 sub     esp, 4
.text:00410BC6                 lea     esp, [ebp-0Ch]
.text:00410BC9                 pop     ebx
.text:00410BCA                 pop     esi
.text:00410BCB                 pop     edi
.text:00410BCC                 pop     ebp
.text:00410BCD                 retn
.text:00410BCD ; ---------------------------------------------------------------------------
.text:00410BCE                 align 10h
.text:00410BD0
.text:00410BD0 loc_410BD0:                             ; CODE XREF: ege::_getkey(ege::_graph_setting *) [clone]+AD↑j
.text:00410BD0                 mov     edx, [ebp+var_30]
.text:00410BD3                 movzx   eax, di
.text:00410BD6                 or      eax, 10000h
.text:00410BDB                 and     edx, 40000000h
.text:00410BE1                 cmp     edx, 1
.text:00410BE4                 sbb     edx, edx
.text:00410BE6                 and     edx, 80000h
```



```
.text:00410D0C                 mov     eax, [ebp+var_34]
.text:00410D0F                 test    eax, eax
.text:00410D11                 jz      loc_410C40
.text:00410D17
.text:00410D17 loc_410D17:                             ; CODE XREF: ege::peekallkey(ege::_graph_setting *,int) [clone]+185↓j
.text:00410D17                 mov     eax, [ebp+var_2C]
.text:00410D1A                 mov     ebx, [eax]
.text:00410D1C                 mov     [esp], ebx      ; lpCriticalSection
.text:00410D1F                 call    _EnterCriticalSection@4 ; EnterCriticalSection(x)
.text:00410D24                 mov     eax, [ebx+7038h]
.text:00410D2A                 mov     ecx, [ebx+7034h]
.text:00410D30                 add     eax, 1
.text:00410D33                 cdq
.text:00410D34                 shr     edx, 16h
.text:00410D37                 sub     esp, 4
.text:00410D3A                 add     eax, edx
.text:00410D3C                 and     eax, 3FFh
.text:00410D41                 sub     eax, edx
.text:00410D43                 cmp     ecx, eax
.text:00410D45                 jz      short loc_410D65
.text:00410D47                 add     ecx, 3FFh
.text:00410D4D                 mov     eax, ecx
.text:00410D4F                 sar     eax, 1Fh
.text:00410D52                 shr     eax, 16h
.text:00410D55                 add     ecx, eax
.text:00410D57                 and     ecx, 3FFh
.text:00410D5D                 sub     ecx, eax
.text:00410D5F                 mov     [ebx+7034h], ecx
.text:00410D65
.text:00410D65 loc_410D65:                             ; CODE XREF: ege::peekallkey(ege::_graph_setting *,int) [clone]+125↑j
.text:00410D65                 mov     [esp], ebx      ; lpCriticalSection
```

得到 key

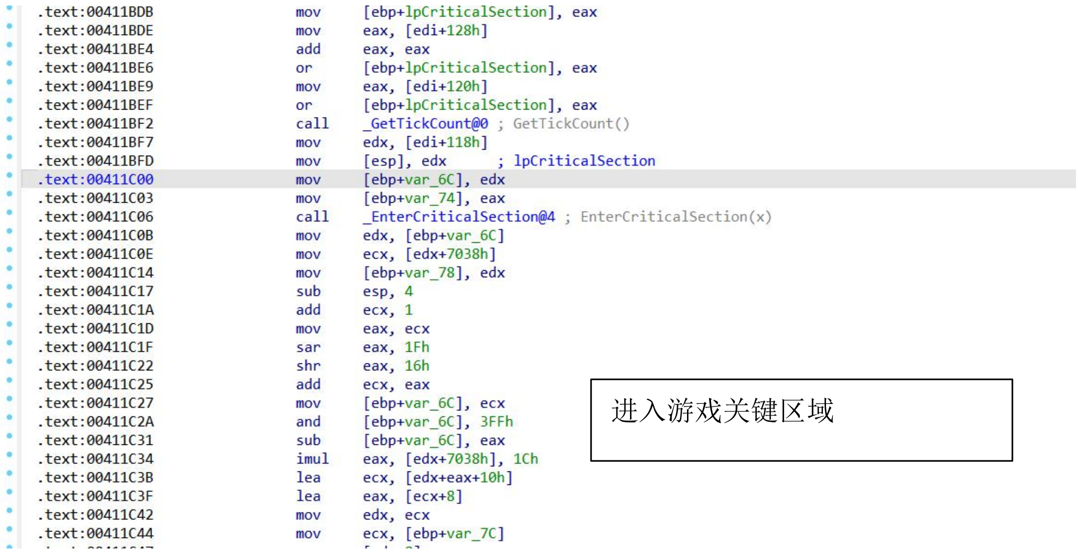
（9）



```
.text:004110F5                 jnz     loc_411067
.text:004110FB                 mov     esi, [edi+144h]
.text:00411101                 test    esi, esi
.text:00411103                 jnz     short loc_411158
.text:00411105                 lea     ebx, [ebp+Point]
.text:00411108                 mov     [esp], ebx      ; lpPoint
.text:0041110B                 call    _GetCursorPos@4 ; GetCursorPos(x)
.text:00411110                 mov     eax, [ebp+hWnd]
.text:00411113                 sub     esp, 4
.text:00411116                 mov     [esp+4], ebx    ; lpPoint
.text:0041111A                 mov     [esp], eax      ; hWnd
.text:0041111D                 call    _ScreenToClient@8 ; ScreenToClient(x,x)
.text:00411122                 lea     eax, [ebp+Rect]
.text:00411125                 sub     esp, 8
.text:00411128                 mov     [esp+4], eax    ; lpRect
.text:0041112C                 mov     eax, [ebp+hWnd]
.text:0041112F                 mov     [esp], eax      ; hWnd
.text:00411132                 call    _GetClientRect@8 ; GetClientRect(x,x)
.text:00411137                 mov     eax, [ebp+Point.x]
.text:0041113A                 sub     esp, 8
.text:0041113D                 cmp     eax, [ebp+Rect.left]
.text:00411140                 jl      short loc_411158
.text:00411142                 cmp     eax, [ebp+Rect.right]
.text:00411145                 jge     short loc_411158
.text:00411147                 mov     eax, [ebp+Point.y]
.text:0041114A                 cmp     eax, [ebp+Rect.top]
.text:0041114D                 jl      short loc_411158
.text:0041114F                 cmp     eax, [ebp+Rect.bottom]
.text:00411152                 jle     loc_411E98
.text:00411158
```

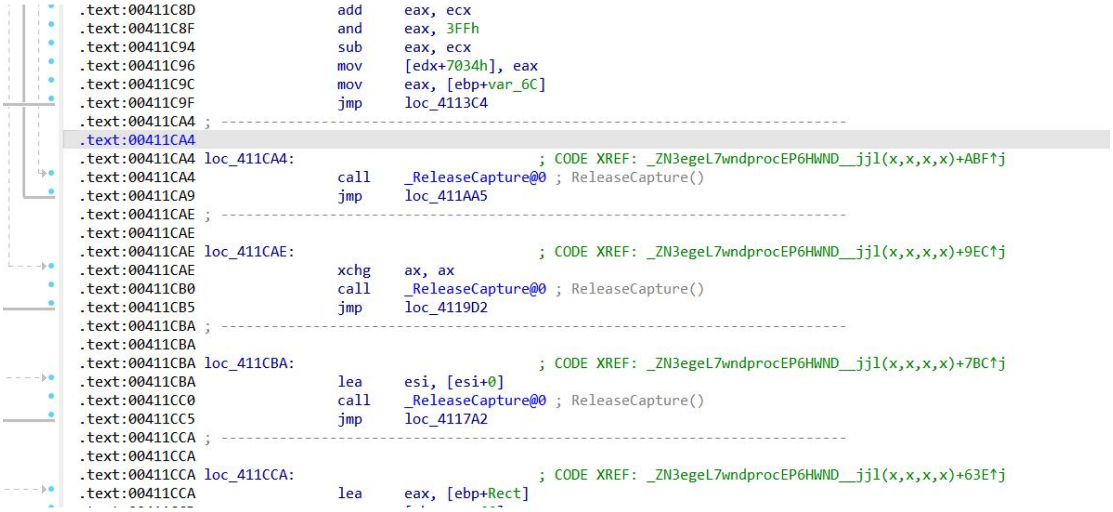
获得坐标和游戏界面大小

（10）



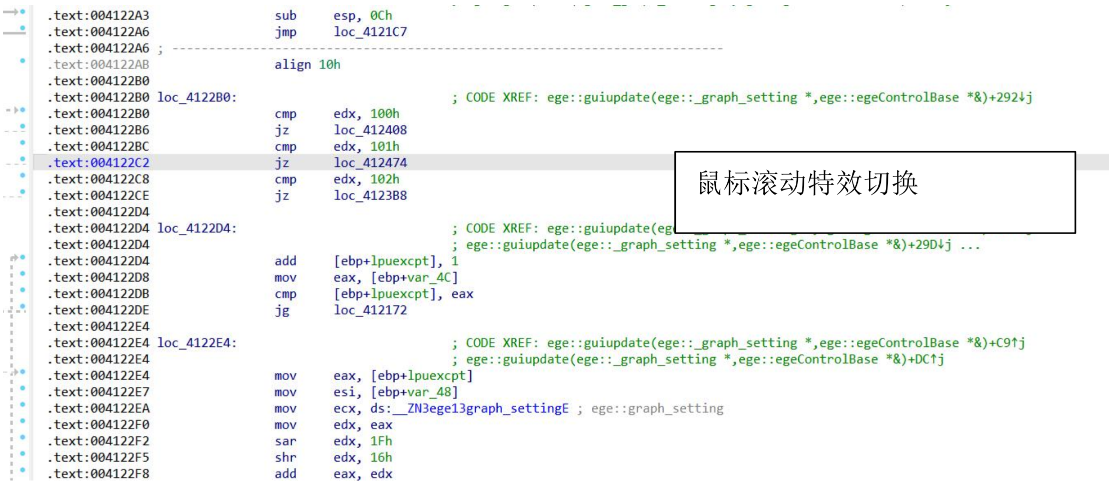
```
.text:00411BDB                 mov     [ebp+lpCriticalSection], eax
.text:00411BDE                 mov     eax, [edi+128h]
.text:00411BE4                 add     eax, eax
.text:00411BE6                 or      [ebp+lpCriticalSection], eax
.text:00411BE9                 mov     eax, [edi+120h]
.text:00411BEF                 or      [ebp+lpCriticalSection], eax
.text:00411BF2                 call    _GetTickCount@0 ; GetTickCount()
.text:00411BF7                 mov     edx, [edi+118h]
.text:00411BFD                 mov     [esp], edx      ; lpCriticalSection
.text:00411C00                 mov     [ebp+var_6C], edx
.text:00411C03                 mov     [ebp+var_74], eax
.text:00411C06                 call    _EnterCriticalSection@4 ; EnterCriticalSection(x)
.text:00411C0B                 mov     edx, [ebp+var_6C]
.text:00411C0E                 mov     ecx, [edx+7038h]
.text:00411C14                 mov     [ebp+var_78], edx
.text:00411C17                 sub     esp, 4
.text:00411C1A                 add     ecx, 1
.text:00411C1D                 mov     eax, ecx
.text:00411C1F                 sar     eax, 1Fh
.text:00411C22                 shr     eax, 16h
.text:00411C25                 add     ecx, eax
.text:00411C27                 mov     [ebp+var_6C], ecx
.text:00411C2A                 and     [ebp+var_6C], 3FFh
.text:00411C31                 sub     [ebp+var_6C], eax
.text:00411C34                 imul    eax, [edx+7038h], 1Ch
.text:00411C3B                 lea     ecx, [edx+eax+10h]
.text:00411C3F                 lea     eax, [ecx+8]
.text:00411C42                 mov     edx, ecx
.text:00411C44                 mov     ecx, [ebp+var_7C]
```

进入游戏关键区域



```
.text:00411C8D                 add     eax, ecx
.text:00411C8F                 and     eax, 3FFh
.text:00411C94                 sub     eax, ecx
.text:00411C96                 mov     [edx+7034h], eax
.text:00411C9C                 mov     eax, [ebp+var_6C]
.text:00411C9F                 jmp     loc_4113C4
.text:00411CA4 ; ---------------------------------------------------------------------------
.text:00411CA4
.text:00411CA4 loc_411CA4:                             ; CODE XREF: _ZN3egeL7wndprocEP6HWND__jjl(x,x,x,x)+ABF↑j
.text:00411CA4                 call    _ReleaseCapture@0 ; ReleaseCapture()
.text:00411CA9                 jmp     loc_411AA5
.text:00411CAE ; ---------------------------------------------------------------------------
.text:00411CAE
.text:00411CAE loc_411CAE:                             ; CODE XREF: _ZN3egeL7wndprocEP6HWND__jjl(x,x,x,x)+9EC↑j
.text:00411CAE                 xchg    ax, ax
.text:00411CB0                 call    _ReleaseCapture@0 ; ReleaseCapture()
.text:00411CB5                 jmp     loc_4119D2
.text:00411CBA ; ---------------------------------------------------------------------------
.text:00411CBA
.text:00411CBA loc_411CBA:                             ; CODE XREF: _ZN3egeL7wndprocEP6HWND__jjl(x,x,x,x)+7BC↑j
.text:00411CBA                 lea     esi, [esi+0]
.text:00411CC0                 call    _ReleaseCapture@0 ; ReleaseCapture()
.text:00411CC5                 jmp     loc_4117A2
.text:00411CCA ; ---------------------------------------------------------------------------
.text:00411CCA
.text:00411CCA loc_411CCA:                             ; CODE XREF: _ZN3egeL7wndprocEP6HWND__jjl(x,x,x,x)+63E↑j
.text:00411CCA                 lea     eax, [ebp+Rect]
```

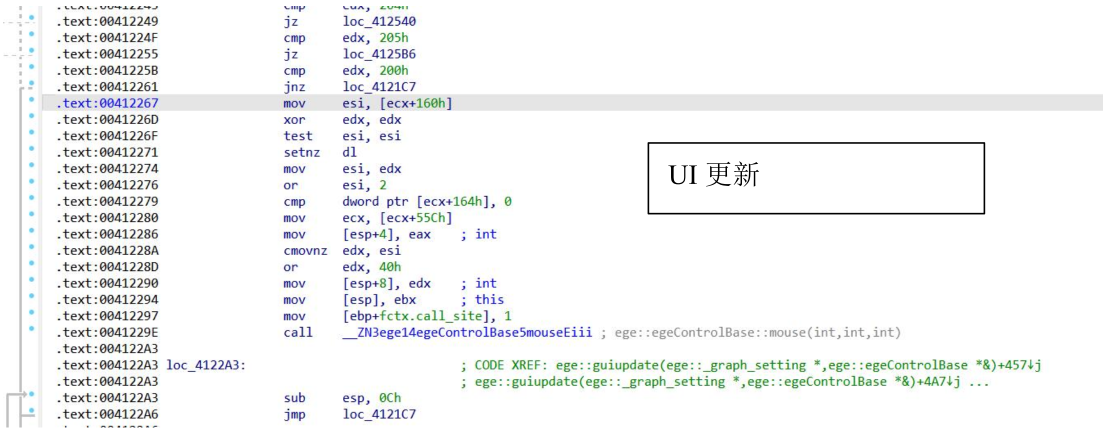
（11）



```
.text:004122A3                 sub     esp, 0Ch
.text:004122A6                 jmp     loc_4121C7
.text:004122A6 ; ---------------------------------------------------------------------------
.text:004122AB                 align 10h
.text:004122B0
.text:004122B0 loc_4122B0:                             ; CODE XREF: ege::guiupdate(ege::_graph_setting *,ege::egeControlBase *&)+292↓j
.text:004122B0                 cmp     edx, 100h
.text:004122B6                 jz      loc_412408
.text:004122BC                 cmp     edx, 101h
.text:004122C2                 jz      loc_412474
.text:004122C8                 cmp     edx, 102h
.text:004122CE                 jz      loc_4123B8
.text:004122D4
.text:004122D4 loc_4122D4:                             ; CODE XREF: ege::guiupdate(eg
.text:004122D4                                         ; ege::guiupdate(ege::_graph_setting *,ege::egeControlBase *&)+29D↓j ...
.text:004122D4                 add     [ebp+lpuexcpt], 1
.text:004122D8                 mov     eax, [ebp+var_4C]
.text:004122DB                 cmp     [ebp+lpuexcpt], eax
.text:004122DE                 jg      loc_412172
.text:004122E4
.text:004122E4 loc_4122E4:                             ; CODE XREF: ege::guiupdate(ege::_graph_setting *,ege::egeControlBase *&)+C9↑j
.text:004122E4                                         ; ege::guiupdate(ege::_graph_setting *,ege::egeControlBase *&)+DC↑j
.text:004122E4                 mov     eax, [ebp+lpuexcpt]
.text:004122E7                 mov     esi, [ebp+var_48]
.text:004122EA                 mov     ecx, ds:__ZN3ege13graph_settingE ; ege::graph_setting
.text:004122F0                 mov     edx, eax
.text:004122F2                 sar     edx, 1Fh
.text:004122F5                 shr     edx, 16h
.text:004122F8                 add     eax, edx
```

鼠标滚动特效切换

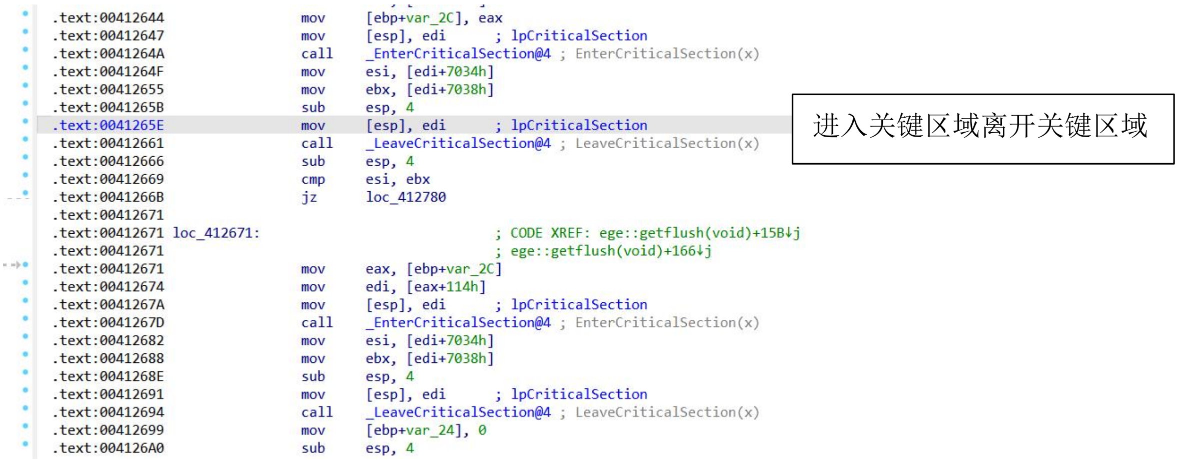
（12）



```
.text:00412249                 jz      loc_412540
.text:0041224F                 cmp     edx, 205h
.text:00412255                 jz      loc_4125B6
.text:0041225B                 cmp     edx, 200h
.text:00412261                 jnz     loc_4121C7
.text:00412267                 mov     esi, [ecx+160h]
.text:0041226D                 xor     edx, edx
.text:0041226F                 test    esi, esi
.text:00412271                 setnz   dl
.text:00412274                 mov     esi, edx
.text:00412276                 or      esi, 2
.text:00412279                 cmp     dword ptr [ecx+164h], 0
.text:00412280                 mov     ecx, [ecx+55Ch]
.text:00412286                 mov     [esp+4], eax    ; int
.text:0041228A                 cmovnz  edx, esi
.text:0041228D                 or      edx, 40h
.text:00412290                 mov     [esp+8], edx    ; int
.text:00412294                 mov     [esp], ebx      ; this
.text:00412297                 mov     [ebp+fctx.call_site], 1
.text:0041229E                 call    __ZN3ege14egeControlBase5mouseEiii ; ege::egeControlBase::mouse(int,int,int)
.text:004122A3
.text:004122A3 loc_4122A3:                             ; CODE XREF: ege::guiupdate(ege::_graph_setting *,ege::egeControlBase *&)+457↓j
.text:004122A3                                         ; ege::guiupdate(ege::_graph_setting *,ege::egeControlBase *&)+4A7↓j ...
.text:004122A3                 sub     esp, 0Ch
.text:004122A6                 jmp     loc_4121C7
```

UI 更新

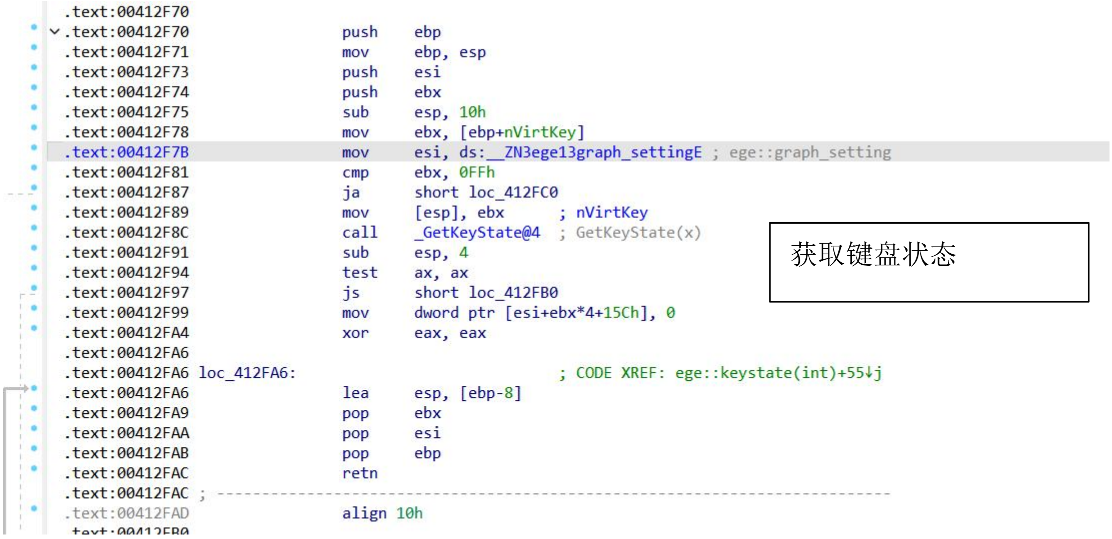
（13）



```
.text:00412644                 mov     [ebp+var_2C], eax
.text:00412647                 mov     [esp], edi      ; lpCriticalSection
.text:0041264A                 call    _EnterCriticalSection@4 ; EnterCriticalSection(x)
.text:0041264F                 mov     esi, [edi+7034h]
.text:00412655                 mov     ebx, [edi+7038h]
.text:0041265B                 sub     esp, 4
.text:0041265E                 mov     [esp], edi      ; lpCriticalSection
.text:00412661                 call    _LeaveCriticalSection@4 ; LeaveCriticalSection(x)
.text:00412666                 sub     esp, 4
.text:00412669                 cmp     esi, ebx
.text:0041266B                 jz      loc_412780
.text:00412671
.text:00412671 loc_412671:                             ; CODE XREF: ege::getflush(void)+15B↓j
.text:00412671                                         ; ege::getflush(void)+166↓j
.text:00412671                 mov     eax, [ebp+var_2C]
.text:00412674                 mov     edi, [eax+114h]
.text:0041267A                 mov     [esp], edi      ; lpCriticalSection
.text:0041267D                 call    _EnterCriticalSection@4 ; EnterCriticalSection(x)
.text:00412682                 mov     esi, [edi+7034h]
.text:00412688                 mov     ebx, [edi+7038h]
.text:0041268E                 sub     esp, 4
.text:00412691                 mov     [esp], edi      ; lpCriticalSection
.text:00412694                 call    _LeaveCriticalSection@4 ; LeaveCriticalSection(x)
.text:00412699                 mov     [ebp+var_24], 0
.text:004126A0                 sub     esp, 4
```

进入关键区域离开关键区域

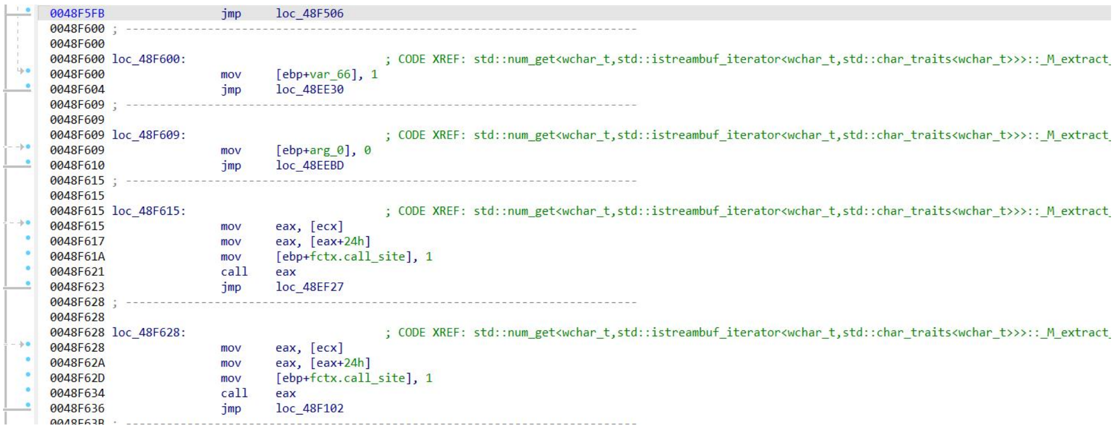
（14）



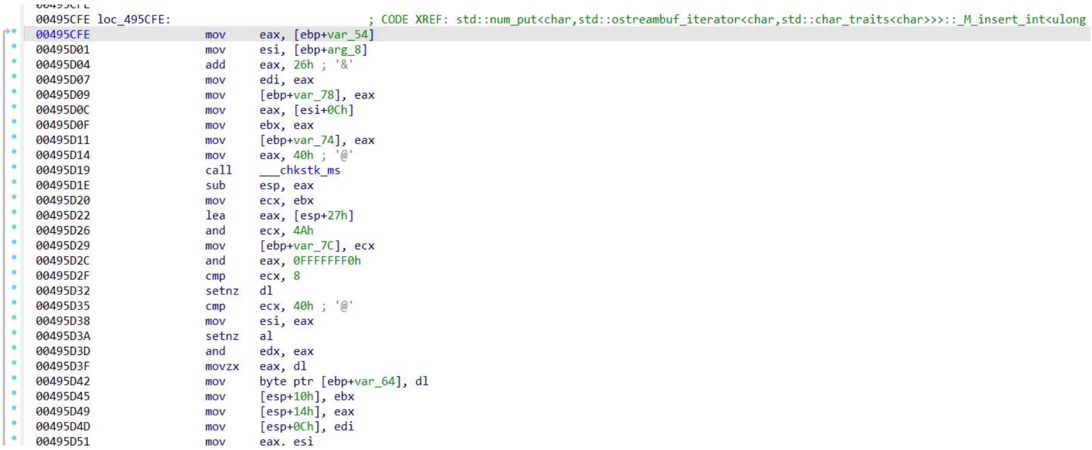
```
.text:00412F70
.text:00412F70                 push    ebp
.text:00412F71                 mov     ebp, esp
.text:00412F73                 push    esi
.text:00412F74                 push    ebx
.text:00412F75                 sub     esp, 10h
.text:00412F78                 mov     ebx, [ebp+nVirtKey]
.text:00412F7B                 mov     esi, ds:__ZN3ege13graph_settingE ; ege::graph_setting
.text:00412F81                 cmp     ebx, 0FFh
.text:00412F87                 ja      short loc_412FC0
.text:00412F89                 mov     [esp], ebx      ; nVirtKey
.text:00412F8C                 call    _GetKeyState@4  ; GetKeyState(x)
.text:00412F91                 sub     esp, 4
.text:00412F94                 test    ax, ax
.text:00412F97                 js      short loc_412FB0
.text:00412F99                 mov     dword ptr [esi+ebx*4+15Ch], 0
.text:00412FA4                 xor     eax, eax
.text:00412FA6
.text:00412FA6 loc_412FA6:                             ; CODE XREF: ege::keystate(int)+55↓j
.text:00412FA6                 lea     esp, [ebp-8]
.text:00412FA9                 pop     ebx
.text:00412FAA                 pop     esi
.text:00412FAB                 pop     ebp
.text:00412FAC                 retn
.text:00412FAC ; ---------------------------------------------------------------------------
.text:00412FAD                 align 10h
```

获取键盘状态

（15）

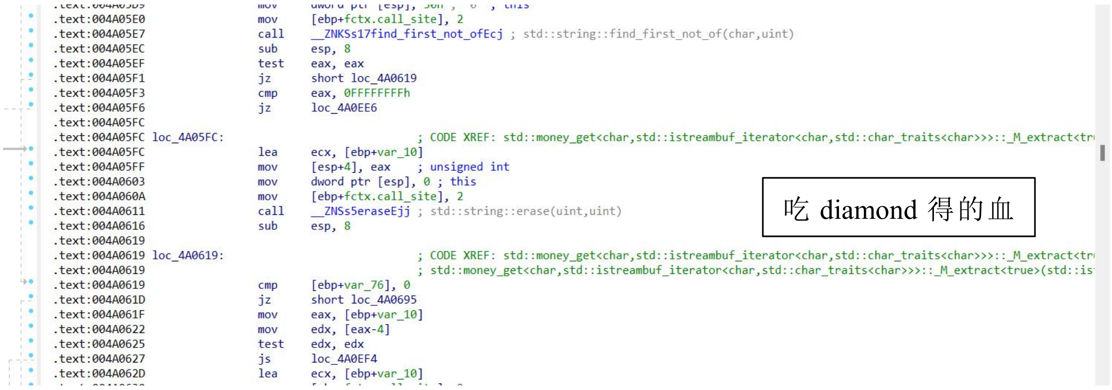


```
0048F5FB                 jmp     loc_48F506
0048F600 ; ---------------------------------------------------------------------------
0048F600
0048F600 loc_48F600:                             ; CODE XREF: std::num_get<wchar_t,std::istreambuf_iterator<wchar_t,std::char_traits<wchar_t>>>::_M_extract_
0048F600                 mov     [ebp+var_66], 1
0048F604                 jmp     loc_48EE30
0048F609 ; ---------------------------------------------------------------------------
0048F609
0048F609 loc_48F609:                             ; CODE XREF: std::num_get<wchar_t,std::istreambuf_iterator<wchar_t,std::char_traits<wchar_t>>>::_M_extract_
0048F609                 mov     [ebp+arg_0], 0
0048F610                 jmp     loc_48EEBD
0048F615 ; ---------------------------------------------------------------------------
0048F615
0048F615 loc_48F615:                             ; CODE XREF: std::num_get<wchar_t,std::istreambuf_iterator<wchar_t,std::char_traits<wchar_t>>>::_M_extract_
0048F615                 mov     eax, [ecx]
0048F617                 mov     eax, [eax+24h]
0048F61A                 mov     [ebp+fctx.call_site], 1
0048F621                 call    eax
0048F623                 jmp     loc_48EF27
0048F628 ; ---------------------------------------------------------------------------
0048F628
0048F628 loc_48F628:                             ; CODE XREF: std::num_get<wchar_t,std::istreambuf_iterator<wchar_t,std::char_traits<wchar_t>>>::_M_extract_
0048F628                 mov     eax, [ecx]
0048F62A                 mov     eax, [eax+24h]
0048F62D                 mov     [ebp+fctx.call_site], 1
0048F634                 call    eax
0048F636                 jmp     loc_48F102
```



```
00495CFE loc_495CFE:                             ; CODE XREF: std::num_put<char,std::ostreambuf_iterator<char,std::char_traits<char>>>::_M_insert_int<ulong
00495CFE                 mov     eax, [ebp+var_54]
00495D01                 mov     esi, [ebp+arg_8]
00495D04                 add     eax, 26h ; '&'
00495D07                 mov     edi, eax
00495D09                 mov     [ebp+var_78], eax
00495D0C                 mov     eax, [esi+0Ch]
00495D0F                 mov     ebx, eax
00495D11                 mov     [ebp+var_74], eax
00495D14                 mov     eax, 40h ; '@'
00495D19                 call    ___chkstk_ms
00495D1E                 sub     esp, eax
00495D20                 mov     ecx, ebx
00495D22                 lea     eax, [esp+27h]
00495D26                 and     ecx, 4Ah
00495D29                 mov     [ebp+var_7C], ecx
00495D2C                 and     eax, 0FFFFFFF0h
00495D2F                 cmp     ecx, 8
00495D32                 setnz   dl
00495D35                 cmp     ecx, 40h ; '@'
00495D38                 mov     esi, eax
00495D3A                 setnz   al
00495D3D                 and     edx, eax
00495D3F                 movzx   eax, dl
00495D42                 mov     byte ptr [ebp+var_64], dl
00495D45                 mov     [esp+10h], ebx
00495D49                 mov     [esp+14h], eax
00495D4D                 mov     [esp+0Ch], edi
00495D51                 mov     eax. esi
```

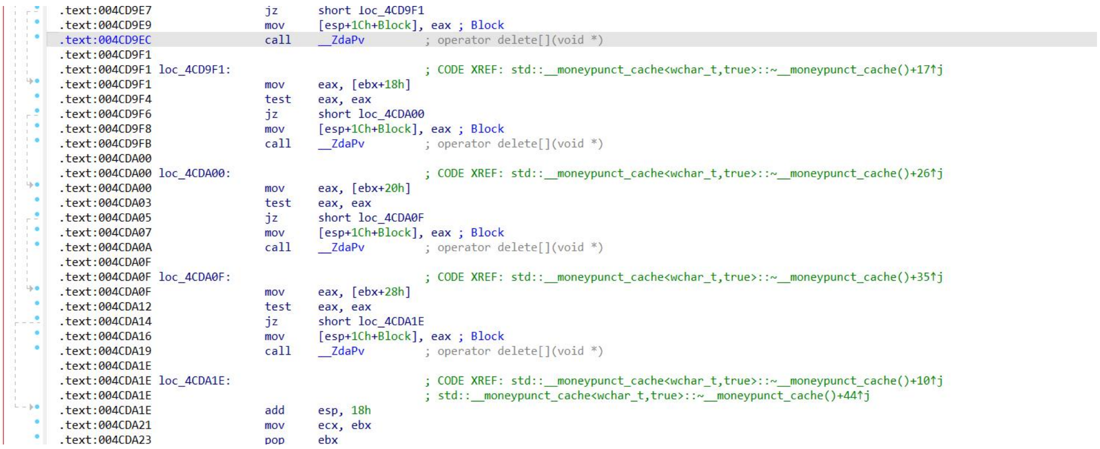
（16）



```
.text:004A05E0                 mov     [ebp+fctx.call_site], 2
.text:004A05E7                 call    __ZNKSs17find_first_not_ofEcj ; std::string::find_first_not_of(char,uint)
.text:004A05EC                 sub     esp, 8
.text:004A05EF                 test    eax, eax
.text:004A05F1                 jz      short loc_4A0619
.text:004A05F3                 cmp     eax, 0FFFFFFFFh
.text:004A05F6                 jz      loc_4A0EE6
.text:004A05FC
.text:004A05FC loc_4A05FC:                             ; CODE XREF: std::money_get<char,std::istreambuf_iterator<char,std::char_traits<char>>>::_M_extract<tr
.text:004A05FC                 lea     ecx, [ebp+var_10]
.text:004A05FF                 mov     [esp+4], eax    ; unsigned int
.text:004A0603                 mov     dword ptr [esp], 0 ; this
.text:004A060A                 mov     [ebp+fctx.call_site], 2
.text:004A0611                 call    __ZNSs5eraseEjj ; std::string::erase(uint,uint)
.text:004A0616                 sub     esp, 8
.text:004A0619
.text:004A0619 loc_4A0619:                             ; CODE XREF: std::money_get<char,std::istreambuf_iterator<char,std::char_traits<char>>>::_M_extract<tr
.text:004A0619                                         ; std::money_get<char,std::istreambuf_iterator<char,std::char_traits<char>>>::_M_extract<true>(std::is
.text:004A0619                 cmp     [ebp+var_76], 0
.text:004A061D                 jz      short loc_4A0695
.text:004A061F                 mov     eax, [ebp+var_10]
.text:004A0622                 mov     edx, [eax-4]
.text:004A0625                 test    edx, edx
.text:004A0627                 js      loc_4A0EF4
.text:004A062D                 lea     ecx, [ebp+var_10]
```

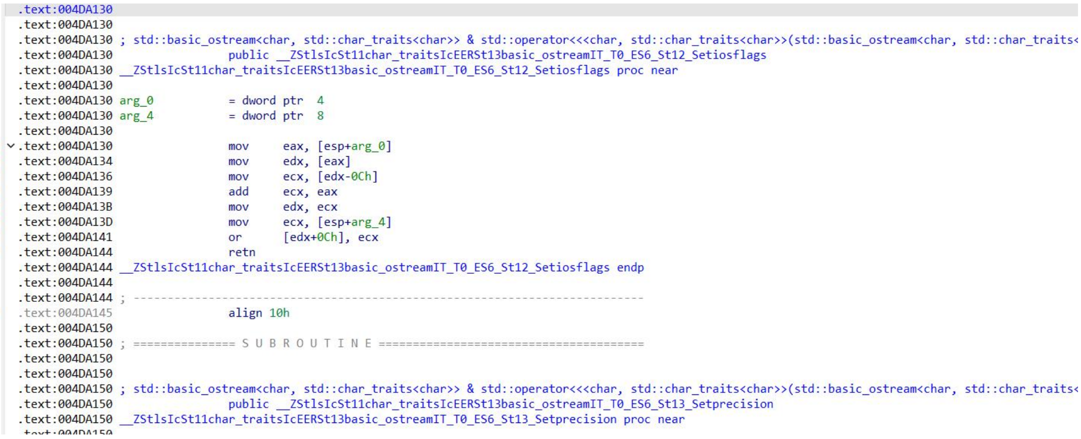
吃 diamond 得的血

（17）



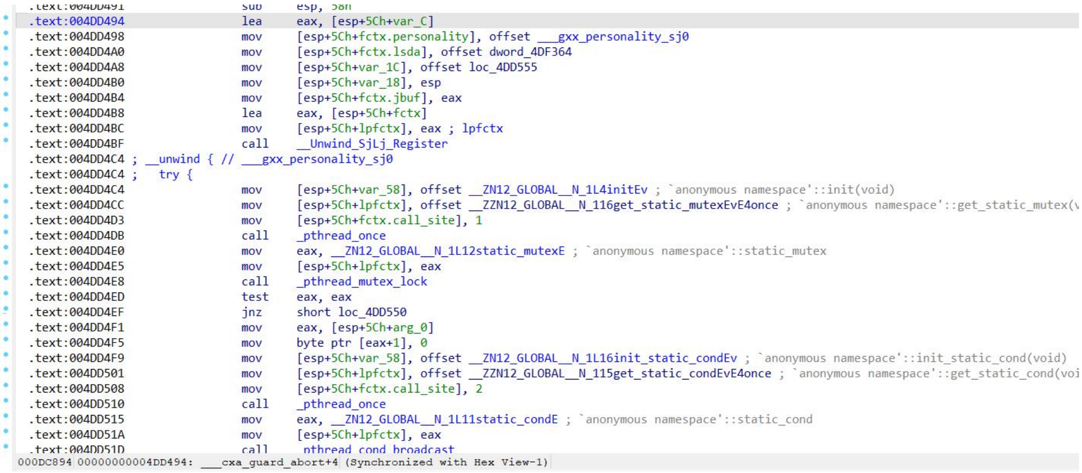
```
.text:004CD9E7                 jz      short loc_4CD9F1
.text:004CD9E9                 mov     [esp+1Ch+Block], eax ; Block
.text:004CD9EC                 call    __ZdaPv         ; operator delete[](void *)
.text:004CD9F1
.text:004CD9F1 loc_4CD9F1:                             ; CODE XREF: std::__moneypunct_cache<wchar_t,true>::~__moneypunct_cache()+17↑j
.text:004CD9F1                 mov     eax, [ebx+18h]
.text:004CD9F4                 test    eax, eax
.text:004CD9F6                 jz      short loc_4CDA00
.text:004CD9F8                 mov     [esp+1Ch+Block], eax ; Block
.text:004CD9FB                 call    __ZdaPv         ; operator delete[](void *)
.text:004CDA00
.text:004CDA00 loc_4CDA00:                             ; CODE XREF: std::__moneypunct_cache<wchar_t,true>::~__moneypunct_cache()+26↑j
.text:004CDA00                 mov     eax, [ebx+20h]
.text:004CDA03                 test    eax, eax
.text:004CDA05                 jz      short loc_4CDA0F
.text:004CDA07                 mov     [esp+1Ch+Block], eax ; Block
.text:004CDA0A                 call    __ZdaPv         ; operator delete[](void *)
.text:004CDA0F
.text:004CDA0F loc_4CDA0F:                             ; CODE XREF: std::__moneypunct_cache<wchar_t,true>::~__moneypunct_cache()+35↑j
.text:004CDA0F                 mov     eax, [ebx+28h]
.text:004CDA12                 test    eax, eax
.text:004CDA14                 jz      short loc_4CDA1E
.text:004CDA16                 mov     [esp+1Ch+Block], eax ; Block
.text:004CDA19                 call    __ZdaPv         ; operator delete[](void *)
.text:004CDA1E
.text:004CDA1E loc_4CDA1E:                             ; CODE XREF: std::__moneypunct_cache<wchar_t,true>::~__moneypunct_cache()+10↑j
.text:004CDA1E                                         ; std::__moneypunct_cache<wchar_t,true>::~__moneypunct_cache()+44↑j
.text:004CDA1E                 add     esp, 18h
.text:004CDA21                 mov     ecx, ebx
.text:004CDA23                 pop     ebx
```

（18）



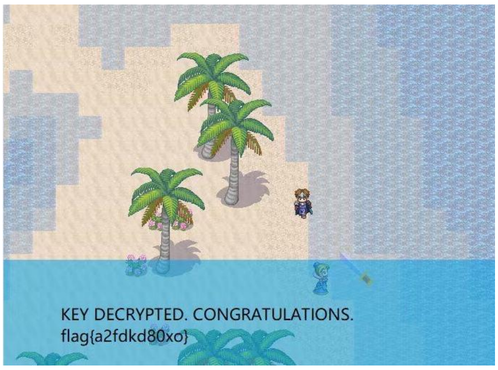
```
.text:004DA130
.text:004DA130
.text:004DA130 ; std::basic_ostream<char, std::char_traits<char>> & std::operator<<<char, std::char_traits<char>>(std::basic_ostream<char, std::char_traits<
.text:004DA130                 public __ZStlsIcSt11char_traitsIcEERSt13basic_ostreamIT_T0_ES6_St12_Setiosflags
.text:004DA130 __ZStlsIcSt11char_traitsIcEERSt13basic_ostreamIT_T0_ES6_St12_Setiosflags proc near
.text:004DA130
.text:004DA130 arg_0           = dword ptr  4
.text:004DA130 arg_4           = dword ptr  8
.text:004DA130
.text:004DA130                 mov     eax, [esp+arg_0]
.text:004DA134                 mov     edx, [eax]
.text:004DA136                 mov     ecx, [edx-0Ch]
.text:004DA139                 add     ecx, eax
.text:004DA13B                 mov     edx, ecx
.text:004DA13D                 mov     ecx, [esp+arg_4]
.text:004DA141                 or      [edx+0Ch], ecx
.text:004DA144                 retn
.text:004DA144 __ZStlsIcSt11char_traitsIcEERSt13basic_ostreamIT_T0_ES6_St12_Setiosflags endp
.text:004DA144
.text:004DA144 ; ---------------------------------------------------------------------------
.text:004DA145                 align 10h
.text:004DA150
.text:004DA150 ; =============== S U B R O U T I N E =======================================
.text:004DA150
.text:004DA150
.text:004DA150 ; std::basic_ostream<char, std::char_traits<char>> & std::operator<<<char, std::char_traits<char>>(std::basic_ostream<char, std::char_traits<
.text:004DA150                 public __ZStlsIcSt11char_traitsIcEERSt13basic_ostreamIT_T0_ES6_St13_Setprecision
.text:004DA150 __ZStlsIcSt11char_traitsIcEERSt13basic_ostreamIT_T0_ES6_St13_Setprecision proc near
.text:004DA150
```

（19）

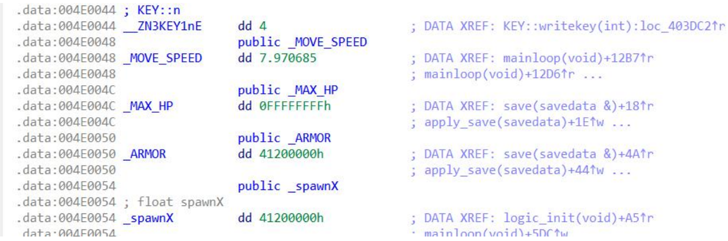


```
.text:004DD491                 sub     esp, 58h
.text:004DD494                 lea     eax, [esp+5Ch+var_C]
.text:004DD498                 mov     [esp+5Ch+fctx.personality], offset ___gxx_personality_sj0
.text:004DD4A0                 mov     [esp+5Ch+fctx.lsda], offset dword_4DF364
.text:004DD4A8                 mov     [esp+5Ch+var_1C], offset loc_4DD555
.text:004DD4B0                 mov     [esp+5Ch+var_18], esp
.text:004DD4B4                 mov     [esp+5Ch+fctx.jbuf], eax
.text:004DD4B8                 lea     eax, [esp+5Ch+fctx]
.text:004DD4BC                 mov     [esp+5Ch+lpfctx], eax ; lpfctx
.text:004DD4BF                 call    __Unwind_SjLj_Register
.text:004DD4C4 ; __unwind { // ___gxx_personality_sj0
.text:004DD4C4 ;   try {
.text:004DD4C4                 mov     [esp+5Ch+var_58], offset __ZN12_GLOBAL__N_1L4initEv ; `anonymous namespace'::init(void)
.text:004DD4CC                 mov     [esp+5Ch+lpfctx], offset __ZZN12_GLOBAL__N_116get_static_mutexEvE4once ; `anonymous namespace'::get_static_mutex(v
.text:004DD4D3                 mov     [esp+5Ch+fctx.call_site], 1
.text:004DD4DB                 call    _pthread_once
.text:004DD4E0                 mov     eax, __ZN12_GLOBAL__N_1L12static_mutexE ; `anonymous namespace'::static_mutex
.text:004DD4E5                 mov     [esp+5Ch+lpfctx], eax
.text:004DD4E8                 call    _pthread_mutex_lock
.text:004DD4ED                 test    eax, eax
.text:004DD4EF                 jnz     short loc_4DD550
.text:004DD4F1                 mov     eax, [esp+5Ch+arg_0]
.text:004DD4F5                 mov     byte ptr [eax+1], 0
.text:004DD4F9                 mov     [esp+5Ch+var_58], offset __ZN12_GLOBAL__N_1L16init_static_condEv ; `anonymous namespace'::init_static_cond(void)
.text:004DD501                 mov     [esp+5Ch+lpfctx], offset __ZZN12_GLOBAL__N_115get_static_condEvE4once ; `anonymous namespace'::get_static_cond(voi
.text:004DD508                 mov     [esp+5Ch+fctx.call_site], 2
.text:004DD510                 call    _pthread_once
.text:004DD515                 mov     eax, __ZN12_GLOBAL__N_1L11static_condE ; `anonymous namespace'::static_cond
.text:004DD51A                 mov     [esp+5Ch+lpfctx], eax
.text:004DD51D                 call    _pthread_cond_broadcast
000DC894 00000000004DD494: ___cxa_guard_abort+4 (Synchronized with Hex View-1)
```

五、 修改 game.exe 二进制代码，获得最后的 Flag。

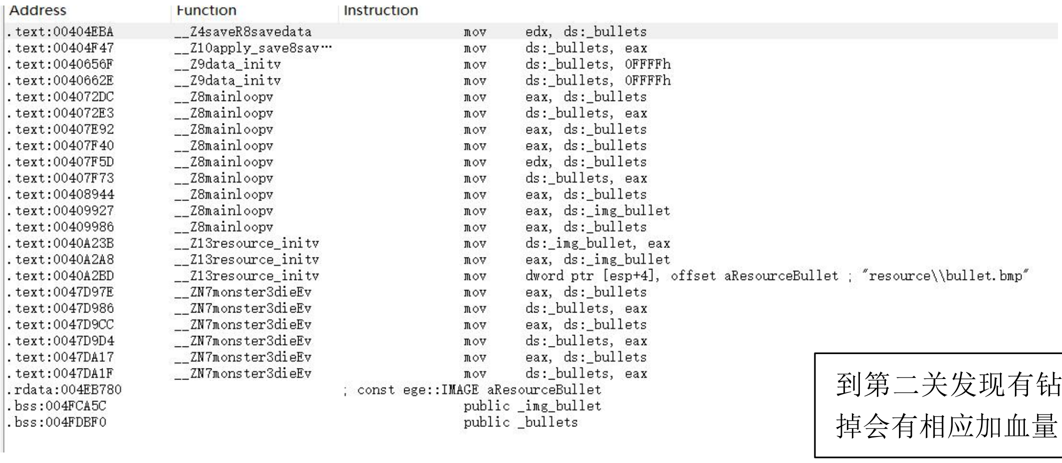


KEY DECRYPTED. CONGRATULATIONS.
flag{a2fdkd80xo}



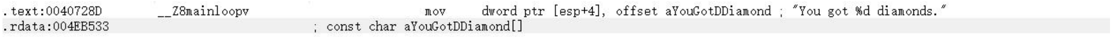
```
.data:004E0044 ; KEY::n
.data:004E0044 __ZN3KEY1nE     dd 4                    ; DATA XREF: KEY::writekey(int):loc_403DC2↑r
.data:004E0048                 public _MOVE_SPEED
.data:004E0048 _MOVE_SPEED     dd 7.970685             ; DATA XREF: mainloop(void)+12B7↑r
.data:004E0048                                         ; mainloop(void)+12D6↑r ...
.data:004E004C                 public _MAX_HP
.data:004E004C _MAX_HP         dd 0FFFFFFFFh           ; DATA XREF: save(savedata &)+18↑r
.data:004E004C                                         ; apply_save(savedata)+1E↑w ...
.data:004E0050                 public _ARMOR
.data:004E0050 _ARMOR          dd 41200000h            ; DATA XREF: save(savedata &)+4A↑r
.data:004E0050                                         ; apply_save(savedata)+44↑w ...
.data:004E0054                 public _spawnX
.data:004E0054 ; float spawnX
.data:004E0054 _spawnX         dd 41200000h            ; DATA XREF: logic_init(void)+A5↑r
.data:004E0054                                         ; mainloop(void)+5DC↑w
```

更改血量，移动速度，最大血量,双方掉血量等

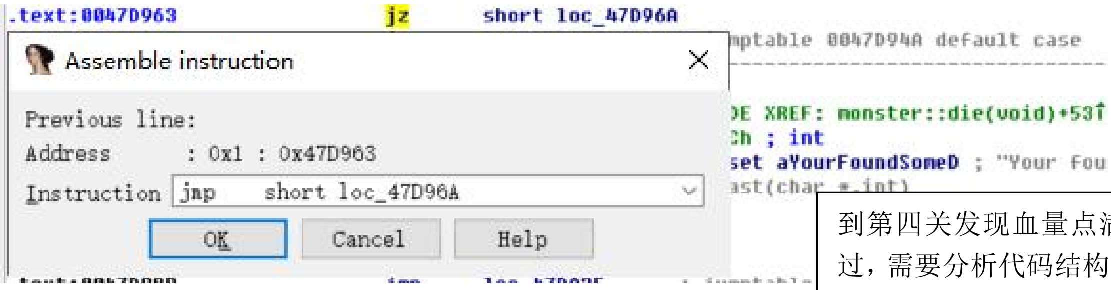


```
Address            Function                 Instruction
.text:00404EBA     __Z4saveR8savedata                      mov     edx, ds:_bullets
.text:00404F47     __Z10apply_save8sav…                    mov     ds:_bullets, eax
.text:0040656F     __Z9data_initv                          mov     ds:_bullets, 0FFFFh
.text:0040662E     __Z9data_initv                          mov     ds:_bullets, 0FFFFh
.text:004072DC     __Z8mainloopv                           mov     eax, ds:_bullets
.text:004072E3     __Z8mainloopv                           mov     ds:_bullets, eax
.text:00407E92     __Z8mainloopv                           mov     eax, ds:_bullets
.text:00407F40     __Z8mainloopv                           mov     eax, ds:_bullets
.text:00407F5D     __Z8mainloopv                           mov     edx, ds:_bullets
.text:00407F73     __Z8mainloopv                           mov     ds:_bullets, eax
.text:00408944     __Z8mainloopv                           mov     eax, ds:_bullets
.text:00409927     __Z8mainloopv                           mov     eax, ds:_img_bullet
.text:00409986     __Z8mainloopv                           mov     eax, ds:_bullets
.text:0040A23B     __Z13resource_initv                     mov     ds:_img_bullet, eax
.text:0040A2A8     __Z13resource_initv                     mov     eax, ds:_img_bullet
.text:0040A2BD     __Z13resource_initv                     mov     dword ptr [esp+4], offset aResourceBullet ; "resource\\bullet.bmp"
.text:0047D97E     __ZN7monster3dieEv                      mov     eax, ds:_bullets
.text:0047D986     __ZN7monster3dieEv                      mov     ds:_bullets, eax
.text:0047D9CC     __ZN7monster3dieEv                      mov     eax, ds:_bullets
.text:0047D9D4     __ZN7monster3dieEv                      mov     ds:_bullets, eax
.text:0047DA17     __ZN7monster3dieEv                      mov     eax, ds:_bullets
.text:0047DA1F     __ZN7monster3dieEv                      mov     ds:_bullets, eax
.rdata:004EB780                          ; const ege::IMAGE aResourceBullet
.bss:004FCA5C                                              public _img_bullet
.bss:004FDBF0                                              public _bullets
```

到第二关发现有钻石，应该吃掉会有相应加血量



```
.text:0040728D        __Z8mainloopv                       mov     dword ptr [esp+4], offset aYouGotDDiamond ; "You got %d diamonds."
.rdata:004EB533                          ; const char aYouGotDDiamond[]
```



```
.text:0047D963                  jz      short loc_47D96A
```

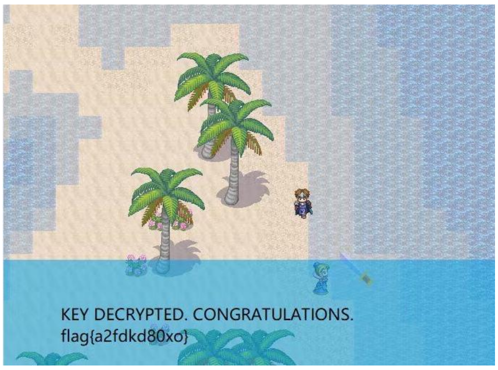
Assemble instruction

Previous line:
Address   : 0x1 : 0x47D963
Instruction jmp   short loc_47D96A

OK
Cancel
Help
mptable 0047D94A default case
DE XREF: monster::die(void)+53↑
Ch ; int
set aYourFoundSomeD ; "Your Fou
ast(char *,int)

到第四关发现血量点满也无法通过，需要分析代码结构。根据游戏第四关代码的分析和实际操作，应将 jz 换成 jnp

Flag：



KEY DECRYPTED. CONGRATULATIONS.
flag{a2fdkd80xo}