

# 扩展战斗系统说明

Beta 0.20

## 目次

1. 新增指令格式.....	1
1.1. 战斗序号.....	1
1.2. 28 指令：取当前人物战斗序号.....	1
1.3. 29 指令：选择攻击目标.....	1
1.4. 30 指令：读取人物战斗属性.....	1
1.5. 31 指令：写入人物战斗属性.....	1
1.6. 44 指令：播放效果.....	2
1.7. 45 指令：显示数字.....	2
1.8. 46 指令：设定效果层.....	2
1.9. 47 指令：重置战场贴图.....	2
1.10. 49 指令：调用任意子程.....	2
2. 扩展事件系统.....	4
2.1. 物品调用事件.....	4
2.2. 武功调用事件.....	5
2.3. 伤害公式事件和状态回复事件.....	6
2.4. 进入游戏预处理与战斗预处理.....	9
Appendix. 1. 常用内存地址.....	10

## 1. 新增指令格式

在游泳的鱼制作的 50 指令集基础上，补充几条战斗中专用的指令。配合原有读取和保存人物属性的指令可实现一些战斗特效，如任意修改敌方属性，偷窃物品等。

这些指令在普通事件中均可以使用，但会造成不正常后果，故不要这样做。

### 1.1. 战斗序号

战斗序号：在战斗之前，所有参战人员会被按轻功编号。编号为从 0 开始的整数，在战斗结束后即失效。

每个回合之后此序号都会重新编排。可以根据战斗序号获取人物序号并加以操作。

50 指令共 8 位，以下指令如不足 8 位需用任意值补全。标志位（type）的含义与 50 原有指令相同。

### 1.2. 28 指令：取当前人物战斗序号

格式：50 28 x

取当前人物的战斗序号，存于变量 x。

实际 28005 变量已经保存了当前人物的战斗序号。

暂时保留。

### 1.3. 29 指令：选择攻击目标

格式：50 29 type 战斗序号 步数 x 是否显示选择画面

Type 的含义与原有 50 指令相同，对“战斗序号”（0 位）和步数（1 位）有效，指定其为数字或变量。返回值保存于变量 x。当选择到某个人物的时候，返回值为该人物的战斗序号。如果返回值为-1，则为空地。

29 号指令会改变变量 28927 的值。如果选择攻击目标（包括空地）时使用空格或回车，该变量为 1；如中途按 Esc 返回，该变量为 0。但无论如何变量 x 都会取得光标所在地点的人物战斗序号（包括-1 的情况），如没有跳转指令，后面的指令在两种情况下均会被执行。所以应在此指令后面立即加上判断 28927 变量和相关跳转指令。

实际上，不仅此指令会使 28927 改变，所有选择目标相关的部分均会改写此变量。

参数“是否显示选择画面”的含义：在选择目标之后，指令会计算出光标的位置的人物战斗序号。但是如果“是否显示选择画面”不为 0，则不会经过选择目标这步，直接根据光标位置计算人物战斗序号，这种情况适用于武功对应事件，因为武功事件的插入位置是在选择目标之后，使用此指令可以直接从光标的位置找到人物序号。增加此参数是为武功事件可以不直接读内存而仅使用指令。

### 1.4. 30 指令：读取人物战斗属性

格式：50 30 type 战斗序号 偏移 x

Type 对“战斗序号”（0 位）和偏移（1 位）有效，指定其为数字或变量。返回值保存于变量 x。

与 28 和 29 指令配合可取得自身和攻击对象的人物序号。

（30 指令与原来的取人物序号并不兼容，需对原有指令略作修改）。

### 1.5. 31 指令：写入人物战斗属性

格式：50 31 type 战斗序号 偏移 x

Type 对“战斗序号”（0 位），偏移（1 位）和 x（2 位）有效，指定其为数字或变量。

人物的战斗属性共有 14 个，以 0~26 的偶数（即偏移）表示，其中 0，2，4，6，10，14，18，26 比较常用，其他属性可能改了也没用。

0：参战人物代号，即 R 文件中的序号，修改此值可以改变战场上人物。

- 2: 敌我判定, 0 为我方, 1 为敌人, 修改此值可以改变角色在战场上的立场。
- 4: 人物坐标  $x$ 。
- 6: 人物坐标  $y$ 。
- 8: 人物固定贴图面对方向 (初始值: 我方 2, 敌人 1)。
- 10: 人物是否死亡。
- 12: 可移动步数。
- 14: 人物是否行动结束, 仅对我方手动人员有效, 决定是否返回此角色的战斗主选单。通常战斗指令中均需要改写此变量。
- 16: 人物贴图, 可能会随 0 属性改变。
- 18: 人物头上将显示的数字。
- 20: 自动战斗动作。
- 22: 自动攻击对手。
- 24: 自动用毒对手。
- 26: 本场战斗获得的经验值。

#### 1.6. 44 指令: 播放效果

格式: 50 44 type 战斗序号 人物动作类型 效果编号

Type 对 3 个参数有效, 指定其为数字或变量。播放效果编号指定的动画及音效, 同时指定人物做对应的动作, 相当于使用武功的动画效果。

如果人物动作类型设定为负值, 则人物不做任何动作, 相当于暗器动画效果。

#### 1.7. 45 指令: 显示数字

格式: 50 45 type 数字颜色 是否闪烁 闪烁颜色

Type 对 3 个参数有效。

数字颜色: 1 为红色, 2 为绿色, 3 为蓝色, 4 为黄色, 5 为紫色, 为 0 不显示。其中黄色数字前面是正号。

另两个参数决定人物贴图的闪烁形式。

是否闪烁: 0 为闪烁, 1 为不闪烁。

闪烁颜色: 1 为白色, 2 为绿色, 3 为紫色。

#### 1.8. 46 指令: 设定效果层

格式: 50 46 type  $x$  起始点  $y$  起始点  $x$  长度  $y$  长度 数值

Type 对 5 个参数有效, 指定其为数字或变量。

设定“ $x$  起始点  $y$  起始点  $x$  长度  $y$  长度”指定的范围内为对应的数值。数值只能为 0 或 1, 0 表示该点无效果, 1 表示该点有效果。本指令应使用在 44 指令之前。

长度必须大于 0。

战场地图范围是 0~63, 指令本身没有溢出保护, 注意不要越界。

#### 1.9. 47 指令: 重置战场贴图

格式: 50 47 type 战斗序号

Type 对战斗序号有效。

将人物的贴图恢复为静止贴图, 否则人物会停留在动作的最后一帧。

#### 1.10. 49 指令: 调用任意子程

格式: 50 49 子程地址低位 子程地址高位 参数起始编号 参数个数  $x$

这个指令用来调用游戏中其它部分的任意子程，但需要知道游戏中原子程的位置及各个参数的含义。参数入栈的顺序与原子程入栈的顺序相同，如有返回值则保存于变量 x。

注意：

无论原子程是否有返回值，x 的值均会有所改变。在确定参数无返回值的时候，请将 x 设定为没有用途的变量。

本指令不仅可以调用战斗时的子程，也可以调用任意其他子程，甚至可以与 25 号指令配合运行某个子程的一部分。但是在调用前请阅读游戏的反汇编文件以确定子程入口及参数含义，在不明白原理时不要使用！

**错误使用本指令会导致游戏出错!! 无特殊要求时不要使用!!**

49 指令实例：转变场景。

以下指令可以无限制地转变场景，通过与改写主地图坐标和场景内坐标配合可以实现场景的任意跳转。

;Kdefnum=1022

1 0 0 0 ; 1(1):[xxx]说：小兄弟，我们现在去桃花岛。

;这句是对话指令，无关紧要。

50 25 0 0 10590 29 75 0 ; 50(32):保存给定地址数据 [1D-295E+0]=75(Int)

;75 为桃花岛的编号，根据实际需求修改。

50 25 0 1 -27969 2 195 0 ; 50(32):保存给定地址数据 [2-92BF+0]=195(Byte)

50 49 -28218 2 0 1 0 0 ; 50(32):新指令，暂时没有解释

50 25 0 1 -27969 2 233 0 ; 50(32):保存给定地址数据 [2-92BF+0]=233(Byte)

0 ; 0(0)::空语句(清屏)

-1 ; -1(FFFF):事件结束

这里并没有改写主地图坐标和内场景坐标，可根据实际需求添加。

在这里所调用的实际上是进入场景子程中处理跳转口的一部分。原内场景子程处理跳转的逻辑是：

判断是否走到跳转口；如是，清屏，重画场景，重载入事件点，流程返回到进入场景的状态。

以上跳转均在内场景子程中，如直接调用整个内场景子程并不合适，仅调用一部分反而更好。

1D295E 是游戏中的当前场景编号，但仅改变这里并不能重画场景和重新载入事件点。

195 (Dec) 在汇编中是从子程中返回指令，用这个指令代替处理跳转部分结束后的下一条指令，从开始处理跳转的位置开始调用子程，这样在遇到 195 (Dec) 之后便会返回，之后立刻将刚才修改的地方还原。

49 号指令应用：“从队伍中选择一个队员”选单

下面的指令片段可以生成一个选择队员的选单，在制作新型装备的时候可以用到。

;Kdefnum=1023

50 0 0 2 0 0 0 0 ; 50(32):变量赋值 [X0]=2

50 49 8336 2 0 1 0 0 ; 50(32):调用任意子程

50 48 0 1 0 0 0 0 ; 50(32):调试指令，显示变量 [X0] -- [X0]

0 ; 0(0)::空语句(清屏)

-1 ; -1(FFFF):事件结束

此事件直接调用游戏中 22090 子程，该子程有一个自变量，表示选单的样式：

0-医疗，1-解毒，2-状态，3-装备，4-修炼，5-使用，6-离队。

返回值为队伍内序号，范围是 0~5，如按了 esc 键，返回值为-1。

如果需要一个通用的选单，建议将原版中“要谁离队”4个字改为“选择队友”，需要选择队友时即调用 6 号样式的选单。尽管真实的离队选单也会有相同变化，但不会产生误解。

## 2. 扩展事件系统

### 2.1. 物品调用事件

在剧情物品上添加场景，大地图事件，与普通事件基本相同。当剧情物品的未知 7 属性大于 0 时，执行该属性指定事件，否则执行原功能。

在暗器物品上添加战斗专用事件，当暗器物品的未知 7 属性大于 0 时，执行该属性指定事件，否则执行原暗器功能，但如果自动战斗时被放掉不负责。

选择这两种物品进行扩充的原因是剧情物品在战斗时会被屏蔽，而暗器在场景和大地图没有效果，可以明确区分战斗特殊物品和剧情特殊物品。

同时，原有的清屏指令和对话指令只针对场景地图，所以将这两个指令也进行重写。

在大地图上引发战斗会导致返回时贴图错误，在战斗中调用战斗会导致本场战斗错误，尽量不要设置这样的事件。

在游戏中，系统功能均有对应子程。故理论上所有系统功能均可以改为调用事件，即系统完全可以由事件改写。但是系统有些特殊功能不明，不建议进行大型修改，目前可以使用所谓“系统类物品”进行功能扩充。

在编写含有战斗的事件的时候，如果确定在本场战斗中会使用此类物品，需要在战斗开始之前留下一定的空间。这个空间大约为 4 个字长，相当于一条对话指令，所以只要不是在事件开始即战斗应不会出现问题。在事件结束判定是否需要返回战斗主选单时，请修改战斗属性偏移 14 的值。为 1 表示占用回合，即会切换至下个角色的战斗主选单，为 0 表示不占用回合。

战斗指令实例：一击必杀的法宝。

制作一个物品，类型为暗器，设置未知 7 属性为 1020。事件如下：

;Kdefnum=1020

50 46 0 0 0 64 64 0 ; 50(32):[战斗中]设定效果层

;实际是清空效果层

50 29 1 28005 5 5 0 0 ; 50(32):[战斗中]选择目标

;在 28005 人物周围生成 5 步的选择范围，选择作用地点，将该地点的人物的战斗序号保存于 x5

50 30 1 5 0 6 0 0 ; 50(32):[战斗中]读取人物战斗属性

;将被选择人物的 R 序号保存至 x6

50 26 0 0 22234 5 1 0 ; 50(32):读取给定地址数据 [X1]=[5-56DA+0](Int)

50 26 0 0 22236 5 2 0 ; 50(32):读取给定地址数据 [X2]=[5-56DC+0](Int)

;这两个地址是光标所在位置

50 46 3 1 2 1 1 1 ; 50(32):[战斗中]写入人物战斗属性

;设定刚才光标所选择的位置为有效效果

50 4 0 0 5 0 0 0 ; 50(32):变量判断 If[X5]<0 then JMP=0 else JMP=1

36 256 60 0 ; 36(24):跳转变量 JMP 是否为 0 是则跳转到:Label0

;如果选到空地则跳转到结尾

50 4 0 2 28927 0 0 0 ; 50(32):变量判断 If[X28927]=0 then JMP=0 else JMP=1

36 256 48 0 ; 36(24):跳转变量 JMP 是否为 0 是则跳转到:Label1

;如果刚才按下的是 esc 键则跳转到结尾

50 17 1 0 6 34 7 0 ; 50(32):读取属性 [X7]=人物[X6]属性偏移 34

```

50 16 1 0 6 34 0 0 ; 50(32):保存属性 人物[X6]属性偏移 34=0
;设置人物 x6 生命为 0
50 31 5 5 18 7 0 0 ; 50(32):[战斗中]写入人物战斗属性
;设置人物头上将显示的数字
50 44 0 0 -1 20 0 0 ; 50(32):[战斗中]播放效果
50 45 0 5 0 2 0 0 ; 50(32):[战斗中]显示数字
50 31 1 28005 14 1 0 0 ; 50(32):[战斗中]写入人物战斗属性
;设置该人物回合结束
;:Label1
;:Label0
-1 ; -1(FFFF):事件结束

```

## 2.2. 武功调用事件

只在点和面类武功上添加了事件，另外两种没有添加。当点面类武功的第 5 个未知属性大于 0 时，会在选择完攻击地点之后执行该数字指定事件，事件执行之后会转至增加武功使用时获得武功经验部分。

以下武功的用于演示，作用是将敌人一击必杀，但是所减的血会刚好与敌人生命相等。

需注意此类武功所定义的事件自动战斗时也会使用。

注意这里有些部分并没有使用新指令，而是直接操作内存相应位置。

武功 R 序号会被预先保存在变量 x28928，而武功在角色中的序号则会被预先保存在变量 x28929。实际上这里利用了原来的 50 43 指令。

```

;Kdefnum=1025
50 26 0 0 22234 5 1 0 ; 50(32):读取给定地址数据 [X1]=[5-56DA+0](Int)
50 26 0 0 22236 5 2 0 ; 50(32):读取给定地址数据 [X2]=[5-56DC+0](Int)
;以上是所选择位置的纵横坐标.
50 3 0 2 1 1 2 0 ; 50(32):四则运算 [X1]=[X1]*2
50 3 0 2 2 2 128 0 ; 50(32):四则运算 [X2]=[X2]*128
50 3 1 0 3 1 2 0 ; 50(32):四则运算 [X3]=[X1]+[X2]
;以上是坐标的处理方法,1E4A04+X*2+Y*128 相当于整个地图平面,如果有人物则该位置是人物战斗序号,没有人物的地点是-1 或-2.
50 26 1 0 18948 30 4 3 ; 50(32):读取给定地址数据 [X4]=[1E-4A04+[X3]](Int)
;读取所选择位置的人物,这里只读了所选择地点的人物
50 30 1 4 0 5 0 0 ; 50(32):[战斗中]读取人物战斗属性
;注意这个并不严格,应判断是否选到了空地.按游戏设定,选到空地也是有效的,但是应跳过一系列设置人物属性的指令.
50 17 1 0 5 34 7 0 ; 50(32):读取属性 [X7]=人物[X5]属性偏移 34
;读取人物的生命值.
50 3 0 2 6 4 28 0 ; 50(32):四则运算 [X6]=[X4]*28
50 25 3 0 -14530 29 7 6 ; 50(32):保存给定地址数据 [1D-C73E+[X6]]=[X7](Int)
;设定人物头上显示的数字,1DC73E+战斗序号*28 保存本回合将显示的数字,这个数字与人物属性的变化无关.

```

```

50 3 0 2 8 28005 28 0      ; 50(32):四则运算 [X8]=[X28005]*28
50 26 1 0 -14522 29 9 8    ; 50(32):读取给定地址数据 [X9]=[1D-C746+[X8]] (Int)
50 3 1 0 9 9 7 0          ; 50(32):四则运算 [X9]=[X9]+[X7]
50 25 3 0 -14522 29 9 8    ; 50(32):保存给定地址数据 [1D-C746+[X8]]= [X9] (Int)

```

;攻击者的经验增加,增加的值等于攻击伤害. 1DC746+战斗序号\*28 保存人物本回合获得的经验,但不清楚人物被更换后是否仍然有效.

```

50 16 1 0 5 34 0 0        ; 50(32):保存属性 人物[X5]属性偏移 34=0

```

;设定人物生命为 0

```

50 25 2 0 -5628 29 1 3    ; 50(32):保存给定地址数据 [1D-EA04+[X3]]=1 (Int)

```

;设定动画效果层, 1DEA04+X\*2+Y\*128 也是整个地图平面, 如果某点值为 0 则没有效果, 如果值为 1 该点会在播放时有效果.

```

50 3 0 0 13 28005 0 0     ; 50(32):四则运算 [X13]=[X28005]+0

```

;这里利用的是人物行动前将战斗序号保存到 X28005, 实际上此变量可取代 28 号指令.

```

50 0 11 9 0 0 0 0        ; 50(32):变量赋值 [X11]=9
50 0 12 1 0 0 0 0        ; 50(32):变量赋值 [X12]=1
50 49 -31330 3 11 3 14 0 ; 50(32):调用任意子程

```

;即播放武功效果子程, 参数是 x11 开始的 3 个, 含义依次为:效果编号 (这里是 9 号), 武功类型 (1), 攻击者战斗序号. 使用无用的 x14, x15 保存可能的返回值.

```

50 45 0 1 1 1 0 0        ; 50(32):[战斗中]显示数字
50 47 1 28005 0 0 0 0     ; 50(32):[战斗中]回复人物静止贴图
-1                          ; -1 (FFFF):事件结束

```

此武功并没有判别敌友, 所以对自己人也会有效。

### 2.3. 伤害公式事件和状态回复事件

每个人都对状态和状态如何影响伤害有不同想法, 如果直接改写 z.dat 并不适合所有人, 因此采取在计算伤害时插入事件的做法。

在计算伤害公式子程中增加一个事件。事件会预先将 4 个数据保存在 28001 开始的 4 个变量中, 即:

28001: 攻击者战斗序号;

28002: 攻击者使用武功序号;

28003: 防御者人物序号;

28004: 按照原伤害公式计算得到的结果。

利用前 3 个变量即可以任意改写伤害公式, 可以对原版伤害公式进行补充, 也可以完全重新写一个算法, 当然也能为特殊人物, 特殊武功甚至特殊装备编写伤害公式。最终只要把结果放在变量 28004 里面就会取代原本的计算结果。如果想对敌人的其他属性或者自己的某些属性 (反噬?) 进行操作, 也可以直接写到这里。

由于已经可以实现武功调用事件, 所以此事件建议用来保存通用伤害公式, 包括各类影响伤害的特殊效果。在特殊效果中, 反噬是最复杂的, 因此需要特别处理。

5C5F7 改为想要的事件编号;

5C5FD: 90 90 90 90 90 改为 E8 B6 1E 00 00。

每回合结束的位置插入一个事件，在这里可以添加一些状态属性的自动回复。

如果想让某状态在战斗后即消失，可以把它写在变量空间里面（建议使用人物序号加一个固定值作为对应变量，变量有近 3 万个不会不够用），在战斗结束清除它们；如果想让某些状态在战斗后仍然保持，就用人物的某个原版未使用的属性位置。只要改写伤害公式，添加这些变量的影响，就可以体现状态效果了。

这个事件的执行时机是在人物“每回合因中毒损血”后立刻执行。敌人的 AI 由于难以控制，所以对状态的设想还需要考虑敌我双方的平衡。

其余如解毒，医疗等子程也可类似添加对应事件。但是敌人 AI 使用的子程和手动操作时的并不完全一致，因此会比较麻烦。

5C63E 改为想要的事件编号；

5C644: 90 90 90 90 90 改为 E8 6F 1E 00 00。

战斗类事件中尽量不要加入清屏指令。

为使这类事件编写方便，每个人物行动之前均会将自身战斗编号取值到 x28005。可以用 30 号指令获取人物编号。

例：反噬的实现。

这里反噬的功能比较简单，只是 50%反弹，当然也可以自行编写其他计算方法。

特殊状态通常需要 3 个部分合作实现：状态对伤害的影响，每回合的回复，引发状态的条件。

以下 3 个部分仅为示例，可以比较完美实现反噬效果，如果需要其他状态，还需在这些事件中添加相关指令。

回合结束位置插入的事件：

;Kdefnum=1026

50 0 10 0 0 0 0 0 ; 50(32):变量赋值 [X10]=0

::Label1

50 30 1 10 0 1 0 0 ; 50(32):[战斗中]读取人物战斗属性

50 3 0 2 4 10 28 0 ; 50(32):四则运算 [X4]=[X10]\*28

50 25 2 0 -14530 29 0 4 ; 50(32):保存给定地址数据 [1D-C73E+[X4]]=0(Int)

;首先清空攻击人物头上将显示的数字。

50 2 1 0 27000 1 2 0 ; 50(32):取数组变量值 [X2]=数组 27000([X1])(Int)

;用 27000+人物序号的变量表示是否存在反噬效果。

;用 27000+战斗序号可以节省空间，但取到被攻击者战斗序号比较困难，而且目前不清楚战斗序号会不会随时重排。

50 4 0 1 2 0 0 0 ; 50(32):变量判断 If[X2]<=0 then JMP=0 else JMP=1

36 256 16 0 ; 36(24):跳转变量 JMP 是否为 0 是则跳转到:Label0

50 3 0 1 2 2 1 0 ; 50(32):四则运算 [X2]=[X2]-1

50 1 3 0 27000 1 2 0 ; 50(32):数组变量赋值 数组 27000([X1])=[X2](Int)

;如果发现表示反噬效果的这个变量大于 0，则将其减 1，即数回合后可自动回复。

::Label0

50 3 0 0 10 10 1 0 ; 50(32):四则运算 [X10]=[X10]+1

50 4 0 0 10 26 0 0 ; 50(32):变量判断 If[X10]<26 then JMP=0 else JMP=1

36 256 -80 0 ; 36(24):跳转变量 JMP 是否为 0 是则跳转到:Label1

;用 x10 控制循环，访问 26 个战斗角色，这里可能要考虑某个战斗编号之下没有角色的情况

-1 ; -1(FFFF):事件结束



伤害公式处插入的事件:

;Kdefnum=1027

```
50 30 1 28001 0 1 0 0      ; 50(32):[战斗中]读取人物战斗属性
50 3 0 0 2 28003 0 0      ; 50(32):四则运算 [X2]=[X28003]+0
50 2 1 0 27000 2 10 0     ; 50(32):取数组变量值 [X10]=数组 27000([X2])(Int)
50 4 0 1 10 0 0 0         ; 50(32):变量判断 If[X10]<=0 then JMP=0 else JMP=1
36 256 8 0                ; 36(24):跳转变量 JMP 是否为 0 是则跳转到:Label0
50 43 0 1028 0 0 0 0      ; 50(32):调用其它事件 Call Sub1028(0,0,0,0)
```

::Label0

;如果被攻击一方的反噬变量大于 0,则调用另一事件,具体处理反噬效果.

```
-1                          ; -1(FFFF):事件结束
```

;Kdefnum=1028

```
50 3 0 3 11 28004 2 0      ; 50(32):四则运算 [X11]=[X28004]/2
50 3 1 1 28004 28004 11 0   ; 50(32):四则运算 [X28004]=[X28004]-[X11]
```

;只需将新的伤害值赋给 28004,即可取代原先的计算.

```
50 17 1 0 1 34 12 0       ; 50(32):读取属性 [X12]=人物[X1]属性偏移 34
50 3 1 1 12 12 11 0       ; 50(32):四则运算 [X12]=[X12]-[X11]
50 4 0 4 12 0 0 0         ; 50(32):变量判断 If[X12]>=0 then JMP=0 else JMP=1
36 256 8 0                ; 36(24):跳转变量 JMP 是否为 0 是则跳转到:Label0
50 0 12 0 0 0 0 0         ; 50(32):变量赋值 [X12]=0
```

;28004 是伤害公式的返回值,令其减半.减少的部分保存于 x11.同时攻击一方的生命减掉 x11.

::Label0

```
50 16 5 0 1 34 12 0       ; 50(32):保存属性 人物[X1]属性偏移 34=[X12]
```

;攻击人物也会掉生命.

```
50 3 0 2 4 28001 28 0      ; 50(32):四则运算 [X4]=[X28001]*28
50 26 1 0 -14544 29 1 4    ; 50(32):读取给定地址数据 [X1]=[1D-C730+[X4]](Int)
50 26 1 0 -14542 29 2 4    ; 50(32):读取给定地址数据 [X2]=[1D-C732+[X4]](Int)
50 3 0 2 1 1 2 0          ; 50(32):四则运算 [X1]=[X1]*2
50 3 0 2 2 2 128 0         ; 50(32):四则运算 [X2]=[X2]*128
50 3 1 0 3 1 2 0          ; 50(32):四则运算 [X3]=[X1]+[X2]
50 25 2 0 -5628 29 1 3     ; 50(32):保存给定地址数据 [1D-EA04+[X3]]=1(Int)
```

;战斗效果层.

```
50 26 1 0 -14530 29 13 4   ; 50(32):读取给定地址数据 [X13]=[1D-C73E+[X4]](Int)
50 3 1 0 13 13 11 0        ; 50(32):四则运算 [X13]=[X13]+[X11]
50 25 3 0 -14530 29 13 4   ; 50(32):保存给定地址数据 [1D-C73E+[X4]]=X13(Int)
```

;设定攻击人物头上显示的数字,用加法是考虑可能一次打中多个有反噬效果的角色.该值将在回合结束时的事件中清零.

;以上都是处理效果的,攻击一方的身上会出现与所使用武功同样的动画效果,并且出现减掉生命的数字.

```
-1                          ; -1(FFFF):事件结束
```

武功“斗转星移”对应事件.

;Kdefnum=1029

```
50 26 0 0 22234 5 1 0     ; 50(32):读取给定地址数据 [X1]=[5-56DA+0](Int)
```

```

50 26 0 0 22236 5 2 0 ; 50(32):读取给定地址数据 [X2]=[5-56DC+0] (Int)
50 3 0 2 1 1 2 0 ; 50(32):四则运算 [X1]=[X1]*2
50 3 0 2 2 2 128 0 ; 50(32):四则运算 [X2]=[X2]*128
50 3 1 0 3 1 2 0 ; 50(32):四则运算 [X3]=[X1]+[X2]
50 26 1 0 18948 30 4 3 ; 50(32):读取给定地址数据 [X4]=[1E-4A04+[X3]] (Int)
50 30 1 4 0 5 0 0 ; 50(32):[战斗中]读取人物战斗属性

```

;获取作用对象,可以事先将此武功的移动范围设为1,使其只能选择到自身.但这类武功在AI中不会被正确使用,也许可以考虑改为对敌人没有效果.

```

50 1 1 0 27000 5 3 0 ; 50(32):数组变量赋值 数组 27000([X5])=3(Int)

```

;设定反噬变量.

```

50 25 2 0 -5628 29 1 3 ; 50(32):保存给定地址数据 [1D-EA04+[X3]]=1(Int)
50 3 0 0 13 4 0 0 ; 50(32):四则运算 [X13]=[X4]+0
50 0 11 44 0 0 0 0 ; 50(32):变量赋值 [X11]=44
50 0 12 4 0 0 0 0 ; 50(32):变量赋值 [X12]=4
50 49 -31330 3 11 3 0 0 ; 50(32):调用任意子程
50 47 1 28005 0 0 0 0 ; 50(32):[战斗中]回复人物静止贴图
-1 ; -1(FFFF):事件结束

```

#### 2.4. 进入游戏预处理与战斗预处理

进入游戏预处理实际是读档预处理。在读取存档之后添加一个事件，可以对内存中的一些数据进行处理。

当然通常预处理是需要一些判断条件的，这些条件可以保存在R文件或D文件里面，可以自己变通。

如果游戏开始时想做什么可以写到开场事件。

5C6E6 改为想要的事件编号；

5C6EA: 90 90 90 90 90 改为 E8 2A FC FC FF

在读取战场之后添加一个事件，可以对战斗中的人物进行预处理。

5C662 改为想要的事件编号；

5C668: 90 90 90 90 90 改为 E8 4B 1E 00 00。

战场预读数据由1E6A10开始，其中1E6A46开始为敌人编号，1E6A04为战斗编号。

例如将战斗预处理事件这样设置，当战斗编号为48时，出场的第一个人物会被更换为0~99中的任意一个人物。

;Kdefnum=1024

```

50 26 0 0 27140 30 2 0 ; 50(32):读取给定地址数据 [X2]=[1E-6A04+0] (Int)
50 4 0 2 2 48 0 0 ; 50(32):变量判断 If[X2]=48 then JMP=0 else JMP=1
36 256 1 0 ; 36(24):跳转变量 JMP 是否为0 是则跳转到:Label0
7 ; 7(7):无条件返回

```

;:Label0

```

50 38 0 100 1 0 0 0 ; 50(32):随机数 [X1]=Random(100)
50 25 1 0 27206 30 1 0 ; 50(32):保存给定地址数据 [1E-6A46+0]=[X1] (Int)
-1 ; -1(FFFF):事件结束

```

以上适合于特殊战斗少的情况，如果特殊战斗太多可以这样设置：在包含战斗的事件中设置某变量值，而战斗预处理事件中只写一个执行该变量指定子程的指令。多数情况时指定到一个空子程。

## Appendix. 1. 常用内存地址

上述的指令已经可以进行大部分普通战斗事件的编写，但是如需要更为丰富的效果，有时必须直接操作内存地址。

以下列出战斗中会用到的内存地址。

556D2 参战人物个数，改变此值同时改写对应人物战斗属性可以添加人物。

556DA 光标 x 位置，重要。

556DC 光标 y 位置，重要。

556F2 是否自动战斗，0 否，1 是。

1D295E 当前所处场景编号，参考 49 号指令中改变场景的实例。

战场地图：

战场地图层共 5 层，其中有一层只在自动战斗有效。战斗指令可以直接改写其中最常用的效果层，如需改写其他层必须直接改写内存。

计算方法为基址+x\*2+y\*128。

1DCA04 战斗地图框架，仅用在自动战斗时 AI 获取目标，通常无效。

1DEA04 战斗地图效果层，为 0 表示无效果，为 1 表示有效果。

1E0A04 战斗地图地面层，记录地面贴图编号，通常不必修改。

1E2A04 战斗地图建筑层，记录建筑贴图编号，为 0 表示没有，注意数字是在 Fish 修改器中查看到的 2 倍。

1E4A04 战场人物层起始位置，如果小于 0 表示没有人，在重画战场时会据此计算人物贴图的位置。

战斗预处理数据：

1E6A04 当前战斗编号，用于战斗预处理。

1E6A06 战斗名。

1E6A10 战斗地图编号。

1E6A14 音乐编号。

1E6A16 手动选择参战人。

1E6A22 自动选择参战人。

1E6A2E 我方 x 坐标。

1E6A3A 我方 y 坐标。

1E6A46 参战敌人。

1E6A6E 敌方 x 坐标。

1E6A96 敌方 y 坐标。

1E6EBE 输后是否有经验。

1E6ED2 战斗状态，0 继续，1 失败，2 胜利，改变此值可以强制结束战斗。

1E6ED6 当前选择武功，0~9。

例 1：召唤新人物出场，如需立刻看到效果需要操作人物层。

```

;Kdefnum=1032
50 29 1 28005 3 0 0 0 ; 50(32):[战斗中]选择目标
50 26 0 0 22234 5 1 0 ; 50(32):读取给定地址数据 [X1]=[5-56DA+0] (Int)
50 26 0 0 22236 5 2 0 ; 50(32):读取给定地址数据 [X2]=[5-56DC+0] (Int)
50 26 0 0 22226 5 0 0 ; 50(32):读取给定地址数据 [X0]=[5-56D2+0] (Int)
50 31 1 0 0 50 0 0 ; 50(32):[战斗中]写入人物战斗属性
50 31 1 0 2 0 0 0 ; 50(32):[战斗中]写入人物战斗属性
50 31 5 0 4 1 0 0 ; 50(32):[战斗中]写入人物战斗属性
50 31 5 0 6 2 0 0 ; 50(32):[战斗中]写入人物战斗属性
50 31 1 0 8 2 0 0 ; 50(32):[战斗中]写入人物战斗属性
50 3 0 2 11 2 128 0 ; 50(32):四则运算 [X11]=[X2]*128
50 3 0 2 12 1 2 0 ; 50(32):四则运算 [X12]=[X1]*2
50 3 1 0 3 11 12 0 ; 50(32):四则运算 [X3]=[X11]+[X12]
50 25 3 0 18948 30 0 3 ; 50(32):保存给定地址数据 [1E-4A04+[X3]]=[X0] (Int)
50 3 0 0 4 0 1 0 ; 50(32):四则运算 [X4]=[X0]+1
50 25 1 0 22226 5 4 0 ; 50(32):保存给定地址数据 [5-56D2+0]=[X4] (Int)
50 31 1 28005 14 1 0 0 ; 50(32):[战斗中]写入人物战斗属性
50 47 1 0 0 0 0 0 ; 50(32):[战斗中]回复人物静止贴图
-1 ; -1(FFFF):事件结束

```

例 2: “神树掌” 对应武功。

武功对应事件不必考虑选择目标和人物是否行动结束。

该武功会在所选择的位置的四周打出 4 棵树，完全困住对方，不过请保证自己有面或线攻击武功。

```

;Kdefnum=1031
50 26 0 0 22234 5 1 0 ; 50(32):读取给定地址数据 [X1]=[5-56DA+0] (Int)
50 26 0 0 22236 5 2 0 ; 50(32):读取给定地址数据 [X2]=[5-56DC+0] (Int)
50 3 0 2 11 2 128 0 ; 50(32):四则运算 [X11]=[X2]*128
50 3 0 2 12 1 2 0 ; 50(32):四则运算 [X12]=[X1]*2
50 3 1 0 3 11 12 0 ; 50(32):四则运算 [X3]=[X11]+[X12]
50 46 0 0 0 63 63 0 ; 50(32):[战斗中]设定效果层
50 46 3 1 2 1 1 1 ; 50(32):[战斗中]设定效果层
50 44 1 28005 4 14 0 0 ; 50(32):[战斗中]播放效果
50 3 0 0 4 3 2 0 ; 50(32):四则运算 [X4]=[X3]+2
50 25 2 0 10756 30 2804 4 ; 50(32):保存给定地址数据 [1E-2A04+[X4]]=2804 (Int)
50 3 0 1 4 3 2 0 ; 50(32):四则运算 [X4]=[X3]-2
50 25 2 0 10756 30 2804 4 ; 50(32):保存给定地址数据 [1E-2A04+[X4]]=2804 (Int)
50 3 0 0 4 3 128 0 ; 50(32):四则运算 [X4]=[X3]+128
50 25 2 0 10756 30 2804 4 ; 50(32):保存给定地址数据 [1E-2A04+[X4]]=2804 (Int)
50 3 0 1 4 3 128 0 ; 50(32):四则运算 [X4]=[X3]-128
50 25 2 0 10756 30 2804 4 ; 50(32):保存给定地址数据 [1E-2A04+[X4]]=2804 (Int)
50 47 1 28005 0 0 0 0 ; 50(32):[战斗中]回复人物静止贴图
50 37 0 100 0 0 0 0 ; 50(32):延时 延时 100
-1 ; -1(FFFF):事件结束

```