链滴

输入系统

作者：xu365082218

原文链接：https://ld246.com/article/1509943534855

来源网站：链滴

流星的输入和招式的原理分析

以下都是个人推测并非事实

输入系统代码已经OK，能正常使用

此文章是按照知乎

格斗游戏的招式指令实际代码是如何判断的? 提问里 Thinkraft的回答来分析的，流星应该也是这个理设计的

输入是 2下 4左 6右 8上 以及 攻击J组成的，在游戏开始的时候角色切换到Idle动作，在这个动作下，以往很多动作切换

在Idle动作下可输入8 8 让角色前冲 同理还有左闪 右闪 后闪 那么当按下8 8时 是如何识别到底是输前冲 还是输入匕首的 88J呢 88J是 匕首上上攻击

从游戏测试,可得到 如果在输入 88之后，等待几帧再输入 J，会出现，角色先前冲，之后在前冲动作束前，切换到 88J动作

这表明了，这二个招式，都满足而且识别了，更说明了，一个输入，对应的输出动作，可能是多个（为88动作和88J动作 2者共享了2个8按键），而如果像把输入缓冲区当作树处理那样，那么头节点确了，子节点确定了，那么就唯一对应了一个输入，就不会同时触发2个动作

这个是有本质区别的，区别在于对输入的缓冲的识别

A处理方式是集中处理，所有招式查询一个输入缓冲区，任意一个较短的输入（88）满足了，可能长（88J）就会被截断而不执行了，如果短的满足了却不删除输入缓冲区，那么没办法确定什么时候该除缓冲区并重置状态。

B方式是每一个招式都有自己的输入状态，让输入状态按照每次的按键输入更新，那么当动作88自己状态满足了，那么同时动作88J的状态处于等待J按键 也就是和88一样拥有2个8按键输入，而且这个候会执行88动作，并且重置88动作的输入状态为初始状态

而若在88J动作等待J按键开始后的N帧之内，若输入了J，那么88J满足，就输出88J动作，而若超过了定的N帧，那么会重置88J动作的状态，也就是88J动作清除掉88的输入，因为在期望得到输入J的时段内没有等到J输入

而原先设计的时候，是想针对当前动作包含的所有下一个可连接招式集合，做一个招式输入状态更新也就是在任意当前动作下，能转到的目标动作会有一个集合，这个集合一共接收多少种输入招式，就这些招式形成一个接受输入的，每一次输入，都遍历里面的每一招

让每一招更新自己的输入缓冲，任意一个长的招式识别了，就不在遍历，而直接输出。

有一个问题是（需要怒气的大招没有怒气时全部会变成普通攻击，这一点是在无怒气状态下，使用大或者小绝，都只会触发攻击，而不会触发其他招式推测的，类似匕首 288J大招 与88J具有重叠，按常，使用不出288J只会重置288J动作的状态，从而执行88与88J，可是游戏里不是这样，那么只有当288没有怒气的时候，重置全部招式的状态，并且执行普通攻击才符合游戏现在的样子）。

另外还有一个特殊例子，就是流星里一个防御取消跳的，原作应该是一个bug,就是按住跳后不要松开快按住防御键，那么防御按下后，就会立即切换到防御，如果松开防御，这时会自动的跳起来。而不松开跳键

这说明跳具有一个延迟响应的设定，为什么这么说，因为其他动作，按下时刻就已经触发，而不会等，而跳跃是比较特殊的动作，他有一个类似压力反馈的机制，就是原作里，跳跃是有轻重的，是通过住跳，在很短的几帧内（10帧左右）如果抬起跳，那么会进入低段跳或者小跳（最少等待这么多帧，且松开了就立即执行低段跳），而按住时间超过了这个10帧，会认为是一个完整的跳，也会立即执行（完整跳最少要等这10帧），这也是说，按住跳的（OnKeyDown）时刻，系统在等，等你在10帧以是否释放跳按键，如果释放了，就会有一个跳跃高度的缩放比例，来减少跳跃高度模拟一个低段跳，

超过这个10帧，则是一个完整的跳。

而原作的跳还有最极端的跳，就是0高度跳跃，这个就是小跳，这个小跳触发的情况，应该在按住跳后，设定一个帧数，在经过这么多帧之前只要释放跳按键，就是小跳，而这个帧要比之前的10小，假是5，那么 按住跳之后1-5帧以内释放则产生小跳

6-10帧之内就按照 比例算完整跳的缩放比，完整跳，高度是75，小跳高度是0 低段跳，按照比例算7的百分比

当释放防御键时，跳招式会恢复更新状态，那么延迟响应的动作跳，就会更新自己的计时器，判断，前按住跳的时间已经超过了10帧，那么就会触发跳而不用抬起跳键，还有一点，是在按住防御的过程，释放了跳OnKeyUp，那么跳招式的输入缓冲区，会被刷新，要么识别了跳但是从当前防御动作是法切到跳跃动作的，要么就是直接的重置了跳动作的输入缓冲区，从而使之后释放防御后，无法触发跳

现在的代码招式识别都是写死的，是按照行号，对应招式，来进行一个判定的，也没有对每个招式维一个自己的缓冲，所以说代码架构非常的垃圾，就是不知道能不能改成上面说的这样，首先的问题是原作里character.act文件里的行号，到底意味着什么。

因为一个行号，可以对应一个group,这个group是一个招式，或者多个招式，也就是说，某一行，既算作输入A招式，也可以算作输入B招式，只有当前动作的可切换动作集合里包含了当前输入的招式，可以切换

而且不同行号，可能输入也会一致，就是因为武器系统的原因，拿刀的下上A 28J和拿剑的 下上A 28 还有拿匕首的 都是同一个输入，但是行号却不一样，所以对应的招式输出也不一样

如果要改成每个招式都维护自己的输入状态，那么是针对每个pose维护自己的输入状态，还是每个行维护自己的输入状态，还是对每一种输入维护自己的输入状态呢

在这个游戏里的输入 忽略防御 跳跃 只看攻击招式有（单个方向键或者仅方向键构成的招式，是由普输入识别方式识别的，连招和普通攻击另外用连招输入识别方式 ，因为攻击会依据武器来算Pose）

J

2J

4J

6J

8J

22删

44删

66删

88删

22J

88J

28J

82J

46J

64J

462J （刀大绝）

246J（双刺大绝）

468J （剑大绝）

228J （枪前砸）

288J (枪匕首大招)

888J (枪破防绝)

似乎就这么多了

但是每一个呢，又可能包含不同的行号

不同的行号，可能对应同样的输出，因为武器不同，这使得如果装备了不同的武器用同样的输入，得的是不同的行号。

关键是现在怎么解决这个问题

想到的解决办法是，用输入项 就是以上列举的这么多项，把每个符合此输入的行号放到一个集合里（一步是人工的），当有此输入时，一个个集合里的行号就依次去测试这个行号包含的全部动作。而每行号拥有多个动作，就会从第一个动作遍历到最后一个，查到任意一个动作，在当前动作的可切换目动作中，那么就调用此动作，同时还要重置此输入项的状态

还有一个问题就是，输入帧限定，限定的是任意时刻都允许更新连招缓冲区，但是出招前必须看是否输入限定帧范围内，这样会使得能够成功输入招式，但是无法使用招式，而不会出现没法办成功输入式。

还有一个问题就是，可以缓存输入指令，让一些动作一旦到达切换帧，就可以立即识别，类似格斗游里的提前输入，等前一个动作刚完，就立即使用招式。

那现在首先要做的是 把招式对应的行号，写到xls表里，程序开始读取这张表，然后根据表的每一行创建一个输入状态，整个表的所有行构成整个输入状态构成输入模块，在动作的限定输入帧之间，更输入模块的每一项的输入状态。一旦有一个满足了输入，那么就会遍历输入状态里所有行号的所有动，去测试当前动作是否含有切换到所有行号包含的任意一个动作，如果含有，那么就测试该动作能否行（怒气值，武器等外部条件的判断），（不存在一个动作可以调用同样招式的2个动作，设计阶段免）

那么现在思路一切清晰，下一步开始写代码

调了一天代码终于OK，动作输入能准确识别了

遇见的问题还是多个动作都成立的问题

类似 288A 同时会使288A动作 88动作 88A动作 3个动作都成立，怎么知道想调用的到底是哪个动作现在的做法是，一旦长动作识别了，就不会继续对短动作进行识别，这个方法可能会有问题

武器行号使用表

飞镖

1， 2， 3， 4， 85， 87， 143， 144

火铳

5，6，7，8，147

飞轮

9， 10， 11， 12，145，146

双刺

[13-24]， 148，149

匕首

[25-34]，84，88，150

剑

[35-47]，151， 152

枪

[48-60]，153，154

刀

[61-72]，155，156

锤子

[73-83]，157，158

乾坤

[89-106]，159，160

指虎

[107-121]

忍刀

[122-142]

中间缺少了86【爆气pose 367】

一共160个行号，每个武器有自己对应的行号，每个行号负责这个武器的一部分招式

这样子，当处于一个攻击Pose的时候，完全可以知道这个Pose在哪一行，这行归属于哪一种武器处理

现在的动作问题

1，远程武器的处理

枪需要进入预备姿态才能开始发出招式

飞轮需要用代码控制攻击过程

2，动画位移是自己控制，还是从动画读取，要怎么配合重力

现在从动画读取位移，发现动作表现和实际游戏不一致，匕首原地跳空中下刺，动画位移 Y轴下68，前 18，游戏中原地跳下刺 Y轴一直到地面约75，向前3左右.可见若读动画位移，那么是错的

可是若自己设定，原游戏中仅有Menu.res存储了招式的攻击力，却没有说明位移多少，是否忽略重等。难道每个动作的各个阶段都是程序里写死的。我想应该不会吧。

3乾坤刀涉及到3个武器姿态的转换，会导致同武器不同模型切换

而最大的问题是重力和动作位移。还有轻功。这部分解决了就可以打怪了，再后面就是怪物AI，关卡本，关卡的机关，怪物设定，关卡与关卡间的连接

主线和支线，道具和装备，buff，技能，佣兵，职业，相机特写，赏金首设定，（合成，强化，重铸可能加一部分，商店，Npc，固定功能点（武器店（武器及其他装备），杂货店（材料-药品），客）

（合成，重铸，强化都需要NPC和建筑），还有存档点，还有这么多，啊西吧