Klotski

3\*3

8块碎片/数字，一块空格

然后移动每一块都只能左右或者上下移动到空格中

数字：

形成

123

456

78

图片：（表盘底部印好了图片）

形成完整图片

代码

cell，每一块{pos , 现在这块的编号，标准答案的编号，（是否移动）}

整个表盘 list of cell

初始位置：5种状态，random

点击pos

List.filter 现在这块编号=-1 cell->pos

Y= x+/-1

X= y+/-1

点击完，现在的编号互换

Click a cell, judge whether valid, if valid -> cell->blank

List.map 在移动，有，click invalid，没有

Update

List.map 在移动？有-》pos valid改状态-》没有移动

有-》pos invalid 改pos

无-》不操作

判断游戏输赢 List.filter now=standard 数量8个

（块的移动，pos+-）

颜色游戏

代码

橙、绿、蓝（三个大list)

小三角相邻判定

方向相反，pos可以定

小三角形{平行于y轴边的上顶点pos，方向左/右}

撞球游戏

类别一

投篮游戏

每一轮篮筐位置随机

每一轮篮筐位置固定，press space的时间越长，抛得越远，抛的话就判定球的位置，在一个合适的范围内成功

总共五轮（篮筐位置变化，不做难度升级）投中三轮得胜

代码

{开始按space的elapsed，停止按space的elapsed，球pos}

Click -> 1.开始按space的elapsed为0 -》 记录此刻的elapsed为开始按space的elapsed

1. 开始按space的elapsed不为0 ，停止按space的elapsed为0-》不操作
2. 开始按space的elapsed不为0 ，停止按space的elapsed不为0-》根据球的elapsed差-》轨迹，+球的pos-》球的下一个pos，改pos

没有click->1.开始按space的elapsed为0 -》不操作

1. 开始按space的elapsed不为0 ，停止按space的elapsed为0-》记录此刻的elapsed为停止按space的elapsed
2. 开始按space的elapsed不为0 ，停止按space的elapsed不为0-》根据球的elapsed差-》轨迹，+球的pos-》球的下一个pos，改pos

判断：球的位置在框的合理位置范围内

在-》胜利

不在-》球的位置在屏幕的合理位置范围内

在-》不操作（不进行胜利与否的变化）

不在-》输

体现球在press过程中的变化 by音效/球的大小变化

--新的feature

--体现球被press

打靶游戏

A simple shooting game. Balls are automatically launched at a fixed time interval. The player controls the shooting angle of the ball to hit the target, with the purpose of preventing the ball from hurting the fragile mirror behind the target. In a round, the mirror will be broken if it is struck for three times.

球{angle,pos,move,begin\_time，输赢}

Model{输次数，赢次数，第几轮，输赢}

Begin\_time:initball的时候的elapsed

Left/right key -> adjust angle

Ball automatically launch, time interval : 3 sec (screen: countdown)

Update

Left/right key->球在move?->不在->adjust angle

在->无效

Elapsed->球在move?->不在->begin\_time+3sec<=elapsed->状态改成在move

->在->angle+pos->新的pos

球的pos和靶子pos判定->合理范围内->胜利，model里的ball重新init，轮数+1，赢次数+1

->不合理范围->输，model里的ball重新init，轮数+1，输次数+1

->不在判断范围内->不操作

model轮的次数<=5 -> 输的次数>=3 -> 输

-> 赢的次数>=3 -> 赢

拦色块

投球入洞

球{angle,pos,move}

Model{第几球，球}

Left/right key -> adjust angle

Space key -> launch ball

Update

Left/right key->球在move?->不在->adjust angle

在->无效

Elapsed->球在move?->不在->不操作

->在->angle+pos->新的pos

球的pos和洞pos判定->合理范围内->胜利，结束游戏

->不合理范围->输，model里的ball重新init，轮数+1

->不在判断范围内->不操作

model轮的次数>3 -> 输

Riddle

例子：

标答

输入的

ShowResult=

Nothing

| Maybe String

Result:string (“correct”/”wrong”)

Update

1. 把输入的东西-》update进model里面一个field叫输入的-》（view再显现出来输入的）
2. 每一个时刻都去判断标答？=输入的，把结果存入model里一个field叫result，不显示
3. 如果是收到Msg（输出结果）-》showresult和result同步

View

1. button（confirm）

点击-》传Msg（输出结果）

1. 显示showresult

View

1. Button
2. Exit,Restart,Continue,Pause,Next,Skip,End
3. Click buildings/stuff/npc
4. Confirm(GS points)
5. MessageBox
6. hint
7. Day x
8. 行动点
9. GSpoints
10. Fetched stuff (photograph fragments/ stroke/ memory)
11. Buildings
12. Roadmap
13. Npc
14. Logo