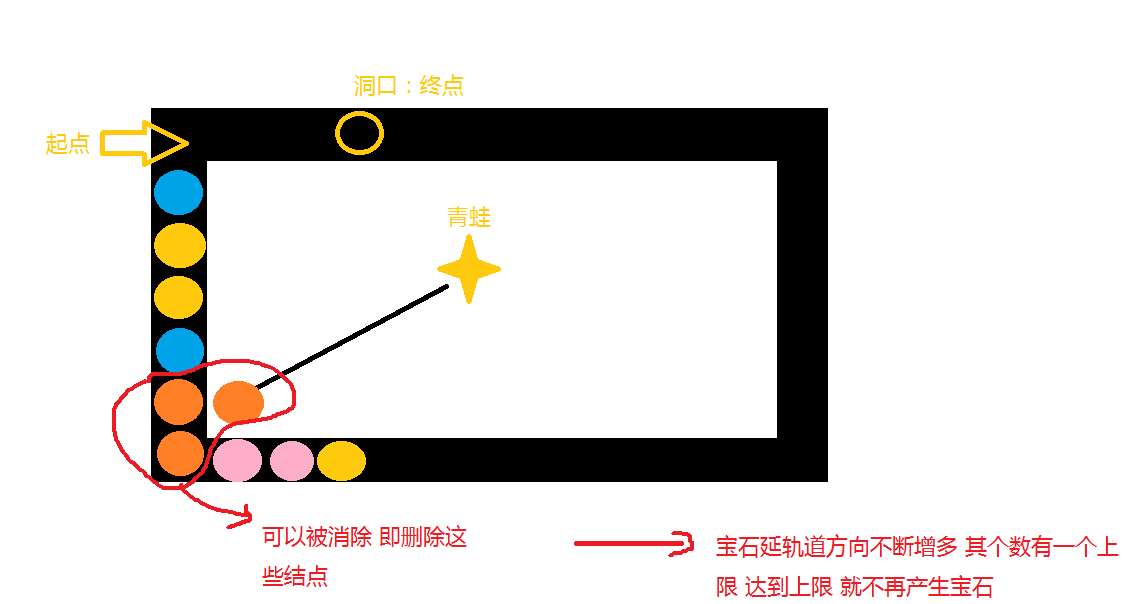
《青蛙祖玛》小游戏设计方案

**一、游戏规则**

有一矩形（其它形状也ok）轨道，轨道有两个端点，一个是出发点，一个是终止点，从轨道一端开始流动宝石（四种颜色），每个宝石都是相邻紧挨的，青蛙处于界面正中，鼠标单击时，青蛙会朝着鼠标所指方向向轨道吐出颜色随机的宝石，当相邻宝石颜色相同且大于等于三个以上时，就会消掉，在规定时间内轨道宝石未掉进洞里游戏胜利，一旦宝石进入终点的洞口则game over。如下图所示：



**二、游戏实现（类似于飞机大战）（此设计暂时不包含ui方面）**

数据结构：链表

定义宝石的结构体：

//宝石类型

typedef enum \_STONE\_TYPE

{

TRACK\_STONE,//轨道上原有的宝石

FROG\_STONE,//青蛙吐出来的宝石

}STONE\_TYPE;

//宝石所处的状态

typedef enum \_STONE\_STATE

{

STONE\_OUT,

STONE\_HIT,

STONE\_MOVED,

}STONE\_STATE;

//宝石的颜色

typedef enum \_STONE\_COLOR

{

YELLOW,

ORANGE,

BLUE,

PINK，

}STONE\_COLOR;

//宝石的属性

typedef struct STONE

{

STONE\_TYPE type;

int x;

int y;

double dir; // 移动时，x相对于y

int y\_step; // y方向每个时间周期移动的距离

STONE\_STATE state;

STONE\_COLOR color;

int ratio\_x; // 在这个范围内击中都有效。

int ratio\_y; // 在这个范围内击中都有效。

}STONE, \*LPSTONE;

;

以及一系列类似于飞机大战的接口函数

LPSTONE create\_stone(

STONE\_TYPE type,

int x,

int y,

double dir, // 移动时，x相对于y

int y\_step, // y方向每个时间周期移动的距离

STONE\_STATE state, //FLY\_STATE state,

STONE\_COLOR color,

unsigned int power,

int ratio\_x, // 在这个范围内击中都有效。

int ratio\_y // 在这个范围内击中都有效。

);

void stones\_destory();

void destory\_stone\_at(unsigned int i);

unsigned int get\_stone\_num();

LPSTONE get\_stone\_at(unsigned int i);

void stone\_move(LPSTONE auto\_fly);

unsigned int get\_stone\_x(LPSTONE auto\_fly);

unsigned int get\_stone\_y(LPSTONE auto\_fly);

void stone\_out(LPSTONE auto\_fly);

void stones\_move\_step();

void destory\_stone\_by\_state();