## 扫雷

## 程序流程

- 1. 进入程序, 初始化界面
- 2. 选择难度或退出
- 3. 初始化扫雷面板
- 4. 读取鼠标行为
  - 1. 若点到重玩,回到3
  - 2. 若点到未开启的格子,则打开格子
  - 3. 若右键未开启的格子,则切换状态:空/旗子
- 5. 确认胜利或失败后结算

## 打开格子

若格子是雷,则打开所有格子,游戏失败结束。

若格子上的数字为空,则打开与之相连的所有格子。

若格子上的数字等于格子周围插旗的格子数,则打开与之相连的所有非旗格子。

若开完之后只剩下n (n =总雷数) 个没打开的格子, 游戏胜利结束。

## 接口说明

类共分: map 抽象地图、view 图形化界面、over 游戏结束

view 只用覆盖, map 需要读取信息, over 处理游戏流程。

void view::view(int h,int w) 是构造函数,初始化的时候创造一个 view 对象。

void view::drawBlock(int x,int y,int t) 在格子 (x,y) 画什么元素, t=-3 代表雷,-2 代表插旗,-1 代表没开过,0 代表开过但是周围没雷, $1\sim 8$ 代表周围雷数。

void map::map(int n,int m,int num) 创建一个  $n \times m$  的地图,随机生成 num 个雷。

void map::flag(int x,int y) 在格子 (x,y) 插旗。

void map::open(int x,int y) 打开格子 (x,y), 如果格子周围雷数=旗子数,则打开周围所有格子。

int map::getBomb(int x,int y) 返回格子 (x,y) 是否有雷,0 没雷,1 有雷。

int map::getStatus(int x,int y) 返回格子 (x,y) 的状态,-3代表雷,-2 代表插旗,-1 代表没开过,0 代表开过但是周围没雷, $1\sim 8$ 代表周围雷数。

void over::win() 记时停止, 弹窗胜利。

void over::lose(int x, int y) 打开所有格子,游戏输在 (x,y) 的位置,在上面打叉。