

吃豆人用户手册

操作方法

- 空格键 - 开始/暂停
- W - 上
- S - 下
- A - 左
- D - 右

游戏规则

1. 游戏中的大黄圆点是玩家
2. 玩家有3条生命
3. 游戏中的小点是玩家要吃的豆子
4. 游戏中的大圆空心是能量豆
5. 玩家吃掉一个豆子得1分（包括能量豆）
6. 游戏中的五角星是鬼，鬼会追捕玩家，鬼追到玩家玩家将丧失一条生命
7. 地图中心是鬼的屋子，玩家不能进入
8. 玩家吃完地图上所有的豆子才能获胜（包括能量豆）
9. 玩家吃能量豆之后有10秒的时间可以吃掉鬼，其余时间都是鬼吃掉玩家
10. 玩家吃掉能量豆后，四只鬼会变成黑色，在能量豆消失前两秒，鬼会交替变色
11. 玩家吃掉鬼后，鬼将变成另一种形态且无法吃掉玩家，直到回到鬼屋才会恢复原形
12. 玩家吃鬼的分数是递增的，第一次50分，以后吃掉的每只鬼加上一次1.5倍的分数
13. 一开始鬼的速度是慢于玩家的，玩家每得100分鬼将升一级，升级后鬼的速度会加快，最高4级，速度最快时和玩家相同
14. 玩家和鬼都不能穿墙，鬼可以随时进鬼屋
15. 鬼和鬼是不会碰撞的，也就是他们面对面行走时会穿过对方

附：游戏设计 - 类设计图

吃豆人 - 类设计图

注：single表示单例模式创建的类

by KevinsBobo <https://kevins.pro>

```
CGameCtrl - single
- m_gameUI: CGameUI
- m_map: CGameMap
- m_moveObj: IGameElement**
- m_ghoObj: IGameElement**
```

```
CGame - （提供全局变量、宏定义和枚举）
// 全局变量
const clock_t g_nInitGhoSpead = 400;
const clock_t g_nInitPlayerSpead = 200;
const int g_nMapRow = 31;
const int g_nMapCol = 28;
```

```

- m_player: CPlayer*

// 游戏开始
+ gameStart(): void
// 游戏逻辑开始
+ gameLoop(): void
// 游戏结束
+ gameStop(): int
// 初始化游戏数据
+ initGameData(): void
// 改变鬼的方向
+ changeGhostAct(): void
// 寻路算法
+ findAction(CPostion, CPostion): int
// 检查状态
+ checkStatus(): void
// 当玩家吃掉超级豆
+ whenPlayerEatSuperPean(): void
// 当玩家碰撞恐惧时的鬼
+ whenPlayerEatGhost(): void
// 当玩家碰撞正常鬼
+ whenGhostEatPlayer(): void
// 当玩家吃完所有豆子
+ whenAllPeanBeEat(): void
// 检查玩家分数、升级鬼
+ checkPlayerScore(): void

```

CGameMap - single
- m_map[MAP_ROW * MAP_COL + 4 + 1]: IGameElement*
<pre> // 初始化地图 + initMap(): void // 重载[]运算符 + operator[](int): IGameElement** // 获取需要移动的对象 + getMoveObj(): IGameElement** // 调用传入UI对象函数指针输出地图中的每个对象 + show(CGameUI*, echoMapElement): void </pre>

CGameUI - single
- m_szElement[4 + 4 + 1]
<pre> // 对外接口 // 打印地图元素 + echoMap(int, int, int): void // 输出游戏信息 + echoGameInfo(int, int, int): void // 在消息区域显示消息 + echoGameMessage(char*): void // 内部接口 + echoMapByPrintf(int, int, int): void + echoMapByApi(int, int, int): void </pre>

CPostion
<pre> - nPosRow: int - nPosCol: int + CPostion(int, int) + CPostion(CPostion&) + operator=(CPostion&): CPostion& + operator+(CPostion&): CPostion + operator-(CPostion&): CPostion + operator*(int): CPostion + operator==(CPostion&): int + operator[](int): int& // 修正坐标 (防止越界) + amend(): void // 设置坐标 + set(int, int): void // 获取这个方向下个坐标 + getActionPostion(int): CPostion // 根据方向修改坐标 + changePostion(int): void </pre>

IGameElement
<pre> // 获取对象坐标 + operator[](int): int& // 与其他对象比较碰撞等级 + operator==(IGameElement&): const int // 获取坐标对象 + getPos(): CPostion& // 获取类型 + getType(): const int // 设置类型 + changeTypet(): void </pre>

```

const int g_nPlayer = 1;
const int g_nGhost = 4;
const int g_nPean = 300;
CGameMap* g_gameMap = NULL;
const int g_nFirstGhoRow = 14;
const int g_nFirstGhoCol = 12;
const int g_nPlayerRow = 23;
const int g_nPlayerCol = 13;
int g_nPlayerLife = 3;
// 玩家已得分数
int g_nAddScore = 0;
// 玩家吃鬼分数
int g_nPlayerEatGhostScore = 50;
// 玩家吃鬼加分倍率
const float g_fMagScore = 1.5;
// 玩家被吃标志
int g_isBeEat = 0;
// 吃掉一只鬼
int g_nGhostBeEat = 0;
// 玩家吃掉豆子的数量 (包括超级豆), 最多300
int g_nEatPeanNum = 1;
// 超级豆被吃标志, 吃一个加1
int g_nSuperPeanBeEat = 0;
// 吃超级豆分数
const int g_nSuperPeanBeEatScore = 1;
// 吃豆子分数
const int g_nPeanBeEatScore = 1;
// 鬼恐惧时间
const clock_t g_nGhostFearTime = 10000;
// 鬼开始恐惧时间
clock_t g_nFearStartTime = 0;
// 鬼等级
int g_nGhostLevel = 1;

// 宏定义
#define MAPROW 31
#define MAPCOL 28
#define NPLAYER 1
#define NGHOST 4
#define NCLASS 14 // 游戏元素数量
#define PLAYEREATGHOSTSCORE 50

// 枚举
// 方向
enum gameAction
{
    actUp = 0,
    actDown,
    actLeft,
    actRight
};
// 位置
enum gamePostion
{
    posRow = 0,
    posCol
};
// 碰撞状态
enum gameCrash
{
    craBeHit = -1,
    craNo = 0,
    craHitAn,
    craPass
};
// 所有元素
enum gameElement
{
    itemPean = 0,
    itemWall,
    itemSurperPean,
    itemRoad,
    itemGhoHome,
    itemGhoNormal,
    itemGhoRed,
    itemGhoYell,
    itemGhoBlue,
    itemGhoPink,
    itemGhoFear,
    itemGhoTremble,
    itemGhoDie,
    itemPlayer
};

```

