第一组详细设计说明书

组员：

一．游戏简介

1、游戏背景

勇士听说国王的女儿被一群怪物抓走了，那群怪物就位于一座魔塔当中，很多人去营救公主，但都没人能够回来。勇士决定去营救公主，于是来到了魔塔，可刚进入魔塔，就被魔塔的首领红衣魔王暗算，勇士的剑和盾都被夺走了，还被关入了监牢。幸好和勇士一同坐牢的还有一个小偷，小偷挖出了一条密道，和勇士一起逃出了监牢，勇士展开了反击，在塔里不停的打怪升级……

2、游戏玩法介绍

通过移动获取钥匙、宝石、武器防具等道具，击败怪物获得金币升级，向更高层进发！

3、游戏特色

魔塔游戏虽不大，但是制作精美，道具很多，而且难度不低，对智商是一次艰巨的考验，所以如果你喜欢挑战，请一定不要错过！

二．模块划分

林展辉：

数据结构设计；绘制系统编写；

移动、门与物体判定；特殊情节编写；

吕远：

从文件中读取地形，创建地图功能；

战斗系统设计；对话与商店功能设计；

三．系统组织结构

四.数据结构设计

class ID

{

public:

ID();

void ChangeID(int GiveID);

void SetMonster(int GiveID);

void SetObject(int GiveID);

int FloorType; //0为地板 1为墙壁或门 2为怪物

int ObjectType; //-2为上楼 -3为下楼 0为空 1为黄钥匙 2为蓝钥匙 3为红钥匙 4为红药水 5为蓝药水 6为红宝石 7为蓝宝石

int DoorType; //0为空 1为黄 2为蓝 3为红 4为青 -5为商店 6为土地爷 7为商人 8为小偷 9为暗墙

int MonsterType; //0为空 1为绿色史莱姆 2为红色史莱姆 3为巨型史莱姆 4为小蝙蝠 5为大蝙蝠 6为骷髅兵 7为骷髅战士 8为骷髅队长 9为巫师 10为高级巫师 29为魔法警卫 30为魔王

int Row, Column, shape;

TCHAR name[30];

int health = NULL;

int offense = NULL;

int defense = NULL;

int money = NULL;

};

地图形式：对象构成的三维数组

1. 函数设计

地板属性.h

void SetValues(int FloorType ,int ObjectType,int DoorOrMonsterType,int Row ,int Column ,int shape,ID \*ptr)

void ID::ChangeID(int GiveID)

void ID::SetMonster(int GiveID)

void ID::SetObject(int GiveID) //通过这几个函数创建与改变区域类型

特殊情节.h

void MonsterMove(int DepRow, int DepCol,int DesRow,int DesCol) //怪物和小偷移动的动画

void SpecialAction()

void SpecialActionDoor(int f) //关键情节与剧情判断与内容

门与道具.h

void OpenDoor(ID\* ptr)

void GetObject(ID\* ptr) //门与道具交互判定

创造地形.h

void Creat()

void SetMatrix() //初始化地图

绘画函数.h（函数较多只罗列关键部分）

void DrawMonsterGuides()

void DrawKnight()

void Draw() //绘制游戏内容

魔塔.cpp（函数较多只罗列关键部分）

void Battle(ID \*ptr) //战斗系统

void move(int operNum) //人物移动

bool talk() //对话系统

void OpenMonsterGuides() //怪物笔记系统

1. 算法设计
2. 算法描述

战斗系统：

伤害值=防御方生命值-攻击方攻击力+防御方防御力。

攻击怪物无危险，以下2种情况：

1、 主角攻击类>怪物防御力且主角防御力≥怪物攻击力；即防杀

2、 主角攻击力≥怪物防御力+怪物生命力；即秒杀

怪物不可被攻击，以下2种情况：

1、 主角攻击力≤怪物防御力，无法对怪物造成伤害；

2、主角攻击力>怪物防御力，但主角生命值不足以打败怪物。

商店系统：

新的售价=旧的价格+（20 \*已购次数）

1. 算法实现

void Battle(ID \*ptr) {

if (ptr->FloorType == 2) {

int K\_health = Knight\_Value[0];

int K\_offense = Knight\_Value[1];

int K\_defense = Knight\_Value[2];

int M\_health = ptr->health;

int M\_offense = ptr->offense;

int M\_defense = ptr->defense;

while (1)

{

M\_health = M\_health - Knight\_Value[1] + M\_defense;

if (M\_health <= 0) {

M\_health = 0;

Knight\_Possessions[0] += ptr->money;

\_sntprintf\_s(info, sizeof(info), \_T("%s被击败了，你获得了%d金币"), ptr->name,ptr->money);//ptr->name

ptr->ChangeID(0);

break;

}

Knight\_Value[0] = Knight\_Value[0] - M\_offense + Knight\_Value[2];

if (Knight\_Value[0] <= 0) { break; }

Draw();

if (((clo2 - clo1) % 800) < (200)) DrawKnight();

else if (((clo2 - clo1) % 800) > (600)) DrawOneSquare(15, 10, 1, Knight\_x, Knight\_y);

else if (((clo2 - clo1) % 800 > 200)& ((clo2 - clo1) % 800 <400))

DrawOneSquare(15, 10, 1, Knight\_x, Knight\_y);

else

{ DrawOneSquare(ptr->Row, ptr->Column, ptr->shape, Knight\_x, Knight\_y); }

FlushBatchDraw();

Sleep(100);

}

}

}

#####################################################

void shop() {

char c2;

shopboxflag = 1;

while (1) {

int UseMoney;

UseMoney=10 \* buy\_n \* (buy\_n - 1) + 20;

\_sntprintf\_s(ManyText, sizeof(ManyText), L"商店\n\n\n消耗%d个金币\n\n按下数字1获得200生命\n\n按下数字2提升2点攻击\n\n按下数字3提升2点防御\n\n按下数字4退出", UseMoney);

if (\_kbhit()) {

c2 = \_getch();

if (c2 == '1') {

if(Knight\_Possessions[0]>=UseMoney){

Knight\_Possessions[0] -= UseMoney;

buy\_n += 1;

Knight\_Value[0] += 200;

}

else

{

\_sntprintf\_s(info, sizeof(info), L"你的金币不够");

}

shopboxflag = 0;

break;

}

if (c2 == '2') {

if (Knight\_Possessions[0] >= UseMoney) {

Knight\_Possessions[0] -= UseMoney;

buy\_n += 1;

Knight\_Value[1] += 2;

}

else

{