

一 . 项目目的和要求（说明通过本项目希望达到的目的和要求）

二 . 项目开发环境（说明本项目需要的环境）

- Ubuntu 12.04
- Linux QT
- Linux GCC
- Phonon

三 . 项目内容（说明本项目的内容，如：问题分析、设计方案、算法、设计图等）

1. 算法

1. AI (即小怪和 BOSS) 的移动算法

a) 由于我们并没有使用游戏引擎作为开发的辅助工具，因而处于对于游戏流畅度的考虑，我们最终并没有使用需要 $O(n^2)$ 这么高的时间复杂度的 Dijkstra 算法，而是使用了较为简单的对时间复杂度要求并不是太高 ($O(1)$ 复杂度) 的贪心算法。

b) 具体实现方法为：AI 每过 Frame (帧间隔) 大小的时间单位对玩家的位置进行判定。

首先，AI 判定距离距离哪一个玩家距离自己的距离更为接近。假设 player1 距离 AI 更为接近，那么 AI 再对 player1 与自己的距离的横纵坐标进行判断：如果 AI 和 player1 之间的横坐标之差比 AI 和 player1 之间的纵坐标之差更大，那么 AI 优先横向靠近玩家；反之则优先纵向靠近玩家。

其次，假设 AI 和 player1 之间的横坐标之差比 AI 和 player1 之间的横坐标之差更大，那么 AI 判断玩家在自己的左边还是在右边，假设是在左边，那么 AI 左边没有石头，炸弹，则 AI 向左移动；否则 AI 纵向移动，如果 player1 在 AI 上面，那么 AI 向上移动；否则向下移动。

最后，如果 AI 不满足以上任意一种情况，那么 AI 会向自己的右边移动。

c) 详细代码见附录一。

2. 炸弹的实现算法

a) 首先我们通过一个链表串来引爆炸弹，每个炸弹拥有自己的引爆时间，用 FIFO 的方式来引爆所有的炸弹。

b) 对于每一个正在爆炸的炸弹，我们首先计算它的爆炸范围，判定范围内部是否还有别的炸弹，如果判定范围内是否还有别的炸弹，如果还有炸弹在正在爆炸的炸弹的爆炸范围内，那么该颗炸弹会直接引爆。

b) 详细代码见附录二

2. 设计图

UML 图：

Bombman
<ul style="list-style-type: none"> - lifeLabel1 : QLabel - lifeLabel2 : QLabel - lifeNum1 : QLCDNumber - lifeNum2 : QLCDNumber - manView : QLabel - timer : QTimer - mdLay : Phonon::MediaObject - mdBomb:Phonon::MediaObject - adLay:Phonon::AudioOutput - adBomb:Phonon::AudioOutput - player : Player[] - head : Bomb - item : Item - plant : Plant[] - normalEnemy : Enemy[] - boss : Enemy[] - sthExist : int[] - enemySize : int - bossSize : int + mode : int + player1 : int + player2 : int + BombMan(QWidget *parent) + initGame() : void + initWidget() : void + initconnect() : void + layBomb(master : int) : void + gameOver(res : int) : void + playerMove() : void + bombAbout() : void + itemAbout() : void + enemyAbout() : void + judgement() : void + enemyMove(enemy : Enemy) : void + gameStart() : void + setmode(m : int) : void + setplayer1(p : int) : void + setplayer2(p : int) : void - process() : void # paintEvent(a : QPaintEvent) : void # keyPressEvent(event : QKeyEvent) : void

preFace
<ul style="list-style-type: none"> - startLabel : QLabel - helpLabel : QLabel - inforLabel : QLabel - ModelLabel : QLabel - RoleLabel1 : QLabel - RoleLabel2 : QLabel - EndLabel : QLabel - startButton : QPushButton - helpButton : QPushButton - inforButton : QPushButton - quitButton : QPushButton - ModeButton1 : QPushButton - ModeButton2 : QPushButton - ModeButton3 : QPushButton - RoleButton1 : QPushButton - RoleButton2 : QPushButton - RoleButton3 : QPushButton - RoleButton4 : QPushButton - yes : QPushButton - music1:Phonon::MediaObject - music2:Phonon::MediaObject - returnButton : QPushButton - yesButton : QPushButton - noButton : QPushButton + bombMan : BombMan + mode : int - player1 : int - player2 : int - music1:Phonon:_0 : MediaObject

<ul style="list-style-type: none"> + go() : void + start() : void + role2() : void + Role2() : void + gameStart() : void + gameEnd() : void - closeStart() : void - closeHelp() : void - closeInfo() : void - closeMode() : void - closeRole1() : void - closeRole2() : void - help() : void - infor() : void - Mode() : void - role1_1choosed() : void - role1_2choosed() : void - role1_3choosed() : void - role1_4choosed() : void - role2_1choosed() : void - role2_2choosed() : void - role2_3choosed() : void - role2_4choosed() : void - role1choosed() : void - role2choosed() : void - role3choosed() : void - role4choosed() : void - mode1() : void - mode2() : void - mode3() : void - restart() : void - closeGameEnd() : void

Player	Enemy
<pre>+ x : int + y : int + restLife : int + direction : int + defDir : int + moveSpeed : int + attackRange : int + bombCountLimit : int + bombCount : int + Player(x : int, y : int)</pre>	<pre>+ x : int + y : int + isBoss : int + moveTime : int + isAlive : int + Enemy(x : int, y : int, isBoss : int, moveTime : int)</pre>

Plant	Item	Bomb
<pre>+ x : int + y : int + Plant(x : int, y : int)</pre>	<pre>+ id : int + lastTime : int + master : int + x : int + y : int + Item(id : int, x : int, y : int)</pre>	<pre>+ x : int + y : int + master : int + lastTime : int + next : Bomb + Bomb(x : int, y : int, master : int)</pre>

四 . 项目结果及分析

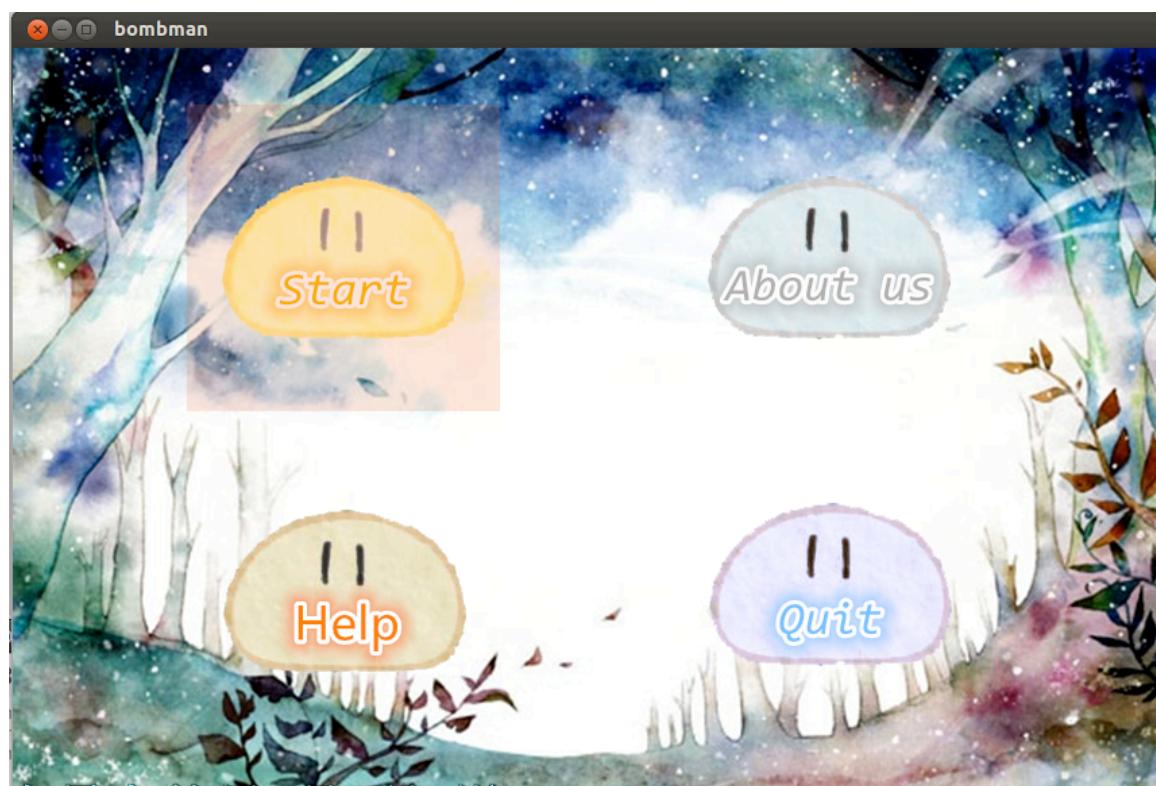
1) 进入游戏等待界面，显示游戏名称——团子炸弹人



2) 进入游戏等待界面 2, 显示游戏 LOGO——“Nine Dangos”
以及主创人员名称——Production Staff



3) 进入游戏主界面, 共四个菜单选项



4) 点击进入 About us, 显示游戏版本号已经游戏名称



5) 点击进入 Help, 显示萌化的游戏帮助信息



6) 点击进入模式选择, 共四个模式: Leisure (休闲模式),

Entertainment (娱乐模式), Hell (地狱模式), Ghost (幽灵模式)。



7) 点击进入任意一个模式后都会进入角色选择界面：

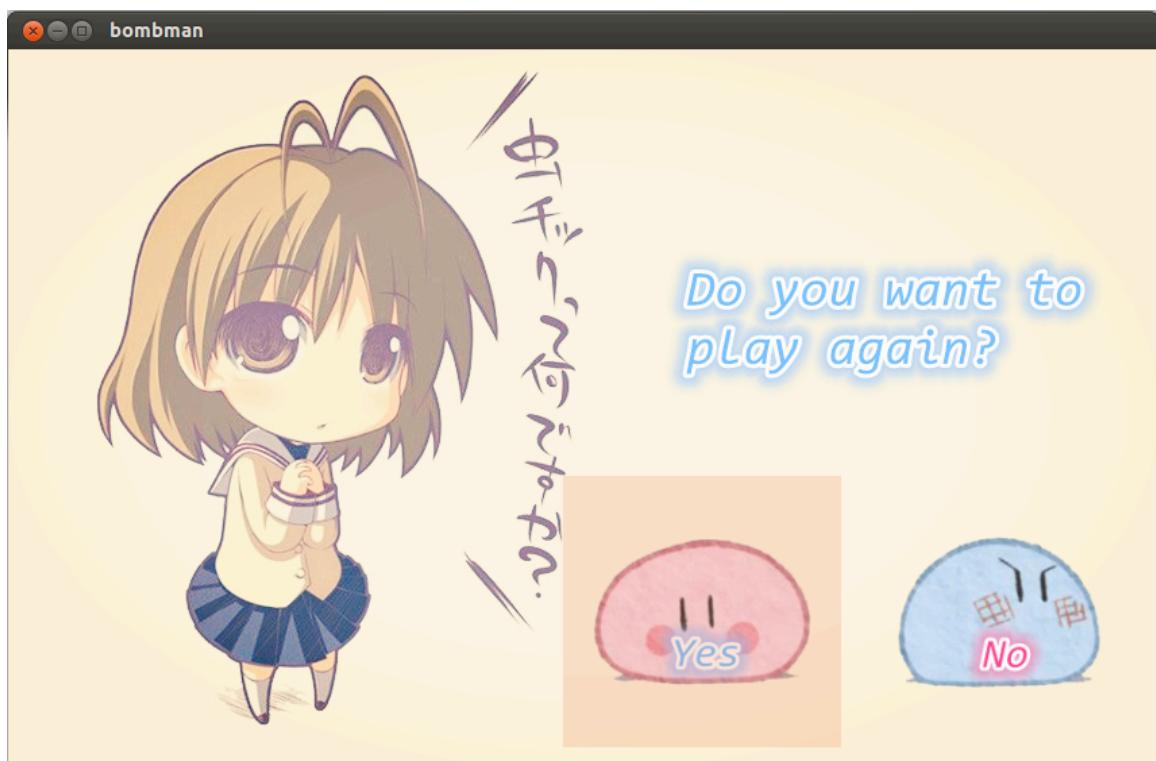


(选择角色 1)



(选择角色 2)

- 8) 任意一次游戏失败后会进入询问界面，询问玩家是否想要继续玩耍，如果选择 Yes 就重新开始游戏；选择 No 就返回主菜单界面。



9) 进入休闲模式



10) 检验炸弹效果, 炸弹是可以炸掉植物和道具的



(炸弹爆炸之前)



(放下炸弹)

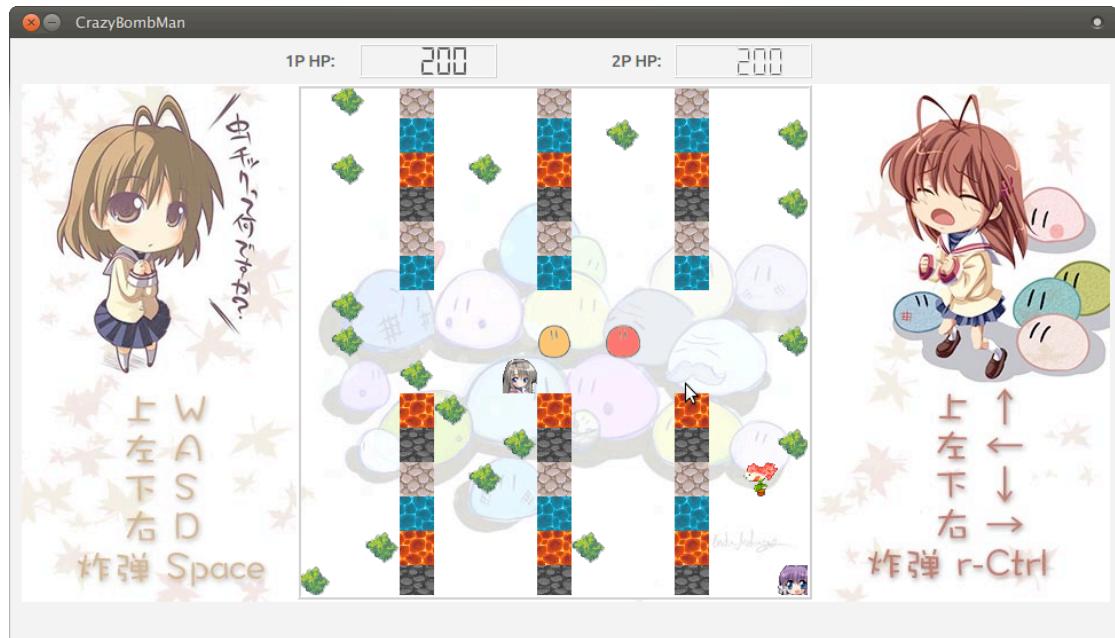


(炸弹爆炸之后)

11) 道具一：盆栽鱼



(盆栽鱼出现)



(放盆栽鱼之前最多能放两个泡泡)



(放完泡泡之后最多能放三个泡泡)

12) 道具二：小鱼干

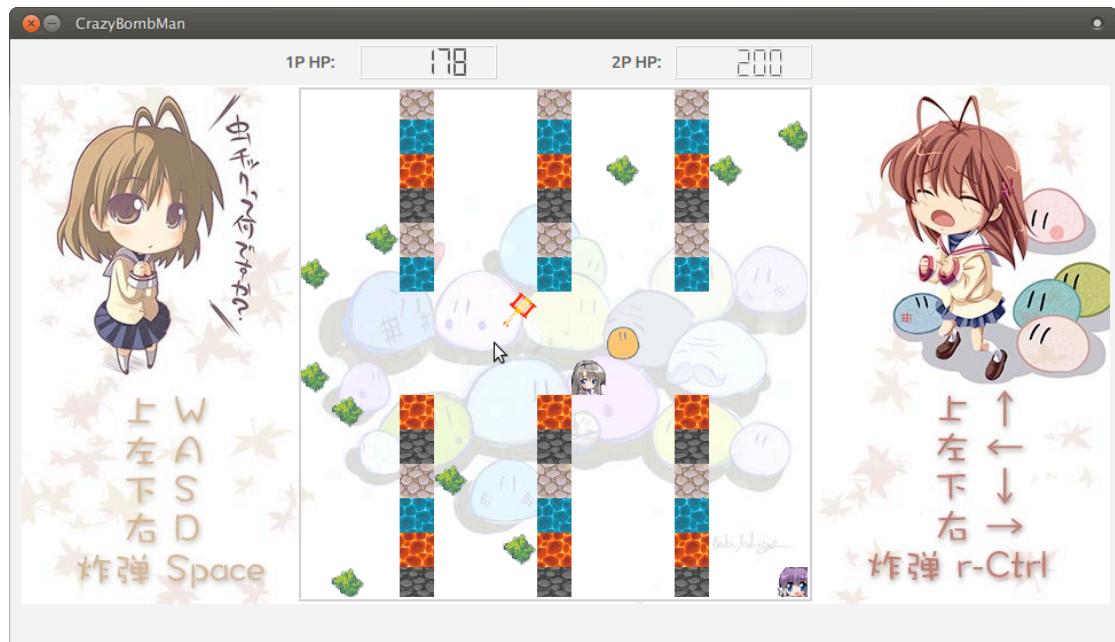


(吃小鱼干之前的血量)



(吃完小鱼干之后的血量)

13) 道具三：小铁锤



(用于增加泡泡威力)



14) 四炮连放的动画情况



15) 进入娱乐模式

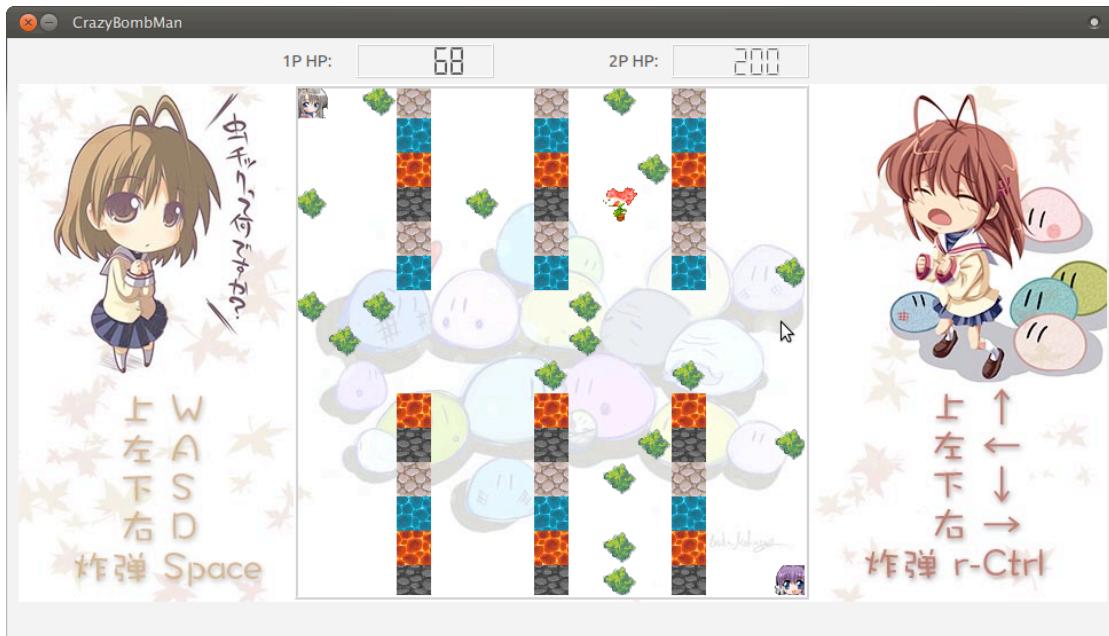
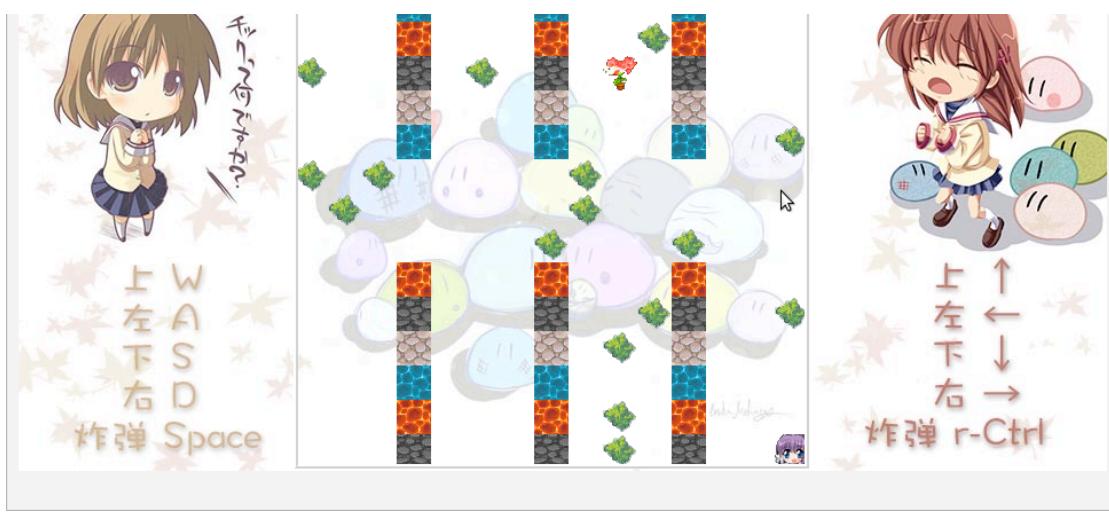


16) 进入地狱模式



17) 进入幽灵模式，所有的怪物都是隐形怪物。







(虽然看不见怪物，但是如果不动的话，仍然会被怪物攻击进而掉血）

五 . 项目人员、进度安排及完成过程

1. 项目人员：

项目人员	人员分工
李琛	<ol style="list-style-type: none">整体游戏框架的设计照料组员的生活学习以及心理状态调控项目进度同步成员间代码协助美工组进行选图，美化工作

	<p>6. 协助算法组修改算法，调试代码</p> <p>7. 参与了最终的文档写作</p>
杨璨	<p>1. 整体游戏框架的设计</p> <p>2. 作为界面组成员修改界面，调试代码</p> <p>3. 协助算法组修改算法，调试代码</p> <p>4. 文档设计</p> <p>5. 类之间的结构设计</p> <p>6. 为大家搭建 QT 环境</p> <p>7. 调研了 Phonon, 定时器，以及 QT Designer 等工具</p> <p>8. 参与了最终的文档写作</p>
姜博耀	<p>1. 整体游戏算法设计</p> <p>2. 游戏算法改进，优化</p> <p>3. 作为算法组组长修改算法，调试代码</p> <p>4. 协助界面组修改界面，调试代码</p>
张从越	<p>1. 整体游戏界面的架构</p> <p>2. 作为界面组组长进行大量的界面架构，调试了大量的代码</p> <p>3. 协助美工组进行选图，制作图片等工作</p> <p>4. 作为最终游戏展示时的主主持人</p>
朱翌辰	<p>1. 整体游戏画风的设计以及调控界面与界面之间，控件与控件之间的风格协调。</p> <p>2. 作为美工组组长进行了大量的选图，抠图，美图</p>

	<p>等重要工作。</p> <ol style="list-style-type: none"> 3. 同时进行挑选游戏音乐，剪辑背景音乐等 I/O 类的繁琐工作 4. 作为最终游戏展示时的主持人之一
朱泓睿	<ol style="list-style-type: none"> 1. 协助美工组进行选图，制作图片等工作 2. 在游戏设计初期调研考察代码 3. 参与架构了游戏整体风格的设定，道具的设定
李思贝	<ol style="list-style-type: none"> 1. 协助美工组进行选图，制作游戏图片等工作 2. 作为最终游戏展示时的游戏解说员，活跃了全场气氛。 3. 很好的调节了组内的气氛，促进了成员之间感情的增进。
郑翔军	<ol style="list-style-type: none"> 1. 在游戏设计初期调研考察代码 2. 修改了键盘 I/O 的映射机制 3. 参与了最终的文档写作
何志鹏	<ol style="list-style-type: none"> 1. 作为调研组组长在游戏设计初期调研考察代码 2. 为最终的游戏设计提供了思路以及模板。 3. 为大家搭建 Ubuntu 环境

2. 进度安排：

时间	任务	人员安排
5月8日-5月11日	游戏设计和开发	

	环境学习	
5月12日-5月14日	核心算法和程序框架生成	
5月15日-5月17日	游戏主题的确定和代码的细化，图形化界面的构建	
5月17日-21日	美工继续完善，游戏主题和设计的完善	
5月21日-22日	最后的优化和调试	

3. 完成过程：

在本次整个项目的前期设计，中期分块编写，后期整合和调试都集合了全组所有同学的力量和劳动。

本组同学会共同抽出时间集中讨论思想和编写代码，之后由组长给大家进行明确的分工，大家进行自己的代码实现等。

总体来说，我们分为美工组，算法组，界面组，整合组。美工组负责给界面的图片进行 PS 等优化工作，算法组负责游戏核心算法的实现和优化，界面组负责提供给用户一个良好的接口，整合组负责链接所有的代码形成有机的整体。

最终在 Deadline 之前完成了 Game Edition10.0 的代码同步以及调试工作。

六 . 项目心得及体会

这虽然只是一个小小的游戏，但是凝聚了我们组耗时长达两周的工作，也是我们组同学力推的合作完成的第一个游戏项目。项目在最开始往往是最艰难的，因为没有经验，所以不知道如何选择项目题目以及设计好游戏的每个模块。在大量调研和了解 Linux 系统在游戏方面的特性后，我们决定使用 QT 和 C++ 来完成整个项目。之后我们组的同学进行了一个大致的人员分类。一部分同学去学习了 QT，一部分同学学习如何编写游戏代码，一部分同学负责游戏创意和主题的设计。在项目的构建过程中，我们组同学都认真积极，每次的集中敲代码之前，本组人员都会自己提出对本次项目看法或者是困惑，在大家讨论之后进行帮助和创新，帮助了我们解决一些工程商的问题。

我们的游戏是基于炸弹人想法的 Clannad 主题的团子炸弹人，在游戏选择界面的设计，和主界面的设计都有我们美工组同学的心血；在游戏主界面的设计上有我们算法组同学和设计组同学的灵感和汗水，从绚丽的背景、动听的音乐、精美的图片上都足矣展现我们组同学的热情和对用户的考虑，频繁的测试和严整的文档工作也为我们的进井有条创造了有利的条件。

总之，这次项目的整个过程让我们所有的成员受益匪浅，项目的完成和效果也让我们欢欣鼓舞，从开始的讨论到自始至终的合作以及最后的磨合，所有成员都付出了巨大的努力，我们也很感激老师给我们这次机会，让我们提高自己的能力，也让我们更加了解 LINUX，以及对团队合作有了更加深刻的认识。

七 . 附录（附上项目其他文档，具体内容根据项目要求确定

附录一：

```
void BombMan::enemyMove(Enemy *enemy)
{
    enemy->moveTime=enemy->moveTime-1;
    /* if(enemy->moveTime<10)
    {
        QMessageBox::about(0,"info",QString::number(enemy->x));
    }*/
    if(enemy->moveTime==0)
    {
        //QMessageBox::about(0,"info","here");
        if(enemy->isBoss)
        {
            enemy->moveTime=300/FRAME;           //重装值， Boss 的移动速度为
300/Frame, 越小越快
        }
        else
        {
            enemy->moveTime=400/FRAME;           //重装值， 小怪的移动速度为
400/Frame, 越小越快
        }
    }
    int x[2],y[2],z[2];
    for(int i=0;i<2;i++)                  //计算与两个玩家的距离
    {
        if((player[i]->x)>(enemy->x))
        {
            x[i]=(player[i]->x)-(enemy->x);
        }
        else
        {
            x[i]=(enemy->x)-(player[i]->x);
        }
        if((player[i]->y)>(enemy->y))
        {
            y[i]=(player[i]->y)-(enemy->y);
        }
        else
        {
            y[i]=(enemy->y)-(player[i]->y);
        }
    }
}
```

```

        }
        z[i]=x[i]*x[i]+y[i]*y[i];
    }

    if(z[0]<z[1]) //距离玩家 1 较近,则向玩家 1
移动
{
    if(x[0]>y[0]) //横坐标更长, 横向靠近
    {
        if((player[0]->x)<(enemy->x)) //玩家 1 在 enemy 左边
        {
            if(sthExist[enemy->x-1][enemy->y]<6)//如果 enemy 左边没石头, 炸弹,
则向左移动
        }

if((sthExist[enemy->x-1][enemy->y]==0)|| (sthExist[enemy->x-1][enemy->y]==5))
{
    sthExist[enemy->x-1][enemy->y]=4;
}

if((sthExist[enemy->x][enemy->y]==4)|| (sthExist[enemy->x][enemy->y]==5))
{
    sthExist[enemy->x][enemy->y]=0;
}
enemy->x=enemy->x-1;
}

else //看看纵向移动
{
    if((player[0]->y)<(enemy->y)) //如果玩家 1 在 enemy 上面
    {
        if(sthExist[enemy->x][enemy->y-1]<6)//如果 enemy 上边没石
头, 炸弹,则向上移动
    }

if((sthExist[enemy->x][enemy->y-1]==0)|| (sthExist[enemy->x][enemy->y-1]==5))
{
    sthExist[enemy->x][enemy->y-1]=4;
}

if((sthExist[enemy->x][enemy->y]==4)|| (sthExist[enemy->x][enemy->y]==5))
{
    sthExist[enemy->x][enemy->y]=0;
}
enemy->y=enemy->y-1;
}

```

```

        }
    }
else //如果玩家 1 在 enemy 下面
{
    if(sthExist[enemy->x][enemy->y+1]<6)//如果 enemy 下边没
石头，炸弹，则向下移动
    {

if((sthExist[enemy->x][enemy->y+1]==0)||(sthExist[enemy->x][enemy->y+1]==5))
{
    sthExist[enemy->x][enemy->y+1]=4;
}

if((sthExist[enemy->x][enemy->y]==4)||(sthExist[enemy->x][enemy->y]==5))
{
    sthExist[enemy->x][enemy->y]=0;
}
enemy->y=enemy->y+1;
}

}

}

else //玩家 1 在 enemy 右边
{
    if(sthExist[enemy->x+1][enemy->y]<6)//如果 enemy 右边没石头，炸弹，
则向右移动
    {

if((sthExist[enemy->x+1][enemy->y]==0)||(sthExist[enemy->x+1][enemy->y]==5))
{
    sthExist[enemy->x+1][enemy->y]=4;
}

if((sthExist[enemy->x][enemy->y]==4)||(sthExist[enemy->x][enemy->y]==5))
{
    sthExist[enemy->x][enemy->y]=0;
}
enemy->x=enemy->x+1;
}

else //看看纵向移动
{
    if((player[0]->y)<(enemy->y)) //如果玩家 1 在 enemy 上面
    {
        if(sthExist[enemy->x][enemy->y-1]<6)//如果 enemy 上边没石

```

头，炸弹,则向上移动

```
{
```

```
if((sthExist[enemy->x][enemy->y-1]==0)||(sthExist[enemy->x][enemy->y-1]==5))
```

```
{
```

```
    sthExist[enemy->x][enemy->y-1]=4;
```

```
}
```

```
if((sthExist[enemy->x][enemy->y]==4)||(sthExist[enemy->x][enemy->y]==5))
```

```
{
```

```
    sthExist[enemy->x][enemy->y]=0;
```

```
}
```

```
    enemy->y=enemy->y-1;
```

```
}
```

```
}
```

```
else //如果玩家 1 在 enemy 下面
```

```
{
```

```
    if(sthExist[enemy->x][enemy->y+1]<6)//如果 enemy 下边没
```

石头，炸弹,则向下移动

```
{
```

```
if((sthExist[enemy->x][enemy->y+1]==0)||(sthExist[enemy->x][enemy->y+1]==5))
```

```
{
```

```
    sthExist[enemy->x][enemy->y+1]=4;
```

```
}
```

```
if((sthExist[enemy->x][enemy->y]==4)||(sthExist[enemy->x][enemy->y]==5))
```

```
{
```

```
    sthExist[enemy->x][enemy->y]=0;
```

```
}
```

```
    enemy->y=enemy->y+1;
```

```
}
```

```
}
```

```
}
```

```
else //纵坐标更长，纵向靠近
```

```
{
```

```
    if((player[0]->y)<(enemy->y)) //玩家 1 在 enemy 上边
```

```
{
```

if(sthExist[enemy->x][enemy->y-1]<6)//如果 enemy 上边没石头，炸弹,
 则向左移动

```
{
```

```

if((sthExist[enemy->x][enemy->y-1]==0)||(sthExist[enemy->x][enemy->y-1]==5))
{
    sthExist[enemy->x][enemy->y-1]=4;
}

if((sthExist[enemy->x][enemy->y]==4)||(sthExist[enemy->x][enemy->y]==5))
{
    sthExist[enemy->x][enemy->y]=0;
}
enemy->y=enemy->y-1;

else //看看横移动
{
    if((player[0]->x)<(enemy->x)) //如果玩家 1 在 enemy 左面
    {
        if(sthExist[enemy->x-1][enemy->y]<6)//如果 enemy 左边没石头, 炸弹,则向左移动
    }

if((sthExist[enemy->x-1][enemy->y]==0)||(sthExist[enemy->x-1][enemy->y]==5))
{
    sthExist[enemy->x-1][enemy->y]=4;
}

if((sthExist[enemy->x][enemy->y]==4)||(sthExist[enemy->x][enemy->y]==5))
{
    sthExist[enemy->x][enemy->y]=0;
}
enemy->x=enemy->x-1;

}
else //如果玩家 1 在 enemy 右面
{
    if(sthExist[enemy->x+1][enemy->y]<6)//如果 enemy 右边没石头, 炸弹,则向右移动
}

if((sthExist[enemy->x+1][enemy->y]==0)||(sthExist[enemy->x+1][enemy->y]==5))
{
    sthExist[enemy->x+1][enemy->y]=4;
}

if((sthExist[enemy->x][enemy->y]==4)||(sthExist[enemy->x][enemy->y]==5))
{
}

```

```

                sthExist[enemy->x][enemy->y]=0;
            }
            enemy->x=enemy->x+1;
        }
    }
}
else //玩家 1 在 enemy 下边
{
    if(sthExist[enemy->x][enemy->y+1]<6)//如果 enemy 下边没石头, 炸弹,
则向下移动
    {
        if((sthExist[enemy->x][enemy->y+1]==0)||(sthExist[enemy->x][enemy->y+1]==5))
        {
            sthExist[enemy->x][enemy->y+1]=4;
        }

        if((sthExist[enemy->x][enemy->y]==4)||(sthExist[enemy->x][enemy->y]==5))
        {
            sthExist[enemy->x][enemy->y]=0;
        }
        enemy->y=enemy->y+1;
    }
    else //看看横向移动
    {
        if((player[0]->x)<(enemy->x)) //如果玩家 1 在 enemy 左面
        {
            if(sthExist[enemy->x-1][enemy->y]<6)//如果 enemy 左边没石
头, 炸弹,则向左移动
            {
                if((sthExist[enemy->x-1][enemy->y]==0)||(sthExist[enemy->x-1][enemy->y]==5))
                {
                    sthExist[enemy->x-1][enemy->y]=4;
                }

                if((sthExist[enemy->x][enemy->y]==4)||(sthExist[enemy->x][enemy->y]==5))
                {
                    sthExist[enemy->x][enemy->y]=0;
                }
                enemy->x=enemy->x-1;
            }
        }
    }
}

```



```

else //看看纵向移动
{
    if((player[1]->y)<(enemy->y)) //如果玩家 2 在 enemy 上面
    {
        if(sthExist[enemy->x][enemy->y-1]<6)//如果 enemy 上边没石
头, 炸弹,则向上移动
    }

if((sthExist[enemy->x][enemy->y-1]==0)||(sthExist[enemy->x][enemy->y-1]==5))
{
    sthExist[enemy->x][enemy->y-1]=4;
}

if((sthExist[enemy->x][enemy->y]==4)||(sthExist[enemy->x][enemy->y]==5))
{
    sthExist[enemy->x][enemy->y]=0;
}
enemy->y=enemy->y-1;
}

else //如果玩家 2 在 enemy 下面
{
    if(sthExist[enemy->x][enemy->y+1]<6)//如果 enemy 下边没
石头, 炸弹,则向下移动
}

if((sthExist[enemy->x][enemy->y+1]==0)||(sthExist[enemy->x][enemy->y+1]==5))
{
    sthExist[enemy->x][enemy->y+1]=4;
}

if((sthExist[enemy->x][enemy->y]==4)||(sthExist[enemy->x][enemy->y]==5))
{
    sthExist[enemy->x][enemy->y]=0;
}
enemy->y=enemy->y+1;
}

else //玩家 2 在 enemy 右边
{
    if(sthExist[enemy->x+1][enemy->y]<6)//如果 enemy 右边没石头, 炸弹,
则向右移动
}

```

```

if((sthExist[enemy->x+1][enemy->y]==0)||(sthExist[enemy->x+1][enemy->y]==5))
{
    sthExist[enemy->x+1][enemy->y]=4;
}

if((sthExist[enemy->x][enemy->y]==4)||(sthExist[enemy->x][enemy->y]==5))
{
    sthExist[enemy->x][enemy->y]=0;
}
enemy->x=enemy->x+1;
}

else //看看纵向移动
{
    if((player[1]->y)<(enemy->y)) //如果玩家 2 在 enemy 上面
    {
        if(sthExist[enemy->x][enemy->y-1]<6)//如果 enemy 上边没石
头, 炸弹,则向上移动
    }

if((sthExist[enemy->x][enemy->y-1]==0)||(sthExist[enemy->x][enemy->y-1]==5))
{
    sthExist[enemy->x][enemy->y-1]=4;
}

if((sthExist[enemy->x][enemy->y]==4)||(sthExist[enemy->x][enemy->y]==5))
{
    sthExist[enemy->x][enemy->y]=0;
}
enemy->y=enemy->y-1;
}

else //如果玩家 2 在 enemy 下面
{
    if(sthExist[enemy->x][enemy->y+1]<6)//如果 enemy 下边没
石头, 炸弹,则向下移动
}

if((sthExist[enemy->x][enemy->y+1]==0)||(sthExist[enemy->x][enemy->y+1]==5))
{
    sthExist[enemy->x][enemy->y+1]=4;
}

```

```

if((sthExist[enemy->x][enemy->y]==4)|| (sthExist[enemy->x][enemy->y]==5))
{
    {
        sthExist[enemy->x][enemy->y]=0;
    }
    enemy->y=enemy->y+1;
}
}

}

}

}

else //纵坐标更长, 纵向靠近
{
    if((player[1]->y)<(enemy->y)) //玩家 2 在 enemy 上边
    {
        if(sthExist[enemy->x][enemy->y-1]<6)//如果 enemy 上边没石头, 炸弹,
则向左移动
    }

if((sthExist[enemy->x][enemy->y-1]==0)|| (sthExist[enemy->x][enemy->y-1]==5))
{
    {
        sthExist[enemy->x][enemy->y-1]=4;
    }
}

if((sthExist[enemy->x][enemy->y]==4)|| (sthExist[enemy->x][enemy->y]==5))
{
    {
        sthExist[enemy->x][enemy->y]=0;
    }
    enemy->y=enemy->y-1;
}
else //看看横移动
{
    if((player[1]->x)<(enemy->x)) //如果玩家 2 在 enemy 左面
    {
        if(sthExist[enemy->x-1][enemy->y]<6)//如果 enemy 左边没石
头, 炸弹,则向左移动
    }

if((sthExist[enemy->x-1][enemy->y]==0)|| (sthExist[enemy->x-1][enemy->y]==5))
{
    {
        sthExist[enemy->x-1][enemy->y]=4;
    }
}

if((sthExist[enemy->x][enemy->y]==4)|| (sthExist[enemy->x][enemy->y]==5))
{
}

```

```

        sthExist[enemy->x][enemy->y]=0;
    }
    enemy->x=enemy->x-1;
}
}
else //如果玩家 2 在 enemy 右面
{
    if(sthExist[enemy->x+1][enemy->y]<6)//如果 enemy 右边没
石头，炸弹,则向右移动
{
}

if((sthExist[enemy->x+1][enemy->y]==0)||(sthExist[enemy->x+1][enemy->y]==5))
{
    sthExist[enemy->x+1][enemy->y]=4;
}

if((sthExist[enemy->x][enemy->y]==4)||(sthExist[enemy->x][enemy->y]==5))
{
    sthExist[enemy->x][enemy->y]=0;
}
enemy->x=enemy->x+1;
}

}

}

else //玩家 2 在 enemy 下边
{
    if(sthExist[enemy->x][enemy->y+1]<6)//如果 enemy 下边没石头,炸弹,
则向下移动
{
}

if((sthExist[enemy->x][enemy->y+1]==0)||(sthExist[enemy->x][enemy->y+1]==5))
{
    sthExist[enemy->x][enemy->y+1]=4;
}

if((sthExist[enemy->x][enemy->y]==4)||(sthExist[enemy->x][enemy->y]==5))
{
    sthExist[enemy->x][enemy->y]=0;
}
enemy->y=enemy->y+1;
}

else //看看横向移动
{
}

```

附录二：

```

/*炸弹相关函数
*用一个链表串起炸弹，用 FIFO 的方式爆炸炸弹
*/
void BombMan::bombAbout(void)
{
    Bomb *p=head->next;
    while(p)
    {
        p->lastTime=p->lastTime-1;
        int tmp=p->lastTime;
        if(tmp < (1000/FRAME))
        {
            mdBomb->play();
            sthExist[p->x][p->y]=2;//爆炸的炸弹
            //之后模拟炸弹爆炸
            if(tmp < (900/FRAME))//给上下左右 4 个位置放上炸弹的影响
            {
                if(((p->x)>0)&&(sthExist[p->x-1][p->y]<7))
                    sthExist[(p->x)-1][p->y]=2;
                if(((p->x)<14)&&(sthExist[p->x+1][p->y]<7))
                    sthExist[(p->x)+1][p->y]=2;
                if(((p->y)>0)&&(sthExist[p->x][p->y-1]<7))
                    sthExist[p->x][(p->y)-1]=2;
                if(((p->y)<14)&&(sthExist[p->x][p->y+1]<7))
                    sthExist[p->x][(p->y)+1]=2;
                if(tmp < (800/FRAME))//给距离为 2 的上下左右 4 个位置放上炸弹的影响
                {
                    if(((p->x)>1)&&(sthExist[p->x-2][p->y]<7))
                        sthExist[(p->x)-2][p->y]=2;
                    if(((p->x)<13)&&(sthExist[p->x+2][p->y]<7))
                        sthExist[(p->x)+2][p->y]=2;
                    if(((p->y)>1)&&(sthExist[p->x][p->y-2]<7))
                        sthExist[p->x][(p->y)-2]=2;
                    if(((p->y)<13)&&(sthExist[p->x][p->y+2]<7))
                        sthExist[p->x][(p->y)+2]=2;
                    if(((player[p->master]->attackRange) > 2)&&(tmp < (700/FRAME)))//给
                    距离为 3 的上下左右 4 个位置放上炸弹的影响
                    {
                        if(((p->x)>2)&&(sthExist[p->x-3][p->y]<7))
                            sthExist[(p->x)-3][p->y]=2;
                        if(((p->x)<12)&&(sthExist[p->x+3][p->y]<7))
                            sthExist[(p->x)+3][p->y]=2;
                        if(((p->y)>2)&&(sthExist[p->x][p->y-3]<7))
                            sthExist[p->x][(p->y)-3]=2;
                    }
                }
            }
        }
    }
}

```

```

        if(((p->y)<12)&&(sthExist[p->x][p->y+3]<7))
            sthExist[p->x][(p->y)+3]=2;
    }
    if(tmp == 0)//清除炸弹的影响
    {
        sthExist[p->x][p->y]=0;
        if(((p->x)>0)&&(sthExist[p->x-1][p->y]<7))
            sthExist[(p->x)-1][p->y]=0;
        if(((p->x)<14)&&(sthExist[p->x+1][p->y]<7))
            sthExist[(p->x)+1][p->y]=0;
        if(((p->y)>0)&&(sthExist[p->x][p->y-1]<7))
            sthExist[p->x][(p->y)-1]=0;
        if(((p->y)<14)&&(sthExist[p->x][p->y+1]<7))
            sthExist[p->x][(p->y)+1]=0;
        if(((p->x)>1)&&(sthExist[p->x-2][p->y]<7))
            sthExist[(p->x)-2][p->y]=0;
        if(((p->x)<13)&&(sthExist[p->x+2][p->y]<7))
            sthExist[(p->x)+2][p->y]=0;
        if(((p->y)>1)&&(sthExist[p->x][p->y-2]<7))
            sthExist[p->x][(p->y)-2]=0;
        if(((p->y)<13)&&(sthExist[p->x][p->y+2]<7))
            sthExist[p->x][(p->y)+2]=0;
        if((player[p->master]->attackRange) > 2)
        {
            if(((p->x)>2)&&(sthExist[p->x-3][p->y]<7))
                sthExist[(p->x)-3][p->y]=0;
            if(((p->x)<12)&&(sthExist[p->x+3][p->y]<7))
                sthExist[(p->x)+3][p->y]=0;
            if(((p->y)>2)&&(sthExist[p->x][p->y-3]<7))
                sthExist[p->x][(p->y)-3]=0;
            if(((p->y)<12)&&(sthExist[p->x][p->y+3]<7))
                sthExist[p->x][(p->y)+3]=0;
        }
        //玩家炸弹数减一
    }

player[p->master]->bombCount=(player[p->master]->bombCount)-1;
//释放炸弹
Bomb *tp=head;
while((tp->next)!=p)
{
    tp=tp->next;
    tp->next=p->next;
    free(p);
}
}

```

```
    }  
    }  
    p=p->next;  
}  
}
```