设计报告

**南京理工大学计算机科学与工程学院**

**C++课程设计 报告**

**班 级 22404班 \_\_\_\_**

**学生姓名 杨靖翔，赵硕\_\_\_\_\_\_**

**学 号 922106840441**

**起止时间 \_\_\_\_\_\_ 8.14-9.22 \_\_**

**指导教师 \_\_\_\_ 娄震 \_ \_\_**

**南京理工大学计算机科学与工程学院制**

目录

1课题二：Baba Is You 游戏复刻 2

1.1课题功能及来源 3

1.2开发平台 3

1.3 总体方案 4

1.3.1总体流程图 4

1.3.2相关数据结构 6

1.4详细设计 6

1.4.1 guide\_scene函数 6

1.4.2 choose\_scene函数 7

1.4.3 game\_scene函数 9

1.4.4 pause\_scene函数 12

1.4.5 setting\_scene函数 13

1.4.6 函数

1.4.7 函数

1.4.8 函数

1.5测试与运行

1.6调试记录

1.7版本记录

1.8 分工情况

1.1课题功能及来源

课题实现了词条功能的结合，合法判断，对方块的属性修改与实际属性生效。实现了键盘输入的移动方式，提供暂停功能，撤回功能与重新开始功能。实现移动，破坏等游戏音效，场景切换时的音乐切换以及音量控制。实现了屏幕抖动与闲置动画的功能与开关选项。

能够读取外部文件实现地图初始化，也能使用地图编辑器生成自定义地图的txt文件。地图编辑器提供所有游戏内方块用于制作地图，能够直接在游戏内测试，读取自定义关卡。提供了搜索功能来查找具有关键词的关卡。

游戏原型——Baba Is You是Arvi Teikari参加2017年Nordic Game Jam制作的参赛作品，其创作灵感来源于推箱子游戏和对逻辑操作符的联想。baba Is You 是一款屡获殊荣的益智游戏，您可以在其中更改游戏规则。在每个关卡中，玩家都会看到一个由各种物体、角色和可移动单词块组成的单屏谜题。规则本身都以块的形式呈现，您可以与之交互；通过操纵它们，可以改变关卡的运作方式并引起意想不到的互动！

1.2 开发平台

Qt是一款跨平台的应用程序开发框架，被广泛用于开发图形用户界面（GUI）和嵌入式系统。Qt最初由挪威的Trolltech公司开发，现在由Digia公司维护和开发。Qt的目标是提供一种简单、直观、高效的开发环境，使开发人员能够快速构建出功能丰富、可靠稳定的应用程序。

其次，Qt提供了丰富的功能和组件，使开发人员能够轻松构建出各种复杂的应用程序。Qt包含了大量的预定义控件和类，例如按钮、文本框、列表框等，开发人员可以直接使用这些控件来构建用户界面。此外，Qt还提供了强大的图形渲染引擎和图形效果，使应用程序界面更加美观和吸引人。

另外，Qt还提供了一套完善的工具和开发环境，帮助开发人员进行代码编辑、调试和测试。Qt Creator是Qt的官方集成开发环境（IDE），它提供了直观的界面和丰富的功能，使开发人员能够更加高效地开发和调试应用程序。

1.3总体方案

1.3.1 总体流程图

图表, 箱线图

描述已自动生成

1.3.2 相关数据结构

1. Block类

- 简介：核心结构，所有方块都是其实例

1. #include "config.h"
2. #include "animation.h"
4. #include <QPixmap>
5. #include <QLabel>
7. **class** Block
8. {
9. //函数
10. **public**:
11. Block();
12. //坐标更新
13. **void** setPosition(**int** block\_to);
14. //载入图片
15. **void** loadPix(**int** type, **int** name, **bool** openAni);
17. //对象
18. **public**:
19. //图片资源
20. QPixmap block\_pix;
22. //动图
23. Animation block\_animation;
25. //坐标
26. **int** block\_x;
27. **int** block\_y;
29. //方块边长
30. **int** sideLength;
32. //是否可见
33. **bool** block\_isVisible;
34. //是否可触碰
35. **bool** block\_isTouch;
36. //是否可操作
37. **bool** block\_isMove;
38. //是否可推动
39. **bool** block\_isPush;
40. //是否是胜利方块
41. **bool** block\_isWin;
42. //是否是危险方块
43. **bool** block\_isDanger;
44. //是否是自爆方块
45. **bool** block\_isBomb;
46. //是否M/H
47. **bool** block\_isMelt;
48. **bool** block\_isHot;
49. //是否S/O
50. **bool** block\_isShut;
51. **bool** block\_isOpen;
52. //是否漂浮
53. **bool** block\_isFloat;
54. //是否溺水
55. **bool** block\_isSink;
56. //是否是特效方块
57. **bool** block\_isDuang;
59. //是否被动移动
60. **bool** block\_isBeMove;
62. //是否生效
63. **bool** block\_isTextEffect;
65. //是否占用（name修改只能生效一个词条）
66. **bool** block\_isWork;
67. //朝向
68. **int** block\_faceTo;
70. //方块类型
71. **int** block\_type;     //ty\_block=1 ty\_subject=1 ty\_predicate=2
72. //方块名
73. **int** block\_name;   //详情见配置文件
75. //主角
76. **int** block\_moveType;
78. };

2. Animation类

- 简介：配合计时器实现动画

1. #include "config.h"
3. #include <QPixmap>
4. #include <QVector>
5. #include <QString>
6. #include <QDebug>


10. **class** Animation
11. {
12. //函数
13. **public**:
14. Animation();
16. //参数初始化
17. **void** iniAni(**int** interval,**int** maxIndex,QString str,**int** height,**int** width);
18. //更新下标
19. **void** updateIndex();

22. //对象
23. **public**:
24. //图库
25. QVector<QPixmap> ani\_pixVector;
26. //切图计时
27. **int** ani\_recorder;
28. //切图间隔
29. **int** ani\_interval;
30. //图片下标
31. **int** ani\_index;
32. //图片最大下标
33. **int** ani\_maxIndex;
35. };

1.4 详细设计

1.4.1 guide\_scene函数

1. 概要

1. //启动刷新
2. **void** playAni();
3. //绘图事件\*
4. **void** paintEvent(QPaintEvent \*event);
5. //读行文件
6. **int** readLineInt(QFile &file);
7. //初始化积分
8. **void** iniScore();
9. //上传积分
10. **void** uploadScore();
11. //音乐播放
12. **void** musicPlay(**int** type);

2. connect：负责处理不同窗口的切换与切屏动画播放



1.4.2 choose\_scene函数

1. 概要

1. //背景动图开始
2. **void** playAni();
3. //绘图事件
4. **void** paintEvent(QPaintEvent \*event);
5. //地图初始化
6. **void** iniMap(**int** map,**int** scores);
7. //行读取
8. **int** readLineInt(QFile &file);
9. //键盘事件
10. **void** keyPressEvent(QKeyEvent \*event);
11. //获取当前名
12. **void** getNowName();

2. 地图初始化函数

1. //——————————————————————————————————————从文件读取地图————————————————————————————————————
2. //根据关卡初始化以下属性：isVisible,type,name,x,y,pix
3. **void** Choose\_Scene::iniMap(**int** map,**int** scores)
4. {
5. score=scores;
6. //初始化
7. **for**(**int** i=0;i<LBLOCK\_NUM;i++)
8. {
9. m\_lblocks[i].lblock\_x=-LENGTH\_OF\_SIDE\_L;
10. m\_lblocks[i].lblock\_y=-LENGTH\_OF\_SIDE\_L;
11. m\_lblocks[i].lblock\_isVisible=**false**;
12. m\_lblocks[i].lblock\_map=MAP\_ISLAND;
13. }
14. //载入地图
15. QFile file;
16. **switch** (map) {
17. **case** MAP\_ISLAND:
18. file.setFileName(MAP\_ISLAND\_PATH);
19. **break**;
20. **case** MAP\_LAVA:
21. file.setFileName(MAP\_LAVA\_PATH);
22. **break**;
23. **case** MAP\_WATER:
24. file.setFileName(MAP\_WATER\_PATH);
25. cout<<"已转到";
26. **break**;
27. **default**:
28. **break**;
29. }
31. back\_mark=map;
32. game\_map = map;
34. **bool** isOk=file.open(QIODevice::ReadOnly);
35. **if**(isOk == **true**)
36. {
37. **int** t\_type;
38. **int** t\_name;
39. **int** t\_num;
40. **int** t\_blocks=0;
41. **while**(file.atEnd() == **false**)
42. {
43. //读类型
44. t\_type=readLineInt(file);
45. //读名字
46. t\_name=readLineInt(file);
47. //读数量
48. t\_num=readLineInt(file);
49. **for**(**int** i=0;i<t\_num;i++)
50. {
51. //对文档输入方块初始化isVisble,type,name,x,y
52. m\_lblocks[t\_blocks].lblock\_isVisible = **true**;
53. m\_lblocks[t\_blocks].lblock\_type = t\_type;
54. m\_lblocks[t\_blocks].lblock\_name = t\_name;
55. m\_lblocks[t\_blocks].loadPix(m\_lblocks[t\_blocks].lblock\_name);
56. //读x坐标
57. m\_lblocks[t\_blocks].lblock\_x = readLineInt(file);
58. //读y坐标
59. m\_lblocks[t\_blocks].lblock\_y = readLineInt(file);
60. //加载所属地图
61. m\_lblocks[t\_blocks].lblock\_map=map;
62. t\_blocks++;
63. }
64. }
65. }
67. }

1.4.3 game\_scene函数

1. 概要

1. //启动游戏 开始利用计数器刷新画面和坐标\*(加星号的是实时刷新的)
2. **void** playGame();
3. //绘图事件\*
4. **void** paintEvent(QPaintEvent \*event);
5. //键盘事件
6. **void** keyPressEvent(QKeyEvent \*event);
7. //方块移动和推动\*
8. **bool** blockMove(**int** key\_value,**int** i,**bool** isFirst);
9. //边界判断
10. **bool** isBorder(**int** key\_value,**int** i);
12. //初始化词条
13. **void** iniText();
14. //地图初始化
15. **void** iniMAP(**int** map, **int** name, QString qName, **int** mode);
16. //文件行int读取
17. **int** readLineInt(QFile &file);
19. //方块修改
20. **void** updateBlock();
21. //属性（效果）更新
22. **void** updateEffect();
23. //属性（效果）修改
24. **void** changeEffect(**int** k,**int** j);
25. //效果实现
26. **void** effectWork();
27. //方块清除
28. **void** killBlock(**int** i);
29. //胜利效果
30. **void** winGame();
31. //屏幕抖动
32. **void** shakeWindow();
34. //撤销功能
35. //记录信息
36. **void** pushStack(**int** x,**int** y,**int** num);
37. //返回
38. **void** popStack();
40. //提示按键
41. **void** isDeadGame();

2. 地图初始化函数

1. //——————————————————————————————————————从文件读取地图————————————————————————————————————
2. //根据txt文本初始化以下属性：isVisible,type,name,x,y,pix
3. **void** Game\_Scene::iniMAP(**int** map,**int** name,QString qName,**int** mode)
4. {
5. nowMap=map;
6. nowLevel=name;
8. effectTextNum=0;
9. //清除所有占用
10. **for**(**int** i=0;i<100;i++)
11. {
12. allName[i]=**false**;
13. }
15. //赢！
16. canWin = **true**;
18. //清空栈
19. remStack.clear();
20. numStack.clear();
21. isWin=**false**;
22. //清空地图（不清空会留存到下局）
23. **for**(**int** i=0;i<BLOCK\_NUM;i++)
24. {
25. m\_blocks[i].block\_isVisible = **false**;
26. //坐标
27. m\_blocks[i].block\_x=-sideLength;
28. m\_blocks[i].block\_y=-sideLength;
30. //是否可见
31. m\_blocks[i].block\_isVisible=**false**;
32. //是否可触碰
33. m\_blocks[i].block\_isTouch=**false**;
34. //是否可操作
35. m\_blocks[i].block\_isMove=**false**;
36. //是否可推动
37. m\_blocks[i].block\_isPush=**false**;
38. //是否是胜利方块
39. m\_blocks[i].block\_isWin=**false**;
40. //是否是危险方块
41. m\_blocks[i].block\_isDanger=**false**;
42. //是否是自爆方块
43. m\_blocks[i].block\_isBomb=**false**;
44. //是否M/H
45. m\_blocks[i].block\_isMelt=**false**;
46. m\_blocks[i].block\_isHot=**false**;
47. //是否S/O
48. m\_blocks[i].block\_isShut=**false**;
49. m\_blocks[i].block\_isOpen=**false**;
50. //是否漂浮
51. m\_blocks[i].block\_isFloat=**false**;
52. //是否溺水
53. m\_blocks[i].block\_isSink=**false**;
54. //是否特效
55. m\_blocks[i].block\_isDuang=**false**;
57. //是否被动移动
58. m\_blocks[i].block\_isBeMove=**false**;
60. //是否生效
61. m\_blocks[i].block\_isTextEffect=**false**;
62. }

65. **if**(map == MAP\_ISLAND)
66. {
67. m\_ubackground\_1.load(":/res/uback2.png");
68. }**else** **if**(map == MAP\_WATER)
69. {
70. m\_ubackground\_1.load(":/res/uback3.png");
71. }**else** **if**(map == MAP\_LAVA)
72. {
73. m\_ubackground\_1.load(":/res/uback1.png");
74. }
76. QString str;
77. **if**(mode == 0)
78. {
79. //初始化游戏地图路径
80. **switch** (map) {
81. **case** MAP\_ISLAND:
82. str = LEVEL\_ISLAND\_PATH;
83. str = str.arg(name);
84. **break**;
85. **case** MAP\_WATER:
86. str = LEVEL\_WATER\_PATH;
87. str = str.arg(name);
88. **break**;
89. **case** MAP\_LAVA:
90. str = LEVEL\_LAVA\_PATH;
91. str = str.arg(name);
92. **break**;
93. **default**:
94. str = FILE\_PATH\_2;
95. **break**;
96. }
97. cout<<"打开关卡地图："<<str;
98. }
99. **else**
100. {
101. str=FILE\_PATH\_EDITOR;
102. str=str.arg(qName);
104. }
106. cout<<map<<" "<<name<<" "<<"地图路径为"<<str;

109. //以下没问题
110. QFile file(str);
111. **bool** isOk=file.open(QIODevice::ReadOnly);
112. cout<<"文件正常打开："<<isOk;
113. **if**(isOk == **true**)
114. {
115. **int** t\_type;
116. **int** t\_name;
117. **int** t\_num;
118. **int** t\_blocks=0;
119. cout<<"文件正常读取："<<file.atEnd();
120. columNum = readLineInt(file);
121. rowNum = readLineInt(file);
122. sideLength = readLineInt(file);
123. cout<<"动态屏幕宽，长方块数+边长"<<columNum<<" "<<rowNum<<" "<<sideLength;
125. inWidth=columNum\*sideLength;
126. inHeight=rowNum\*sideLength;
127. originX=(SCENE\_M\_WIDTH-inWidth)/2;
128. originY=(SCENE\_M\_HEIGHT-inHeight)/2;
130. cout<<"动态屏幕原点："<<originX<<" "<<originY;
131. **while**(file.atEnd() == **false**)
132. {
133. //读类型
134. t\_type=readLineInt(file);
135. //读名字
136. t\_name=readLineInt(file);
137. //读数量
138. t\_num=readLineInt(file);
139. cout<<t\_type<<t\_name<<t\_num;
140. **for**(**int** i=0;i<t\_num;i++)
141. {
142. //对文档输入方块初始化isVisble,type,name,x,y
143. m\_blocks[t\_blocks].block\_isVisible = **true**;
144. m\_blocks[t\_blocks].block\_type = t\_type;
145. m\_blocks[t\_blocks].block\_name = t\_name;
146. m\_blocks[t\_blocks].sideLength = sideLength;
147. m\_blocks[t\_blocks].loadPix(m\_blocks[t\_blocks].block\_type,m\_blocks[t\_blocks].block\_name,openAni);
149. //读x坐标
150. m\_blocks[t\_blocks].block\_x = readLineInt(file)\*sideLength+originX;
151. //读y坐标
152. m\_blocks[t\_blocks].block\_y = readLineInt(file)\*sideLength+originY;
153. t\_blocks++;
154. }
155. }
156. }
158. updateEffect();
159. //统计生效词条数
160. **for**(**int** i=0;i<BLOCK\_NUM;i++)
161. {
162. **if**(m\_blocks[i].block\_isVisible == **true** && m\_blocks[i].block\_isTextEffect == **true**)
163. {
164. effectTextNum++;
165. }
166. }
168. cout<<"初始化最终值"<<effectTextNum;
170. }

1.4.4 pause\_scene函数

1. 概要

1. **void** iniName();
3. **void** quickRestart();

2. 槽与信号相关

1. //信号
2. signals:
3. **void** gameToChoose\_signal();
4. **void** exitToGame\_signal();
5. **void** gameToGuide\_signal();
6. **void** restart\_signal();
7. **void** testLevelToEdit\_signal();
9. **private** slots:
10. **void** on\_b2\_exitToChoose\_released();
11. **void** on\_b1\_start\_released();
12. **void** on\_b5\_exitToGuide\_released();
13. **void** on\_b3\_reStart\_released();

1.4.5 setting\_scene函数

1. 概要

1. //绘图事件\*
2. **void** paintEvent(QPaintEvent \*event);
3. //刷新
4. **void** playAni();

1.4.6

1.4.7

1.4.8

1.5 测试与运行

1. guide\_scene界面



2. choose\_scene界面

电脑游戏的截图

描述已自动生成

3. game\_scene界面

电脑萤幕画面

描述已自动生成

4. pause\_scene界面

电脑萤幕画面

描述已自动生成

5. Editor\_level界面

文本

中度可信度描述已自动生成

6. Editor界面

图形用户界面, 文本, 应用程序

描述已自动生成

1.6 调试记录

1. 检测音效正常

文本, 白板

描述已自动生成

2. 检测方块移动

背景图案

低可信度描述已自动生成

3. 检测地图切换情况

徽标, 公司名称

描述已自动生成

4. 检测键盘输入

图形用户界面, 文本, 应用程序

描述已自动生成

5. 检测撤销功能入栈正常

图形用户界面, 文本, 应用程序, 电子邮件

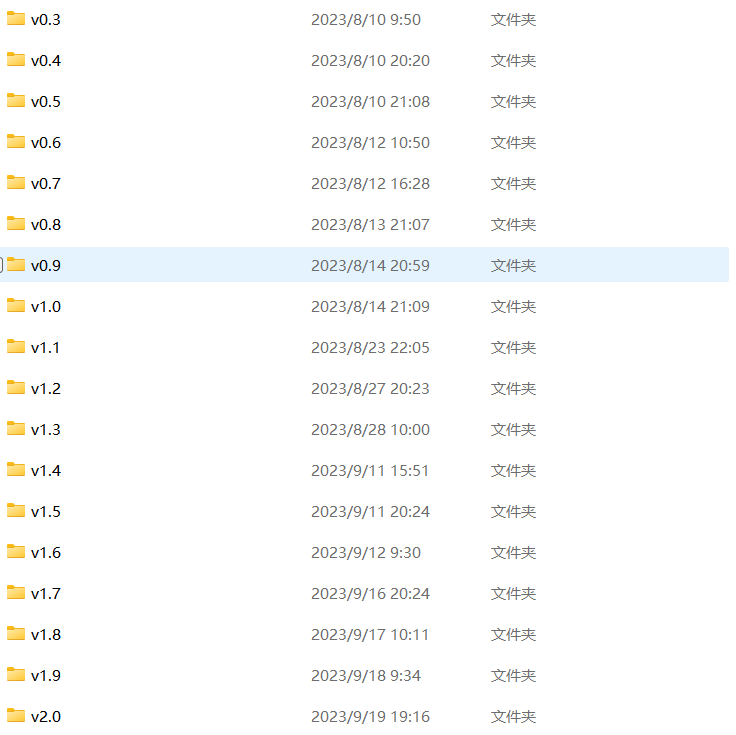
描述已自动生成

6. 检测地图打开情况

文本

描述已自动生成

1.7 版本记录



表格

中度可信度描述已自动生成

1.8 分工情况

杨靖翔：负责animation(.h及.cpp，下同)，block，change\_scene，choose\_scene，config，game\_scene，guide\_scene，lblock，pause\_scene，setting\_scene的代码，部分美工，部分UI设计，代码量2000-2500行

赵硕：负责config，edit\_level，editor，search，set\_level\_name，term\_board，bolck的代码，大部分美工，部分UI设计，代码量1000-1200行