QT 大作业报告

小组队长：黄子烨
小组成员：曹备权
小组成员：黄一鸣

一、程序功能介绍

本小组实现了一个推箱子小游戏。在此游戏中，玩家将扮演猪猪侠，将棒棒糖送给小呆呆，将奖牌送给迷糊老师。在传统推箱子的基础上，本程序允许一次推动两个箱子，同时加入了传送门元素。玩家需要在规定步数内完成游戏谜题。由于主打一个休闲等待时玩的小游戏，因此游戏并没有设置为全屏（虽然玩家可以这样做），以方便玩家同时关注其他信息。

二、项目各模块与类设计细节

（1）主界面设计：

红色的“GGBOX”标题部分，采用 QLabel 类；

利用信号和槽，Start 按钮部分，点击会进入关卡选择界面，Exit 按钮部分，点击会退出程序；

利用 QMediaPlayer 实现背景音乐的播放。

这两部分，设置了字体和大小，并且将普通状态下的按钮背景设定为透明；

利用了 QPainter 绘制了背景图片部分；

（2）关卡选择界面；

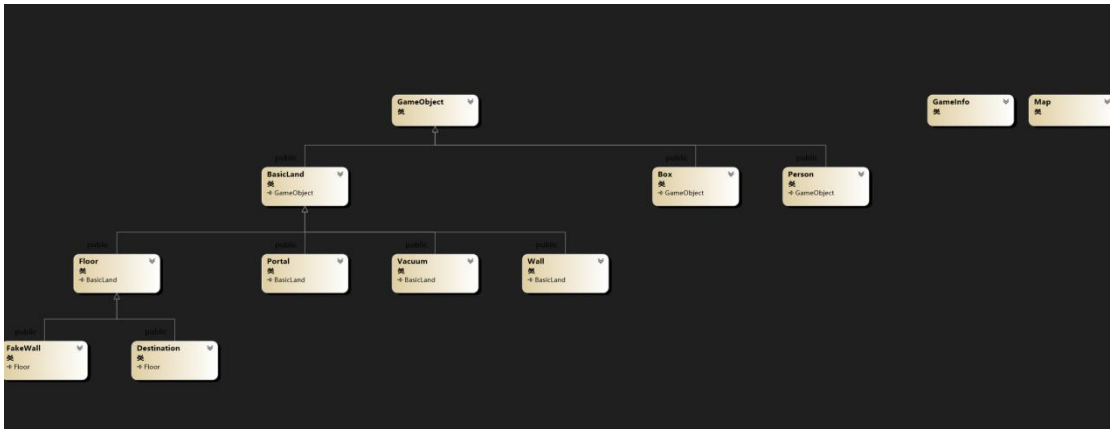
竖排“levels”标签，利用 QLabel 绘制，为了适配背景设置了字体和大小；

八个关卡选择的按钮，点击任意一个会进入相应的关卡，将普通状态的按钮背景设定为透明；

背景图片部分；

其中，（1）中的“start”“exit”和（2）中的 8 个按钮均采用设计的新型 QPushButton 子类“NewButton”，利用 enterevent 和 leaveevent，当鼠标放置在上面的时候颜色和按钮的背景会产生变化，这可以极大增强交互感；

（3）后端模块



以上为后端类继承图

GameObject 为游戏中出现的一切物体的基类，拥有坐标属性 x 和 y。游戏地图分为上下两层。下层为地面，在整局游戏中，地面不会发生任何变化。

地面的基类为 BasicLand 类，继承自 GameObject，并派生出 Floor 类，Portal 类，Vacuum

类，Wall 类。

Floor 类是基础的地板类，任何物体可在上部穿行。

Portal 类是传送门类，每个 Portal 类实例会有一个指向一个 BasicLand 类的指针，当物体在 Portal 类实例上部时，若其所指向的 BasicLand 类上无其他物体，则将该物体传送至所指向的 BasicLand 类上。

Vacuum 类是虚空类，即表明这一块地不在游戏地图范围内，通过添加 Vaccum 类，可以实现不规则地图。

Wall 类是基础的墙类，游戏实体无法在上部穿行。

Floor 类又派生出 Destination 类与 FakeWall 类。

Destination 类是游戏的目的地类，当每个箱子的下方都是 Destination 类时，即可判断游戏胜利。

FakeWall 类是假墙类，拥有一个 bool 类型成员变量 found，当没有物体经过其上方时，found 为 false，表示假墙还没有被发现，在图片界面显示的是墙的图形，当有物体经过其上方时，found 类的值改为 true，表示假墙已被发现，此后在图形界面均显示普通的地板图形。

Person 类与 Box 类派生自 GameObject 类，是游戏主体。

Gameinfo 类是游戏信息类，存储了一关游戏所需要的全部信息，包括地图元素、箱子个数、箱子位置、人位置、目的地位置等。

Map 类是游戏地图类，用以处理游戏过程中几个类交互的过程。checkout 函数用以判断坐标是否超出边界。person_move 函数用以描述人物移动过程。move_box 函数用以描述箱子被推动以及带动其他箱子的过程。move_box 采用递归的写法，返回值表示包括当前箱子，朝向某一个方向有几个连续的箱子。当箱子个数大于 2 时，便不能推动这么多连续的箱子。win_judge 函数用以判断游戏是否结束。

在 person_move 函数中，我们调用了 BasicLand 类的虚函数 accept 函数。之所以使用虚函数，是因为 BasicLand 类的几个派生类对于上方物体经过的权限都有所不同，如在 Wall 类的 accept 函数中，不需要做任何操作，在 Floor 类的 accept 函数中，我们需要改变 person 或者 box 的位置。在 Portal 类的 accept 函数中，我们需要判断是否传送，如果需要，并完成传送。通过 accept 函数的使用，我们提高了代码的可读性与可扩充性，这是面向对象设计的优越性的体现。

（4）关卡界面

每一关对应一个 game 类，其中关卡的信息被保存在源文件里的静态 gameinfo 类对象中。以下是代码的详细介绍：

1）构造函数和析构函数：

game1:game1(QWidget *parent)：该构造函数是 game1 类的构造函数，它继承自 QWidget 类。在构造函数中，调用了 map_initialization 函数来初始化游戏界面，并连接了一些信号和槽函数。

game1::~game1()：这是 game1 类的析构函数，用于释放动态分配的资源。

2)map_initialization 函数：

该函数用于初始化游戏界面。它设置了窗口的大小、标题和图标，并创建了一个返回按钮。将按钮添加到布局管理器中，使其位于界面的底部。

3)paintEvent 函数：

这个函数是一个事件处理函数，当界面需要重绘时被调用。在函数内部，通过 QPainter 类来绘制游戏界面的各个元素，如箱子、人物、目的地、墙等。还绘制了剩余步数的文本，并根据游戏状态绘制相应的提示信息。

4)keyPressEvent 函数：

该函数用于处理键盘按键事件。当玩家按下 W、A、S、D 键时，游戏会调用 testmap 对象的相关方法来移动人物的位置，并更新游戏界面。同时，步数计数器会增加，表示玩家的移动步数，上一步的位置信息也会被保存在一个 gameinfo 类的对象中。
当玩家按下 R 键时，游戏会重新开始当前关卡。对象成员指针指向内容发生改变。
当玩家按下 E 键时，游戏会回退到上一步的状态。根据一个 gameinfo 类的对象改变对象成员指针指向内容，使其回到原先状态。

5）游戏结束和成功的信号处理：

gameOver_after 槽函数：在游戏结束时被调用，将 end 标志设置为 1，断开了刷新界面的连接，即禁止了界面的更新。
gameSuccess_after 槽函数：在游戏成功时被调用，将 end 标志设置为 1，断开了刷新界面的连接，即禁止了界面的更新。

6）返回按钮点击事件：

returnclicked 槽函数：处理返回按钮的点击事件。在该槽函数中，创建一个新的主界面对象 Widget，显示主界面，并关闭当前的游戏窗口。

这段代码实现了一个简单的关卡 1 的游戏界面，玩家可以通过键盘操作人物移动，推动箱子到目的地上。界面会根据游戏状态实时绘制，并显示剩余步数。游戏结束或成功后，会显示相应的提示信息，并禁用界面的更新。玩家可以通过返回按钮返回到主界面。

三、小组成员分工情况

黄子烨同学负责所有后端类与游戏机制的实现，以及带传送门的关卡设计，以及添加背景音乐。曹备权同学负责推箱子界面的实现以及前端与后端的整合工作。黄一鸣同学负责主界面及选关界面的实现。

四、项目总结与反思

本次大作业，我小组三位成员分工明确，将推箱子游戏划分为几个部分，方便同时开发。在这次开发中，我们感受到了面向对象思想在程序开发中的重要作用。通过运用面向对象思

想，我们的程序拥有良好的可扩充性，可以很方便地增加关卡，同时也可以很方便地在地图中加入新的元素，有利于之后的更新与维护。在功能方面，我们没有局限于传统推箱子，而是增加了传送门，假墙等新元素，增加了游戏的趣味性。同时传送门的加入也使得我们必须自己设计关卡（因为网上找不到带传送门的推箱子关卡）。

由于时间与能力的不足，在动画方面并没有实现得很好，导致视觉效果欠佳。另外，由于种种原因，导致项目初期各小组成员的 QT 版本并不相同，导致了开发合作上的困难。