广东东软学院

学生实验报告

**实验课程名称：移动平台开发技术**

**实验项目名称：推箱子游戏**

**实验类型：设计性实验**

**指导教师：王千秋**

**实验日期：2017年 12月 10日**

|  |  |  |  |
| --- | --- | --- | --- |
| **学生姓名** | **张振云** | **学 号** | **15210121002** |
| **班 级** | **软件工程10班** | **专业名称** | **软件工程** |
| **实验组**  **其他成员** | **无** | | |
| **实验地点** | **A410** | | |
| **实验成绩**  **（教师签名）** |  | | |

|  |
| --- |
| **实验目的与要求**  1. 熟练掌握Activity、常用Widget的使用及常用布局的设计  2. 熟练掌握事件监听机制及Android绘图知识  3. 熟悉开发工具Android Studio的使用  4. 通过实验提高分析问题和解决问题的能力 |
| **实验内容**  1. Activity的创建、布局设计及Intent的使用  --游戏开始界面、游戏说明界面及选关卡界面的设计与切换  2. Android绘图知识  --创建游戏区域界面，绘制游戏角色  3. 监听器的使用  --实现搬运工的移动  4. 综合应用--实现地图的存储并按地图绘制游戏区域  5. 综合应用--实现搬运工移动时的逻辑处理 |
| **实验设备与软件环境**  1. Android studio |
| **实验过程与结果（可贴图）**  **1. 创建游戏开始界面、游戏说明界面及选关卡界面**  创建新项目，并在其中：  1) 创建游戏开始主界面MainActivity及其布局文件acivity\_main.xml:    **2) 创建游戏说明界面，并实现界面切换**  在项目中创建GameIntroActivity及其布局文件，可采用RelativeLayout或LinearLayout实现。  游戏说明界面的设计如下：    **3) 创建选关卡界面**  在项目中创建GameLevelActivity及其布局文件activity\_game\_level.xml，这里使用了GridView组件，并采用RelativeLayout布局实现。  该界面布局设计如下：    首先，在项目中创建GameActivity游戏活动，这里不需要创建布局文件，而是通过创建一个自定义的View组件，在该组件中绘制游戏的界面(如网格、角色等)，之后将该组件作为GameActivty的ContentView显示出来。  其次，需要创建GameView组件类，该类继承了View。该组件需要重写onDraw方法来绘制游戏的界面。该方法的原型为**protected void** onDraw(Canvas canvas) ，参数为画布，通过重写该方法在界面上绘制出网络线和搬运工角色。  画布Canvas类的常用方法有：  1) 绘制直线  Paint linePaint = **new** Paint(); linePaint.setColor(Color.***BLACK***); **for** (**int** r=0;r<=***CELL\_NUM\_PER\_LINE***;r++)  canvas.drawLine(0,r\***mCellWidth**,getWidth(),r\***mCellWidth**,linePaint); **for** (**int** c=0;c<=***CELL\_NUM\_PER\_LINE***;c++)  canvas.drawLine(c\***mCellWidth**,0,c\***mCellWidth**,***CELL\_NUM\_PER\_LINE***\***mCellWidth**,linePaint);  2) 绘制矩形  public void drawRect(float left, float top, float right, float bottom,  Paint paint)  参数说明：  left:第一个点距离x轴多少距离(单位像素)  top:第一个点距离y轴多少距离  right:第二个距离x轴多少距离  bottom:第二个点距离y轴多少距离  矩形的长度=right-left   矩形的高度为=bottom-top  **3、实现搬运工的移动**  通过重写GameView的两个方法实现图片的移动：  **public boolean** onTouchEvent(MotionEvent event) {  **if**(isGameOver()){  **return false**;  }  **if** (event.getAction() != MotionEvent.***ACTION\_DOWN***)  **return true**;  **int** touch\_x = (**int**) event.getX();  **int** touch\_y = (**int**) event.getY();   **if** (touch\_below\_to\_man(touch\_x, touch\_y, **mManRow**, **mManColumn**)) {  handleDown();  }  **if** (touch\_up\_to\_man(touch\_x, touch\_y, **mManRow**, **mManColumn**)){  handleUp();  }  **if** (touch\_right\_to\_man(touch\_x, touch\_y, **mManRow**, **mManColumn**)){  handleRight();  }  **if** (touch\_left\_to\_man(touch\_x, touch\_y, **mManRow**, **mManColumn**)){  handleLeft();  }   postInvalidate();   **return true**; }  **4、实现地图的存储并按地图绘制游戏区域**  **public class** GameLevels {  **public static final int *DEFAULT\_ROW\_NUM*** = 12;  **public static final int *DEFAULT\_COLUMN\_NUM*** = 12;   *//游戏区单元格放了什么？* **public static final char *NOTHING*** =**' '**;*//该单元格表示啥也没有* **public static final char *BOX*** =**'B'**;*//g该单元格放的是箱子* **public static final char *FLAG***=**'F'**;*//红旗，表示箱子的目的的* **public static final char *MAN*** =**'M'**;*//搬运工* **public static final char *WALL*** = **'W'**;*//樯* **public static final char *MAN\_FALG*** = **'R'**;*//搬运工+红旗* **public static final char *BOX\_FALG*** = **'X'**;*//箱子加红旗* **public static final** String[] ***LEVEL\_1***={  **"WWWWWWWWWWWW"**,  **"W FW"**,  **"W W"**,  **"W W"**,  **"W WWWW W"**,  **"W W"**,  **"W B W"**,  **"W M W"**,  **"W W"**,  **"W W"**,  **"W W"**,  **"WWWWWWWWWWWW"** };   **public static final** String[] ***LEVEL\_2***={  **"WWWWWWWWWWWW"**,  **"W W"**,  **"W W"**,  **"W WWWWWWW W"**,  **"W W FFB W W"**,  **"W W W B W W"**,  **"W W W W W W"**,  **"W W BMW W W"**,  **"W WFB W W"**,  **"W WFWWWW W"**,  **"W WW W"**,  **"WWWWWWWWWWWW"** };  **public static final** String[] ***LEVEL\_3***={  **"WWWWWWWWWWWW"**,  **"W F W"**,  **"W W"**,  **"W W"**,  **"W WWWW W"**,  **"W M W"**,  **"W B W"**,  **"W B F W"**,  **"W W"**,  **"W W"**,  **"W W"**,  **"WWWWWWWWWWWW"** };    **public static** ArrayList<String[]> *OriginalLevels* = **new** ArrayList<>();*//存储多个开局的列表  //loadGameLevels()的作用是加载关卡列表* **public static void** loadGameLevels(){  **if**(*OriginalLevels*.isEmpty()){  *OriginalLevels*.add(***LEVEL\_1***);*//把第一关的开局添加到开局列表中  OriginalLevels*.add(***LEVEL\_2***);  *OriginalLevels*.add(***LEVEL\_3***);   }  }  *//getLevel()是根据关卡号level得到该关卡的开局（用String【】实现的矩阵）* **public static** String[] getLevel(**int** level){*//level参数是关卡号  loadGameLevels*();  **return** *OriginalLevels*.get(level-1);  }  **public static** List<String> getLevelList(){  *loadGameLevels*();*//加载关卡列表* List<String> levelList = **new** ArrayList<>();*//创建关卡名列表对象levelList ,并分配存储空间* **int** levelNum = *OriginalLevels*.size();*//得到关卡数目* **for**(**int** i = 1;i<=levelNum;i++){ *//对 每一关i* levelList.add(**new** String(**"第"**+i+**"关"**));*//把关卡名* }  **return** levelList;   }  }  **5、实现搬运工移动时的逻辑处理**  **箱子向左为例子：**  **private void** handleLeft() {  **if** (isBoxLeftMan()) {  **if** (**mBoxColumn** >0  && **labelInCells**[**mBoxRow** ].charAt(**mBoxColumn** - 1) != **'W'** && **labelInCells**[**mBoxRow**].charAt(**mBoxColumn**- 1) != **'B'**) {  **labelInCells**[**mManRow**].setCharAt(**mManColumn**, **' '**);  **labelInCells**[**mBoxRow**].setCharAt(**mBoxColumn**, **' '**);  **mBoxColumn**--;  **mManColumn**--;  **mBox**[**boxIndex**][1] = **mBoxColumn**;  **labelInCells**[**mManRow**].setCharAt(**mManColumn**, **'M'**);  **labelInCells**[**mBoxRow**].setCharAt(**mBoxColumn**, **'B'**);  **mSoundPool**.play(**mSoundOneSte**,(**float**)0.5, (**float**)0.5, 1, 0, 1f);  }  } **else if** (**mManColumn** >0&& **labelInCells**[**mManRow** ].charAt(**mManColumn**- 1) != **'W'**) {  **labelInCells**[**mManRow**].setCharAt(**mManColumn**, **' '**);  **mManColumn**--;  **labelInCells**[**mManRow**].setCharAt(**mManColumn**, **'M'**);  **mSoundPool**.play(**mSoundOneSte**,(**float**)0.5, (**float**)0.5, 1, 0, 1f);  } }  **6、实现搬运工箱子和红旗初始化位置：**  *//根据游戏开局初始化搬运工的位置* **public void** get\_gongren\_chushi\_weizhi() { *// StringBuffer[] labelInCells = mGameActivity.getCurrentState().getLabelInCells(); //得到表示局面的二维矩阵* **for** (**int** r = 0; r < GameView.***CELL\_NUM\_PER\_LINE***; r++) *//逐行地扫描* **for** (**int** c = 0; c < GameView.***CELL\_NUM\_PER\_LINE***; c++){ *//对当前行，逐列地扫描* **if** (**labelInCells**[r].charAt(c) == **'M'**) { *//找到搬运工所在位置* **mManRow** = r; *//记住搬运工的行号* **mManColumn** = c; *//记住搬运工的列号* **return**; *//停止扫描。函数返回。* }  }  }   *//根据开局初始化箱子的位置* **public void** get\_XiangZi\_ChuShi\_WeiZhi() {  **int** temp = 0; *// StringBuffer[] labelInCells = mGameActivity.getCurrentState().getLabelInCells(); //得到表示局面的二维矩阵* **for** (**int** r = 0; r < GameView.***CELL\_NUM\_PER\_LINE***; r++) *//从上到下扫描一行* **for** (**int** c = 0; c < GameView.***CELL\_NUM\_PER\_LINE***; c++) { *//对当前行，从左到右扫描* **if** (**labelInCells**[r].charAt(c) == **'B'**) { *//遇到箱子 // mBoxRow = r; //记住箱子的行号 // mBoxColumn = c; //记住箱子的列号 // return; //停止扫描。返回。* temp++;  }  }   **mBox** = **new int**[temp][2];  temp = 0;  **for** (**int** r = 0; r < GameView.***CELL\_NUM\_PER\_LINE***; r++) *//从上到下扫描一行* **for** (**int** c = 0; c < GameView.***CELL\_NUM\_PER\_LINE***; c++) { *//对当前行，从左到右扫描* **if** (**labelInCells**[r].charAt(c) == **'B'**) { *//遇到箱子 // mBoxRow = r; //记住箱子的行号 // mBoxColumn = c; //记住箱子的列号* **mBox**[temp][0] = r;  **mBox**[temp][1] = c;  temp++;  }  }  }  **7、箱子到达目的地时不再移动：**  **public boolean** onTouchEvent(MotionEvent event) {  **if**(isGameOver()){  **return false**;  }  ...  }  **8、游戏操作结果测试：** |
| **操作异常问题与解决方案**  尚未完成箱子的撤销一步等功能。 |
| **实验总结** |