Coursework Report

HONG HONG
40333300@napier.ac.uk
Edinburgh Napier University - Mobile Application Development (SET08114)

Abstract

With time goes, people's material life level is increasing day by day, more and more people are in pursuit of high quality of life. One specific reflect is various categories of mobile apps. In recent years, technology of mobile have made great progress. Mobile apps have become an important part of people's daily life. These applications are in various industries and fields because they have different functions. For example,on the one hand, people can use mobile applications to pay products, book a taxi or restaurant, and they are also helpful on working field. On the other hand, there are many entertainment applications for people to relax. The entertainment applications are an integral pare of people's lives, especially in the fast pace society. In summary, an entertainment application-2048 game is select to build. Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Keywords–Android Studio, 2048, Game, Numbers, Android app

1 Introduction

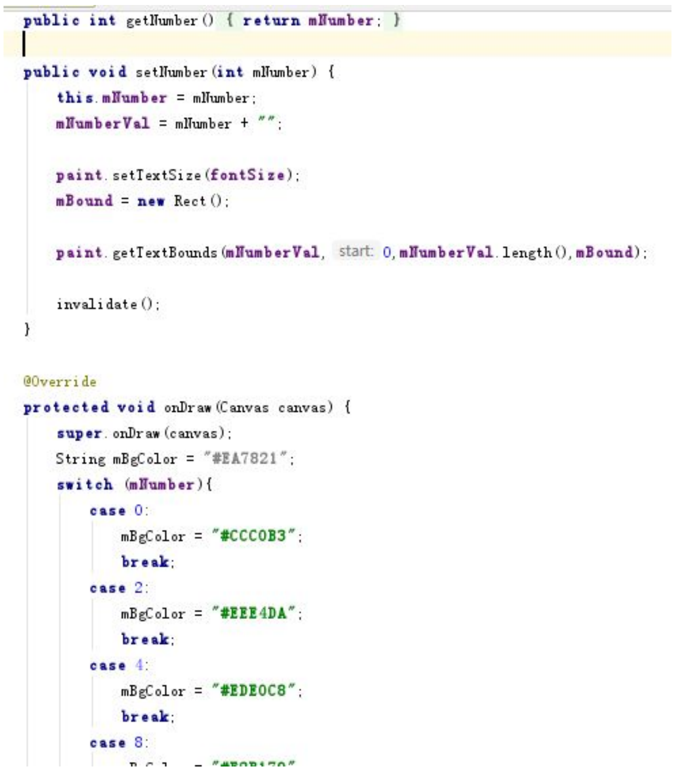
2048 Game is a digital puzzle game. Its rule is simple. At the beginning, a number will generate on the chessboard randomly, the number may be 2 or 4. The player can select four directions to move the number. If it is moved successfully or it is combined with the beside number, the movement you did is regards effective. If the number on the target position is same as moving number, the two numbers will be combined together and the sum is effective score of this step. After each effective step, the empty position will generate a new number(also may be 2 or 4). You will lose the game if the chessboard is filled with numbers and you can not move effectively. Oppositely, you will win the game if you get the final sum-2048.
This game application is built in java language and the platform is Android Studio. In this report, I will introduce the rules of 2048 game at first, and than the processes of its realisation in details.

2 Processes of realisation

First of all, we should define two Views. One is Game2048Item, it is the View of chessboard grids. Another is Game2048Layout, it is used to set all chessboard grids. Below will introduce the processes of realisation in details.

2.1 Game2048Item

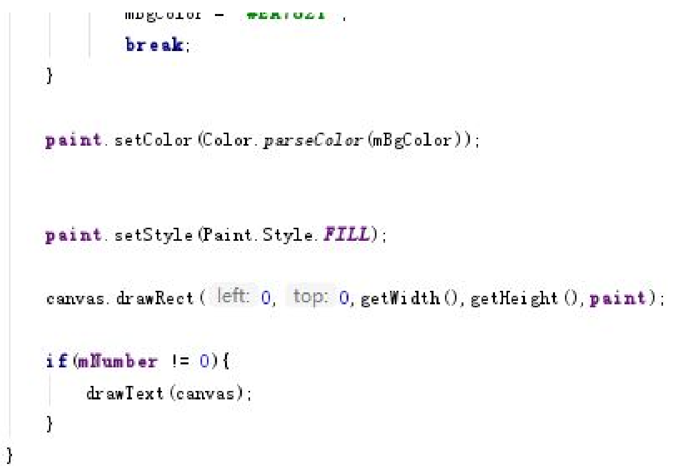
The main function of Game2048Item is to display numbers, so it needs a number attribute to record the current number on chessboard. To make the interface more beautiful, different number is set to have different background colour. After setting background colours, we should print a rectangle which has five parameters, four directions and printer tool. Then we should display numbers on it. One method called Paint.getTextBounds can help to achieve this purpose. This method is used to get corresponding minimum rectangle to the drawn string, and put the minimum rectangle in Rect object. So that we can use Rect object to set the position where to display numbers.



```
public int getNumber() { return mNumber; }

public void setNumber(int mNumber) {
    this.mNumber = mNumber;
    mNumberVal = mNumber + "";

    paint.setTextSize(fontSize);
    mBound = new Rect();

    paint.getTextBounds(mNumberVal, start: 0,mNumberVal.length(),mBound);

    invalidate();
}

@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    String mBgColor = "#EA7821";
    switch (mNumber){
        case 0:
            mBgColor = "#CCC0B3";
            break;
        case 2:
            mBgColor = "#EEE4DA";
            break;
        case 4:
            mBgColor = "#EDE0C8";
            break;
        case 8:
```

Figure 1: Game2048Item

In the onDraw() method, we first determine whether mNumber is 0. The rectangle will be drawn only when mNumber is not 0, if it is 0, it will not be drawn out.

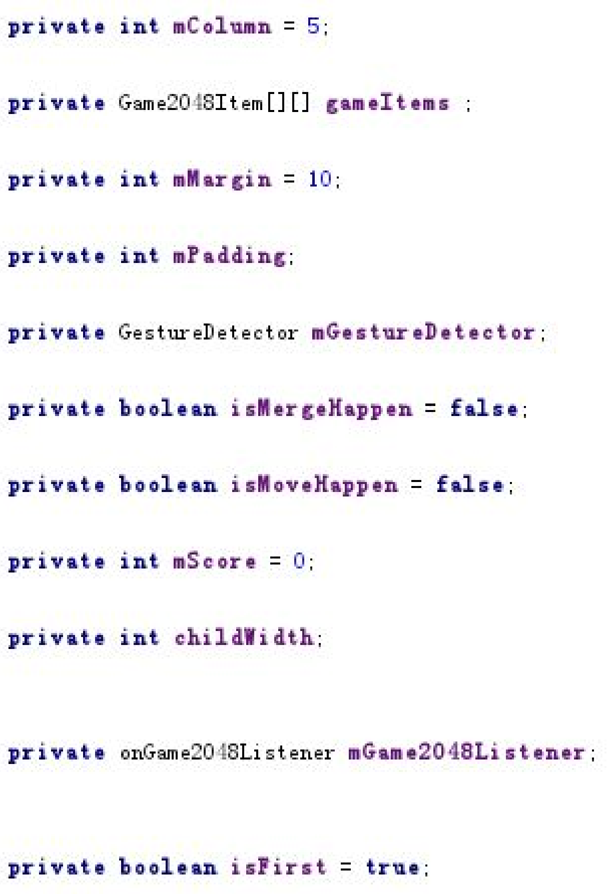


```
            mBgColor = [illegible];
            break;
    }

    paint.setColor(Color.parseColor(mBgColor));


    paint.setStyle(Paint.Style.FILL);

    canvas.drawRect( left: 0,  top: 0, getWidth(), getHeight(), paint);

    if(mNumber != 0){
        drawText(canvas);
    }
}
```

Figure 2: onDrawn method

2.2 Game2048Layout

Game2048Layout is a view to put all chessboard grids. First of all, it needs a attribute to set the number of grids per row. Then we can new correspond number of Game2048Item depends on this variable, and put them in a two-dimensional array. In addition, we need two boolean objects. One is used to make sure if the combination happened, another is used to make sure if the movement happened. Finally, we do corresponding operations rely on the two variables.
Next step is to draw the grid and numbers on the grid during initialisation. Setting the number and width of grids in onMeasure() method at first. Secondly, draw the grids and numbers on grids based on the variables set above in the onLayout() method. And generateNum() method is used to generate number on random position randomly.
After drawing a grid, we need to monitor the grids. According to the user's gestures, to perform the corresponding operation. Although the system provides us with an interface to monitor touch events, namely OnTouchListener, we can use it to monitor some simple operations, such as down, up and so on. But if the gestures we want to monitor are more complex, this interface is not helpful anymore. At this time we can use the GestureDetector class provided by Android, through which we can monitor some complex gestures. Firstly,an enumeration class was prepared to save the corresponding gestute,and then logically judge the user's gesture in the onFling() methon of the GestureDetector class.Then the gesture is passed into the action() method for the specific move merge operation.The specific logic is using for loop on one row/column and get the value of array.By getRowIndexByAction() method and getColIndexByAction() method,based on user's gesture,travel grids to get coordinate on x and y axis of the array which does not equal 0. Then based on these two coordinates to take out the value of array and put it into a temporary array.This array will be compared with previous array,if they are not same ,isMoveHapped will be set to true. Then two adjacent values will be compared, if they are same, they will be merged and isMergeHappen will be set to true. After looping through one row/column, it will enter the next loop and repeat above operation.



```
private int mColumn = 5;

private Game2048Item[][] gameItems ;

private int mMargin = 10;

private int mPadding;

private GestureDetector mGestureDetector;

private boolean isMergeHappen = false;

private boolean isMoveHappen = false;

private int mScore = 0;

private int childWidth;


private onGame2048Listener mGame2048Listener;


private boolean isFirst = true;
```

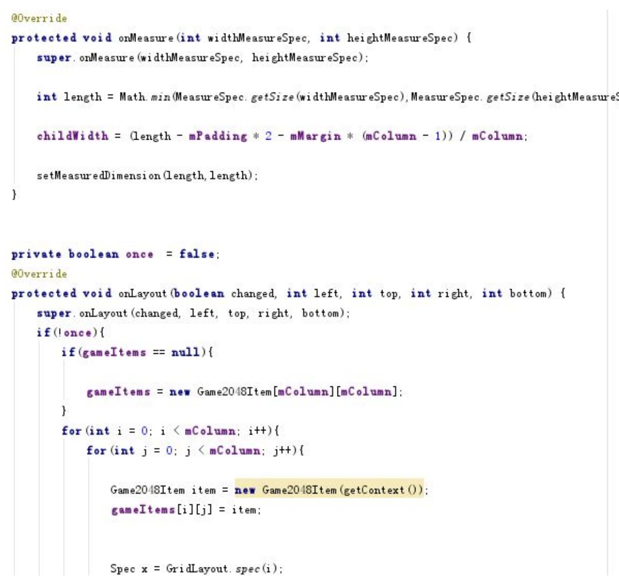
Figure 3: Game2048Layout

```
private boolean ifFull() {
  for(int i = 0; i < mColumn; i++){
      for(int j = 0 ; j < mColumn; j++){
          Game2048Item item = gameItems[i][j];
          if(item.getNumber() == 0){
              return false;
          }
      }
  }
  return true;
}
```

Finally, we need a method to determine if the game is ended. isFull() method is used to check if the chessboard is full and game will be over when the chessboard is full.

3 Conclusion

In conclusion, logic is the most important part of this application. It has two views-Game2048Item and Game2048Item. Game2048Item has an important attribute-number. The onDraw method draws background and display numbers by number attribute. And number travel setNumber for setting. The Game2048Layout is the main class for functions. We stored all items in this class and set some attributes of item. It also implements the monitor function and corresponding operations. For instance, it is able to determine if need to generation a new number. To implement the function of redraw item, we used for loop. In essence,it is a two-layer cycle



```
@Override
protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
    super.onMeasure(widthMeasureSpec, heightMeasureSpec);

    int length = Math.min(MeasureSpec.getSize(widthMeasureSpec),MeasureSpec.getSize(heightMeasureSpec));

    childWidth = (length - mPadding * 2 - mMargin * (mColumn - 1)) / mColumn;

    setMeasuredDimension(length,length);
}


private boolean once  = false;
@Override
protected void onLayout(boolean changed, int left, int top, int right, int bottom) {
    super.onLayout(changed, left, top, right, bottom);
    if(!once){
        if(gameItems == null){

            gameItems = new Game2048Item[mColumn][mColumn];
        }
        for(int i = 0; i < mColumn; i++){
            for(int j = 0; j < mColumn; j++){

                Game2048Item item = new Game2048Item(getContext());
                gameItems[i][j] = item;


                Spec x = GridLayout.spec(i);
```

Figure 4: Generate grids and numbers

with an outer loop code cycle number,and there are three loops in the inner layer. The first loop is used to take out each row with numbers and temporarily store it. The second one is used for looping to determine if a move has occurred. Finally, determine if it has empty grids on chessboard, full means game is over.