横向滑动列表——未来天气

效果图

步骤

1.自定义横向ListView：新建HorizontalListView.java
2.修改weather_info.xml
3.新建未来天气类——FutureWeather.java
4.新建适配器——HorizontalAdapter.java
5.修改parseXML函数与TodayWeather类
6.修改MainActivity.java

一·自定义横向ListView：新建HorizontalListView.java

代码：

```
import android.content.Context;
import android.database.DataSetObserver;
import android.graphics.Rect;
import android.util.AttributeSet;
```

```
import android.view.GestureDetector;
import android.view.GestureDetector.OnGestureListener;
import android.view.MotionEvent;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListAdapter;
import android.widget.Scroller;

import java.util.LinkedList;
import java.util.Queue;

public class HorizontalListView extends AdapterView<ListAdapter> {

    public boolean mAlwaysOverrideTouch = true;
    protected ListAdapter mAdapter;
    private int mLeftViewIndex = -1;
    private int mRightViewIndex = 0;
    protected int mCurrentX;
    protected int mNextX;
    private int mMaxX = Integer.MAX_VALUE;
    private int mDisplayOffset = 0;
    protected Scroller mScroller;
    private GestureDetector mGesture;
    private Queue<View> mRemovedViewQueue = new LinkedList<View>();
    private OnItemSelectedListener mOnItemSelected;
    private OnItemClickListener mOnItemClicked;
    private OnItemLongClickListener mOnItemLongClicked;
    private boolean mDataChanged = false;


    public HorizontalListView(Context context, AttributeSet attrs) {
        super(context, attrs);
        initView();
    }

    private synchronized void initView() {
        mLeftViewIndex = -1;
        mRightViewIndex = 0;
        mDisplayOffset = 0;
        mCurrentX = 0;
        mNextX = 0;
        mMaxX = Integer.MAX_VALUE;
        mScroller = new Scroller(getContext());
        mGesture = new GestureDetector(getContext(), mOnGesture);
    }

    @Override
    public void setOnItemSelectedListener(OnItemSelectedListener listener)
{
        mOnItemSelected = listener;
    }
```

```
55
56      @Override
57      public void setOnItemClickListener(OnItemClickListener listener){
58          mOnItemClicked = listener;
59      }
60
61      @Override
62      public void setOnItemLongClickListener(OnItemLongClickListener
    listener) {
63          mOnItemLongClicked = listener;
64      }
65
66      private DataSetObserver mDataObserver = new DataSetObserver() {
67
68          @Override
69          public void onChanged() {
70              synchronized(HorizontalListView.this){
71                  mDataChanged = true;
72              }
73              invalidate();
74              requestLayout();
75          }
76
77          @Override
78          public void onInvalidated() {
79              reset();
80              invalidate();
81              requestLayout();
82          }
83
84      };
85
86      @Override
87      public ListAdapter getAdapter() {
88          return mAdapter;
89      }
90
91      @Override
92      public View getSelectedView() {
93          //TODO: implement
94          return null;
95      }
96
97      @Override
98      public void setAdapter(ListAdapter adapter) {
99          if(mAdapter != null) {
100             mAdapter.unregisterDataSetObserver(mDataObserver);
101         }
102         mAdapter = adapter;
103         mAdapter.registerDataSetObserver(mDataObserver);
104         reset();
```

```
105     }
106 
107     private synchronized void reset(){
108         initView();
109         removeAllViewsInLayout();
110         requestLayout();
111     }
112 
113     @Override
114     public void setSelection(int position) {
115         //TODO: implement
116     }
117 
118     private void addAndMeasureChild(final View child, int viewPos) {
119         LayoutParams params = child.getLayoutParams();
120         if(params == null) {
121             params = new LayoutParams(LayoutParams.FILL_PARENT,
LayoutParams.FILL_PARENT);
122         }
123 
124         addViewInLayout(child, viewPos, params, true);
125         child.measure(MeasureSpec.makeMeasureSpec(getWidth(),
MeasureSpec.AT_MOST),
126                 MeasureSpec.makeMeasureSpec(getHeight(),
MeasureSpec.AT_MOST));
127     }
128 
129 
130 
131     @Override
132     protected synchronized void onLayout(boolean changed, int left, int
top, int right, int bottom) {
133         super.onLayout(changed, left, top, right, bottom);
134 
135         if(mAdapter == null){
136             return;
137         }
138 
139         if(mDataChanged){
140             int oldCurrentX = mCurrentX;
141             initView();
142             removeAllViewsInLayout();
143             mNextX = oldCurrentX;
144             mDataChanged = false;
145         }
146 
147         if(mScroller.computeScrollOffset()){
148             int scrollx = mScroller.getCurrX();
149             mNextX = scrollx;
150         }
151 
```

```
        if(mNextX <= 0){
            mNextX = 0;
            mScroller.forceFinished(true);
        }
        if(mNextX >= mMaxX) {
            mNextX = mMaxX;
            mScroller.forceFinished(true);
        }

        int dx = mCurrentX - mNextX;

        removeNonVisibleItems(dx);
        fillList(dx);
        positionItems(dx);

        mCurrentX = mNextX;

        if(!mScroller.isFinished()){
            post(new Runnable(){
                @Override
                public void run() {
                    requestLayout();
                }
            });

        }
    }

    private void fillList(final int dx) {
        int edge = 0;
        View child = getChildAt(getChildCount()-1);
        if(child != null) {
            edge = child.getRight();
        }
        fillListRight(edge, dx);

        edge = 0;
        child = getChildAt(0);
        if(child != null) {
            edge = child.getLeft();
        }
        fillListLeft(edge, dx);


    }

    private void fillListRight(int rightEdge, final int dx) {
        while(rightEdge + dx < getWidth() && mRightViewIndex <
mAdapter.getCount()) {

```

```
            View child = mAdapter.getView(mRightViewIndex,
mRemovedViewQueue.poll(), this);
            addAndMeasureChild(child, -1);
            rightEdge += child.getMeasuredWidth();

            if(mRightViewIndex == mAdapter.getCount()-1) {
                mMaxX = mCurrentX + rightEdge - getWidth();
            }

            if (mMaxX < 0) {
                mMaxX = 0;
            }
            mRightViewIndex++;
        }

    }

    private void fillListLeft(int leftEdge, final int dx) {
        while(leftEdge + dx > 0 && mLeftViewIndex >= 0) {
            View child = mAdapter.getView(mLeftViewIndex,
mRemovedViewQueue.poll(), this);
            addAndMeasureChild(child, 0);
            leftEdge -= child.getMeasuredWidth();
            mLeftViewIndex--;
            mDisplayOffset -= child.getMeasuredWidth();
        }
    }

    private void removeNonVisibleItems(final int dx) {
        View child = getChildAt(0);
        while(child != null && child.getRight() + dx <= 0) {
            mDisplayOffset += child.getMeasuredWidth();
            mRemovedViewQueue.offer(child);
            removeViewInLayout(child);
            mLeftViewIndex++;
            child = getChildAt(0);

        }

        child = getChildAt(getChildCount()-1);
        while(child != null && child.getLeft() + dx >= getWidth()) {
            mRemovedViewQueue.offer(child);
            removeViewInLayout(child);
            mRightViewIndex--;
            child = getChildAt(getChildCount()-1);
        }
    }

    private void positionItems(final int dx) {
        if(getChildCount() > 0){
            mDisplayOffset += dx;
```

```
            int left = mDisplayOffset;
            for(int i=0;i<getChildCount();i++){
                View child = getChildAt(i);
                int childWidth = child.getMeasuredWidth();
                child.layout(left, 0, left + childWidth,
child.getMeasuredHeight());
                left += childWidth + child.getPaddingRight();
            }
        }
    }

    public synchronized void scrollTo(int x) {
        mScroller.startScroll(mNextX, 0, x - mNextX, 0);
        requestLayout();
    }

    @Override
    public boolean dispatchTouchEvent(MotionEvent ev) {
        boolean handled = super.dispatchTouchEvent(ev);
        handled |= mGesture.onTouchEvent(ev);
        return handled;
    }

    protected boolean onFling(MotionEvent e1, MotionEvent e2, float
velocityX,
                              float velocityY) {
        synchronized(HorizontalListView.this){
            mScroller.fling(mNextX, 0, (int)-velocityX, 0, 0, mMaxX, 0,
0);
        }
        requestLayout();

        return true;
    }

    protected boolean onDown(MotionEvent e) {
        mScroller.forceFinished(true);
        return true;
    }

    private OnGestureListener mOnGesture = new
GestureDetector.SimpleOnGestureListener() {

        @Override
        public boolean onDown(MotionEvent e) {
            return HorizontalListView.this.onDown(e);
        }

        @Override
        public boolean onFling(MotionEvent e1, MotionEvent e2, float
velocityX,
```

```
                                float velocityY) {
            return HorizontalListView.this.onFling(e1, e2, velocityX,
velocityY);
        }

        @Override
        public boolean onScroll(MotionEvent e1, MotionEvent e2,
                                float distanceX, float distanceY) {

            synchronized(HorizontalListView.this){
                mNextX += (int)distanceX;
            }
            requestLayout();

            return true;
        }

        @Override
        public boolean onSingleTapConfirmed(MotionEvent e) {
            for(int i=0;i<getChildCount();i++){
                View child = getChildAt(i);
                if (isEventWithinView(e, child)) {
                    if(mOnItemClicked != null){

mOnItemClicked.onItemClick(HorizontalListView.this, child, mLeftViewIndex
+ 1 + i, mAdapter.getItemId( mLeftViewIndex + 1 + i ));
                    }
                    if(mOnItemSelected != null){

mOnItemSelected.onItemSelected(HorizontalListView.this, child,
mLeftViewIndex + 1 + i, mAdapter.getItemId( mLeftViewIndex + 1 + i ));
                    }
                    break;
                }

            }
            return true;
        }

        @Override
        public void onLongPress(MotionEvent e) {
            int childCount = getChildCount();
            for (int i = 0; i < childCount; i++) {
                View child = getChildAt(i);
                if (isEventWithinView(e, child)) {
                    if (mOnItemLongClicked != null) {

mOnItemLongClicked.onItemLongClick(HorizontalListView.this, child,
mLeftViewIndex + 1 + i, mAdapter.getItemId(mLeftViewIndex + 1 + i));
                    }
                    break;
```

```
                }

            }
        }

        private boolean isEventWithinView(MotionEvent e, View child) {
            Rect viewRect = new Rect();
            int[] childPosition = new int[2];
            child.getLocationOnScreen(childPosition);
            int left = childPosition[0];
            int right = left + child.getWidth();
            int top = childPosition[1];
            int bottom = top + child.getHeight();
            viewRect.set(left, top, right, bottom);
            return viewRect.contains((int) e.getRawX(), (int) e.getRawY());
        }
    };
}
```

二·修改weather_info.xml

展示图：

代码：

```
<RelativeLayOUT
    android:layout_width="match_parent"
    android:layout_height="140dp"
    android:layout_marginTop="100dp">
    <com.example.mwessj.mlayout.HorizontalListView
        android:id="@+id/future_weather_list"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
</RelativeLayout>
```

三·新建未来天气类——FutureWeather.java

代码：

```
public class FutureWeather {
    private String fengli;
    private String date;
    private String high;
    private String low;
    private String type;
    private int imageID;

    public FutureWeather(String date, String high, String low, String type, String fengli, int imageID){
        this.fengli = fengli;
        this.date = date;
        this.high = high;
        this.low = low;
        this.type = type;
        this.imageID = imageID;
    }

    public String getFengli() {
        return fengli;
    }

    public String getDate() {
        return date;
    }

    public String getHigh() {
        return high;
    }

    public String getLow() {
        return low;
    }

    public String getType() {
        return type;
    }

    public int getImageID() {
        return imageID;
    }
}
```

四·新建适配器——HorizontalAdapter.java

代码：

```
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;

import com.example.mwessj.bean.FutureWeather;
import com.example.mwessj.mwessjweather.R;

import java.util.List;

public class HorizontalAdapter extends BaseAdapter {
    private LayoutInflater mInflater;
    private List<FutureWeather> mDatas;
    public HorizontalAdapter(Context context, List<FutureWeather> datas){
        mInflater = LayoutInflater.from(context);
        mDatas = datas;
    }

    public void  updateListView(List<FutureWeather> datas){
        mDatas = datas;
        notifyDataSetChanged();
    }

    @Override
    public int getCount() {
        return mDatas.size();
    }

    @Override
    public Object getItem(int position) {
        return position;
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent)
{
        ViewHolder holder = null;
        if (convertView == null){
            convertView = mInflater.inflate(R.layout.weather_today,
parent, false);
            holder = new ViewHolder();
```

```
            holder.weatherImgTv = (ImageView)
convertView.findViewById(R.id.weather_img);
            holder.dataTv = (TextView)
convertView.findViewById(R.id.week_today);
            holder.temperatureTv = (TextView)
convertView.findViewById(R.id.temperature);
            holder.climateTv = (TextView)
convertView.findViewById(R.id.climate);
            holder.windTv = (TextView)
convertView.findViewById(R.id.wind);
            convertView.setTag(holder);
        }else {
            holder = (ViewHolder)convertView.getTag();
        }

        FutureWeather futureWeather = mDatas.get(position);
        holder.weatherImgTv.setImageResource(futureWeather.getImageID());
        holder.dataTv.setText(futureWeather.getDate());

holder.temperatureTv.setText(futureWeather.getHigh()+"~"+futureWeather.get
Low());
        holder.climateTv.setText(futureWeather.getType());
        holder.windTv.setText("风力："+futureWeather.getFengli());
        return convertView;
    }

    private class ViewHolder{
        ImageView weatherImgTv;
        TextView dataTv;
        TextView temperatureTv;
        TextView climateTv;
        TextView windTv;
    }
}
```

五.修改parseXML函数与TodayWeather类

首先，修改TodayWeather类，补充下列代码即可：

```
private String date1;
private String high1;
private String low1;
private String type1;
private String fengli1;

private String date2;
private String high2;
private String low2;
private String type2;
private String fengli2;
```

```
12
13  private String date3;
14  private String high3;
15  private String low3;
16  private String type3;
17  private String fengli3;
18
19  private String date4;
20  private String high4;
21  private String low4;
22  private String type4;
23  private String fengli4;
24
25  public String getDate1() {
26      return date1;
27  }
28
29  public String getDate2() {
30      return date2;
31  }
32
33  public String getDate3() {
34      return date3;
35  }
36
37  public String getDate4() {
38      return date4;
39  }
40
41  public String getHigh1() {
42      return high1;
43  }
44
45  public String getHigh2() {
46      return high2;
47  }
48
49  public String getHigh3() {
50      return high3;
51  }
52
53  public String getHigh4() {
54      return high4;
55  }
56
57  public String getLow1() {
58      return low1;
59  }
60
61  public String getLow2() {
62      return low2;
```

```
}

public String getLow3() {
    return low3;
}

public String getLow4() {
    return low4;
}

public String getType1() {
    return type1;
}

public String getType2() {
    return type2;
}

public String getType3() {
    return type3;
}

public String getType4() {
    return type4;
}

public String getFengli1() {
    return fengli1;
}

public String getFengli2() {
    return fengli2;
}

public String getFengli3() {
    return fengli3;
}

public String getFengli4() {
    return fengli4;
}

public void setDate1(String date1) {
    this.date1 = date1;
}

public void setDate2(String date2) {
    this.date2 = date2;
}

public void setDate3(String date3) {
```

```
    this.date3 = date3;
}

public void setDate4(String date4) {
    this.date4 = date4;
}

public void setHigh1(String high1) {
    this.high1 = high1;
}

public void setHigh2(String high2) {
    this.high2 = high2;
}

public void setHigh3(String high3) {
    this.high3 = high3;
}

public void setHigh4(String high4) {
    this.high4 = high4;
}

public void setLow1(String low1) {
    this.low1 = low1;
}

public void setLow2(String low2) {
    this.low2 = low2;
}

public void setLow3(String low3) {
    this.low3 = low3;
}

public void setLow4(String low4) {
    this.low4 = low4;
}

public void setType1(String type1) {
    this.type1 = type1;
}

public void setType2(String type2) {
    this.type2 = type2;
}

public void setType3(String type3) {
    this.type3 = type3;
}

```

```
public void setType4(String type4) {
    this.type4 = type4;
}

public void setFengli1(String fengli1) {
    this.fengli1 = fengli1;
}

public void setFengli2(String fengli2) {
    this.fengli2 = fengli2;
}

public void setFengli3(String fengli3) {
    this.fengli3 = fengli3;
}

public void setFengli4(String fengli4) {
    this.fengli4 = fengli4;
}
```

然后，修改parseXML函数，补充下列代码即可：

```
else if (xmlPullParser.getName().equals("fengli") && fengliCount == 1) {
    eventType = xmlPullParser.next();
    todayWeather.setFengli1(xmlPullParser.getText());
    fengliCount++;
} else if (xmlPullParser.getName().equals("fengli") && fengliCount == 2) {
    eventType = xmlPullParser.next();
    todayWeather.setFengli2(xmlPullParser.getText());
    fengliCount++;
} else if (xmlPullParser.getName().equals("fengli") && fengliCount == 3) {
    eventType = xmlPullParser.next();
    todayWeather.setFengli3(xmlPullParser.getText());
    fengliCount++;
} else if (xmlPullParser.getName().equals("fengli") && fengliCount == 4) {
    eventType = xmlPullParser.next();
    todayWeather.setFengli4(xmlPullParser.getText());
    fengliCount++;
} else if (xmlPullParser.getName().equals("date") && dateCount == 1) {
    eventType = xmlPullParser.next();
    todayWeather.setDate1(xmlPullParser.getText());
    dateCount++;
} else if (xmlPullParser.getName().equals("date") && dateCount == 2) {
    eventType = xmlPullParser.next();
    todayWeather.setDate2(xmlPullParser.getText());
    dateCount++;
} else if (xmlPullParser.getName().equals("date") && dateCount == 3) {
    eventType = xmlPullParser.next();
    todayWeather.setDate3(xmlPullParser.getText());
    dateCount++;
} else if (xmlPullParser.getName().equals("date") && dateCount == 4) {
    eventType = xmlPullParser.next();
```

```
    todayWeather.setDate4(xmlPullParser.getText());
    dateCount++;
} else if (xmlPullParser.getName().equals("high") && highCount == 1) {
    eventType = xmlPullParser.next();
    todayWeather.setHigh1(xmlPullParser.getText().substring(2).trim());
    highCount++;
} else if (xmlPullParser.getName().equals("high") && highCount == 2) {
    eventType = xmlPullParser.next();
    todayWeather.setHigh2(xmlPullParser.getText().substring(2).trim());
    highCount++;
} else if (xmlPullParser.getName().equals("high") && highCount == 3) {
    eventType = xmlPullParser.next();
    todayWeather.setHigh3(xmlPullParser.getText().substring(2).trim());
    highCount++;
} else if (xmlPullParser.getName().equals("high") && highCount == 4) {
    eventType = xmlPullParser.next();
    todayWeather.setHigh4(xmlPullParser.getText().substring(2).trim());
    highCount++;
} else if (xmlPullParser.getName().equals("low") && lowCount == 1) {
    eventType = xmlPullParser.next();
    todayWeather.setLow1(xmlPullParser.getText().substring(2).trim());
    lowCount++;
} else if (xmlPullParser.getName().equals("low") && lowCount == 2) {
    eventType = xmlPullParser.next();
    todayWeather.setLow2(xmlPullParser.getText().substring(2).trim());
    lowCount++;
} else if (xmlPullParser.getName().equals("low") && lowCount == 3) {
    eventType = xmlPullParser.next();
    todayWeather.setLow3(xmlPullParser.getText().substring(2).trim());
    lowCount++;
} else if (xmlPullParser.getName().equals("low") && lowCount == 4) {
    eventType = xmlPullParser.next();
    todayWeather.setLow4(xmlPullParser.getText().substring(2).trim());
    lowCount++;
} else if (xmlPullParser.getName().equals("type") && typeCount == 1) {
    eventType = xmlPullParser.next();
    todayWeather.setType1(xmlPullParser.getText());
    typeCount++;
} else if (xmlPullParser.getName().equals("type") && typeCount == 2) {
    eventType = xmlPullParser.next();
    todayWeather.setType2(xmlPullParser.getText());
    typeCount++;
} else if (xmlPullParser.getName().equals("type") && typeCount == 3) {
    eventType = xmlPullParser.next();
    todayWeather.setType3(xmlPullParser.getText());
    typeCount++;
} else if (xmlPullParser.getName().equals("type") && typeCount == 4) {
    eventType = xmlPullParser.next();
    todayWeather.setType4(xmlPullParser.getText());
    typeCount++;
}
```

六.修改MainActivity.java

变量的定义：

```
private HorizontalListView fw_list;
private List<FutureWeather> fwList;
private HorizontalAdapter horizontalAdapter;
```

在initview函数中补充以下代码：

```
fw_list = (HorizontalListView) findViewById(R.id.future_weather_list);
fwList = new ArrayList<FutureWeather>();
horizontalAdapter = new HorizontalAdapter(MainActivity.this,fwList);
fw_list.setAdapter(horizontalAdapter);
```

添加更新未来天气的相关函数：

```
void updateFutureWeather(TodayWeather todayWeather){
    fwList.clear();

addingData(todayWeather.getDate1(),todayWeather.getHigh1(),todayWeather.ge
tLow1(),todayWeather.getType1(),todayWeather.getFengli1());

addingData(todayWeather.getDate2(),todayWeather.getHigh2(),todayWeather.ge
tLow2(),todayWeather.getType2(),todayWeather.getFengli2());

addingData(todayWeather.getDate3(),todayWeather.getHigh3(),todayWeather.ge
tLow3(),todayWeather.getType3(),todayWeather.getFengli3());

addingData(todayWeather.getDate4(),todayWeather.getHigh4(),todayWeather.ge
tLow4(),todayWeather.getType4(),todayWeather.getFengli4());
    horizontalAdapter.updateListView(fwList);
    Toast.makeText(MainActivity.this,"更新成
功！",Toast.LENGTH_SHORT).show();
}

public void addingData(String data, String high, String low, String type,
String fengli){
    FutureWeather fw =new FutureWeather(data, high, low, type, fengli,
getWeatherImgId(type));
    fwList.add(fw);
    Log.d("myWeather", data+"   "+high+"   "+low+"   "+type+"   "+fengli);
}
```

在线程中添加更新函数即可：

```
private Handler mHandler = new Handler(){
    public void handleMessage(android.os.Message msg){
        switch (msg.what){
            case UPDATE_TODAY_WEATHER:
                updateTodayWeather((TodayWeather) msg.obj);
                updateFutureWeather((TodayWeather) msg.obj); //new add
                break;
```

```
            default:
                break;
        }
    }
};
```