

[登录](#) [注册](#)

欢迎访问萬仟网! (<http://www.10qianwan.com>)



请输入搜索关键字

网站首页 (/)

网页制作

软件设计

IT编程技术

脚本编程

移动编程

网站设计

编程语言

网络编程

当前位置: 萬仟网 (/) > 移动技术 (/list/mobile/) > 移动开发 (/list/mdeve_2/) > Android (/list/android_3/) > Android自定义View实现仿GitHub...

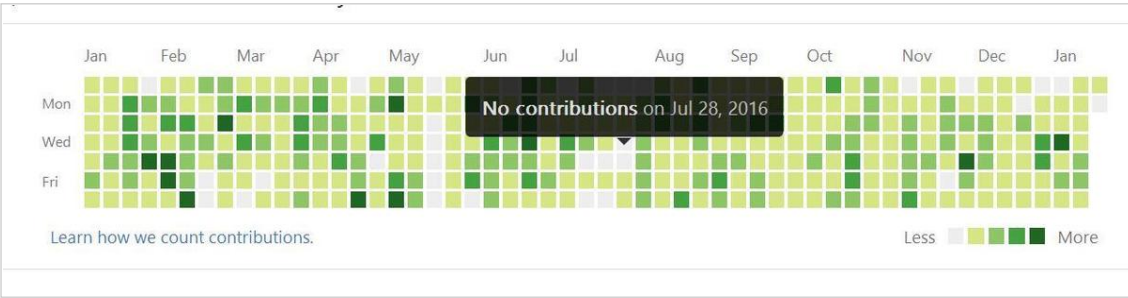
Android自定义View实现仿GitHub的提交活跃表格

2019年07月24日 | 萬仟网移动技术 (/list/mobile/) | [我要评论](#)

说明 本文可能需要一些基础知识点, 如canvas,paint, path, rect等类的基本使用, 建议不熟悉的同学可以学习gcssloop 安卓自定义view教程目录, 会帮助

说明

本文可能需要一些基础知识点, 如canvas,paint, path, rect等类的基本使用, 建议不熟悉的同学可以学习gcssloop 安卓自定义view教程目录, 会帮助很大。



上图就是github的提交表格, 直观来看可以分为几个部分进行绘制:

- (1) 各个月份的小方格子, 并且色彩根据提交次数变化, 由浅到深
- (2) 右下边的颜色标志, 我们右对齐就可以了
- (3) 左边的星期, 原图是从周日画到周六, 我们从周一画到周日
- (4) 上面的月份, 我们只画出1-12月
- (5) 点击时候弹出当天的提交情况, 由一个小三角和圆角矩形组成

需要解决的计算问题:

- (1) 生成任意一年的所有天, 包含年月日周, 提交次数, 色块颜色, 坐标
- (1) 一年中所有的小方格子坐标
- (2) 右下边颜色标志坐标
- (3) 左边星期坐标
- (4) 上面月份坐标
- (5) 点击弹出的提示框和文字坐标

生成某年所有天数

每天的信息我们需要封装成一个类, 代码如下:



嵌入式培训 包装工

双眼皮+开眼角 医美

跨境电商系统 智能

嵌入式培训班 小

嵌入式软件培训 技

最近更新的文章

- Android提高之使
- 【转】基于IC设计
- 关于RxJava的一!
- Android实现简单
- Android开发样式
- ListView展示不同
- Android自定义控
- Android中Java和
- android 实现圆角
- Android 校验em;

推荐阅读



大家感兴趣的文

- 1 Android 组件化
- 2 Android Studio
- 3 【解决】andro
- 4 android10打开
- 5 已解决! PyCh

```

1  /**
2   * created by administrator on 2017/1/13.
3   * 封装每天的属性，方便在绘制的时候进行计算
4   */
5  public class day implements Serializable {
6      /**年**/
7      public int year;
8      /**月**/
9      public int month;
10     /**日**/
11     public int date;
12     /**周几**/
13     public int week;
14     /**贡献次数，默认0**/
15     public int contribution = 0;
16     /**默认颜色，根据提交次数改变**/
17     public int colour = 0xffeeeeee;
18     /**方格坐标，左上点，右下点，确定矩形范围**/
19     public float startx;
20     public float starty;
21     public float endx;
22     public float endy;
23     @Override
24     public String toString() {
25         //这里直接在弹出框中显示
26         return "" + year + "年" + month + "月" + date + "日" + week + ", " + contribution + "次";
27     }
28 }

```

要想先绘制表格，需要计算出所有的天，这里计算一年中所有的天，我们通过从当年1月1日算起，到12月31日，因为星期是连续的，所以我们需要我们提供某年的1月1日是周几，比如2016年1月1日是周五，这里必要的参数是2016和周五，那么我们用一类来实现该方法，代码如下：

```

1  public class datefactory {
2      /**平年map，对应月份和天数**/
3      private static HashMap<Integer,Integer> monthmap = new LinkedHashMap<> (12);
4      /**闰年map，对应月份和天数**/
5      private static HashMap<Integer,Integer> leapmonthmap = new LinkedHashMap<> (12);
6      static {
7          //初始化map，只有2月份不同
8          monthmap.put(1,31); leapmonthmap.put(1,31);
9          monthmap.put(2,28); leapmonthmap.put(2,29);
10         monthmap.put(3,31); leapmonthmap.put(3,31);
11         monthmap.put(4,30); leapmonthmap.put(4,30);
12         monthmap.put(5,31); leapmonthmap.put(5,31);
13         monthmap.put(6,30); leapmonthmap.put(6,30);
14         monthmap.put(7,31); leapmonthmap.put(7,31);
15         monthmap.put(8,31); leapmonthmap.put(8,31);
16         monthmap.put(9,30); leapmonthmap.put(9,30);
17         monthmap.put(10,31); leapmonthmap.put(10,31);
18         monthmap.put(11,30); leapmonthmap.put(11,30);
19         monthmap.put(12,31); leapmonthmap.put(12,31);
20     }
21     /**
22     * 输入年份和1月1日是周几
23     * 闰年为366天，平年为365天
24     * @param year 年份
25     * @param weekday 该年1月1日为周几
26     * @return 该年1月1日到12月31日所有的天数
27     */
28     public static List<day> getdays(int year, int weekday) {
29         List<day> days = new ArrayList<>();
30         boolean isleapyear = isleapyear(year);
31         int daynum = isleapyear ? 366 : 365;
32         day day;
33         int lastweekday = weekday;
34         for (int i = 1; i <= daynum; i++) {
35             day = new day();
36             day.year = year;
37             //计算当天为周几，如果大于7就重置1
38             day.week = lastweekday <= 7 ? lastweekday : 1;
39             //计算当天为几月几号
40             int[] monthandday = getmonthandday(isleapyear, i);
41             day.month = monthandday[0];
42             day.date = monthandday[1];
43             //记录下昨天是周几并+1
44             lastweekday = day.week;
45             lastweekday++;
46             days.add(day);
47         }
48         checkdays(days);
49         return days;
50     }
51     /**
52     * 获取月和日
53     * @param isleapyear 是否闰年
54     * @param currentday 当前天数
55     * @return 包含月和天的数组
56     */
57     public static int[] getmonthandday(boolean isleapyear, int currentday) {

```

6 NanoPi K2 (An

宿舍管理系统

8 Chrome Inspi

LOGO 设计

合 人 工 智 能 物 理



第三方仓储物流



小程序开发

```

58     hashmap<integer, integer> maps = isleapyear?leapmonthmap:monthmap;
59     set<map.entry<integer, integer>> set = maps.entrySet();
60     int count = 0;
61     map.entry<integer, integer> month = null;
62     for (map.entry<integer, integer> entry : set) {
63         count+=entry.getValue();
64         if (currentday<=count){
65             month = entry;
66             break;
67         }
68     }
69     if (month == null){
70         throw new illegalstateexception("未找到所在的月份");
71     }
72     int day = month.getValue()-(count-currentday);
73     return new int[]{month.getKey(), day};
74 }
75 /**
76  * 判断是闰年还是平年
77  * @param year 年份
78  * @return true 为闰年
79  */
80 public static boolean isleapyear(int year) {
81     return year % 4 == 0 && year % 100 != 0 || year % 400 == 0;
82 }
83 /**
84  * 检测生成的天数是否正常
85  * @param days
86  */
87 private static void checkdays(list<day> days) {
88     if (days == null) {
89         throw new illegalargumentexception("天数为空");
90     }
91     if (days.size() != 365 && days.size() != 366) {
92         throw new illegalargumentexception("天数异常:" + days.size());
93     }
94 }
95 public static void main(string[] args){
96     //test
97     list<day> days = datefactory.getdays(2016, 5);
98     for (int i = 0; i < days.size(); i++) {
99         system.out.println(days.get(i).toString());
100     }
101 }
102 }

```

宿舍管理系统

LOGO设计 万年历



第三方仓储物流



小程序开发

具体的计算逻辑可以看看代码，不是很难，这样我们就能得到某年的所有天。

绘制天数格子

因为该view比较长，所以需要横屏显示，方便起见，这里我们也不再进行view的测量计算，也不再进行自定义属性，只关注其核心逻辑即可。

首先我们需要将需要的成员变量定义出来：

```

1  /**灰色方格的默认颜色*/
2  private final static int default_box_colour = 0xffeeeeee;
3  /**提交次数颜色值*/
4  private final static int[] colour_level =
5      new int[]{0xff1e6823, 0xff44a340, 0xff8cc665, 0xffd6e685, default_box_colour};
6  /**星期*/
7  private string[] weeks = new string[]{"mon", "wed", "fri", "sun"};
8  /**月份*/
9  private string[] months =
10     new string[]{"jan", "feb", "mar", "apr", "may", "jun", "jul", "aug", "sep", "oct", "nov", "dec"};
11  /**默认的padding,绘制的时候不贴边画*/
12  private int padding = 24;
13  /**小方格的默认边长*/
14  private int boxside = 8;
15  /**小方格间的默认间隔*/
16  private int boxinterval = 2;
17  /**所有周的列数*/
18  private int column = 0;
19  private list<day> mdays;//一年中所有的天
20  private paint boxpaint;//方格画笔
21  private paint textpaint;//文字画笔
22  private paint infopaint;//弹出框画笔
23  private paint.fontmetrics metrics;//测量文字
24  private float downx;//按下的点的x坐标
25  private float downy;//按下的点的y坐标
26  private day clickday;//按下所对应的天

```

这些提取的变量是慢慢增加的，在自定义的时候一下想不全的时候可以先写，等用到某些变量的时候就提取出来。

然后我们初始化一下数据：

```

1 public githubcontributionview(context context, attributeset attrs, int defstyleattr)
2     super(context, attrs, defstyleattr);
3     initview();
4 }
5 public void initview() {
6     mdays = datefactory.getdays(2016, 5);
7     //方格画笔
8     boxpaint = new paint();
9     boxpaint.setStyle(Paint.Style.FILL);
10    boxpaint.setStrokeWidth(2);
11    boxpaint.setColor(default_box_colour);
12    boxpaint.setAntiAlias(true);
13    //文字画笔
14    textpaint = new paint();
15    textpaint.setStyle(Paint.Style.FILL);
16    textpaint.setColor(Color.GRAY);
17    textpaint.setTextSize(12);
18    textpaint.setAntiAlias(true);
19    //弹出的方格信息画笔
20    infopaint = new paint();
21    infopaint.setStyle(Paint.Style.FILL);
22    infopaint.setColor(0xcc888888);
23    infopaint.setTextSize(12);
24    infopaint.setAntiAlias(true);
25    //将默认值转换px
26    padding = ui.dp2px(getContext(), padding);
27    boxside = ui.dp2px(getContext(), boxside);
28    metrics = textpaint.getFontMetrics();
29 }

```

宿舍管理系统

LOGO设计 万年历



第三方仓储物流



小程序开发

这里我们以2016年来举例，mdays就是获取2016年的所有天的集合（参数可以当作自定义属性提取出来），相关的paint也已经初始化好了，接下来就需要在onDraw方法里画，先画所有的方格子和月份标志：

```

1 /**
2  * 画出1-12月方格小块和上面的月份
3  * @param canvas 画布
4  */
5 private void drawbox(Canvas canvas) {
6     //方格的左上右下坐标
7     float startx, starty, endx, endy;
8     //起始月份为1月
9     int month = 1;
10    for (int i = 0; i < mdays.size(); i++) {
11        day day = mdays.get(i);
12        if (i == 0) {
13            //画1月的文本标记,坐标应该是x=padding,y=padding-boxside/2(间隙),y坐标在表格上面一点
14            canvas.drawText(months[0], padding, padding-boxside/2, textpaint);
15        }
16        if (day.week == 1 && i != 0) {
17            //如果当天是周1,那么说明增加了一列
18            column++;
19            //如果列首的月份有变化,那么说明需要画月份
20            if (day.month > month) {
21                month = day.month;
22                //月份文本的坐标计算,x坐标在变化,而y坐标都是一样的,boxside/2(间隙)
23                canvas.drawText(months[month-1], padding+column*(boxside+boxinterval), padding-boxside/2, textpaint);
24            }
25        }
26        //计算方格坐标点,x坐标随列数的增多而增加,y坐标随行数的增多而变化
27        startx = padding + column * (boxside + boxinterval);
28        starty = padding + (day.week - 1) * (boxside + boxinterval);
29        endx = startx + boxside;
30        endy = starty + boxside;
31        //将该方格的坐标保存下来,这样可以在点击方格的时候计算弹框的坐标
32        day.startx = startx;
33        day.starty = starty;
34        day.endx = endx;
35        day.endy = endy;
36        //给画笔设置当前天的颜色
37        boxpaint.setColor(day.colour);
38        canvas.drawRect(startx, starty, endx, endy, boxpaint);
39    }
40    boxpaint.setColor(default_box_colour); //恢复默认颜色
41 }

```

这里主要是注意下行数列数的变化和月份坐标的计算，格子画好了。

绘制星期文本

我们再画左边的星期文本：

```

1  /**
2   * 画左侧的星期
3   * @param canvas 画布
4   */
5  private void drawweek(canvas canvas) {
6      //文字是左对齐,所以找出最长的字
7      float textlength = 0;
8      for (string week : weeks) {
9          float templength = textpaint.measuretext(week);
10         if (textlength < templength) {
11             textlength = templength;
12         }
13     }
14     //依次画出星期文本,坐标点x=padding-文本长度-文本和方格的间隙,y坐标随行数变化
15     canvas.drawText(weeks[0], padding - textlength - 2, padding + boxside - metrics.des
16     canvas.drawText(weeks[1], padding - textlength - 2, padding + 3 * (boxside + boxint
17     canvas.drawText(weeks[2], padding - textlength - 2, padding + 5 * (boxside + boxint
18     canvas.drawText(weeks[3], padding - textlength - 2, padding + 7 * (boxside + boxint
19 }

```

宿舍管理系统

LOGO设计 万年



第三方仓储物流



小程序开发

绘制颜色深浅标志

然后根据表格的高度再画出右下边的颜色深浅标志:

```

1  /**
2   * 画出右下角的颜色深浅标志, 因为是右对齐的所以需要从右往左画
3   * @param canvas 画布
4   */
5  private void drawtag(canvas canvas) {
6      //首先计算出两个文本的长度
7      float morelength = textpaint.measuretext("more");
8      float lesslength = textpaint.measuretext("less");
9      //画 more 文本,x坐标=padding+ (列数+1) * (方格边长+方格间隙) - 一个方格间隙-文本长度
10     float morex = padding + (column + 1) * (boxside + boxinterval) - boxinterval - more
11     //y坐标=padding+ (方格行数+1,和表格底部有些距离) * (方格边长+方格间隙)+字体的ascent高度
12     float morey = padding + 8 * (boxside + boxinterval) + math.abs(metrics.ascent);
13     canvas.drawText("more", morex, morey, textpaint);
14     //画深浅色块,坐标根据上面的more依次计算就可以了
15     float interval = boxside - 2; //文字和色块间的距离
16     float leftx = morex - interval - boxside;
17     float topy = morey - boxside;
18     float rightx = morex - interval;
19     float bottomy = morey; //色块的y坐标是一样的
20     for (int i = 0; i < colour_level.length; i++) {
21         boxpaint.setcolor(colour_level[i]);
22         canvas.drawRect(leftx - i * (boxside + boxinterval), topy, rightx - i * (boxside +
23     }
24     //最后画 less 文本,原理同上
25     canvas.drawText("less", leftx - 4 * (boxside + boxinterval) - interval - lesslength
26 }

```

这样整个表格主体绘制完成。

处理点击事件

接下来要处理点击事件,判断点击的坐标如果在方格内,那么弹出对于的文本框,先处理点击事件:

```

1  @Override
2  public boolean onTouchEvent(MotionEvent event) {
3      //获取action_down的坐标,用来判断点在哪天,并弹出
4      if (motionEvent.action_down == event.getAction()) {
5          downx = event.getX();
6          downy = event.getY();
7          findclickbox();
8      }
9      //这里因为我们只是记录坐标点,不对事件进行拦截所以默认返回
10     return super.onTouchEvent(event);
11 }

```

判断是否在方格内:

```

1  /**
2   * 判断是否点击在方格内
3   */
4  private void findclickbox() {
5      for (day day : mdays) {
6          //检测点击的坐标如果在方格内，则弹出信息提示
7          if (downx >= day.startx && downx <= day.endx && downy >= day.starty && downy <= day
8              clickday = day; //纪录点击的哪天
9              break;
10         }
11     }
12     //点击完要刷新，这样每次点击不同的方格，弹窗就可以在相应的位置显示
13     refreshview();
14 }
15 /**
16  * 点击弹出文字提示
17  */
18 private void refreshview() {
19     invalidate();
20 }

```

绘制弹出文本框

然后看看弹出文本框的绘制：

```

1  /**
2   * 画方格上的文字弹框
3   * @param canvas 画布
4   */
5  private void drawpopupinfo(canvas canvas) {
6      if (clickday != null) { //点击的天不为null时候才画
7          //先根据方格来画出一个三角形，坐标就是方格的中间
8          path infopath = new path();
9          //先从方格中心
10         infopath.moveTo(clickday.startx + boxside / 2, clickday.starty + boxside / 2);
11         //然后是方格的左上点
12         infopath.lineTo(clickday.startx, clickday.starty);
13         //然后是方格的右上点
14         infopath.lineTo(clickday.endx, clickday.starty);
15         //画出三角
16         canvas.drawpath(infopath, infopaint);
17         //画三角上的圆角矩形
18         textpaint.setColor(color.white);
19         //得到当天的文本信息
20         string popupinfo = clickday.toString();
21         system.out.println(popupinfo);
22         //计算文本的高度和长度用以确定矩形的大小
23         float infoheight = metrics.descent - metrics.ascent;
24         float infolength = textpaint.measureText(popupinfo);
25         log.e("height", infoheight + "");
26         log.e("length", infolength + "");
27         //矩形左上点应该是x=当前天的x+边长/2- (文本长度/2+文本和框的间隙)
28         float leftx = (clickday.startx + boxside / 2) - (infolength / 2 + boxside);
29         //矩形左上点应该是y=当前天的y+边长/2- (文本高度+上下文本和框的间隙)
30         float topy = clickday.starty - (infoheight + 2 * boxside);
31         //矩形的右下点应该是x=leftx+文本长度+文字两边和矩形的间距
32         float rightx = leftx + infolength + 2 * boxside;
33         //矩形的右下点应该是y=当前天的y
34         float bottomy = clickday.starty;
35         system.out.println("x=" + leftx + "y=" + topy + "x=" + rightx + "y=" + bottomy);
36         rectf rectf = new rectf(leftx, topy, rightx, bottomy);
37         canvas.drawRoundRect(rectf, 4, 4, infopaint);
38         //绘制文字，x=leftx+文字和矩形间距，y=topy+文字和矩形上面间距+文字顶部到基线高度
39         canvas.drawText(popupinfo, leftx + boxside, topy + boxside + math.abs(metrics.ascent), text
40         clickday = null; //重新置空，保证点击方格外信息消失
41         textpaint.setColor(color.gray); //恢复画笔颜色
42     }
43 }

```

这样主体逻辑完成，但需要开放设置某天提交次数的方法：

宿舍管理系统

LOGO设计 万年



第三方仓储物流



小程序开发


```

1  /**
2   * 设置某天的次数
3   * @param year 年
4   * @param month 月
5   * @param day 日
6   * @param contribution 次数
7   */
8  public void setdata(int year,int month,int day,int contribution){
9      //先找到是第几天,为了方便不做参数检测了
10     for (day d : mdays) {
11         if (d.year == year && d.month == month && d.date == day){
12             d.contribution = contribution;
13             d.colour = getcolour(contribution);
14             break;
15         }
16     }
17     refreshview();
18 }
19 /**
20  * 根据提交次数来获取颜色值
21  * @param contribution 提交的次数
22  * @return 颜色值
23  */
24 private int getcolour(int contribution){
25     int colour = 0;
26     if (contribution <= 0){
27         colour = colour_level[4];
28     }
29     if (contribution == 1){
30         colour = colour_level[3];
31     }
32     if (contribution == 2){
33         colour = colour_level[2];
34     }
35     if (contribution == 3){
36         colour = colour_level[1];
37     }
38     if (contribution >= 4){
39         colour = colour_level[0];
40     }
41     return colour;
42 }

```

好了,所有逻辑完成,主要涉及到一些计算,完整代码:

```

1  /**
2   * created by administrator on 2017/1/13.
3   * 仿github的提交活跃表
4   * 横屏使用
5   */
6  public class githubcontributionview extends view {
7      /**灰色方格的默认颜色**/
8      private final static int default_box_colour = 0xffeeeeee;
9      /**提交次数颜色值**/
10     private final static int[] colour_level =
11         new int[]{0xff1e6823, 0xff44a340, 0xff8cc665, 0xffd6e685, default_box_colour};
12     /**星期**/
13     private string[] weeks = new string[]{"mon", "wed", "fri", "sun"};
14     /**月份**/
15     private string[] months =
16         new string[]{"jan", "feb", "mar", "apr", "may", "jun", "jul", "aug", "sep", "oct", "nov"}
17     /**默认的padding,绘制的时候不贴边画**/
18     private int padding = 24;
19     /**小方格的默认边长**/
20     private int boxside = 8;
21     /**小方格间的默认间隔**/
22     private int boxinterval = 2;
23     /**所有周的列数**/
24     private int column = 0;
25     private list<day> mdays;//一年中所有的天
26     private paint boxpaint;//方格画笔
27     private paint textpaint;//文字画笔
28     private paint infopaint;//弹出框画笔
29     private paint.fontmetrics metrics;//测量文字
30     private float downx;//按下的点的x坐标
31     private float downy;//按下的点的y坐标
32     private day clickday;//按下所对应的天
33     public githubcontributionview(context context) {
34         this(context, null);
35     }
36     public githubcontributionview(context context, attributeset attrs) {
37         this(context, attrs, 0);
38     }
39     public githubcontributionview(context context, attributeset attrs, int defstyleattr
40     super(context, attrs, defstyleattr);
41     initview();
42 }
43 public void initview() {
44     mdays = datefactory.getdays(2016, 5);
45     //方格画笔
46     boxpaint = new paint();

```

宿舍管理系统

LOGO设计 万年历



第三方仓储物流



小程序开发



第三方仓储物流



小程序开发

```

47 boxpaint.setStyle(Paint.Style.FILL);
48 boxpaint.setStrokeWidth(2);
49 boxpaint.setColor(default_box_colour);
50 boxpaint.setAntiAlias(true);
51 //文字画笔
52 textpaint = new Paint();
53 textpaint.setStyle(Paint.Style.FILL);
54 textpaint.setColor(Color.GRAY);
55 textpaint.setTextSize(12);
56 textpaint.setAntiAlias(true);
57 //弹出的方格信息画笔
58 infopaint = new Paint();
59 infopaint.setStyle(Paint.Style.FILL);
60 infopaint.setColor(0xcc888888);
61 infopaint.setTextSize(12);
62 infopaint.setAntiAlias(true);
63 //将默认值转换px
64 padding = ui.dp2px(getContext(), padding);
65 boxside = ui.dp2px(getContext(), boxside);
66 metrics = textpaint.getFontMetrics();
67 }
68 @Override
69 protected void onSizeChanged(int w, int h, int oldw, int oldh) {
70     super.onSizeChanged(w, h, oldw, oldh);
71 }
72 @Override
73 protected void onDraw(Canvas canvas) {
74     super.onDraw(canvas);
75     column = 0;
76     canvas.save();
77     drawBox(canvas);
78     drawWeek(canvas);
79     drawTag(canvas);
80     drawPopUpInfo(canvas);
81     canvas.restore();
82 }
83 /**
84  * 画出1-12月方格小块和上面的月份
85  * @param canvas 画布
86  */
87 private void drawBox(Canvas canvas) {
88     //方格的左上右下坐标
89     float startX, startY, endX, endY;
90     //起始月份为1月
91     int month = 1;
92     for (int i = 0; i < mdays.size(); i++) {
93         day day = mdays.get(i);
94         if (i == 0) {
95             //画1月的文本标记, 坐标应该是x=padding, y=padding-boxside/2 (间隙), y坐标在表格上面一点
96             canvas.drawText(months[0], padding, padding-boxside/2, textpaint);
97         }
98         if (day.week == 1 && i != 0) {
99             //如果当天是周1, 那么说明增加了一列
100             column++;
101             //如果列首的月份有变化, 那么说明需要画月份
102             if (day.month > month) {
103                 month = day.month;
104                 //月份文本的坐标计算, x坐标在变化, 而y坐标都是一样的, boxside/2 (间隙)
105                 canvas.drawText(months[month-1], padding+column*(boxside+boxinterval), padding-bo
106             }
107         }
108         //计算方格坐标点, x坐标一致随列数的增多而增加, y坐标随行数的增多而变化
109         startX = padding + column * (boxside + boxinterval);
110         startY = padding + (day.week - 1) * (boxside + boxinterval);
111         endX = startX + boxside;
112         endY = startY + boxside;
113         //将该方格的坐标保存下来, 这样可以在点击方格的时候计算弹框的坐标
114         day.startX = startX;
115         day.startY = startY;
116         day.endX = endX;
117         day.endY = endY;
118         //给画笔设置当前天的颜色
119         boxpaint.setColor(day.colour);
120         canvas.drawRect(startX, startY, endX, endY, boxpaint);
121     }
122     boxpaint.setColor(default_box_colour); //恢复默认颜色
123 }
124 /**
125  * 画左侧的星期
126  * @param canvas 画布
127  */
128 private void drawWeek(Canvas canvas) {
129     //文字是左对齐, 所以找出最长的字
130     float textlength = 0;
131     for (String week : weeks) {
132         float templength = textpaint.measureText(week);
133         if (textlength < templength) {
134             textlength = templength;
135         }
136     }
137     //依次画出星期文本, 坐标点x=padding-文本长度-文本和方格的间隙, y坐标随行数变化
138     canvas.drawText(weeks[0], padding - textlength - 2, padding + boxside - metrics.de

```



```

139 canvas.drawText(weeks[1], padding - textlength - 2, padding + 3 * (boxside + boxin
140 canvas.drawText(weeks[2], padding - textlength - 2, padding + 5 * (boxside + boxin
141 canvas.drawText(weeks[3], padding - textlength - 2, padding + 7 * (boxside + boxin
142 }
143 /**
144  * 画出右下角的颜色深浅标志, 因为是右对齐的所以需要从右往左画
145  * @param canvas 画布
146  */
147 private void drawtag(canvas canvas) {
148     //首先计算出两个文本的长度
149     float morelength = textpaint.measuretext("more");
150     float lesslength = textpaint.measuretext("less");
151     //画 more 文本,x坐标=padding+ (列数+1) * (方格边长+方格间隙) - 一个方格间隙-文本长度
152     float morex = padding + (column + 1) * (boxside + boxinterval) - boxinterval - mor
153     //y坐标=padding+ (方格行数+1,和表格底部有些距离) * (方格边长+方格间隙) +字体的ascent高度
154     float morey = padding + 8 * (boxside + boxinterval) + math.abs(metrics.ascent);
155     canvas.drawText("more", morex, morey, textpaint);
156     //画深浅色块,坐标根据上面的more依次计算就可以了
157     float interval = boxside - 2; //文字和色块间的距离
158     float leftx = morex - interval - boxside;
159     float topy = morey - boxside;
160     float rightx = morex - interval;
161     float bottomy = morey; //色块的y坐标是一样的
162     for (int i = 0; i < colour_level.length; i++) {
163         boxpaint.setcolor(colour_level[i]);
164         canvas.drawRect(leftx - i * (boxside + boxinterval), topy, rightx - i * (boxside
165     }
166     //最后画 less 文本,原理同上
167     canvas.drawText("less", leftx - 4 * (boxside + boxinterval) - interval - lesslengt
168 }
169 @Override
170 public boolean onTouchEvent(MotionEvent event) {
171     //获取点击时候的坐标, 用来判断点在哪天, 并弹出
172     if (motionEvent.getAction() == MotionEvent.ACTION_DOWN) {
173         downx = event.getX();
174         downy = event.getY();
175         findclickbox();
176     }
177     return super.onTouchEvent(event);
178 }
179 /**
180  * 判断是否点击在方格内
181  */
182 private void findclickbox() {
183     for (day day : mdays) {
184         //检测点击的坐标如果在方格内, 则弹出信息提示
185         if (downx >= day.startx && downx <= day.endx && downy >= day.starty && downy <= d
186             clickday = day; //纪录点击的哪天
187         break;
188     }
189 }
190 //点击完要刷新, 这样每次点击不同的方格, 弹窗就可以在相应的位置显示
191 refreshview();
192 }
193 /**
194  * 点击弹出文字提示
195  */
196 private void refreshview() {
197     invalidate();
198 }
199 /**
200  * 画方格上的文字弹框
201  * @param canvas 画布
202  */
203 private void drawpopupinfo(canvas canvas) {
204     if (clickday != null) {
205         //先根据方格来画出一个三角形, 坐标就是方格的中间
206         Path infopath = new Path();
207         //先从方格中心
208         infopath.moveTo(clickday.startx + boxside / 2, clickday.starty + boxside / 2);
209         //然后是方格的左上点
210         infopath.lineTo(clickday.startx, clickday.starty);
211         //然后是方格的右上点
212         infopath.lineTo(clickday.endx, clickday.starty);
213         //画出三角
214         canvas.drawPath(infopath, infopaint);
215         //画三角上的圆角矩形
216         textpaint.setColor(color.white);
217         //得到当天的文本信息
218         String popupinfo = clickday.toString();
219         System.out.println(popupinfo);
220         //计算文本的高度和长度用以确定矩形的大小
221         float infoheight = metrics.descent - metrics.ascent;
222         float infolength = textpaint.measureText(popupinfo);
223         Log.e("height", infoheight + "");
224         Log.e("length", infolength + "");
225         //矩形左上点应该是x=当前天的x+边长/2- (文本长度/2+文本和框的间隙)
226         float leftx = (clickday.startx + boxside / 2) - (infolength / 2 + boxside);
227         //矩形左上点应该是y=当前天的y+边长/2- (文本高度+上下文本和框的间隙)
228         float topy = clickday.starty - (infoheight + 2 * boxside);
229         //矩形的右下点应该是x=leftx+文本长度+文字两边和矩形的间距
230         float rightx = leftx + infolength + 2 * boxside;

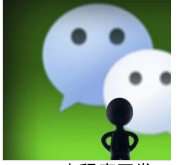
```

宿舍管理系统

LOGO设计 万年



第三方仓储物流



小程序开发

```

231 //矩形的右下点应该是y=当前天的y
232 float bottomy = clickday.starty;
233 system.out.println(""+leftx+"/"+topy+"/"+rightx+"/"+bottomy);
234 rectf rectf = new rectf(leftx, topy, rightx, bottomy);
235 canvas.drawroundrect(rectf,4,4,infopaint);
236 //绘制文字,x=leftx+文字和矩形间距,y=topy+文字和矩形上面间距+文字顶到基线高度
237 canvas.drawText(popupinfo,leftx+boxside,topy+boxside+math.abs(metrics.ascent),tex
238 clickday = null;//重新置空,保证点击方格外信息消失
239 textpaint.setColor(color.gray);//恢复画笔颜色
240 }
241 }
242 /**
243  * 设置某天的次数
244  * @param year 年
245  * @param month 月
246  * @param day 日
247  * @param contribution 次数
248  */
249 public void setdata(int year,int month,int day,int contribution){
250 //先找到是第几天,为了方便不做参数检测了
251 for (day d : mdays) {
252     if (d.year == year && d.month == month && d.date == day){
253         d.contribution = contribution;
254         d.colour = getcolour(contribution);
255         break;
256     }
257 }
258 refreshview();
259 }
260 /**
261  * 根据提交次数来获取颜色值
262  * @param contribution 提交的次数
263  * @return 颜色值
264  */
265 private int getcolour(int contribution){
266     int colour = 0;
267     if (contribution <= 0){
268         colour = colour_level[4];
269     }
270     if (contribution == 1){
271         colour = colour_level[3];
272     }
273     if (contribution == 2){
274         colour = colour_level[2];
275     }
276     if (contribution == 3){
277         colour = colour_level[1];
278     }
279     if (contribution >= 4){
280         colour = colour_level[0];
281     }
282     return colour;
283 }
284 }

```

宿舍管理系统

LOGO设计 万年历



第三方仓储物流



小程序开发

这样弄个布局测试下:

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout
3      xmlns:android="http://schemas.android.com/apk/res/android" (http://schemas.android.co
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:gravity="center"
7      android:orientation="vertical"
8      >
9      <com.franky.custom.view.githubcontributionview
10         android:id="@+id/cc_chart"
11         android:layout_width="match_parent"
12         android:layout_height="match_parent"
13     />
14 </LinearLayout>

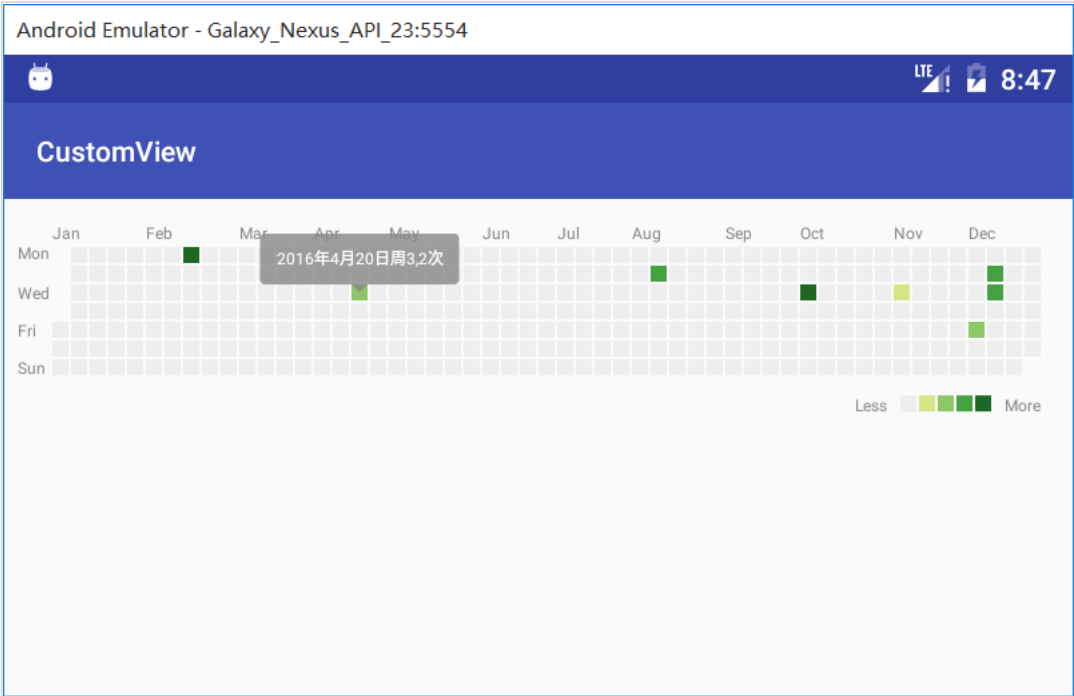
```

随机弄些数据:

```
1 public class MainActivity extends AppCompatActivity {
2     @Override
3     protected void onCreate(Bundle savedInstanceState) {
4         super.onCreate(savedInstanceState);
5         setContentView(R.layout.activity_main);
6         GithubContributionView github = (GithubContributionView) findViewById(R.id.cc_chart);
7         github.setData(2016, 12, 9, 2);
8         github.setData(2016, 10, 5, 10);
9         github.setData(2016, 8, 9, 3);
10        github.setData(2016, 4, 20, 2);
11        github.setData(2016, 12, 13, 3);
12        github.setData(2016, 12, 14, 3);
13        github.setData(2016, 2, 15, 4);
14    }
15 }
16 }
```

效果

gif没有录好，看看图片效果：



[查看源码](#)

以上所述是小编给大家介绍的android自定义view实现仿github的提交活跃表格，希望对大家有所帮助

高端LOGO设计与VI设计，快看这里



版权声明：本文内容由互联网用户贡献，该文观点仅代表作者本人。本站仅提供信息存储服务，不拥有所有权，不承担相关法律责任。如发现本站有涉嫌抄袭侵权/违法违规的内容， 请发送邮件至 2386932994@qq.com 举报，一经查实将立刻删除。



人事管理系统



设计logo



公司拓展活动



it外包

相关文章：

宿舍管理系统

LOGO设计 万年历



第三方仓储物流



小程序开发



做一个小程序大概要多少钱

小程序开发要多少钱

2.1万阅读

AndroidStudio利用android-support-multidex解决64k的各种异常 (/articledetail/490436.html)

64k的各种异常 当你的应用程序和库引用达到一定规模，你遇到构建错误显示你的应用已经达到了一个android应用程序构建体系结构的...
[阅读全文] (/articledetail/490436.html)

ListView实现聊天列表之处理不同数据项 (/articledetail/485853.html)

通常我们用惯的listview每一项的布局都是相同的，只是控件所绑定的数据不同。但单单只是如此...
[阅读全文] (/articledetail/485853.html)

Android解析XML的三种方式SAX、Pull、Dom (/articledetail/491459.html)

在android开发中，经常用到去解析xml文件，常见的解析xml的方式有以下三种：sax、pull、dom解析方式。最近做了一个a...
[阅读全文] (/articledetail/491459.html)



做一个小程序大概要多少钱

小程序开发要多少钱

2.1万阅读

Android签名机制介绍：生成keystore、签名、查看签名信息等方法 (/articledetail/493538.html)

android独有的安全机制，除了权限机制外，另外一个就是签名机制了。签名机制主要用在以下两个主要场合起到其作用：升级app和权限检...
[阅读全文] (/articledetail/493538.html)

Android MVP在项目中的用法！！！ (/articledetail/651055.html)

网上有很多MVP的理解，很多都是偏理论的，我这篇主要讲的是针对于项目构建的实操！gogogo！你知道MVP吗？M...
[阅读全文] (/articledetail/651055.html)

Android getObject与optJSONObject的区别结合源码分析 (/articledetail/488506.html)

android getObject与optJSONObject的区别结合源码分析 json解析常见问题: getjson...
[阅读全文] (/articledetail/488506.html)



什么是ERP系统

生产管理 erp

3739阅读

Android保存图片到相册 (/articledetail/636832.html)

Android保存图片到相册前言随着Android系统的不断升级 从最初的第一个版本 更新到现在 Android...
[阅读全文] (/articledetail/636832.html)

Android开发之基于DialogFragment创建对话框的方法示例 (/articledetail/486639.html)

本文实例讲述了android基于dialogfragment创建对话框的方法。分享给大家供大家参考，具体如下：/** ...
[阅读全文] (/articledetail/486639.html)

Android星级评分条的实现代码 (/articledetail/486195.html)

ratingbar星级评分条 它跟拖白条类似。都允许用户拖动来改变进度，不同的是，星级评分条通过星星图案表示进度。想淘宝，等等...
[阅读全文] (/articledetail/486195.html)

宿舍管理系统

LOGO设计 万年历



第三方仓储物流



小程序开发



详谈Android ListView的选择模式 (/articledetail/487822.html)

效果图：listview 定义了choicemode属性，描述是这样的：用于为视图定义选择行为。默认情况下，列表时没有... [阅读全文] (/articledetail/487822.html)

宿舍管理系统

LOGO设计 万年


第三方仓储物流


小程序开发



做一个小程序大概要多少钱

小程序开发要多少钱

2.1万阅读

高端LOGO设计与VI设计，快看这里



网友评论



您的评论内容...

验证码：



发表评论