

可视化工作——第二部分

1.任务描述

1.1 数据简介

大连海事大学中国欧盟可用性研究中心(Sino European Usability Center)制作的可以安装在移动设备上的用户体验数据采集工具 CAUX(content-aware user experience research), 这个工具采集移动设备的运行状况(手机的电量、亮度、加速度、耳机接口、网络状况等), app 使用情况(打开, 退出, 退回后台的时间等), 通过日志法记录采集到的数据并录入数据库。

这部分的工作目标是围绕师姐的研究课题——场景还原来进行可视化, 根据师姐的描述, 在移动设备上收集到的数据最终将会根据规律进行场景分类, 分成比如游戏场景、购物场景、聊天场景等, 对于。由于场景分类数据处理工作在当时还未完成, 故在 JavaScript 代码中手动分类模拟了“2018-03-23”这一天的场景时间。

1.2 可视化任务

单个用户在一天内场景分布情况的可视化

2.工作概览

由于数据处理工作当时尚未完成, 这部分工作可以分为数据模拟模块、数据处理和可视化模块。

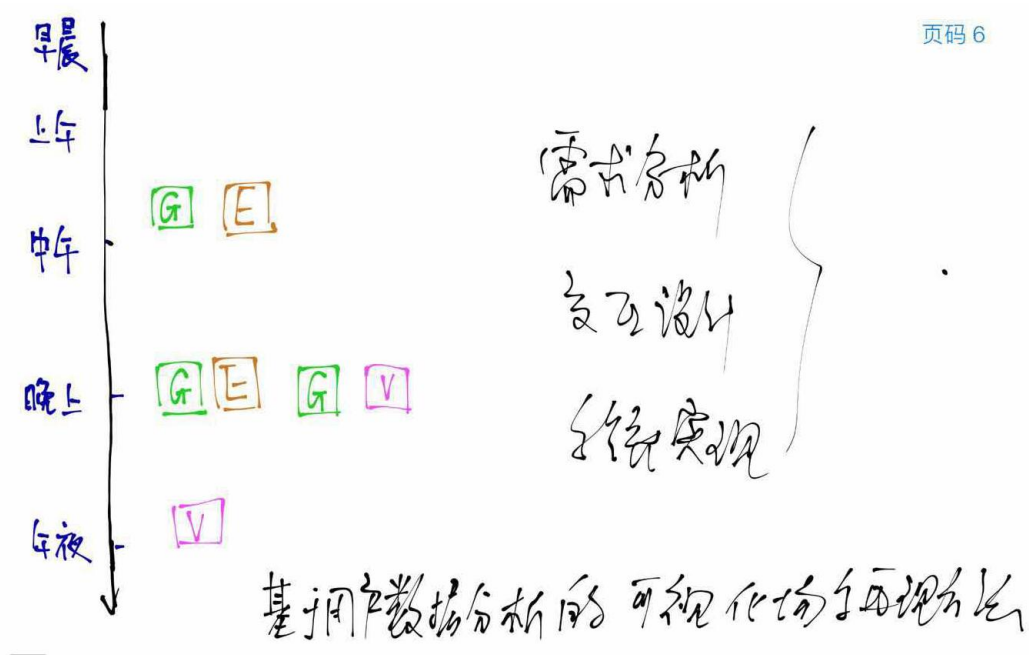
数据模拟模块即在 JavaScript 代码中根据模拟场景分类后的数据(包括场景名称, 场景开始结束时间, 场景内 app 使用及备注), 代替了第一部分工作的数据存储和传输模块。

数据处理模块即 JavaScript 中对于场景时间数据格式的转换、场景持续时间统计、场景总时间计算的代码段、场景时间段(4 小时一段/6 小时一段)的计算。

可视化模块即 JavaScript 中使用 echarts.js 的代码段，这部分工作参考调用了 echarts.js 官方示例中自定义图的代码。

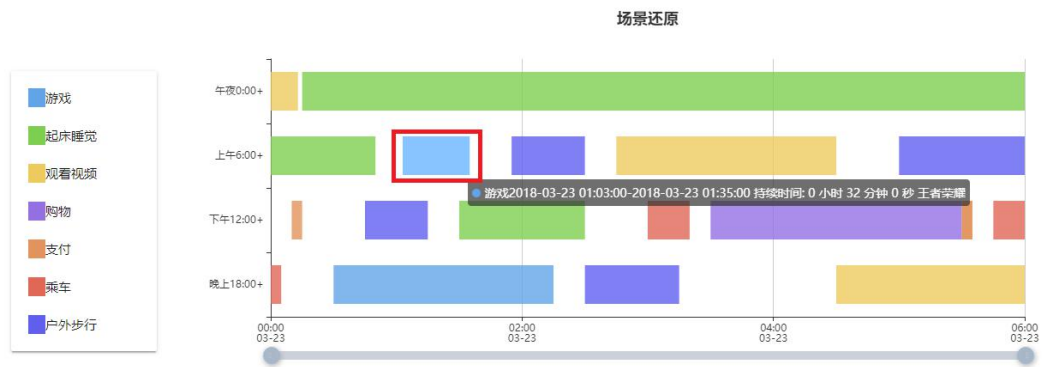
3.可视化方法和具体实现

对于这部分任务，师姐给了我一个设计草稿(如图所示)。这使我意识到可以使用第一部分工作的 gantt 图进行改动，以实现这部分工作的可视化。

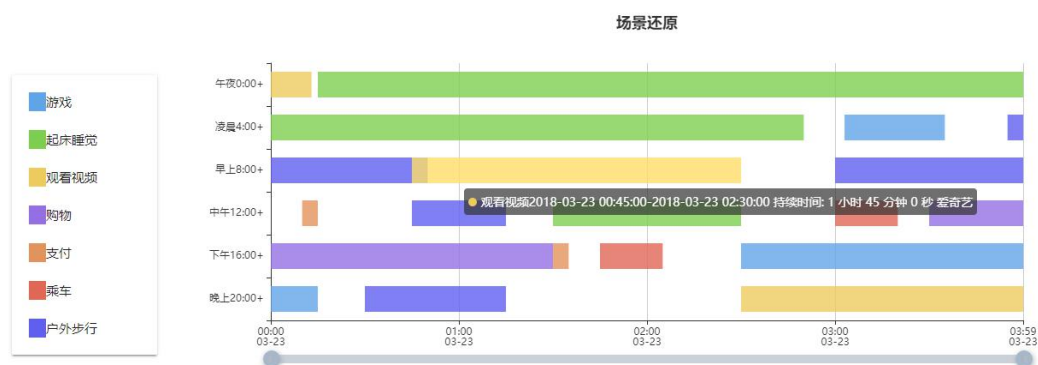


页码 6

如图，将“2018-03-23”当天天分为了午夜，上午，下午，晚上4个时间段。y轴表示所在时间(4小时一段/6小时一段)，x轴表示时间段里的需要增加的时间，以6小时一段的可视化图表为例，光标指向红色方框的柱状形时，提示框代表“游戏场景从上午9:03(8:00+01:03)开始，上午9:35(8:00+01:35)结束，持续时间32分钟，场景中运行的手机app为王者荣耀”。根据图表左侧的图例可以对照找到相应的场景。为了防止因持续时间过短导致柱状过细使得研究人员错漏数据的情况发生，在下方设置数据区域缩放，研究人员可以通过鼠标滚轮缩放，通过拖动缩放条调整查看的时间区间。



为了给研究人员提供多种选择，制作了 4 小时一段的可视化图表版本，在“早上 8:00+”的时间段中，模拟了场景交叉进行的效果。



在图例的颜色选择上，参考了蚂蚁金服可视化团队的视觉设计指引，参考链接

<https://www.yuque.com/mo-college/vis-design/ugbofr>

4. 工作遇到的困难

模拟数据部分，由于暂时没有场景分类的数据，模拟的数据要尽可能地接近真实数据，即通过 JSON 文件传输到 html 页面的字符串格式，这导致了时间数据仍要进行格式转换。

数据处理部分，需要对每个场景时间减去 y 轴对应的时间段(0,6,12,18 小时)，使其能够显示在正确的时间段中。

5. 相关代码段

5.1 将时间戳转换为 x 小时 x 分钟 x 秒

```
function formatDuring(mss) {
    var days = parseInt(mss / (1000 * 60 * 60 * 24));
    var hours = parseInt((mss % (1000 * 60 * 60 * 24)) / (1000 * 60 * 60));
    var minutes = parseInt((mss % (1000 * 60 * 60)) / (1000 * 60));
    var seconds = (mss % (1000 * 60)) / 1000;
    return hours + " 小时 " + minutes + " 分钟 " + seconds + " 秒 ";
}
```

5.2 获取 datetime 格式数据中的小时数

```
function gethour(testarr){
    var getdate = echarts.format.formatTime('yyyy-MM-dd', testarr);
    var dateplustime = getdate + ' 00:00:00';//输出为 2018/01/01 00:00:00
    var gettime = Date.parse(testarr) - Date.parse(dateplustime);
    var gethours = parseInt((gettime % (1000 * 60 * 60 * 24)) / (1000 * 60 * 60));
    return gethours;
}
```

5.3 根据场景时间段减去 0,6,12,18 小时

```
function typing(firstdatetime){
    var hour = gethour(firstdatetime);
    var typevalue = [];
    if (hour >= 0 && hour < 6){
        typevalue.push(3, echarts.format.formatTime('yyyy-MM-dd hh:mm:ss', Date.parse(firstdatetime)));
        //返回数据中 value 的前三项(index 场景时间段,场景的第一个时间,场景的第二个时间)
        return typevalue;
    }
    else if (hour >= 6 && hour < 12){
        typevalue.push(2, echarts.format.formatTime('yyyy-MM-dd hh:mm:ss', Date.parse(firstdatetime) - 1000*60*60*6));
        //返回数据中 value 的前三项(index 场景时间段,场景的第一个时间,场景的第二个时间)
        return typevalue;
    }
    else if (hour >= 12 && hour < 18){
        typevalue.push(1, echarts.format.formatTime('yyyy-MM-dd hh:mm:ss', Date.parse(firstdatetime) - 1000*60*60*12));
        //返回数据中 value 的前三项(index 场景时间段,场景的第一个时间,场景的第二个时间)
        return typevalue;
    }
    else if (hour >= 18 && hour < 24){
```

```

        typevalue.push(0, echarts.format.formatTime('yyyy-MM-dd hh:mm:ss', Date.parse(firstdatetime) -
1000*60*60*18));
        //返回数据中 value 的前三项(index 场景时间段,场景的第一个时间,场景的第二个时间)
        return typevalue;
    }
}

```

5.4 模拟数据格式

```

{
    name: '游戏',
    value: [
        typing("2018-03-23 07:03:00")[0],
        // index:3 是午夜,index:0 是晚上,index:1 是下午,index:2 是上午
        typing("2018-03-23 07:03:00")[1],//baseTime
        typing("2018-03-23 07:35:00")[1],//"2018-03-23 01:35:00",//baseTime += duration,
        formatDuring(Date.parse("2018-03-23 07:35:00")-Date.parse("2018-03-23 07:03:00")) + "王者荣耀",
    ],
    itemStyle: {
        normal: {
            color: findName('游戏')
        }
    }
},

```