**中国地质大学（武汉）物理实验教学中心实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 姓名 | 叶正楠 | 班号 | 07D228 | 学 号 | 20221001555 |
| 日期 | 20220226 | 指导老师 | 杨勇 | 成绩 |  |
| 课程名称 | 大学物理实验A | | |
| 实验项目 | 安卓手机传感器导出软件 | | | | |
| 友情提示：1.实验报告务必按时按要求提交；2.请勿旷课。 | | | | | |
| # 实验目的  大多数 Android 设备都有内置传感器，用来测量运动、屏幕方向和各种环境条件。这些传感器能够提供高度精确的原始数据，非常适合用来监测设备的三维移动或定位，或监测设备周围环境的变化。市场中也有成型的软件将这些传感器的数据导出但大都由国外团队开发且未开源，本实验开发了一款基于开源项目Autox.js的安卓设备传感器数据记录软件，可以供在arm64-v8a，armeabi-v7a，x86，x86\_64等设备平台下运行。  # 实验器材  ## 开发环境  IDE：vscode（主要使用插件https://marketplace.visualstudio.com/items?itemName=aaroncheng.auto-js-vsce-fixed）  ADB版本：Android Debug Bridge version 1.0.41Version 28.0.2-debian  系统：Ubuntu 22.04.2 LTS  ## 编译环境  编译设备：Google Pixel 6，Android 13  编译软件：Autox.js（https://github.com/kkevsekk1/AutoX/compare/6.3.7...6.3.7）  ## 技术栈  ES6，[Rhino 1.7.13](https://github.com/mozilla/rhino)，auto.js，autox.js  # 实验原理及内容  1. 通过调用autox.js提供的api调用android提供的传感器api注册传感器对象，同时检查传感器是否可用  2. 监听传感器参数改变的事件，在事件响应时将值存入临时变量  3. 基于javascript的eventLoop机制，使用serInterval函数在指定时间间隔内将临时变量的值组合符合csv文件标准的字符串，存入文件输入流中  # 主要实验步骤  1. 编写程序，编译安装包  2. 在手机上安装软件  3. 设计实验测试软件性能  4. 进行实验，得出数据  5. 分析结果  # 实验数据  由于实验数据有4MB，一万多项数据，所以请在线查看项目文件中的data.csv  代码见文件末尾  > 这个项目比较小所以使用单个.js文件包含代码和布局部分，样式全用默认  测试实验：将手机平移和旋转，分析角加速度，线性方向加速度，重力加速度得出运动路径，计算总位移是否为0  > 本来我想把实验过程录下来，但最终设计的路径是从宿舍楼走一圈，过程比较长，就不录了  # 数据处理及实验结果  由于测出的物理量大都是加速度，已知初速度为0，所以可以直接把 带成然后硬算算出速度，然后得出位移，旋转角度；  而手机的正方向是固定的，也就是说人握着手机旋转，手机传感器测量的角度会改变但加速度的方向不会改变，也就是说我们需要根据角度修正加速度的值。  在实验过程中，我手持手机在宿舍绕一圈，记录频率选10ms，手机x方向始终为移动方向且相对身体保持不动。计算数据取加速度传感器的x方向加速度和方向传感器的方位角，总的代码如下  % 整理时间间隔矩阵  deT = ([data.time; 0] - [0; data.time])./1000.^2;  time = data.time./1000;  deT(end,:) = [];  % 设备坐标系位移  total\_x = sum(abs(data.xms2).\*deT);  x = cumsum(data.xms2.\*deT);  % 角度修正坐标系位移  x\_fix = x.\*(cos((data.xAng./180).\*pi));  plot(time, x, time, x\_fix, time, -cumsum(data.xms3.\*deT));  d=legend("原始位移","根据角度修正的位移");    由上图结合实验路径可发现，传感器在运动幅度过大，比如转身时表现相当差，具体分析见误差分析。  ## 存在的问题和误差来源分析  \* 我先前在宿舍内进行了多次测试，结果来看数据相当不好，我认为需要适当提升数据记录的频率，在这一版软件中我写的是能有多少传感器就记录多少数据，这样相当耗费设备资源，假如后续开发应该让用户自主选择需要保存哪些数据。  \* 我最开始设计的csv文件表头其实不利于导入matlab进行数据处理，还是应该用全英文。  \* 让人拿着手机移动感觉并不太好，应该设计一个小车，在确定长度的场地拖着手机移动，这样得出来的数据会更加精准，但这种小车并不算随处可见的实验器材。  \* 有一个问题我在阅读Android开发文档后也没有得到准确的答案，为什么加速度传感器会有负值？方向不应该只由方向传感器表达吗？  \* 从图像分析，在微小尺度内，原始位移的图像和修正位移的图像并不重合，这也是因为人手持手机的问题，在y轴方向和z轴方向上手机角度并不为0，这会导致部分加速度被分在其他两个方向上了。  \* 似乎，每次测量之后需要重新打开软件，但这明显是上线环节的事，本软件在开发环境可以就行了，因为这个是物理实验不是编程大作业。 | | | | | |

1. "ui";
2. ui.layout(
3. <vertical w="\*" h="\*">
4. <text textSize="40sp" textStyle="bold" textColor="#000000">Sensorself</text>
5. <text textColor="#000000">你的下一款传感器又何必是传感器</text>
6. <text> </text>
7. <text margin="10 0">注意！距离传感器只有两个值：0/5，0表示前置摄像头附近有物体，5表示反之情况</text>
8. <text margin="10 0">注意！大部分设备没有环境温度和环境湿度传感器</text>
9. <text> </text>
10. <text textColor="#000000">经检测,您的设备缺失以下传感器</text>
11. <list id="mistake\_sensor\_list\_display">
12. <horizontal>
13. <text h="20sp" textColor="#000000" id="context" text="{{context}}"></text>
14. </horizontal>
15. </list>
16. <text> </text>
17. <button w="\*" id="no\_shutdown" style="Widget.AppCompat.Button.Colored" text="开始记录" />
18. <button w="\*" id="shutdown" style="Widget.AppCompat.Button.Colored" text="结束记录" />
19. <text> </text>
20. <text textSize="20sp" textColor="#000000">数据文件存放路径</text>
21. <input inputType="textLongMessage" w="\*" id="path" text="/storage/emulated/0/Documents/data.csv" />
22. <button w="\*" id="path\_change" text="确定" />
23. <text> </text>
24. <text textSize="20sp" textColor="#000000">数据记录间隔</text>
25. <text>不建议改太小,可能会影响时间帧记录,简单观测下来,实际使用过程中可能会有11ms左右的误差</text>
26. <input inputType="number" w="\*" id="interval\_time\_input" text="500" />
27. <button w="\*" id="interval\_time\_change" text="确定" />
28. <text> </text>
29. <text>本软件用于提交物理创新实验作业, 大概率不会有后续版本, 点击下方按钮跳转项目地址</text>
30. <button w="\*" id="turn\_github" text="跳转" />
31. </vertical >
32. );
33. const key = ["accelerometer", "orientation", "gyroscope", "magnetic\_field", "gravity", "linear\_acceleration", "ambient\_temperature", "light", "pressure", "proximity", "relative\_humidity"], sensor\_des = ["加速度传感器", "方向传感器", "陀螺仪传感器", "磁场传感器", "重力传感器", "线性加速度传感器", "环境温度传感器", "光线传感器", "压力传感器", "距离传感器", "湿度传感器"], sensor\_arg = ["x轴加速度m/s^2,y轴加速度m/s^2,z轴加速度m/s^2,", "方位角/度,绕x轴旋转的角度/度,绕y轴顺时针旋转的角度/度,", "x轴角速度P/rad/s,y轴角速度/rad/s,z轴角速度/rad/s,", "x轴磁场强度/uT,y轴磁场强度/uT,z轴磁场强度/uT,", "x轴重力加速度m/s^2,y轴重力加速度m/s^2,z轴重力加速度m/s^2,", "x轴线性加速度m/s^2,y轴线性加速度m/s^2,z轴线性加速度m/s^2,", "环境温度/摄氏度,", "环境光强度/lux,", "大气压/hPa,", "0/5,", "相对湿度/%,"];
34. var sensor\_arr = [];
35. let mistake\_sensor = new Set(), mistake\_msg = "", csv\_path = "/storage/emulated/0/Documents/data.csv", interval\_time = 500;
36. var csv\_temp\_msg = [""];
37. key.forEach((element, index) => {
38. sensor\_arr.push(sensors.register(element, sensors.delay.game));
39. if (!sensor\_arr[index]) mistake\_sensor.add(index);
40. });
41. toastLog(mistake\_sensor);
42. var mistake\_sensor\_list = [];
43. mistake\_sensor.forEach(element => {
44. mistake\_sensor\_list.push({
45. "context": `➰${sensor\_des[element]}不可用\n`
46. });
47. });
48. ui.mistake\_sensor\_list\_display.setDataSource(mistake\_sensor\_list);
49. const begin\_scanf = function () {
50. sensor\_arr.forEach((element, index) => {
51. element ? element.on("change", (event, a, b, c) => {
52. csv\_temp\_msg[index] = "";
53. if (!(a === NaN)) csv\_temp\_msg[index] += util.format("%d,", a) || util.format("0,");
54. if (b) csv\_temp\_msg[index] += util.format("%d,", b);
55. if (c) csv\_temp\_msg[index] += util.format("%d,", c);
56. }) : "";
57. });
58. }
59. var file = open(csv\_path, "w");
60. begin\_scanf();
61. const begin\_write\_data = function (interval\_time) {
62. let csv\_heard = "时间,";
63. toastLog("开始记录数据 " + new Date());
64. var now = Date.now();
65. sensor\_arg.forEach((element, index) => {
66. if (!mistake\_sensor.has(index)) csv\_heard += element;
67. });
68. file.writeline(csv\_heard);
69. return setInterval(() => {
70. let msg = (Date.now() - now) + ",";
71. for (let index in csv\_temp\_msg)
72. if (!mistake\_sensor.has(index))
73. msg += csv\_temp\_msg[index];
74. file.writeline(msg);
75. console.log("写入数据\n" + csv\_temp\_msg);
76. }, interval\_time);
77. }
78. var writer;
79. ui.no\_shutdown.click(() => {
80. begin\_scanf();
81. file = open(csv\_path, "w");
82. writer = begin\_write\_data(interval\_time);
83. });
84. ui.shutdown.click(() => {
85. clearInterval(writer);
86. file.close();
87. sensors.unregisterAll();
88. });
89. ui.path\_change.click(() => {
90. csv\_path = ui.path.getText();
91. console.log(csv\_path);
92. });
93. ui.interval\_time\_change.click(() => {
94. interval\_time = ui.interval\_time\_input.getText();
95. console.log(interval\_time);
96. });
97. ui.turn\_github.click(() => app.openUrl("https://github.com/nan-mu/sensorself"));