**scratch新增block说明**

1. 界面新增电子模块

JS文件路径/scratch-gui-develop/node\_modules/scratch-blocks/dist/vertical.js，该文件是对源码文件的打包压缩生成后的文件（源码文件路径/scratch-gui-develop/node\_modules/scratch-blocks/blocks- vertical），修改后立即生效；根据查看package.json文件，打包压缩执行的是”python build.py && webpack”，我执行python脚本build.py进行压缩时报错” Exception in thread Thread-3:”，可能是python环境没装好，可以先修改vertical.js即可。

1. 修改Blockly.Blocks.defaultToolbox，定义的xml格式，其中：category用于解析成菜单项Motion、Looks…，block用于解析成模块，具体格式如下

|  |
| --- |
| <category name="Motion" colour="#4C97FF" secondaryColour="#3373CC">  <block type="motion\_movesteps" id="motion\_movesteps">  <value name="STEPS">  <shadow type="math\_number">  <field name="NUM">10</field>  </shadow>  </value>  </block>  </category> |

1. Xml中的id字段对应需要新增对象，用于负责传递数据给相应的处理函数及模块上的文字显示，对象定义如下

|  |
| --- |
| Blockly.Blocks.motion = {};  Blockly.Blocks.motion\_movesteps = {  init: function() {  this.jsonInit({  message0: "move %1 steps",  args0: [{  type: "input\_value",  name: "STEPS"  }],  category: Blockly.Categories.motion,  extensions: ["colours\_motion", "shape\_statement"]  })  }  }; |

1. 修改Blockly.Colours，新增模块颜色；
2. 修改Blockly.Categories，新增模块分类；
3. 修改Blockly.ScratchBlocks.VerticalExtensions.registerAll，大致作用是把前面的定义进行解析；
4. 电子模块增加对应的处理函数

JS文件路径/scratch-gui-develop/node\_modules/scratch-vm/dist/node/scratch-vm.js，该文件是对源码文件的打包压缩生成后的文件（源码文件路径/ /scratch-gui-develop/node\_modules/scratch-vm/src/blocks/），修改后立即生效；根据查看package.json文件，打包压缩执行的是” "build": "webpack --progress --colors --bail"”，即在当前目录下执行npm run build就可以把源码打包生成scratch-vm.js文件。

1. 新增模块处理函数，可以参照文件scratch3\_motion.js进行编写，修改getPrimitives ()函数时注意对应id和处理函数；
2. 修改scratch-vm\src\engine\runtime.js中的defaultBlockPackages，增加自己模块scratch3\_emode: require('../blocks/scratch3\_emode')，该步骤用于打包新增文件，若无新增文件，可直接修改scratch-vm.js即可（修改立即生效）；
3. /scratch-gui-develop/node\_modules/scratch-vm目录下执行npm run build即可重新生成scratch-vm.js；
4. 前后台通信交互

Server代码\scratch-gui-develop\node\_modules\webpack-dev-server\lib\Server.js，若是修改该文件，需要重新启动server才能生效：npm start；前后端的交互是通过ajax发送请求和接收数据的（注：scrachPi是利用XMLHttpRequest和server进行交互的，和ajax类似），下面以motion模块的移动多少步为例：

1. scratch-vm.js的moveSteps函数是motion模块的移动多少步对应的处理函数，在其中增加以下处理：

|  |
| --- |
| var url = "http://127.0.0.1:8601/webpack-dev-server";  $.ajax({  url: url,  dataType: "json"  }).done(function (data) {  console.log("ajax test done");  }); |

备注：ajax的使用需要引入jquery，形式类似于const $ = require('../jquery/index');

1. server端接收到webpack-dev-server请求后，可以调用相应的硬件进行处理：

|  |
| --- |
| app.get('/webpack-dev-server', (req, res) => {  //相应的硬件操作部分  }); |

附：scrachPi的电子模块的前后端通信代码：

前端处理\ScratchPi\package.nw\ScratchPi\magicbeans\threads\_ext.js

后端处理\ScratchPi\nodejs\_http\bt-exboard-http-service.js