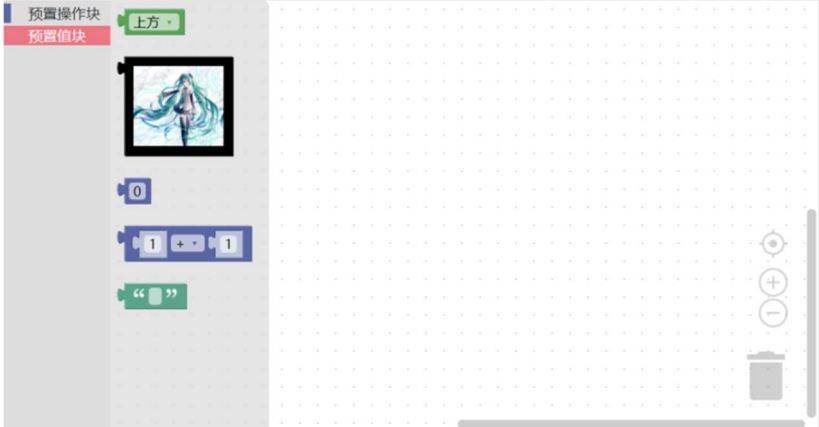


Blockly 开发实战之——吃蘑菇案例

采蘑菇catch the mushroom:

Blockly



Python Codes:

click 'Run' to eat the mushroom



KeyLock Run

任务 1:

要求一：利用 **js** 外部引用的相关知识点,引入对应的自定义 **Blockly** 文件,其文件为 **block_s** 文件夹下的 **block1_definitions.js** 和 **block1_code.js**

要求二：利用 **workspace** 和 **toolbox** 的相关知识点，设计 **toolbox**

要求三：利用代码生成相关知识，设计一个代码输出框

要求四：利用 **canvas** 相关知识，设立一个动画演示框

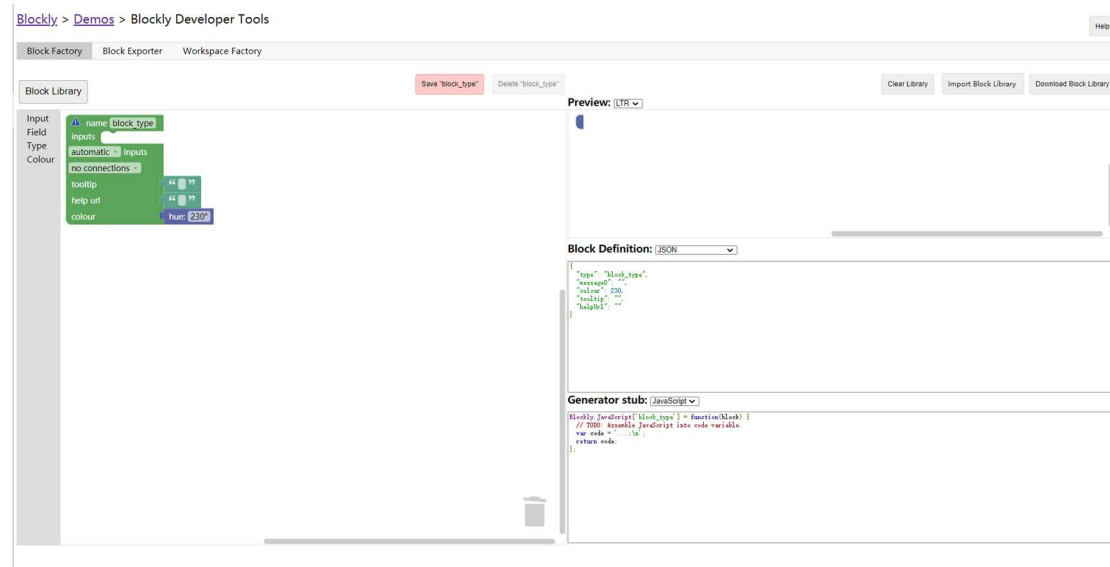
任务 2:

要求一：利用 **blockly** 的块相关知识，设计 **blockly** 块。并且在刚才的网页中，嵌入所需要的 4 个块。

1. 分析所需要的块，分别有选择英雄的 `set_hero`, `block_hero`, `block_move`, `block_direction`
2. 打开开发工具 `blockly-master\demos\blockfactory\index.html`

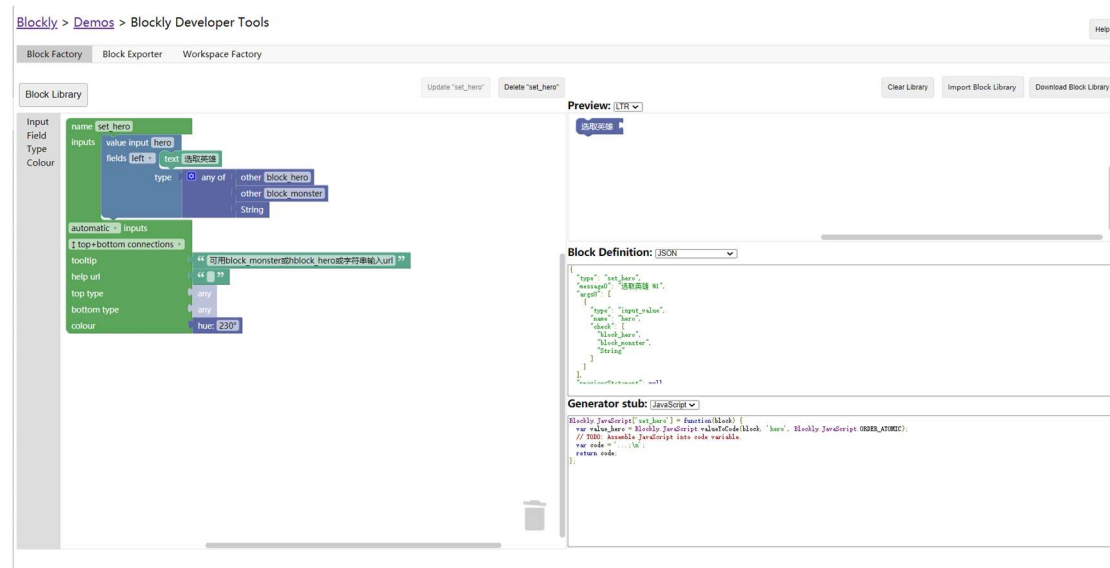
页面介绍：

- Preview：预览区，可以看到自定义块的预览。
- Block Definition：block 的定义，描述块的形状、功能、颜色等，用于在 blockly 中添加自定义块。我们使用的 Json 方式的定义，添加方法见后文。
- Generator stub：js 形式，其中的代码作用是：为自定义块添加 JavaScript 代码和 python 代码。
- Block Library：做好的块可以在这里查看和再次更新。



按照如下的设计进行开发

set_hero：设置英雄形象



block_hero: 描述英雄的块

Blockly > Demos > Blockly Developer Tools

Block Factory Block Exporter Workspace Factory

Block Library Update "block_hero" Delete "block_hero" Clear Library Import Block Library Download Block Library

Input Field Type Colour

name: block_hero

inputs: dummy input

fields: left: Image https://s6.bdstatic.com/94c1d7d_bAaC787mm9GUKT... width: 100 height: 100 alt text: Rip RTL

automatic: inputs

left output: 44 31

tooltip: 44 31

help url: other: block_hero

output type: 44 31

colour: hue: 267

Preview: LTR

Block Definition: JSON

```
{
  "type": "block_hero",
  "message0": "英雄",
  "args0": [
    {
      "type": "field_image",
      "src": "https://s6.bdstatic.com/94c1d7d_bAaC787mm9GUKT...",
      "width": 100,
      "height": 100,
      "alt": "英雄",
      "rtl": false
    }
  ],
  "output": "block_hero",
  "x": 0,
  "y": 0
}
```

Generator stub: JavaScript

```
Blockly.JavaScript["block_hero"] = function(block) {
  // TODO: Assemble JavaScript into code variable
  var code =
    // TODO: Change ORDER_HERE to the correct strength
    return code;
  Blockly.JavaScript.ORDER_HERE;
};
```

block_move: 移动英雄的块, 包含两个 inline 的 input

Blockly > Demos > Blockly Developer Tools

Block Factory Block Exporter Workspace Factory

Block Library Update "block_move" Delete "block_move" Clear Library Import Block Library Download Block Library

Input Field Type Colour

name: block_move

inputs: value input direction

fields: left: text 10 type: any of Number other: block_direction

value input steps

fields: left: text 10 type: Number

dummy input

fields: left: text 10

inline: inputs

top+bottom connections: 44 31 第一个参数: 方向 (1,2,3,4或数字或自定义方向块); 第二个参数: 步数, 每步为横竖一个身长

tooltip: 44 31

help url: any

top type: any

bottom type: any

colour: hue: 3150

Preview: LTR

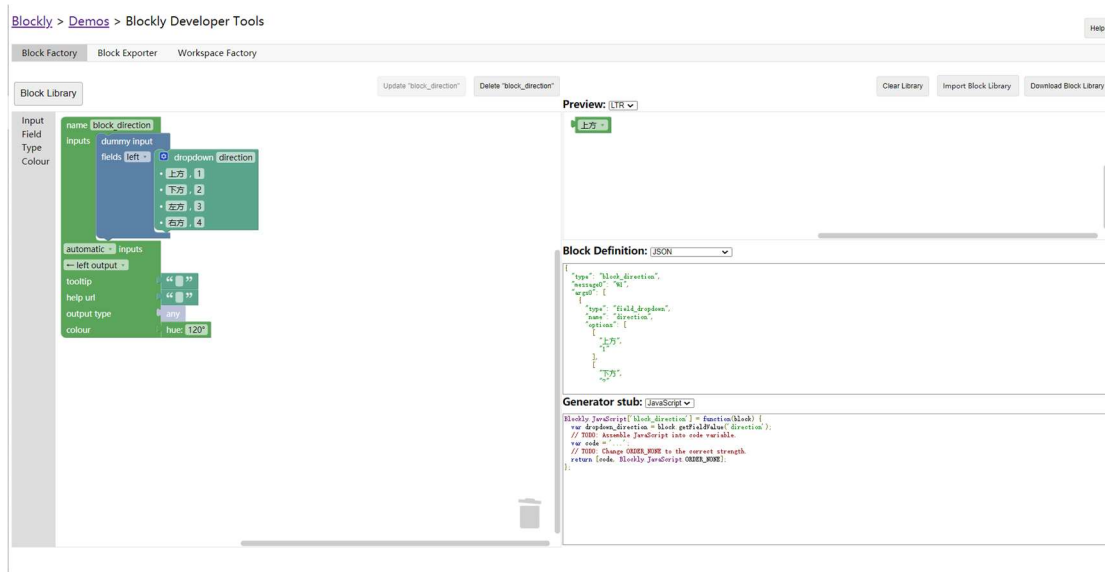
Block Definition: JSON

```
{
  "type": "block_move",
  "message0": "移动英雄",
  "args0": [
    {
      "type": "input_value",
      "name": "direction",
      "type": "Number"
    },
    {
      "type": "input_value",
      "name": "steps"
    }
  ],
  "output": "block_move",
  "x": 0,
  "y": 0
}
```

Generator stub: JavaScript

```
Blockly.JavaScript["block_move"] = function(block) {
  var value_direction = Blockly.JavaScript.valueToCode(block, "direction", Blockly.JavaScript.ORDER_ATOMIC);
  var value_steps = Blockly.JavaScript.valueToCode(block, "steps", Blockly.JavaScript.ORDER_ATOMIC);
  // TODO: Assemble JavaScript into code variable
  var code =
    return code;
};
```

block_direction: 选择方向的下拉菜单块



任务 3:

要求一：利用 **blockly** 的代码生成的相关知识，设计 **blockly** 块的代码显示。即添加一个文本框，并且对 workspace 中 blocks 对应的 Python 代码实时显示。

在 Python Codes 标题下，补充代码至如下：

```
<h2>Python Codes:</h2>
<br>
<textarea id="python-code" rows="20" cols="70"></textarea><br>
```

在 html 末尾的<script></script>标签之间，添加如下代码：

```
function myUpdateFunction(event) {
    var code = Blockly.Python.workspaceToCode(workspace);
    document.getElementById('python-code').value = code;
}
workspace.addChangeListener(myUpdateFunction);
```

完成后的网页中将会出现一个文本框，拖拉块进入工作区，文本框中将会显示 Python 代码。

任务 4:

要求一：利用 **blockly** 的代码生成的相关知识，设计 **blockly** 块的代码显示。

给刚才添加的自定义块，添加块对应的代码(包括 JavaScript 和 Python)。

在 block factory 中，找到刚才自定义的块，拷贝 Generator stub 的代码添加到对应区域，并且编辑 set_hero, block_hero, block_move, block_direction 块的 Code 内容。使其能调用内置的接口，并且操纵英雄，完成项目复现！

小提示：最终成果在 CatchTheMushroom.html 已完成，但是希望各位学员能够独立完成，尽量不要作参考。