在 gui 工程目录下解决 block 的 build 失败问题。

1. /home/bibaodi/demo/scratch-gui/node_modules/scratch-blocks 目录下执行 python build.py 会得到一个错误“Error: Closure not found.  Read this: developers.google.com/blockly/guides/modify/web/closure”

2. 访问上述地址，下载相应的 closure-library 后放到 /home/bibaodi/demo/scratch-gui/node_modules/ 目录

3. 重新执行 python build.py

4. 得到信息

```
 python build.py
SUCCESS: msg/js/en.js

Exception in thread Thread-3:
Traceback (most recent call last):
  File "/usr/lib/python2.7/threading.py", line 801, in __bootstrap_inner
    self.run()
  File "build.py", line 180, in run
    self.gen_core(True)
  File "build.py", line 221, in gen_core
    self.do_compile(params, target_filename, filenames, "")
  File "build.py", line 291, in do_compile
    conn.request("POST", "/compile", urllib.urlencode(params), headers)
  File "/usr/lib/python2.7/httplib.py", line 1042, in request
    self._send_request(method, url, body, headers)
  File "/usr/lib/python2.7/httplib.py", line 1082, in _send_request
    self.endheaders(body)
  File "/usr/lib/python2.7/httplib.py", line 1038, in endheaders
    self._send_output(message_body)
  File "/usr/lib/python2.7/httplib.py", line 882, in _send_output
    self.send(msg)
  File "/usr/lib/python2.7/httplib.py", line 844, in send
    self.connect()
  File "/usr/lib/python2.7/httplib.py", line 1255, in connect
    HTTPConnection.connect(self)
  File "/usr/lib/python2.7/httplib.py", line 821, in connect
    self.timeout, self.source_address)
  File "/usr/lib/python2.7/socket.py", line 575, in create_connection
    raise err
error: [Errno 111] Connection refused

SUCCESS: blockly_uncompressed_horizontal.js
SUCCESS: blockly_uncompressed_vertical.js
```

5. 成功生成 未压缩的文件了，但是依然报错，这个时候需要查看 291 行的错误。原来是请求 url 失败。 *conn = httplib.HTTPSConnection("closure-compiler.appspot.com")*
*conn.request("POST", "/compile", urllib.urlencode(params), headers)*

6. 既然这样那就看看这个链接是干什么的吧？搜索后发现 closure-compiler 是一个 google 提供的 JS 的压缩 API。这个工具不是必须的，不用管，因为这个既可以通过本地程序来代替(附录一就是这个方法)也可以通过其他 API 或者找网络代理等方法。

7. 既然未压缩的生成了，那么压缩的自己解决就好了，利用附录的方法。

---

Appendix I

---

1. Getting Started with the Closure Compiler Application

# The Hello World of the Closure Compiler Application

The Closure Compiler application is a Java command-line utility that compresses, optimizes, and looks for mistakes in your JavaScript. To try out the Closure Compiler application with a simple JavaScript program, follow the steps below.

To work through this exercise you need the Java Runtime Environment version 7.

1. **Download the Closure Compiler package**

   Create a working directory called `closure-compiler`.

   Download the Closure Compiler compiler.jar file and save it in `closure-compiler`.

2. **Create a JavaScript file**

   Create a file named `hello.js` containing the following JavaScript:

   ```
   // A simple function.
   function hello(longName) {
     alert('Hello, ' + longName);
   }
   hello('New User');
   ```

   Save this file in the `closure-compiler` directory.

3. **Compile the JavaScript file**

   Run the following command from the `closure-compiler` directory:

   ```
   java -jar compiler.jar --js hello.js --js_output_file hello-compiled.js
   ```

   This command creates a new file called `hello-compiled.js`, which contains the following JavaScript:

   ```
   function hello(a){alert("Hello, "+a)}hello("New User");
   ```

   Note that the compiler has stripped comments, whitespace and an unnecessary semi-colon. The compiler has also replaced the parameter name `longName` with the shorter name `a`. The result is a much smaller JavaScript file.

   To confirm that the compiled JavaScript code still works correctly, include `hello-compiled.js` in an HTML file like this one:

   ```
   <html>
   <head><title>Hello World</title></head>
   <body>
   <script src="hello-compiled.js"></script>
   </body>
   </html>
   ```

   Load the HTML file in a browser, and you should see a friendly greeting!

## Next Steps

This example illustrates only the most simple optimizations performed by the Closure Compiler. To learn more about the compiler's capabilities, read Advanced Compilation and Externs.

To learn more about other flags and options for the Closure Compiler, execute the jar with the `--help` flag:

```
java -jar compiler.jar --help
```

Except as otherwise noted, the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code samples are licensed under the Apache 2.0 License. For details, see our Site Policies. Java is a registered trademark of Oracle and/or its affiliates.

Last updated July 22, 2015.