为了使用 pytest 检查单元测试覆盖率，我们可以使用 pytest-cov 插件。以下是如何在你的项目中配置和使用 pytest-cov 来检查单元测试覆盖率的步骤：

### **1. 安装 pytest-cov**

首先，确保你已经安装了 pytest-cov 插件。你可以使用以下命令来安装它：

pip install pytest-cov

### **2. 配置和运行测试覆盖率**

在你的项目根目录下运行 pytest 时，使用 --cov 选项来指定要测量覆盖率的包或模块。你可以使用以下命令来运行测试并生成覆盖率报告：

python -m pytest --cov=src tests/

这将运行 tests 目录中的所有测试，并生成 src 目录中代码的覆盖率报告。

### **3. 生成覆盖率报告**

你可以生成不同格式的覆盖率报告，比如在终端输出简要报告，或者生成 HTML 报告来更直观地查看覆盖率细节。

#### **示例命令**

1. ****生成终端简要报告****：

python -m pytest --cov=src tests/

1. ****生成 HTML 报告****：

python -m pytest --cov=src --cov-report=html tests/

生成 HTML 报告后，会在项目根目录下创建一个 htmlcov 目录，你可以在浏览器中打开 htmlcov/index.html 来查看详细的覆盖率报告。

### **4. 项目结构示例**

确保你的项目结构如下：

AutoWulin/│├── src/│ ├── auto\_whack\_a\_mole.py│ └── gui.py│├── tests/│ ├── test\_auto\_whack\_a\_mole.py│ └── \_\_init\_\_.py│├── images/│ └── whack\_a\_mole/│ ├── iron\_ore.png│ ├── female\_mole.png│ └── snake.png│├── logs/│├── config.json5└── README.md

### **5. 示例测试文件**

确保你的测试文件正确配置，以下是一个示例测试文件：

****tests/test\_auto\_whack\_a\_mole.py****

*# tests/test\_auto\_whack\_a\_mole.py*

**import** pytest**from** unittest.mock **import** patch, MagicMock**from** src.auto\_whack\_a\_mole **import** AutoWhackAMole

**@patch('pygetwindow.getWindowsWithTitle')def** **test\_get\_window\_resolution**(mock\_getWindowsWithTitle):

*# 设置模拟返回值*

mock\_window = MagicMock()

mock\_window.width = 800

mock\_window.height = 600

mock\_getWindowsWithTitle.return\_value = [mock\_window]

config = {

'time\_limit': 60,

'threshold': 0.8,

'sleep\_interval': 0.03,

'click\_interval': 0.03

}

game = AutoWhackAMole('test\_game', config)

resolution = game.get\_window\_resolution()

*# 断言*

**assert** resolution == (800, 600)

**assert** game.window **is** **not** **None**

### **6. 运行测试并检查覆盖率**

在项目根目录下运行以下命令：

python -m pytest --cov=src --cov-report=html tests/

这将运行所有测试并生成覆盖率报告，你可以在 htmlcov/index.html 中查看详细的覆盖率报告。

### **总结**

1. 安装 pytest-cov 插件。
2. 使用 --cov 选项运行 pytest 并生成覆盖率报告。
3. 可选择生成 HTML 报告以便更直观地查看覆盖率细节。

通过这些步骤，你可以轻松地检查项目的单元测试覆盖率，并确保代码质量。

是的，Visual Studio Code 可以通过设置和使用插件来可视化显示代码覆盖率。你可以使用插件 Coverage Gutters 来实现这一点。以下是具体的步骤：

### **1. 安装 Coverage Gutters 插件**

1. 打开 VS Code。
2. 进入扩展（Extensions）视图，或者按下 Ctrl+Shift+X。
3. 在搜索栏中输入 Coverage Gutters。
4. 找到并安装 Coverage Gutters 插件。

### **2. 生成代码覆盖率报告**

确保你已经使用 pytest 和 pytest-cov 生成了代码覆盖率报告。你可以使用以下命令生成覆盖率报告：

python -m pytest --cov=src --cov-report=xml --cov-report=html tests/

这将生成 coverage.xml 文件（XML 格式的覆盖率报告）和 htmlcov 目录（HTML 格式的覆盖率报告）。

### **3. 配置 Coverage Gutters**

1. 打开 VS Code 设置（按下 Ctrl+, 或者通过菜单 File > Preferences > Settings）。
2. 搜索 Coverage Gutters。
3. 找到 Coverage Gutters: Coverage File Names 选项，并添加 coverage.xml。
4. 确保 Coverage Gutters: Coverage Report File 选项中包含你的覆盖率报告文件名。

### **4. 在 VS Code 中显示覆盖率**

1. 生成覆盖率报告后，打开命令面板（按下 Ctrl+Shift+P）。
2. 输入并选择 Coverage Gutters: Display Coverage。
3. 选择生成的 coverage.xml 文件。

现在，你应该能够在 VS Code 中看到覆盖率信息，未覆盖的代码将以红色高亮显示，已覆盖的代码将以绿色高亮显示。

### **总结**

1. ****安装** Coverage Gutters **插件****：在 VS Code 中搜索并安装 Coverage Gutters 插件。
2. ****生成代码覆盖率报告****：使用 pytest 和 pytest-cov 生成 coverage.xml 文件。
3. ****配置** Coverage Gutters **插件****：在 VS Code 设置中添加 coverage.xml。
4. ****显示覆盖率****：使用命令面板中的 Coverage Gutters: Display Coverage 选项加载覆盖率报告。

通过这些步骤，你可以在 VS Code 中直观地查看代码覆盖率，方便地识别未覆盖的代码部分。