

[Qt设计工作室手册](#) > [调试Qt快速示例应用程序](#)

调试Qt快速示例应用程序

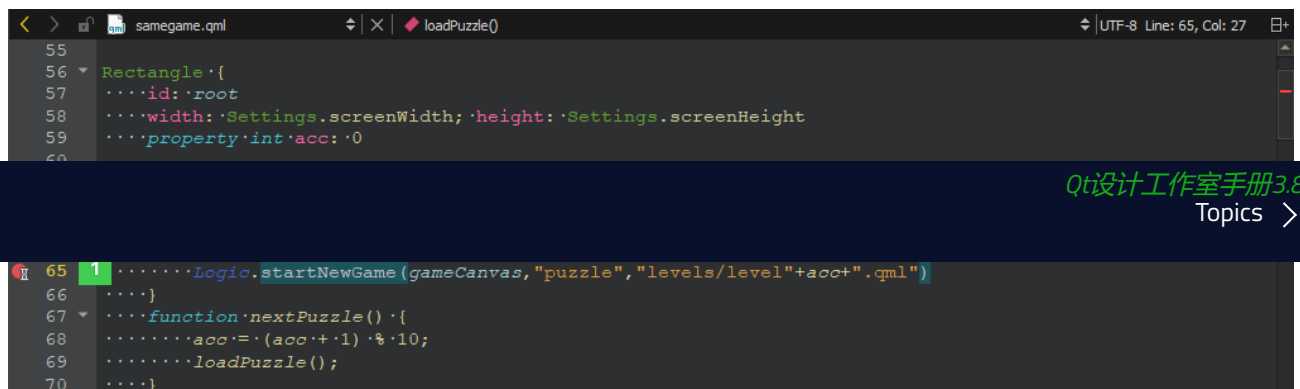
本节使用[同一游戏](#)示例应用程序来说明如何在调试模式下[调试](#)Qt Quick应用程序。

有关您拥有的所有选项的更多信息，请参阅[调试 Qt 快速项目](#)。

注意：在本节中，您将使用高级菜单项。默认情况下，这些是不可见的。要切换高级菜单项的可见性，请参阅[自定义菜单](#)。

同一游戏演示展示了如何在QML中编写游戏，使用JavaScript进行所有游戏逻辑。在Qt Design Studio中打开演示项目进行调试：

1. 要查看启动新游戏的代码，请在 `samegame.qml` 中放置一个断点，方法是在行号和调用函数的行上的窗口边框之间单击 (1)。`startNewGame()`



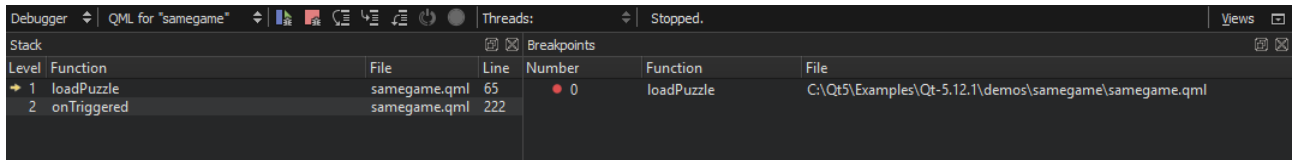
红色圆圈表示现在在该行号上设置了断点。

2. 选择“[调试](#)>[启动调试](#)>[启动项目的调试](#)”或按F5。
3. 同一游戏应用程序启动后，选择拼图以启动新游戏。

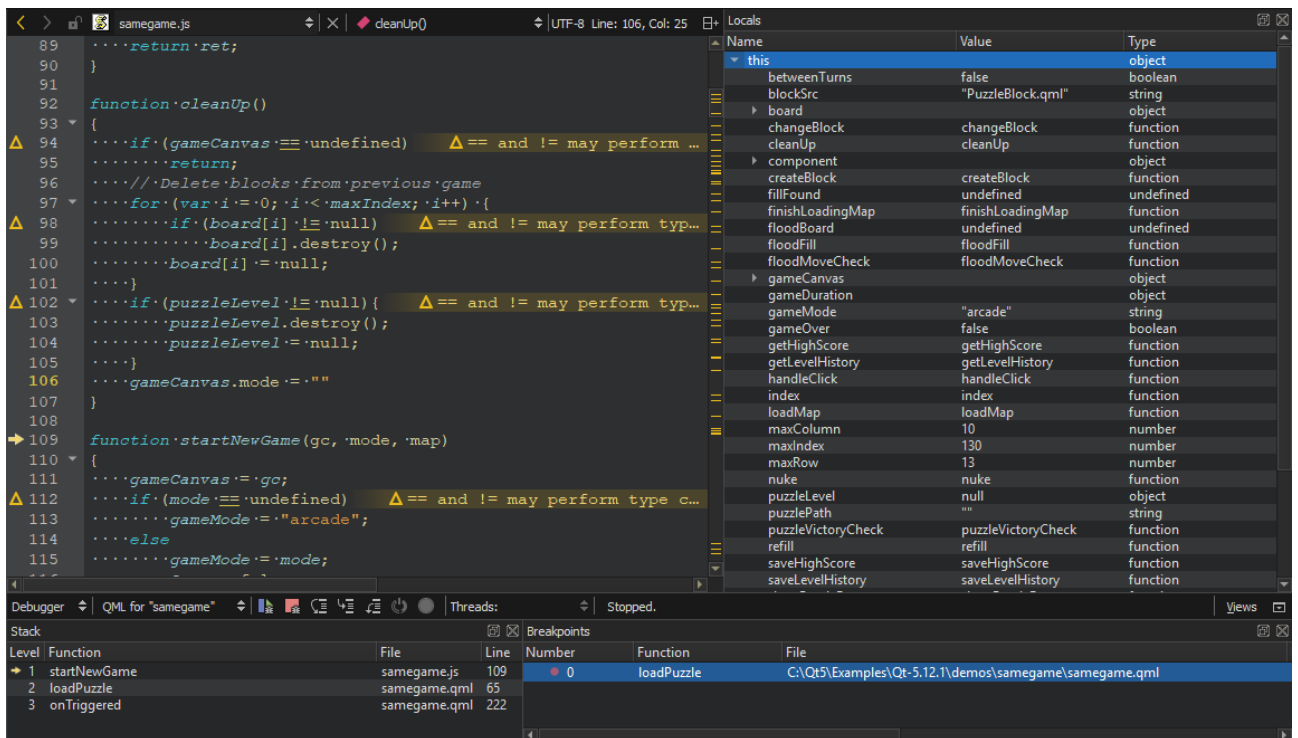





4. 当调试器命中断点时，它会中断应用程序。Qt Design Studio将显示导致当前位置的嵌套函数调用作为调用堆栈跟踪（1）。



5. 单击工具栏上的 （单步执行）按钮或按F11 单步执行堆栈中的代码。相同的game.js文件在代码编辑器中打开，用于启动新游戏的函数。



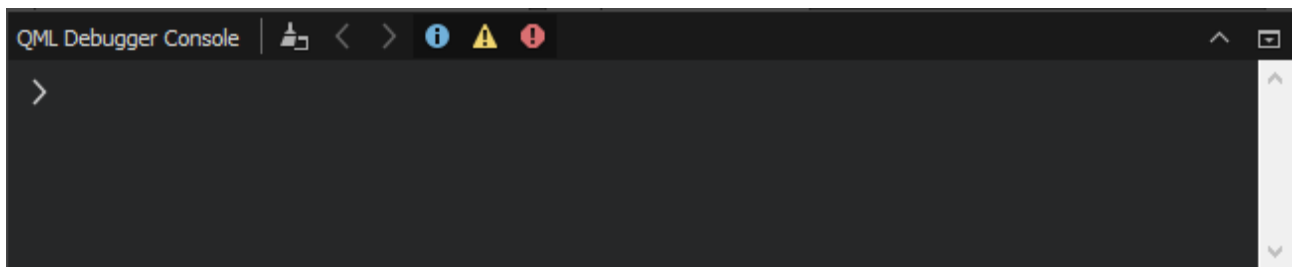
6. 检查“局部变量”视图中的局部变量。单步执行代码以查看视图中的信息如何更改。
7. 在函数末尾添加一个断点，然后单击  继续以命中断点。startNewGame()

```

128
129
130     ...//Initialize Board
131     ...board = new Array(maxIndex);
132     ...gameCanvas.score = 0;
133     ...gameCanvas.score2 = 0;
134     ...gameCanvas.moves = 0;
135     ...gameCanvas.curTurn = 1;
136     ...if (gameMode === "puzzle")
137     ...    loadMap(map);
138     ...else//Note that we load them in reverse order for correct visual stacking
139     ...    for (var column = maxColumn - 1; column >= 0; column--)
140     ...        for (var row = maxRow - 1; row >= 0; row--)
141     ...            createBlock(column, row);
142     ...if (gameMode === "puzzle")
143     ...    getLevelHistory(); //Needs to be after map load
144     ...gameDuration = new Date();
145 }

```

8. 要在当前上下文中执行 JavaScript 命令，请打开QML 调试器控制台。



9. 若要删除断点，请右键单击该断点，然后选择“删除断点”。

10. 在“局部变量”视图中，在运行时浏览对象结构。

Name	Value	Type
this		object
betweenTurns	false	boolean
blockSrc	"PuzzleBlock.qml"	string
board		object
changeBlock	changeBlock	function
cleanUp	cleanUp	function
component		object
createBlock	createBlock	function
fillFound	undefined	undefined
finishLoadingMap	finishLoadingMap	function
floodBoard	undefined	undefined
floodFill	floodFill	function
floodMoveCheck	floodMoveCheck	function
gameCanvas		object
gameDuration	undefined	undefined
gameMode	"puzzle"	string
gameOver	false	boolean
getHighScore	getHighScore	function
getLevelHistory	getLevelHistory	function
handleClick	handleClick	function
index	index	function
loadMap	loadMap	function
maxColumn	10	number
maxIndex	130	number
maxRow	13	number
nuke	nuke	function
puzzleLevel		object
puzzlePath	"levels/level0.qml"	string
puzzleVictoryCheck	puzzleVictoryCheck	function
refill	refill	function
saveHighScore	saveHighScore	function
saveLevelHistory	saveLevelHistory	function

11. 选择“调试>在顶部显示应用程序”，以便在与调试器交互时保持应用程序可见。

12. 选择“调试>选择”以激活选择模式，然后单击“菜单”按钮以移动到“局部变量”视图和代码编辑器中的 menuButton 组件。

13. 在“局部变量”视图中，双击属性的值以对其进行更改。



联系我们

公司

- 关于我们
- 投资者
- 编辑部
- 职业
- 办公地点

支持

- 支持服务
- 专业服务
- 合作伙伴
- 训练

社区

- 为Qt做贡献
- 论坛
- 维基
- 下载
- 市场

发牌

- 条款和条件
- 开源
- 常见问题

对于客户

- 支持中心
- 下载
- Qt登录
- 联系我们
- 客户成功案例