

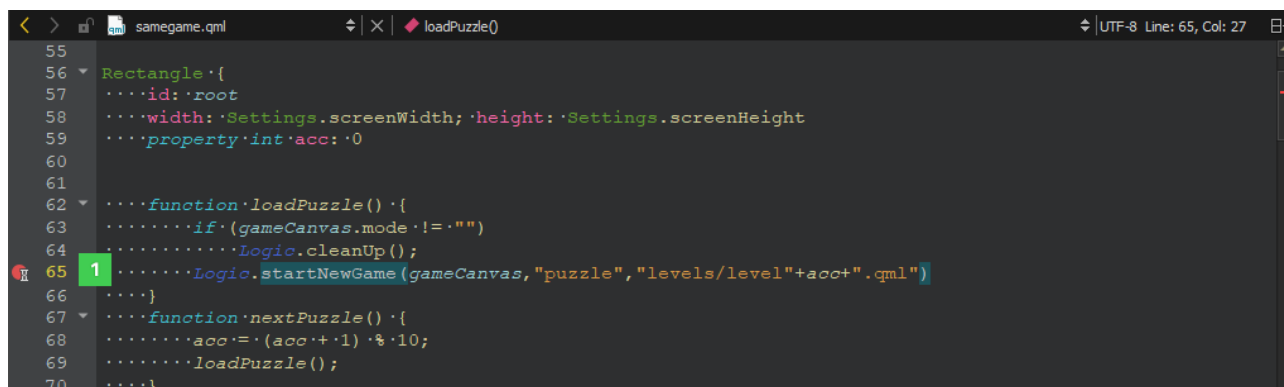
调试 Qt 快速示例应用程序

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，请参见[调试 Qt 快速项目](#)。

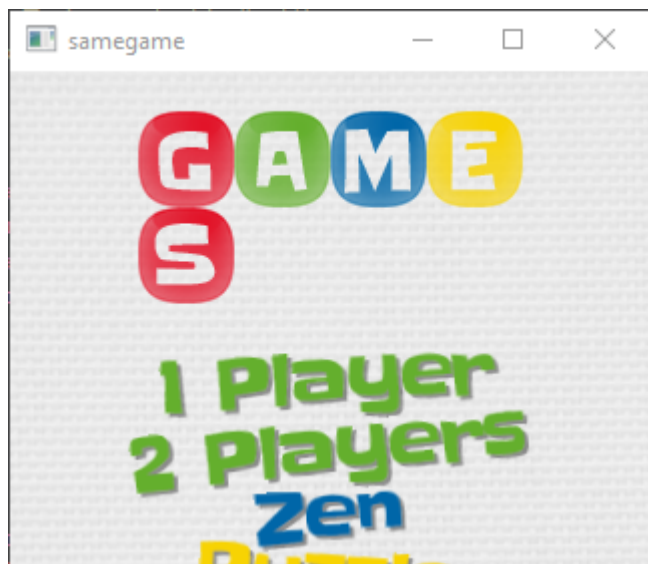
同一游戏演示展示了如何在 QML 中编写游戏，对所有游戏逻辑使用 JavaScript。在 Qt 创建器中打开演示项目进行调试：

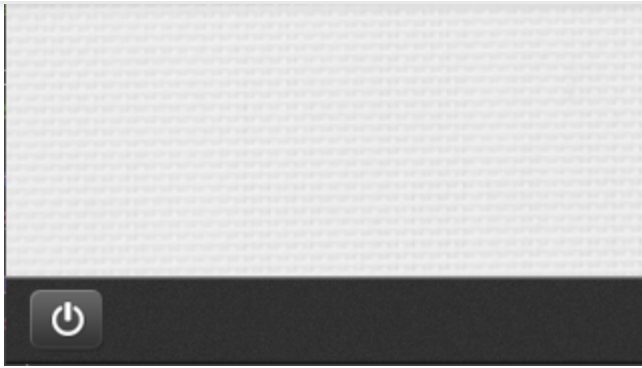
1. 若要查看启动新游戏的代码，请在 `samegame.qml` 中放置一个断点，方法是在行号和调用函数的行上的窗口边框之间单击 (1)。`startNewGame()`



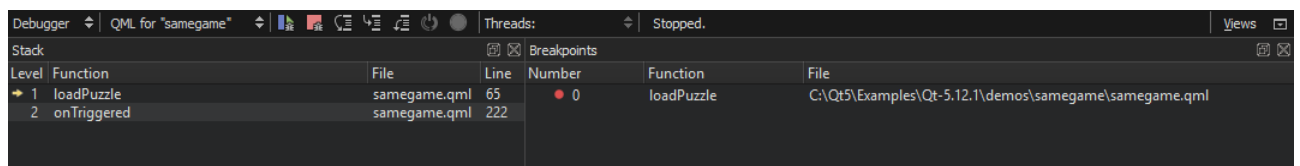
红色圆圈表示现在在该行号上设置了断点。

2. 选择“调试”>“启动调试”>“启动项目”的“启动调试”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。

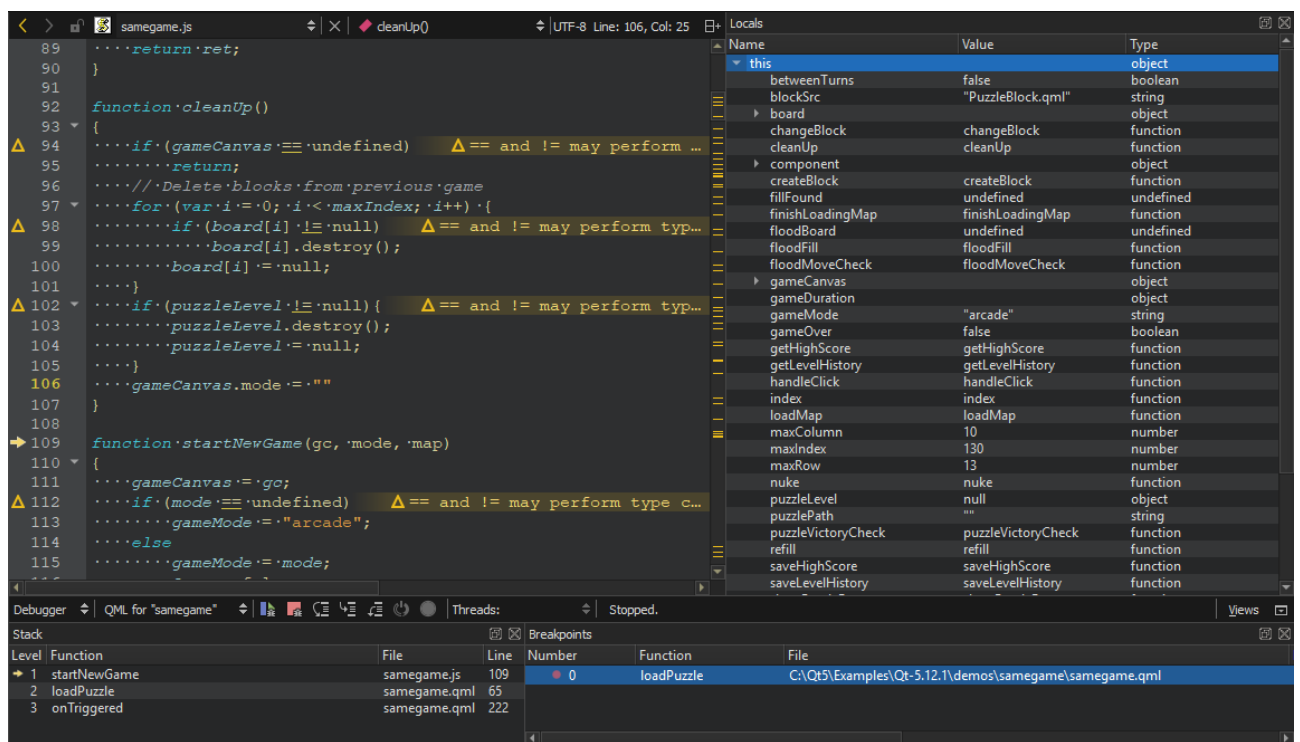




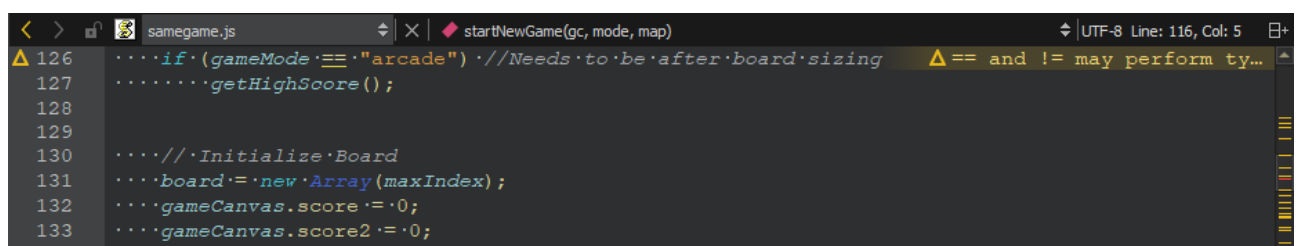
4. 当调试器到达断点时，它将中断应用程序。Qt Creator 将导致当前位置的嵌套函数调用显示为调用堆栈跟踪 (1)。



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。同一游戏.js文件将在代码编辑器中启动新游戏的函数中打开。



6. Examine the local variables in the **Locals** view. Step through the code to see how the information changes in the view.
7. Add a breakpoint at the end of the function, and click  (Continue) to hit the breakpoint. `startNewGame()`

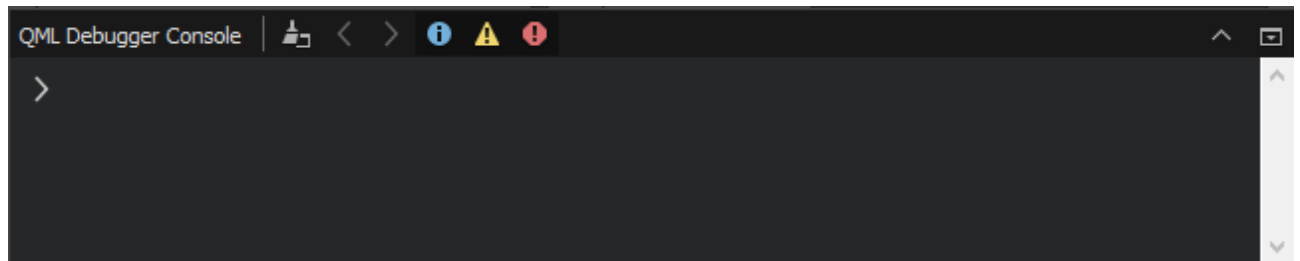


```

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. To execute JavaScript commands in the current context, open the **QML Debugger Console**.



9. To remove a breakpoint, right-click it and select **Delete Breakpoint**.

10. In the **Locals** view, explore the object structure at runtime.

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. Select **Debug > Show Application on Top** to keep the application visible while you interact with the debugger.

12. Select **Debug > Select** to activate selection mode and then click the **Menu** button to move into the **menuButton** component in the **Locals** view and the code editor.

13. In the **Locals** view, double-click the value of a property to change it.

< [Debugging a C++ Example Application](#)

[Troubleshooting Debugger](#) >



Contact Us

Company

- About Us
- Investors
- Newsroom
- Careers
- Office Locations

Support

- Support Services
- Professional Services
- Partners
- Training

Community

- Contribute to Qt
- Forum
- Wiki
- Downloads
- Marketplace

Licensing

- Terms & Conditions
- Open Source
- FAQ

For Customers

- Support Center
- Downloads
- Qt Login
- Contact Us
- Customer Success