

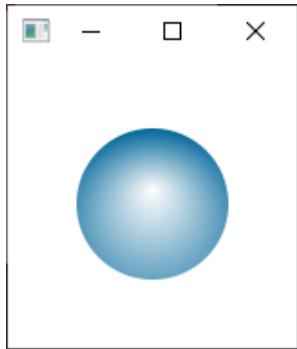
[Qt创作者手册](#) > [创建移动应用程序](#)

创建移动应用程序

本教程描述了当使用Qt 6作为最低Qt版本和CMake作为构建系统时，如何使用Qt Creator为Android和iOS设备开发Qt Quick应用程序。

我们实现了一个Qt Quick应用程序，该应用程序根据不断变化的加速度计值加速SVG（可缩放矢量图形）图像。

注意：您必须安装Qt 6.2或更高版本的Qt传感器模块才能遵循本教程。



设置开发环境

要为移动设备构建应用程序并在移动设备上运行它，您必须为设备平台设置开发环境，并在Qt Creator和移动设备之间配置连接。

要针对 Android 设备进行开发，您必须安装Qt for Android 并按照[连接 Android 设备](#)中的说明设置开发环境。

若要针对 iOS 设备进行开发，必须安装 Xcode 并使用它来配置设备。为此，您需要从Apple收到的Apple开发人员帐户和iOS开发人员计划证书。有关详细信息，请参阅[连接 iOS 设备](#)。

创建项目

1. 选择**文件>新建项目>应用程序（Qt）>Qt 快速应用程序**。
2. 选择“选择”以打开“**项目位置**”对话框。
3. 在“**名称**”字段中，输入应用程序的名称。在命名自己的项目时，请记住，以后无法轻松重命名它们。
4. 在“**创建位置**”字段中，输入项目文件的路径。您可以稍后毫无问题地移动项目文件夹。
5. 选择**下一步**（或在 macOS 上**继续**）以打开定义**构建系统**对话框。
6. 在“生成系统”字段中，选择“**CMake**”作为用于生成和运行项目的**生成系统**。

注意：如果选择qmake，则配置项目的说明将不适用。

7. 选择“**下一步**”以打开“**定义项目详细信息**”对话框。



DOCUMENTATION

10. 选择“**下一步**”以使用默认设置并打开“**套件选择**”对话框。
11. 要为其构建应用程序的平台选择 Qt 6.2 或更高版本的**工具包**。要为移动设备构建应用程序，请同时选择适用于 Android 和 iOS 的工具包。

注意：如果在Edit>Preferences>Kits（在 Windows 和 Linux 上）或Qt Creator>Preferences>Kits（在 macOS 上）中指定了工具包，则会列出它们。有关详细信息，请参阅[添加工具包](#)。

12. 选择“**下一步**”以打开“**项目管理**”对话框。
13. 查看项目设置，然后选择“**完成**”（或在 macOS 上选择“**完成**”）以创建项目。

有关跳过的设置和其他可用的向导模板的更多信息，请参阅[创建Qt快速应用程序](#)。

将图像添加为资源

应用程序的主视图显示一个 SVG 气泡图像，当您倾斜设备时，该图像会在屏幕上移动。

在本教程中，我们使用 *Bluebubble.svg* 但您可以使用任何其他图像或组件。

若要在运行应用程序时显示映像，必须在向导为您创建的 *CMakeLists.txt* 文件的部分中将其指定为资源：RESOURCES

```
qt_add_qml_module(appaccelbubble
    URI accelbubble
    VERSION 1.0
    QML_FILES main.qml
    RESOURCES Bluebubble.svg
)
```

创建加速气泡主视图

我们通过添加一个以 *Bluebubble.svg* 作为源的**图像组件**来创建 *main.qml* 文件中的主视图：

```
Image {
    id: bubble
    source: "Bluebubble.svg"
    smooth: true
```

接下来，我们添加自定义属性以相对于主窗口的宽度和高度定位图像：

```
property real centerX: mainWindow.width / 2
property real centerY: mainWindow.height / 2
property real bubbleCenter: bubble.width / 2
x: centerX - bubbleCenter
y: centerY - bubbleCenter
```

我们现在要添加代码以根据加速度计传感器值移动气泡。首先，我们添加以下导入语句：

```
import QtSensors
```

```
Accelerometer {
    id: accel
    dataRate: 100
    active:true
```

然后，我们添加以下 JavaScript 函数，根据当前加速度计值计算气泡的 x 和 y 位置：

```
function calcPitch(x,y,z) {
    return -Math.atan2(y, Math.hypot(x, z)) * mainWindow.radians_to_degrees;
}
function calcRoll(x,y,z) {
    return -Math.atan2(x, Math.hypot(y, z)) * mainWindow.radians_to_degrees;
}
```

我们为加速度计组件的信号添加了以下 JavaScript 代码，以使气泡在加速度计值更改时移动：onReadingChanged

```
onReadingChanged: {
    var newX = (bubble.x + calcRoll(accel.reading.x, accel.reading.y, accel.reading.z) * .1)
    var newY = (bubble.y - calcPitch(accel.reading.x, accel.reading.y, accel.reading.z) * .1)

    if (isNaN(newX) || isNaN(newY))
        return;

    if (newX < 0)
        newX = mainWindow.width - bubble.width

    if (newY < 18)
        newY = 18

    if (newY > mainWindow.height - bubble.height)
        newY = mainWindow.height - bubble.height

    bubble.x = newX
    bubble.y = newY
}
```

Qt 创建者手册 8.0.2
Topics >

我们希望确保气泡的位置始终在屏幕的范围内。如果加速度计返回的不是数字（NaN），则会忽略该值，并且不会更新气泡位置。

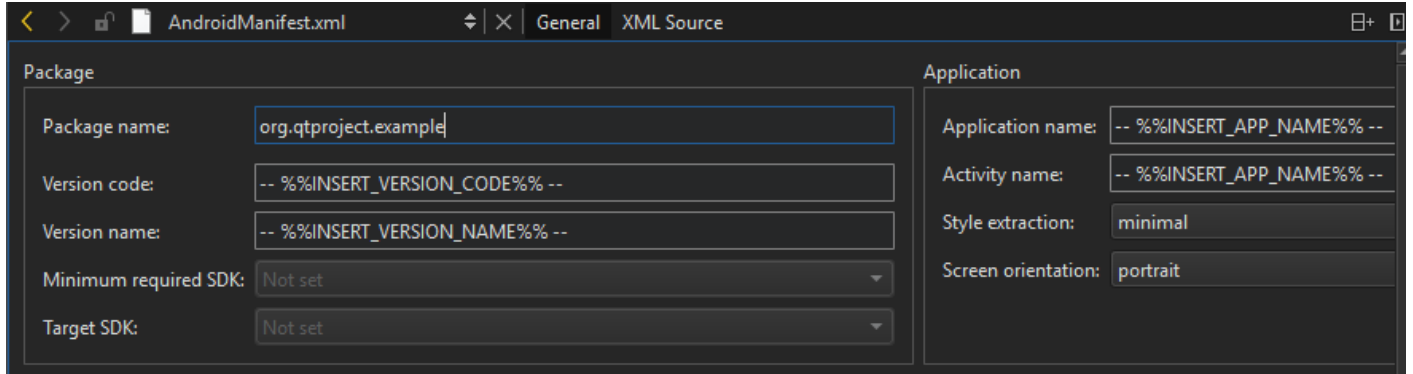
我们在气泡的属性和属性上添加了平滑动画行为，以使其移动看起来更平滑。xy

```
Behavior on y {
    SmoothedAnimation {
        easing.type: Easing.Linear
        duration: 100
    }
}
Behavior on x {
    SmoothedAnimation {
        easing.type: Easing.Linear
```

锁定装置方向

默认情况下，当设备方向在纵向和横向之间更改时，设备显示屏会旋转。对于此示例，最好固定屏幕方向。

要在Android上锁定纵向或横向方向，请在*Android清单*中指定它.xml您可以在Qt Creator中生成。有关更多信息，请参阅[编辑清单文件](#)。



若要生成和使用清单文件，必须在*CMakeLists.txt*文件中指定 Android 包源目录：QT_ANDROID_PACKAGE_SOURCE_DIR

```
set_property(TARGET appaccelbubble APPEND PROPERTY
    QT_ANDROID_PACKAGE_SOURCE_DIR ${CMAKE_CURRENT_SOURCE_DIR}/android
)
```

因为我们的CMake版本早于3.19，所以我们必须在函数中添加一个手动完成步骤：qt_add_executable

```
qt_add_executable(appaccelbubble
    main.cpp
    MANUAL_FINALIZATION
)
```

我们还需要添加函数：qt_finalize_executable

```
qt_finalize_executable(appaccelbubble)
```

在 iOS 上，可以将设备方向锁定在*Info.plist*文件中，该文件在*CMakeLists.txt*文件中指定为变量的值：MACOSX_BUNDLE_INFO_PLIST

```
set_target_properties(appaccelbubble PROPERTIES
    MACOSX_BUNDLE_GUI_IDENTIFIER my.example.com
    MACOSX_BUNDLE_BUNDLE_VERSION ${PROJECT_VERSION}
    MACOSX_BUNDLE_SHORT_VERSION_STRING ${PROJECT_VERSION_MAJOR}.${PROJECT_VERSION_MINOR}
    MACOSX_BUNDLE_INFO_PLIST "${CMAKE_CURRENT_SOURCE_DIR}/Info.plist"
    MACOSX_BUNDLE TRUE
    WIN32_EXECUTABLE TRUE
)
```

您必须通过在项目文件中指定依赖项来告诉构建系统您的应用程序需要哪些Qt模块。选择“**项目**”以使用以下 Qt 模块信息更新 CMake 配置： „SensorsSvgXml

CMakeLists.txt文件应包含以下条目，这些条目告诉 CMake 查找 Qt 安装并导入应用程序所需的 Qt Sensors、Qt SVG 和 Qt XML 模块：

```
find_package(Qt6 6.2 COMPONENTS Quick Sensors Svg Xml REQUIRED)
```

您还需要将 Qt 模块添加到目标链接库列表中。告诉 CMake，accelbubble 可执行文件通过引用上面调用导入的目标来使用 Qt Sensors、Qt SVG 和 Qt XML 模块。这会将必要的参数添加到链接器，并确保将相应的包含目录和编译器定义传递给 C++ 编译器。target_link_libraries(find_package())

```
target_link_libraries(appaccelbubble
    PRIVATE Qt6::Quick Qt6::Sensors Qt6::Svg Qt6::Xml)
```

添加依赖项后，选择“**生成>运行 CMake**”以应用配置更改。

有关 CMakeLists.txt 文件的详细信息，请参阅[CMake 入门](#)。

运行应用程序

应用程序已完成，可以部署到设备：

1. 在安卓设备上启用**USB调试**，或在iOS设备上启用**开发人员模式**。
2. 将设备连接到开发电脑。

如果您使用的是运行Android v4.2.2的设备，它应该提示您验证连接以允许从其连接的PC进行USB调试。若要避免每次连接设备时都出现此类提示，请选中“**始终允许来自此计算机**”复选框，然后选择“**确定**”。

3. 若要在设备上运行应用程序，请按Ctrl+R。

文件：

- › [加速泡泡/蓝泡泡.svg](#)
- › [accelbubble/CMakeLists.txt](#)
- › [accelbubble/main.qml](#)

[‹ 创建基于 Qt Widget 的应用程序](#)

[管理项目 ›](#)

©2022 Qt有限公司 此处包含的文档贡献的版权归 他们各自的所有者。此处提供的文档根据自由软件基金会发布的GNU 自由文档许可证版本1.3的条款进行许可。Qt和相应的徽标是Qt有限公司在芬兰和/或全球其他国家的商标。所有其他商标均为财产 其各自所有者。



[联系我们](#)

关于我们
投资者
编辑部
职业
办公地点

条款和条件
开源
常见问题

支持

支持服务
专业服务
合作 伙伴
训练

对于客户

支持中心
下载
Qt登录
联系我们
客户成功案例

社区

为Qt做贡献
论坛
维基
下载
市场

© 2022 Qt公司

[反馈](#) [登录](#)