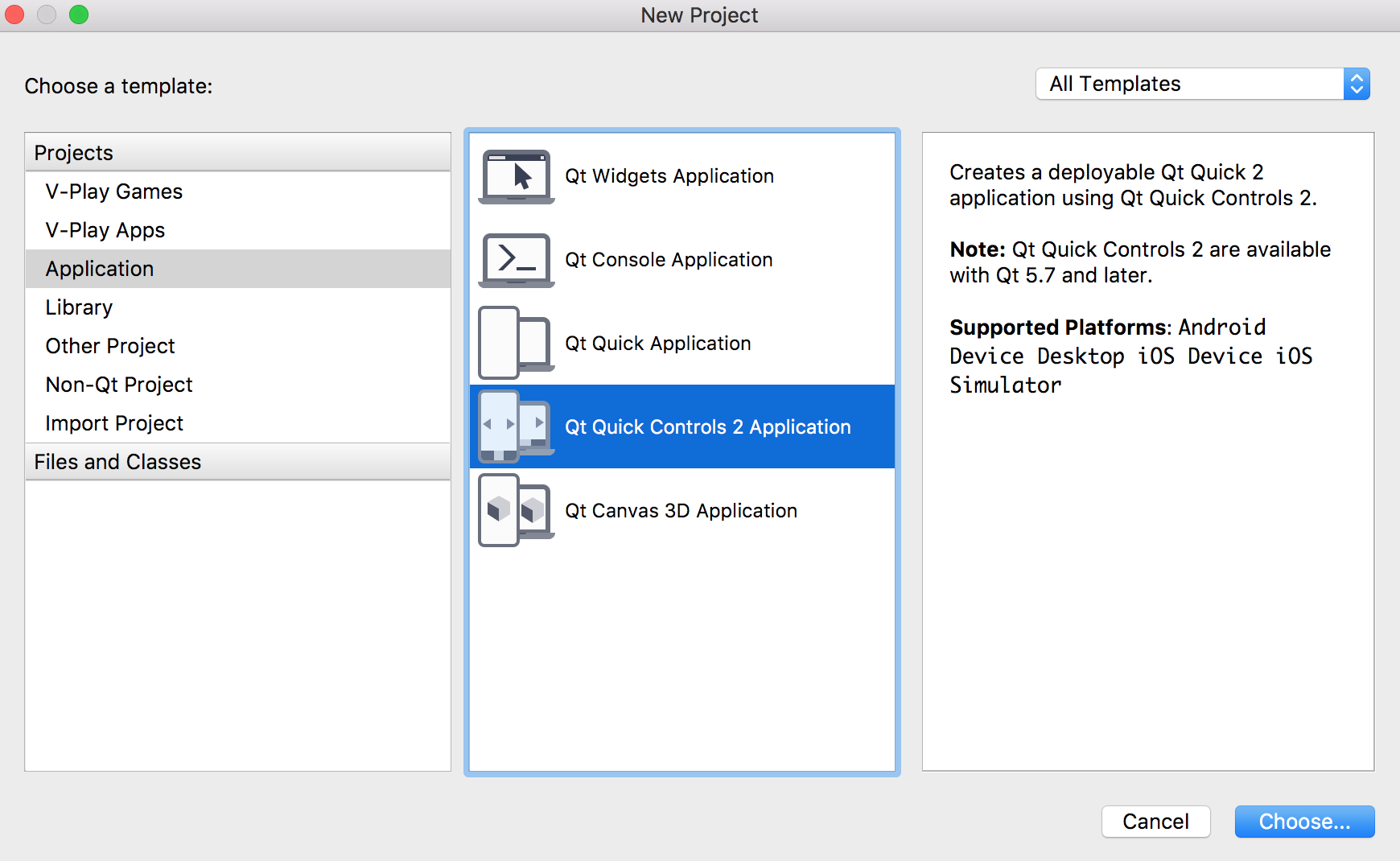
Qt Quick

* UI: qml/js
* Code behind: C++

Loại project sử dụng: Qt Quick Controls 2 Application

(Loại project được xây dựng sẵn một số control cơ bản, sẽ giúp cho quá trình xây dựng ứng dụng nhanh hơn!)



Phần 1: QML

Phần nhiều UI và các logic đơn giản sẽ được hoàn thiện bằng các file .qml

|  |
| --- |
| **//main.qml**  import QtQuick 2.7  import QtQuick.Controls 2.0  import QtQuick.Layouts 1.3  ApplicationWindow {  visible: true  width: 640  height: 480  title: *qsTr*("Hello World")  SwipeView {  id: *swipeView*  anchors.fill: *parent*  currentIndex: *tabBar*.currentIndex  Page1 {  }  Page {  Label {  text: *qsTr*("Second page")  anchors.centerIn: *parent*  }  }  }  footer: TabBar {  id: *tabBar*  currentIndex: *swipeView*.currentIndex  TabButton {  text: *qsTr*("First")  }  TabButton {  text: *qsTr*("Second")  }  }  } |

Cấu trúc của qml cũng gần giống json

|  |
| --- |
| TenControl {  TenThuocTinh: value  TenThuocTinh.SubThuocTinh: value  OtherControl {  }  }  TenThuocTinh: TenControl {  id: *tabBar*  } |

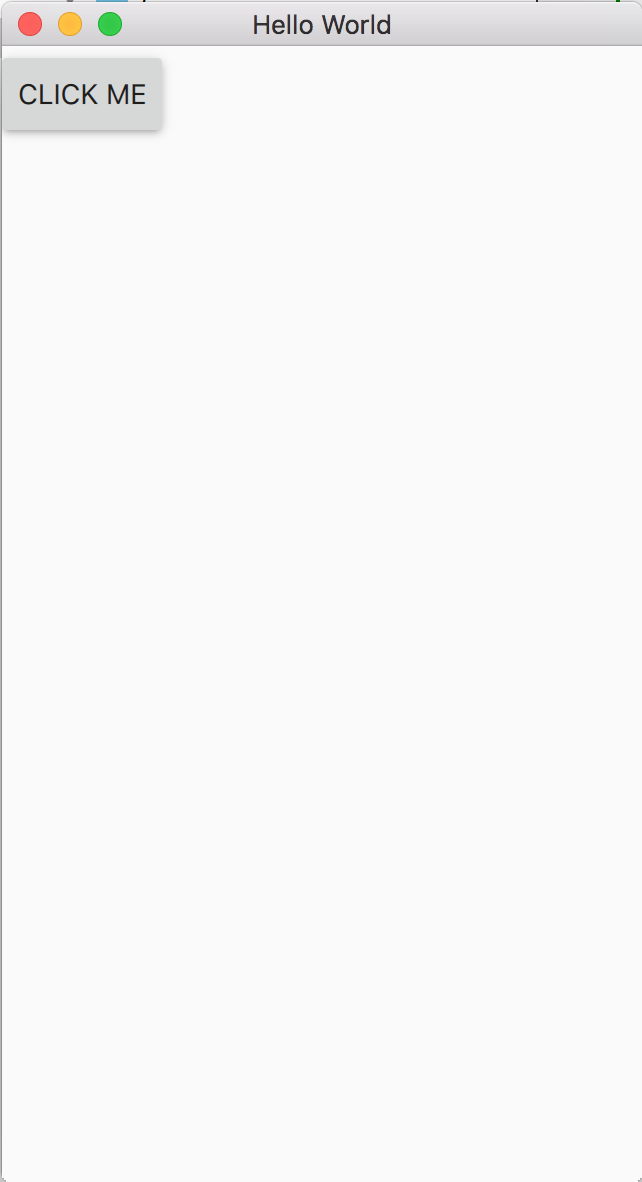
Tạo một button

|  |
| --- |
| Button{  id: *myFirstButton*  text: *qsTr*("Click Me")  } |

Main.qml

|  |
| --- |
| import QtQuick 2.7  import QtQuick.Controls 2.0  import QtQuick.Layouts 1.3  ApplicationWindow {  visible: true  width: 640/2  height: 1136/2  title: *qsTr*("Hello World")  Button{  id: *myFirstButton*  text: *qsTr*("Click Me")  }  } |

Result:



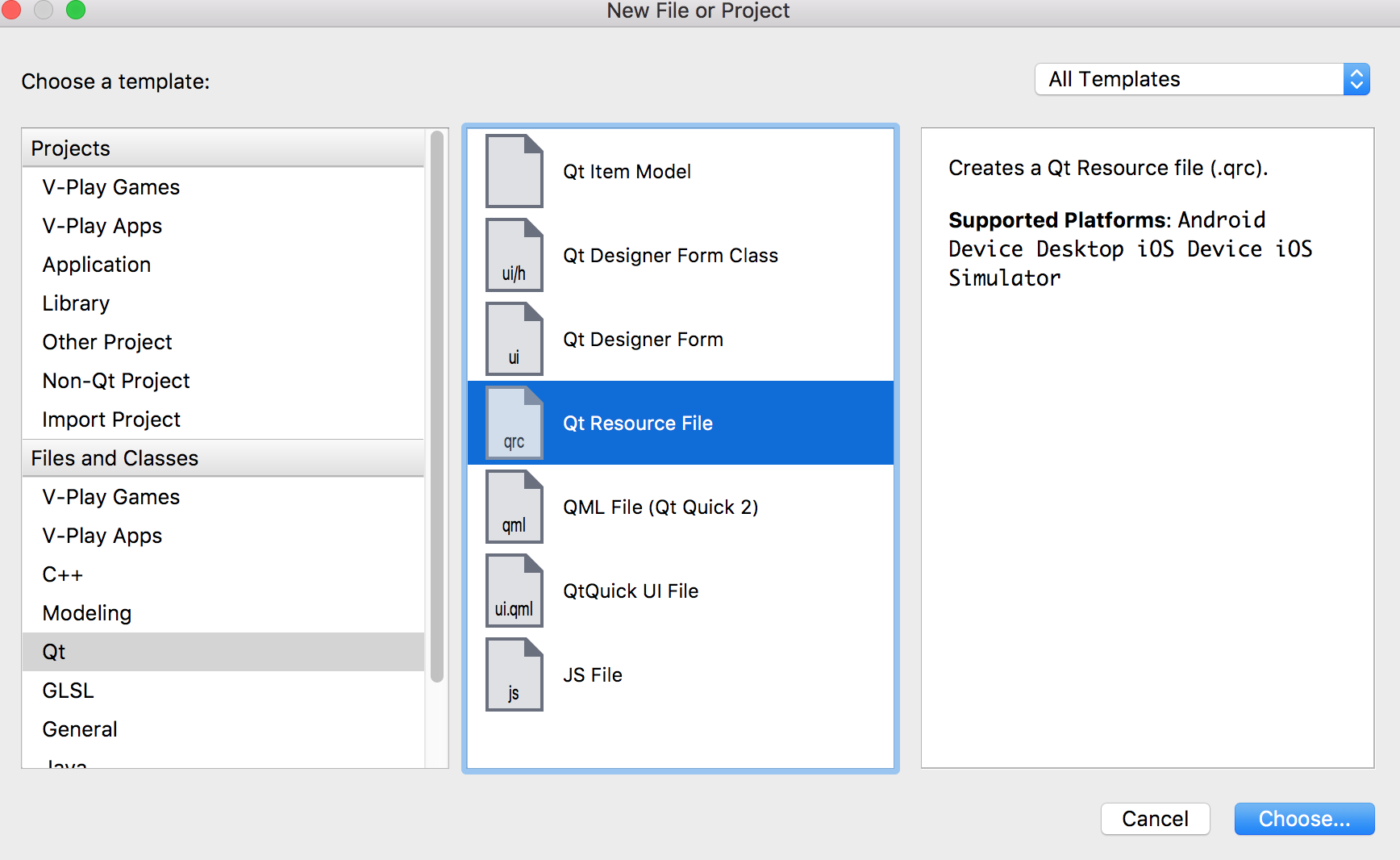
# Ref

## Các control có

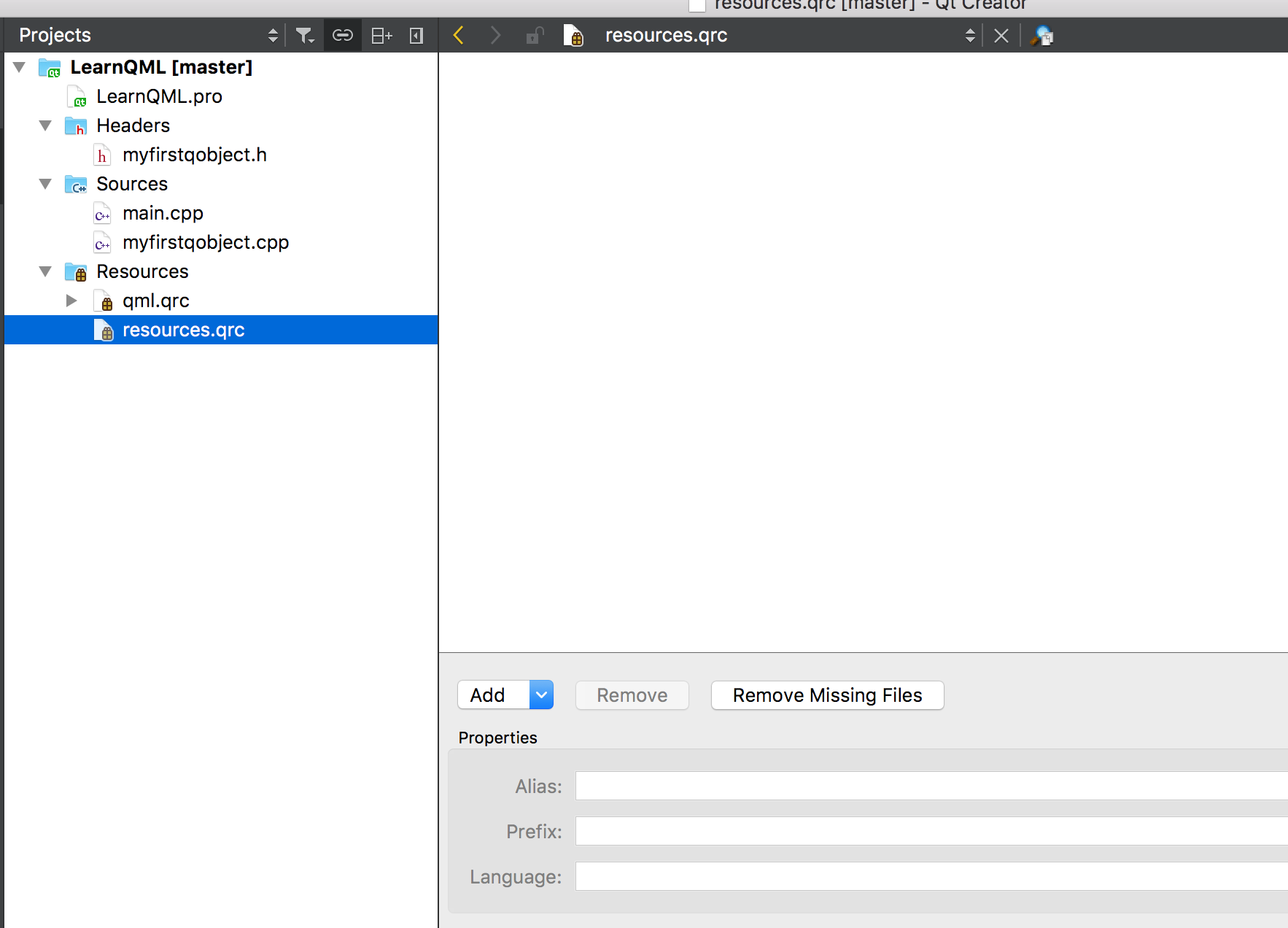
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| QML Types   |  |  | | --- | --- | | [AbstractButton](https://doc.qt.io/qt-5/qml-qtquick-controls2-abstractbutton.html) | Abstract base type providing functionality common to buttons | | [ApplicationWindow](https://doc.qt.io/qt-5/qml-qtquick-controls2-applicationwindow.html) | Styled top-level window with support for a header and footer | | [BusyIndicator](https://doc.qt.io/qt-5/qml-qtquick-controls2-busyindicator.html) | Indicates background activity, for example, while content is being loaded | | [Button](https://doc.qt.io/qt-5/qml-qtquick-controls2-button.html) | Push-button that can be clicked to perform a command or answer a question | | [ButtonGroup](https://doc.qt.io/qt-5/qml-qtquick-controls2-buttongroup.html) | Mutually-exclusive group of checkable buttons | | [CheckBox](https://doc.qt.io/qt-5/qml-qtquick-controls2-checkbox.html) | Check button that can be toggled on or off | | [CheckDelegate](https://doc.qt.io/qt-5/qml-qtquick-controls2-checkdelegate.html) | Item delegate with a check indicator that can be toggled on or off | | [ComboBox](https://doc.qt.io/qt-5/qml-qtquick-controls2-combobox.html) | Combined button and popup list for selecting options | | [Container](https://doc.qt.io/qt-5/qml-qtquick-controls2-container.html) | Abstract base type providing functionality common to containers | | [Control](https://doc.qt.io/qt-5/qml-qtquick-controls2-control.html) | Abstract base type providing functionality common to all controls | | [DelayButton](https://doc.qt.io/qt-5/qml-qtquick-controls2-delaybutton.html) | Check button that triggers when held down long enough | | [Dial](https://doc.qt.io/qt-5/qml-qtquick-controls2-dial.html) | Circular dial that is rotated to set a value | | [Dialog](https://doc.qt.io/qt-5/qml-qtquick-controls2-dialog.html) | Popup dialog with standard buttons and a title, used for short-term interaction with the user | | [DialogButtonBox](https://doc.qt.io/qt-5/qml-qtquick-controls2-dialogbuttonbox.html) | A button box used in dialogs | | [Drawer](https://doc.qt.io/qt-5/qml-qtquick-controls2-drawer.html) | Side panel that can be opened and closed using a swipe gesture | | [Frame](https://doc.qt.io/qt-5/qml-qtquick-controls2-frame.html) | Visual frame for a logical group of controls | | [GroupBox](https://doc.qt.io/qt-5/qml-qtquick-controls2-groupbox.html) | Visual frame and title for a logical group of controls | | [ItemDelegate](https://doc.qt.io/qt-5/qml-qtquick-controls2-itemdelegate.html) | Basic item delegate that can be used in various views and controls | | [Label](https://doc.qt.io/qt-5/qml-qtquick-controls2-label.html) | Styled text label with inherited font | | [Menu](https://doc.qt.io/qt-5/qml-qtquick-controls2-menu.html) | Popup that can be used as a context menu or popup menu | | [MenuItem](https://doc.qt.io/qt-5/qml-qtquick-controls2-menuitem.html) | Presents an item within a Menu | | [MenuSeparator](https://doc.qt.io/qt-5/qml-qtquick-controls2-menuseparator.html) | Separates a group of items in a menu from adjacent items | | [Page](https://doc.qt.io/qt-5/qml-qtquick-controls2-page.html) | Styled page control with support for a header and footer | | [PageIndicator](https://doc.qt.io/qt-5/qml-qtquick-controls2-pageindicator.html) | Indicates the currently active page | | [Pane](https://doc.qt.io/qt-5/qml-qtquick-controls2-pane.html) | Provides a background matching with the application style and theme | | [Popup](https://doc.qt.io/qt-5/qml-qtquick-controls2-popup.html) | Base type of popup-like user interface controls | | [ProgressBar](https://doc.qt.io/qt-5/qml-qtquick-controls2-progressbar.html) | Indicates the progress of an operation | | [RadioButton](https://doc.qt.io/qt-5/qml-qtquick-controls2-radiobutton.html) | Exclusive radio button that can be toggled on or off | | [RadioDelegate](https://doc.qt.io/qt-5/qml-qtquick-controls2-radiodelegate.html) | Exclusive item delegate with a radio indicator that can be toggled on or off | | [RangeSlider](https://doc.qt.io/qt-5/qml-qtquick-controls2-rangeslider.html) | Used to select a range of values by sliding two handles along a track | | [RoundButton](https://doc.qt.io/qt-5/qml-qtquick-controls2-roundbutton.html) | A push-button control with rounded corners that can be clicked by the user | | [ScrollBar](https://doc.qt.io/qt-5/qml-qtquick-controls2-scrollbar.html) | Vertical or horizontal interactive scroll bar | | [ScrollIndicator](https://doc.qt.io/qt-5/qml-qtquick-controls2-scrollindicator.html) | Vertical or horizontal non-interactive scroll indicator | | [ScrollView](https://doc.qt.io/qt-5/qml-qtquick-controls2-scrollview.html) | Scrollable view | | [Slider](https://doc.qt.io/qt-5/qml-qtquick-controls2-slider.html) | Used to select a value by sliding a handle along a track | | [SpinBox](https://doc.qt.io/qt-5/qml-qtquick-controls2-spinbox.html) | Allows the user to select from a set of preset values | | [StackView](https://doc.qt.io/qt-5/qml-qtquick-controls2-stackview.html) | Provides a stack-based navigation model | | [SwipeDelegate](https://doc.qt.io/qt-5/qml-qtquick-controls2-swipedelegate.html) | Swipable item delegate | | [SwipeView](https://doc.qt.io/qt-5/qml-qtquick-controls2-swipeview.html) | Enables the user to navigate pages by swiping sideways | | [Switch](https://doc.qt.io/qt-5/qml-qtquick-controls2-switch.html) | Button that can be toggled on or off | | [SwitchDelegate](https://doc.qt.io/qt-5/qml-qtquick-controls2-switchdelegate.html) | Item delegate with a switch indicator that can be toggled on or off | | [TabBar](https://doc.qt.io/qt-5/qml-qtquick-controls2-tabbar.html) | Allows the user to switch between different views or subtasks | | [TabButton](https://doc.qt.io/qt-5/qml-qtquick-controls2-tabbutton.html) | Button with a look suitable for a TabBar | | [TextArea](https://doc.qt.io/qt-5/qml-qtquick-controls2-textarea.html) | Multi-line text input area | | [TextField](https://doc.qt.io/qt-5/qml-qtquick-controls2-textfield.html) | Single-line text input field | | [ToolBar](https://doc.qt.io/qt-5/qml-qtquick-controls2-toolbar.html) | Container for context-sensitive controls | | [ToolButton](https://doc.qt.io/qt-5/qml-qtquick-controls2-toolbutton.html) | Button with a look suitable for a ToolBar | | [ToolSeparator](https://doc.qt.io/qt-5/qml-qtquick-controls2-toolseparator.html) | Separates a group of items in a toolbar from adjacent items | | [ToolTip](https://doc.qt.io/qt-5/qml-qtquick-controls2-tooltip.html) | Provides tool tips for any control | | [Tumbler](https://doc.qt.io/qt-5/qml-qtquick-controls2-tumbler.html) | Spinnable wheel of items that can be selected | |

## Cách thêm resource vào Project

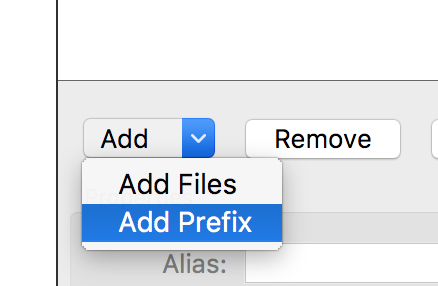
1 – Add new file or Project, chọn Qt/Qt Resource File



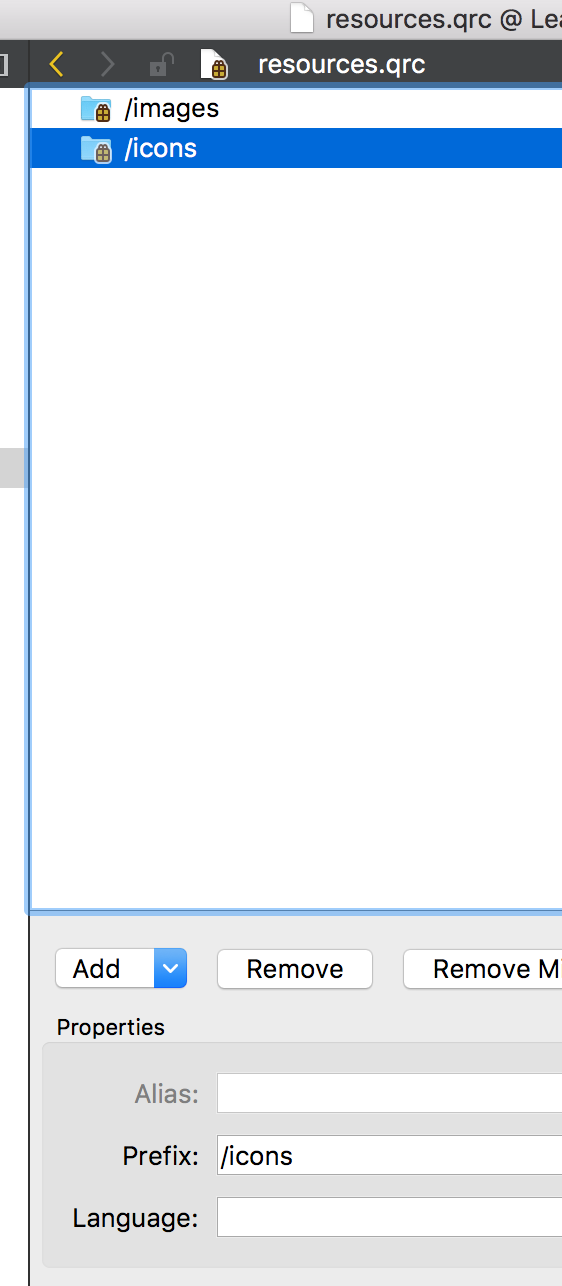
Kết quả:



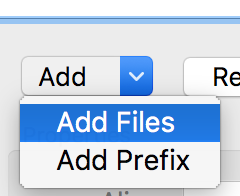
2 – Add prefix, kiểu kiểu giống như add thêm thư mục con vậy!



Kết quả:

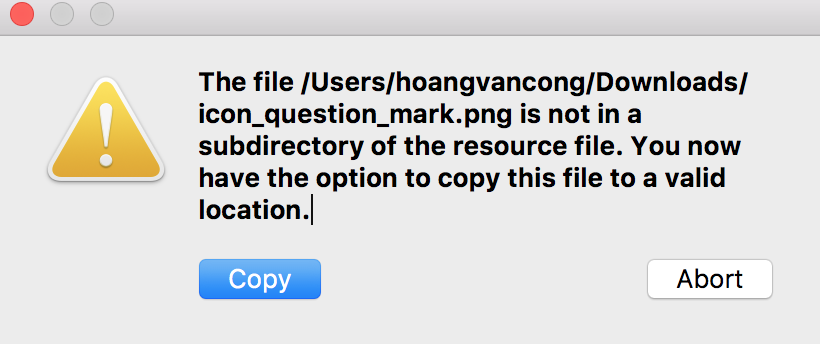


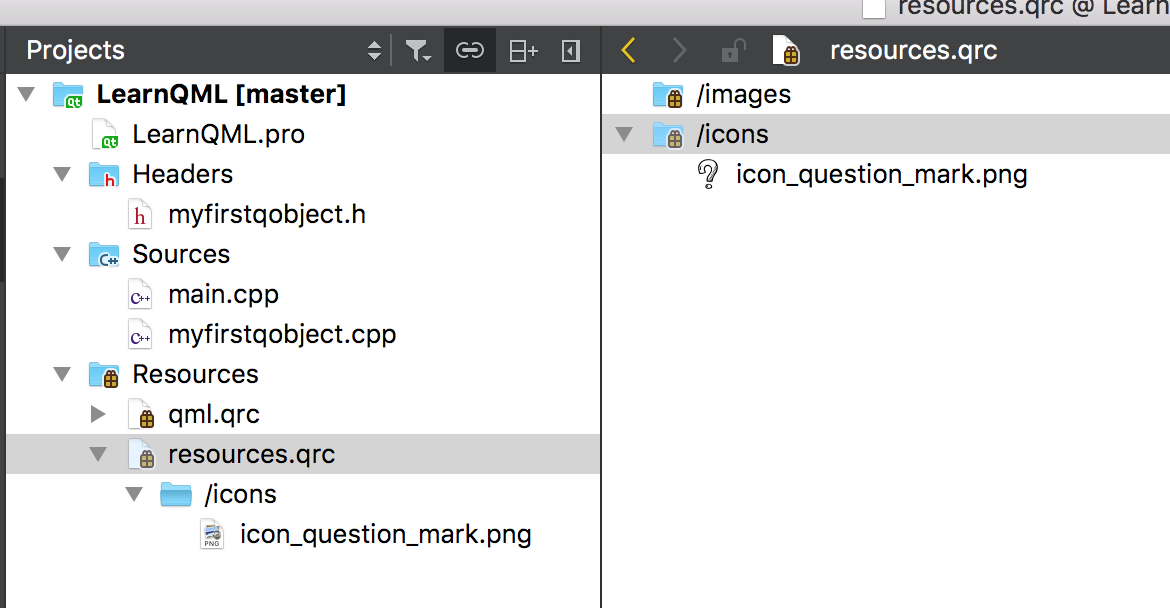
3 – Chọn 1 Prefix rồi add file vào đó



Kết quả:

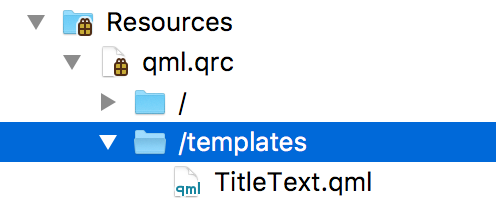
Nhớ chọn copy tới cùng location với Project



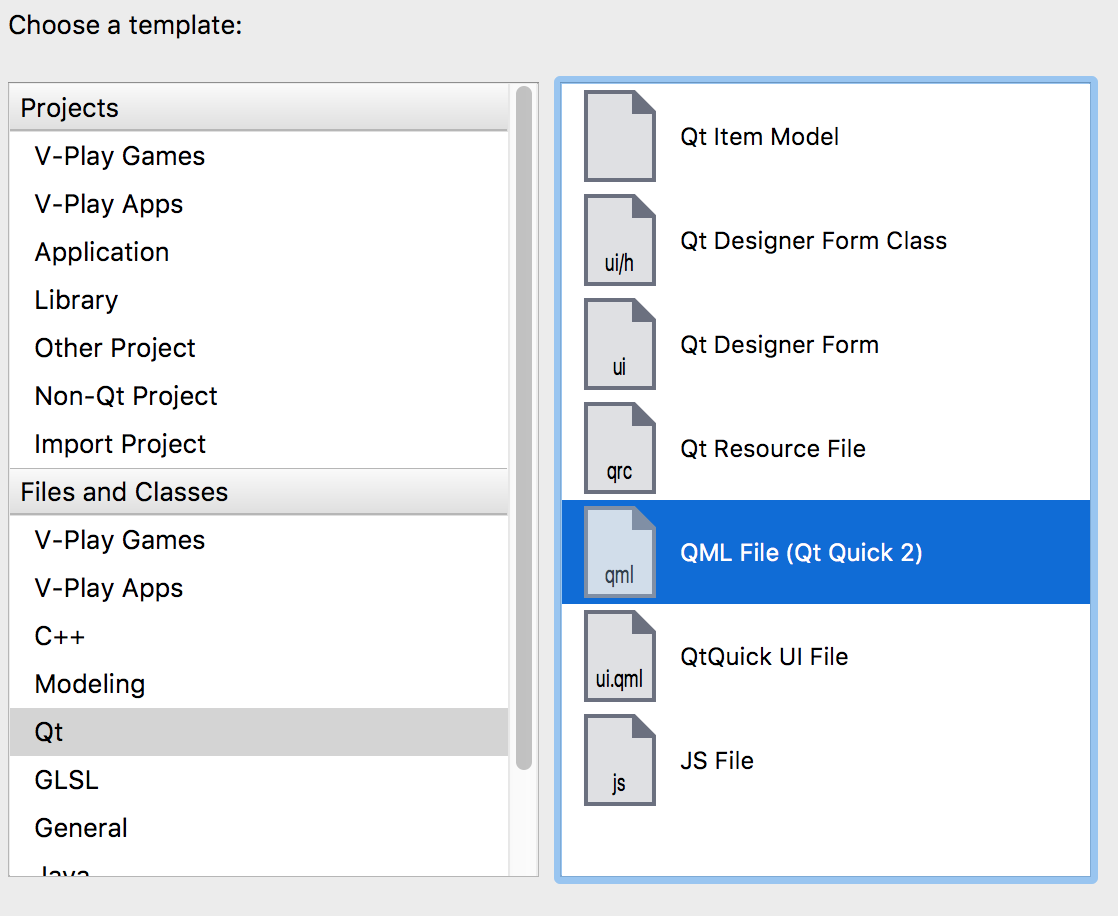


## Cách tạo control template

**B1: Tạo ra thêm 1 prefix cho dễ quản lý**



**B2: Thêm file qml vào**



**B3: Thiết kế control**

|  |
| --- |
| import QtQuick 2.0  Text {  text: "test"  font.family: "Courier"  font.pointSize: 50  } |

**B4: Sử dụng control mới như bình thường**

|  |
| --- |
| ColumnLayout{  id: *layTitle*  x: *thisWindow*/2 - *width*/2  TitleText {  id: *txtTitle\_Basic*  text: "Basic"  }  TitleText {  id: *txtTitle\_English*  text: "English"  }  TitleText {  id: *txtTitle\_Grammar*  text: "Grammar"  }  } |

* Nhưng design và build thì vẫn bình thường, chỉ lúc chạy là nó báo lỗi không định nghĩa TitleText, thế là lại phải chuyển file TitleText vào prefix mặc định về cùng chung prefix với file nó sử dụng! (Cần xem them)

## Ứng dụng nhiều page

Với ứng dụng có nhiều màn hình thì sẽ setting như sau:

* Nên có một cái header toolbar để di chuyển sang cho dễ!

## Kết hợp C++ vs QML

Để thực hiện các công việc nặng, liên quan đến business logic, không liên quan nhiều đến giao diện, ta thường kết hợp sử dụng C++ và Qml/JS tương tác qua lại với nhau để có thể hoàn thành công việc!

Từ C++ cũng có thể gọi và điều khiển các thuộc tính ờ QML, và ngược lại từ Qml cũng có thể gọi và làm việc với các thuộc tính của class C++.