Archived:Como usar QVBoxLayout e QHBoxLayout

Aquivado: Este artigo foi arquivado, pois o conteúdo não é mais considerado relevante para se criar soluções comerciais atuais. Se você achar que este artigo ainda é importante, inclua o template {{ForArchiveReview|*escreva a sua justificativa*}}.

Qt Quick should be used for all UI development on mobile devices. The approach described in this article (based on QWidget) is deprecated.

Introdução

Qt, basicamente suporta muitas layouts básicos, onde você pode organizar os seus widgets para serem mostrados na tela.

Ele suporta os seguinte tipos de layout:

1. Vertical layout
2. Horizontal layout
3. Grid layout
4. Form layout

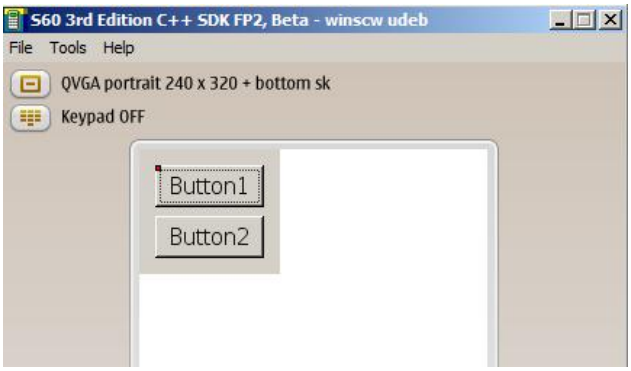
Este artigo demonstra o uso dos dois primeiro, Vertical and Horizontal

Como usar o QVBoxLayout

Código fonte

```
#include <QApplication>
#include <QPushButton>
#include <QVBoxLayout>

int main(int argc, char *argv[])
{
QApplication app(argc, argv);
QWidget* win = new QWidget();
QVBoxLayout* layout = new QVBoxLayout(win);
QPushButton* but1 = new QPushButton("Horizontal");
but1->resize(70,20);
layout->addWidget(but1);
QPushButton* but2 = new QPushButton("Vertical");
but2->resize(70,20);
layout->addWidget(but2);

win->show();

return app.exec();
}
```

Captura de tela



Como usar o QHBoxLayout

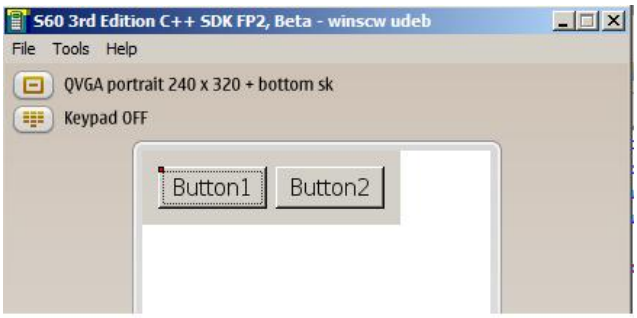
Código fonte

```
#include <QApplication>
#include <QPushButton>
#include <QHBoxLayout>

int main(int argc, char *argv[])
{
QApplication app(argc, argv);
QWidget* win = new QWidget();
```


(C) Copyright Nokia 2013. All rights reserved.

```
QHBoxLayout* layout = new QHBoxLayout(win);
QPushButton* but1 = new QPushButton("Horizontal");
but1->resize(70,20);
layout->addWidget(but1);
QPushButton* but2 = new QPushButton("Vertical");
but2->resize(70,20);
layout->addWidget(but2);
win->show();

return app.exec();
}
```

Captura de tela



(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.