



# MeeGo API — “Visual services”

*Васильев Валентин*



# MeeGo API — Visual Services

## Содержание лекции

---

1. 2D графика: QPainter, SVG.
2. 3D графика: OpenGL ES.
3. Интернационализация: I18n, QtLinguist.



## MeeGo API — Visual Services 2D графика/QPainter

---

- QPainter — универсальный класс, включающий высокооптимизированные функции для рисования практически любой графики;
- QPaintDevice — абстрактное устройство вывода;
- QPaintEngine — интерфейс с помощью которого осуществляется вывод (скрыт);



- QWidget, QImage, QPixmap, QGLPixelBuffer, QPicture и QPrinter в качестве устройств вывода.



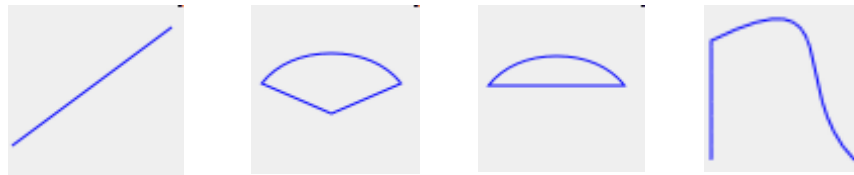
## MeeGo API — Visual Services

### 2D графика/QPainter/Простейший пример

---

```
void SimpleExampleWidget::paintEvent()  
{  
    QPainter paint(this);  
    paint.setPen(Qt::blue);  
    paint.drawText(rect(), Qt::AlignCenter, tr("The Text"));  
}
```

- ▶ Примитивы рисования: **drawPoint()**, **drawPoints()**, **drawLine()**, **drawRect()**, **drawRoundRect()**, **drawEllipse()**, **drawArc()**, **drawPie()**, **drawChord()**, **drawLineSegments()**, **drawPolyline()**, **drawPolygon()**, **drawConvexPolygon()**, **drawCubicBezier()** и т.д.
- ▶ Рисования на пиксельных картах/рисунках: **drawPixmap()**, **drawImage()**, **drawTiledPixmap()**.
- ▶ и так далее...





## MeeGo API — Visual Services 2D графика/QPainter/Настройки

---

- **font()** — текущие установки шрифта;
- **brush()** — текущие установки кисти; цвет или образец используемый для заливки, например, круги;
- **pen()** — текущие настройки пера; цвет или пунктир, используемый для рисования линий и границ;
- **backgroundMode()** — является ли подложка прозрачной;

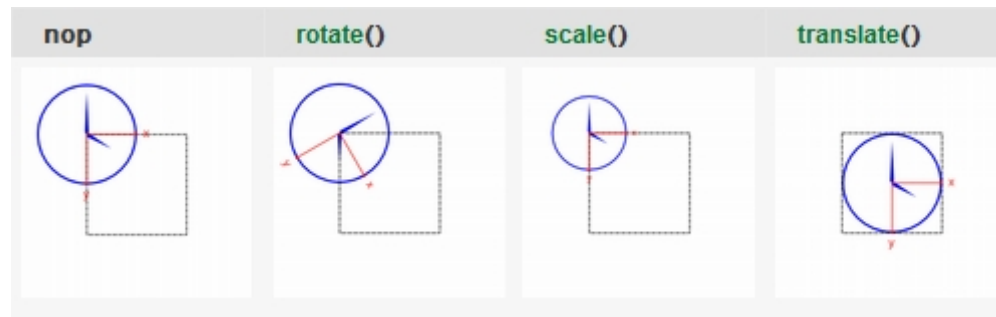


## MeeGo API — Visual Services 2D графика/QPainter/Настройки

---

- **brushOrigin()** — оригинальный узор кисти, обычно оригинальный фон виджета;
- **hasClipping()** — указывает, осуществляет ли класс обрезание по каким-либо границам. Если обрезание осуществляется, то по региону, указанному в методе **clipRegion()**;
- **save()** — сохраняет настройки во внутреннем стеке;
- **restore()** — восстанавливает их.

- Простейшие аффинные преобразования:  
`scale()`, `rotate()`, `translate()`, `shear()`:



- Работа с видимой областью:  
`setViewport()`, `setWindow()`;
- Получение текущей матрицы преобразований:  
`matrice()`, `setMatrice()`.





## MeeGo API — Visual Services 2D графика/QPainter/QPainterPath

---

```
QPainterPath path;  
path.addRect(20, 20, 60, 60);  
path.moveTo(0, 0);  
path.cubicTo(99, 0, 50, 50, 99, 99);  
path.cubicTo(0, 99, 50, 50, 0, 0);  
  
QPainter painter(this);  
painter.fillRect(0, 0, 100, 100, Qt::white);  
painter.setPen(QPen(QColor(79, 106, 25),  
                    1,  
                    Qt::SolidLine,  
                    Qt::FlatCap,  
                    Qt::MiterJoin));  
painter.setBrush(QColor(122, 163, 39));  
painter.drawPath(path);
```



## MeeGo API — Visual Services 2D графика/SVG

---

- SVG (от англ. Scalable Vector Graphics — масштабируемая векторная графика) — язык разметки масштабируемой векторной графики;
- подмножество XML;
- предназначен для описания двумерной векторной и смешанной векторно/растровой графики;
- поддерживает анимированную и интерактивную графику;
- стандарт открытый, является рекомендацией консорциума W3C;
- разрабатывается с 1999 года;
- актуальная версия — 1.1 (2001-го года).



## MeeGo API — Visual Services 2D графика/SVG/Основные классы

---

- **QSvgGenerator** — класс, осуществляющий рисование;
- **QSvgRenderer** — отображение содержимого SVG файлов;
- **QSvgWidget** — соответствующий виджет, через который отображаются SVG данные.



## MeeGo API — Visual Services 2D графика/SVG/QSvgWidget

---

- `widget = QSvgWidget("filename.svg");`

или

```
widget = QSvgWidget();  
widget.load("filename.svg");
```

или

```
widget = QSvgWidget(qbyte);
```

где `qbyte` — экземпляр класса `QByteArray`, содержащий сериализованное представление XML SVG файла.

- `QSvgRenderer` может использоваться для отрисовки SVG графики на любых экземплярах класса `QPainter` (например, `QWidget`, `QImage` и `QGLWidget`).



## МееGo API — Visual Services 2D графика/SVG/QSvgGenerator

---

- QSvgGenerator generator;

```
generator.setFileName(path);
generator.setSize(QSize(200, 200));
generator.setViewBox(QRect(0, 0, 200, 200));
generator.setTitle(tr("SVG Generator Example
Drawing"));

generator.setDescription(
    tr("An SVG drawing created by the SVG
Generator "
        "Example provided with Qt."))
);
```
- QPainter painter;

```
painter.begin(&generator);
painter.fillRect(QRect(0, 0, 200, 200), Qt::darkBlue);
painter.translate(145, 10);
painter.setBrush(Qt::white);
painter.drawPath(moon);
painter.translate(-145, -10);
painter.end();
```



## MeeGo API — Visual Services 3D графика/OpenGL ES

---



- **OpenGL ES** (OpenGL for Embedded Systems) — OpenGL для встраиваемых систем;
- подмножество графического интерфейса OpenGL для встраиваемых систем;
- определяется и продвигается консорциумом Khronos Group;

**OpenGL ES 1.x** (текущая реализация по версии 1.5 стандарта):

- ограниченный набор основных вариантов для рисования и освещение объектов (фиксированный графический конвейер);

**OpenGL ES 2.0:**

- повсеместно используются шейдеры;
- OpenGL ES Shading Language (GLSL ES);
- вывод даже простейшего треугольника требует знаний основ 3d графики.



## MeeGo API — Visual Services Интернационализация

---

«Интернационализация — технологические приёмы разработки, упрощающие адаптацию продукта (такого как программное или аппаратное обеспечение) к языковым и культурным особенностям региона (регионов), отличного от того, в котором разрабатывался продукт.»

*([ru.wikipedia.org/wiki/Интернационализация](http://ru.wikipedia.org/wiki/Интернационализация))*



## MeeGo API — Visual Services Интернационализация

---

- В английском языке для слова “internationalization” принято сокращение “i18n”. При этом число 18 означает количество пропущенных между «i» и «n» букв;
- интернационализация — это адаптация продукта для потенциального использования практически в любом месте на начальных этапах разработки;
- чтобы сделать приложение написанное в России полезным для японских пользователей, требуется чтобы программное обеспечение не только отображало информацию на Японском, но и использовало различные методы ввода, кодировку символов, соглашения о представлении услуг, методы отображения дат и т.д.





## MeeGo API — Visual Services Интернационализация

---

- **Специальное поведение при разрыве строки.** Некоторые из азиатских языков пишутся без пробелов между словами;
- **Двунаправленные письменные формы.** В Арабском и Иврите пишутся справа налево, за исключением номеров и встроенного английского текста (который пишется слева направо);
- **Номера интервалов или диакритических знаков (акцентов или умляутов в европейских языках);**
- **Лигатуры.** В особых контекстах, некоторые пары символов необходимо заменять комбинированным символом.



## MeeGo API — Visual Services Интернационализация/Схема

---

- Все целевые литерные тексты в коде проекта маркируются вызовами функции **tr()**, или макроса **QT\_TR\_NOOP()**;
- выполняется трансляция утилитой **lupdate**, которая «выдирает» маркированные тексты;
- тексты переводятся с использованием утилиты Qt Linguist.



## MeeGo API — Visual Services

### Интернационализация/tr(), QT\_TR\_NOOP()

---

```
LoginWidget::LoginWidget()
{
    QLabel *label = new QLabel(tr("Password:"));
    ...
}
```

Пример подходит для 99% случаев использования.



## MeeGo API — Visual Services

### Интернационализация/tr(), QT\_TR\_NOOP()

---

Пример, который будет работать:

```
QPushButton *ok = new QPushButton( tr("OK"), this );
```

Который не будет работать:

```
QString str = "OK";  
QPushButton *ok = new QPushButton( tr(str), this );
```



## MeeGo API — Visual Services

### Интернационализация/tr(), QT\_TR\_NOOP()

---

Использование подстановки:

```
tr( QString("Cannot open %1").arg(fileName) );
```

или

```
tr( "Cannot open %1" ).arg( fileName );
```



## MeeGo API — Visual Services Интернационализация/Трансляция

---

- Запустить утилиту **lupdate**, которая возьмёт переводимые тексты из исходного C++ кода и поместит их в **.TS** файл;
- отредактировать получившийся TS файл для того, чтобы перевести текст используя утилиту **Qt Linguist**;
- вызвать утилиту **lrelease** для того чтобы получить легковесный файл с сообщениями (**.QM** файл) из **.TS**, пригодный только для конечного использования;
- TS файлы можно рассматривать как исходные файлы, а QM — как объектные.



## MeeGo API — Visual Services

### Интернационализация/Проектный файл

---

```
HEADERS = funnydialog.h \  
        wackywidget.h  
SOURCES = funnydialog.cpp \  
        main.cpp \  
        wackywidget.cpp  
FORMS = fancybox.ui  
TRANSLATIONS = superapp_dk.ts \  
              superapp_fi.ts \  
              superapp_no.ts \  
              superapp_se.ts
```



## MeeGo API — Visual Services Интернационализация/Main

---

```
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    QTranslator qtTranslator;
    qtTranslator.load("qt_" + QLocale::system().name(),
        QLibraryInfo::location(QLibraryInfo::TranslationsPath));
    app.installTranslator(&qtTranslator);
    QTranslator myappTranslator;
    myappTranslator.load("myapp_" + QLocale::system().name());
    app.installTranslator(&myappTranslator);
    ...
    return app.exec();
}
```





## MeeGo API — Visual Services

### Интернационализация/Динамическая трансляция

---

- Событие **LanguageChange** выбрасывается при вызове метода **QCoreApplication::installTranslator()**

- Обработчик события:

```
void MyWidget::changeEvent(QEvent *event)
{
    if (e->type() == QEvent::LanguageChange) {
        titleLabel->setText(tr("Document Title"));
        ...
        okPushButton->setText(tr("&OK"));
    } else
        QWidget::changeEvent(event);
}
```



## MeeGo API — Visual Services

### Интернационализация/Локализация

---

- «Локализация — это процесс адаптации к имеющимся локальным предпочтениям. Например, представления даты и времени в официальном для данного региона формате.»
- Пример реализации:

```
void Clock::setTime(const QTime &time)
{
    if (tr("AMPM") == "AMPM") {
        // 12-х часовое время
    } else {
        // 24-х часовое время
    }
}
```



# Вопросы?

*Валентин Васильев (gnome@bk.ru)*