



DEVELOPER
DAYS **2014**
NORTH AMERICA



Deep Dive into Qt Quick and Qt Quick Controls

A Live Coding Session

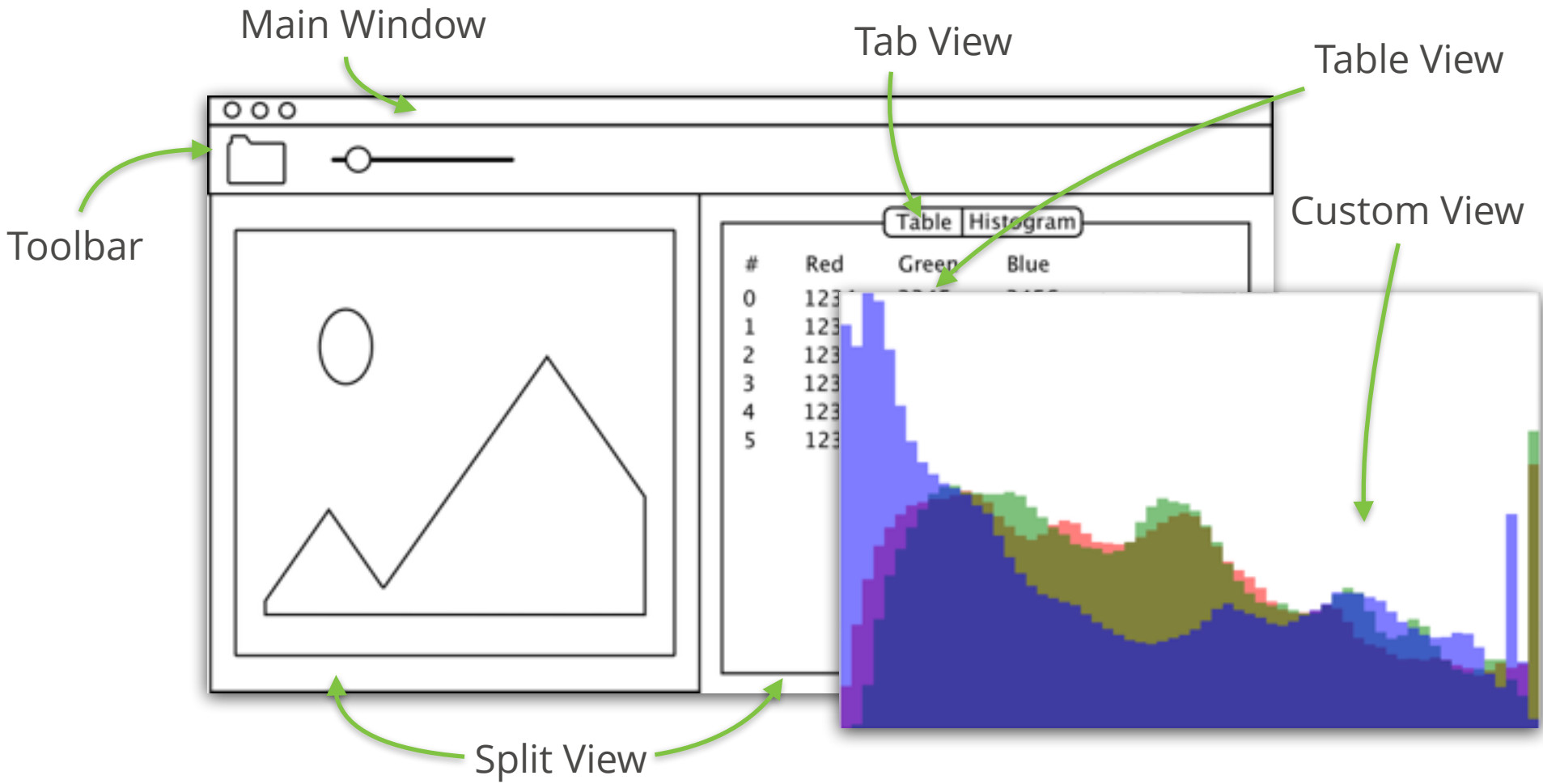
Gabriel de Dietrich, The Qt Company

Who Am I?

- Working on Qt since 2009
- UI team member and pixel perfectionist
- Qt Quick Controls and Qt Mac port
- Living and working in Oslo, Norway
- ½ French + ½ Spanish = Me

Agenda

- Implement an UI with Qt Quick Controls
- Interface QML with C++
- Create a Qt Quick compatible model



```
class RgbHistogram : public QObject
{
    Q_OBJECT

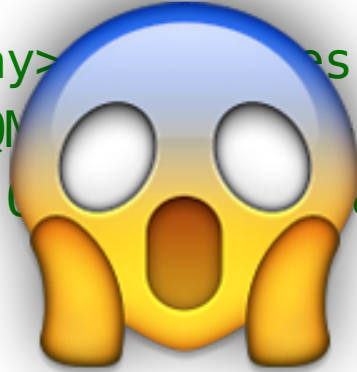
    Q_PROPERTY(int binCount READ binCount WRITE setBinCount NOTIFY binCountChanged)
    ...

public slots:
    void compute();

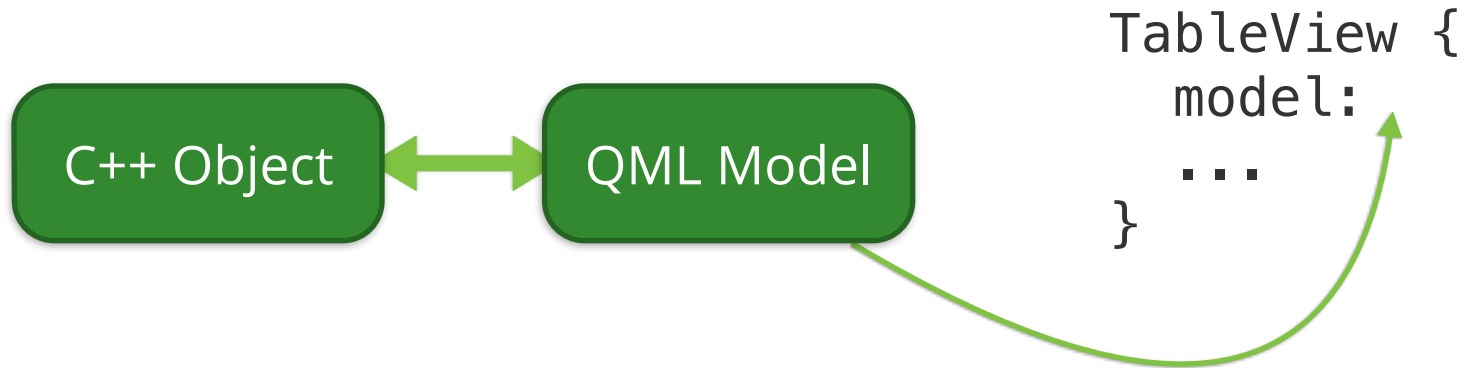
signals:
    void histogramUpdated();
    ...

public:
    Q_INVOKABLE int redCount(int bin) const;
    ...
};
```

```
class RgbHistogram : public QObject QAbstractListModel
{
    ...
};
public:
    QHash<int, QByteArray> keys() const;
    int rowCount(const QModelIndex & parent = QModelIndex()) const;
    QVariant data(const QModelIndex & index, int role) const;
};
```



Proxy Pattern and Qt Quick Model



```
class RgbHistogram : public QObject
{
    ...
};
```

```
class RgbHistogramModel : public QAbstractListModel
{
    Q_PROPERTY(RgbHistogram *histogram READ histogram ...);

public:
    QHash<int, QByteArray> roleNames() const;
    int rowCount(const QModelIndex &) const;
    QVariant data(const QModelIndex &, int role) const;
};
```




Thank you!

www.qt.io

gabriel.dedietrich@theqtcompany.com

Application code available at github.com/drgvond/RGB-Analyzer