Atelier QML

Présentation

Atelier QML

Présentation

QML, c’est quoi ?

Historique

QML : Qt Modeling Language

Historique

QML : Qt Modeling Language

1er sortie : 2009 Qt4.X

Historique

QML : Qt Modeling Language

1er sortie : 2009 Qt4.X

QtQuick 1 (Qt4)

Historique

QML : Qt Modeling Language

1er sortie : 2009 Qt4.X

QtQuick 1 (Qt4)

QtQuick 2 (Qt5)

Pré-requis

QtCreator ~v4.3

Pré-requis

QtCreator ~v4.3

Qt5.9

Pré-requis

QtCreator ~v4.3

Qt5.9

Les slides: https://github.com/obiwankennedy/Learn-Qt.git

Pré-requis

QtCreator ~v4.3

Qt5.9

Les slides: https://github.com/obiwankennedy/Learn-Qt.git

Les exercices: https://github.com/obiwankennedy/TrainingExercices.git

Exemple d’utilisation du QML

Cette présentation

Exemple d'utilisation du QML

Cette présentation

Les fiches de perso de Rolisteam

Exemple d’utilisation du QML

Cette présentation

Les fiches de perso de Rolisteam

Démos Qt

Exemple d’utilisation du QML

Cette présentation

Les fiches de perso de Rolisteam

Démos Qt

Premier programme

Hello world!

Premier programme

```
1:  import QtQuick 2.0
2:  Item {
3:      Text{
4:          id: text
5:          text:qsTr("Hello World!!")"
6:      }
7:  }
8:
```

Premier programme

Hello world!

$ qml helloworld.qml

QtCreator

Visite de QtCreator

QtCreator

Visite de QtCreator

Edition d’un fichier QML

QtCreator

Visite de QtCreator

Edition d’un fichier QML

Intégration de l’aide (F1)

QtCreator

Visite de QtCreator

Edition d’un fichier QML

Intégration de l’aide (F1)

Designer (ctrl+3)

QtCreator

Visite de QtCreator

Edition d'un fichier QML

Intégration de l'aide (F1)

Designer (ctrl+3)

Créer un projet

QtCreator

- Visite de QtCreator
- Edition d’un fichier QML
- Intégration de l’aide (F1)
- Designer (ctrl+3)
- Créer un projet
- Visite du projet

Les éléments de base

Item

Les éléments de base

Item

Text

Les éléments de base

- Item
- Text
- Image

Les éléments de base

- Item
- Text
- Image
- Rectangle

Les éléments de base

- Item
- Text
- Image
- Rectangle
- TextEdit / TextInput

Les éléments de base

- Item
- Text
- Image
- Rectangle
- TextEdit / TextInput
- Timer

Les éléments de base

- Item
- Text
- Image
- Rectangle
- TextEdit / TextInput
- Timer
- MouseArea

Les éléments de base

- Item
- Text
- Image
- Rectangle
- TextEdit / TextInput
- Timer
- MouseArea
- Flickable

Les éléments de base

Item

Text

Image

Rectangle

TextEdit / TextInput

Timer

MouseArea

Flickable

Exercice: reprendre helloworld.qml pour ajouter un rectangle et afficher une trace toutes les secondes (Timer)

Le positionnement

Positionnement en Z

4 types de positionnements

Le positionnement

Positionnement en Z

4 types de positionnements

Relatif au parent

Le positionnement

```
1:  Item {
2:      id: root
3:      width: 1000
4:      height: 800
5:      // relatif au parent
6:      Rectangle {
7:          id: rect
8:          color: "red"
9:          width:  200
10:         height: 200
11:         x:0
12:         y:0
13:         MouseArea {
14:             id: mouse
15:             width: 200
16:             height: 200
17:             y: 100
```

Le positionnement

```
10:         height: 200
11:         x:0
12:         y:0
13:         MouseArea {
14:             id: mouse
15:             width: 200
16:             height: 200
17:             x: 100
18:             y: 100
19:             onClicked:console.log("click on rectangle")
20:         }
21:     }
22:     Text {
23:         id: hello
24:         text: "Hello World!"
25:     }
26: }
```

Le positionnement

Positionnement en Z

4 types de positionnements

Relatif au parent

Ancre

Le positionnement

```
1:
2:    Image {
3:        id: img
4:        source: "bouton.png"
5:        MouseArea {
6:            anchors.fill: parent
7:        }
8:    }
9:    ////
10:   Image {
11:       id: img
12:       source: "logo.png"
13:       x: 200
14:       y: 0
15:   }
16:   Text {
17:       anchors.bottom: img.bottom
```

Le positionnement

```
 7:     }
 8:   }
 9:   ////
10:   Image {
11:     id: img
12:     source: "logo.png"
13:     x: 200
14:     y: 0
15:   }
16:   Text {
17:       anchors.bottom: img.bottom
18:       anchors.left: img.left
19:       width: 250
20:       height: 20
21:       text: "Rolisteam"
22:   }
23:
```

Le positionnement

Positionnement en Z

4 types de positionnements

Relatif au parent

Ancre

Positionner: Flow, Column, Grid, Row

Le positionnement

```
1:    Grid {
2:        x: 535
3:        y: 3
4:        columns: 3
5:        spacing: 2
6:        Rectangle { color: "red"; width: 50; height: 50 }
7:        Rectangle { color: "green"; width: 20; height: 50 }
8:        Rectangle { color: "blue"; width: 50; height: 20 }
9:        Rectangle { color: "cyan"; width: 50; height: 50 }
10:       Rectangle { color: "magenta"; width: 10; height: 10 }
11:   }
12:
13:
```

Le positionnement

Positionnement en Z

4 types de positionnements

Relatif au parent

Ancre

Positionner: Flow, Column, Grid, Row

Layout: ColumnLayout, GridLayout, RowLayout, StackLayout

Le positionnement

```
1:  GridLayout {
2:      id: grid
3:      x: 373
4:      y: 0
5:      columns: 3
6:      Rectangle { color: "red"; width: 50; height: 50 }
7:      Rectangle { color: "green"; width: 20; height: 50 }
8:      Rectangle { color: "blue"; width: 50; height: 20 }
9:      Rectangle { color: "cyan"; width: 50; height: 50 }
10:     Rectangle { color: "magenta"; width: 10; height: 10 }
11:
12:   }
13:
14:
```

Le positionnement

Positionnement en Z

4 types de positionnements

Relatif au parent

Ancre

Positionner: Flow, Column, Grid, Row

Layout: ColumnLayout, GridLayout, RowLayout, StackLayout

Exemple: 03_positionnement/test.qml

Properties et signaux

Les propriétés

Properties et signaux

Les propriétés

Les importantes

Properties et signaux

Les propriétés

Les importantes

Propagation (Opacité/visibilité)

Properties et signaux

```
1:  Rectangle {
2:      x: 0
3:      width: 500
4:      height: 500
5:      opacity: 0.4
6:      color: "blue"
7:      Rectangle {
8:          width: 250
9:          height: 250
10:         color: "yellow"
11:     }
12: }
13:
14: Rectangle {
15:     x: 500
16:     width: 500
17:     height: 500
```

Properties et signaux

```
 9:             height: 250
10:             color: "yellow"
11:         }
12:     }
13:
14:     Rectangle {
15:         x: 500
16:         width: 500
17:         height: 500
18:         visible: false
19:         color: "green"
20:         Rectangle {
21:             width: 250
22:             height: 250
23:             color: "yellow"
24:         }
25:     }
```

Properties et signaux

Les propriétés

Les importantes

Propagation (Opacité/visibilité)

notification

Properties et signaux

Les propriétés

Les importantes

Propagation (Opacité/visibilité)

notification

acces

Properties et signaux

Les propriétés

Les importantes

Propagation (Opacité/visibilité)

notification

acces

Ajouter une propriété

Properties et signaux

```
1:    property int marge: 20
2:    property string text: ""
```

Properties et signaux

Les propriétés

Les importantes

Propagation (Opacité/visibilité)

notification

acces

Ajouter une propriété

Ajouter un alias

Properties et signaux

```
1:    property alias text: titre.text
2:        Text{
3:            id: titre
4:            text: "chocolat"
5:    }
6:
```

Properties et signaux

Les propriétés

Les importantes

Propagation (Opacité/visibilité)

notification

acces

Ajouter une propriété

Ajouter un alias

Exercice: Créer un item contenant deux rectangles clickables,
les boutons changent la valeur d'une propriété de l'item
et cela déclenche un affichage en console.

Binding

Associé une propriété à une autre

Binding

Associé une propriété à une autre

javascript

Binding

Associé une propriété à une autre

javascript

Exercise: afficher un carré

Méthode de travail

Du design au QML

Méthode de travail

Du design au QML

Du QML au QML propre

Méthode de travail

Du design au QML

Du QML au QML propre

Autres méthodes

Les animations

Animer une propriété

Les animations

Animer une propriété

Type: PropertyAnimation, NumberAnimation,RotationAnimation, ColorAnir

Les animations

Animer une propriété

Type: PropertyAnimation, NumberAnimation,RotationAnimation, ColorAnir

Animation Directe

Les animations

```
1:    MouseArea {
2:          anchors.fill: parent
3:          onClicked: {
4:              animateColor.start()
5:          }
6:      }
7:
8:      PropertyAnimation {id: animateColor; target: my_rectangle; properties: "color"; to: "green"; duration: 100}
```

Les animations

Animer une propriété

Type: PropertyAnimation, NumberAnimation,RotationAnimation, ColorAnir

Animation Directe

Corportementale

Les animations

```
1:    Behavior on opacity {
2:        NumberAnimation { duration: 1000 }
3:    }
```

Les animations

Animer une propriété

Type: PropertyAnimation, NumberAnimation,RotationAnimation, ColorAnir

Animation Directe

Corportementale

Transition

Les animations

```
1:    transitions: [
2:            Transition {
3:                NumberAnimation { properties: "x,y" }
4:            }
5:        ]
6:
```

Les animations

Animer une propriété

Type: PropertyAnimation, NumberAnimation,RotationAnimation, ColorAnir

Animation Directe

Corportementale

Transition

Animations Séquentielles ou en Parallèles

Les animations

```
1:    SequentialAnimation {
2:          id: anim
3:          NumberAnimation { target: bullet; property: "x"; from: 0; to: 200; duration: 500 }
4:          NumberAnimation { target: enemy; property: "opacity"; from: 1.0; to: 0; duration: 500 }
5:      }
6:
```

Les animations

Animer une propriété

Type: PropertyAnimation, NumberAnimation,RotationAnimation, ColorAnir

Animation Directe

Corportementale

Transition

Animations Séquentielles ou en Parallèles

Exercice: faire bouger les carrés vers dans le sens horaires (sur un seul axe)
En parallèle puis en sequentiel

Etats et transitions

Explication

Etats et transitions

Explication

Analyse de la syntaxe

Etats et transitions

```
states: [
        State {
            name: "clicked"
            PropertyChanges { target: myRect; color: "red" }
        },
        State {
            name: "disabled"
            PropertyChanges { target: myRect; color: "grey" }
        }
    ]
```

Etats et transitions

Explication

Analyse de la syntaxe

Dans le designer

Etats et transitions

Explication

Analyse de la syntaxe

Dans le designer

Avantages ?

Etats et transitions

Explication

Analyse de la syntaxe

Dans le designer

Avantages ?

Transition

Exercice: Réaliser un feu de traffic: Rouge puis Rouge et Orange puis vers, puis Orange puis Rouge

Etats et transitions

Explication

Analyse de la syntaxe

Dans le designer

Avantages ?

Transition

Exercice: Réaliser un feu de traffic: Rouge puis Rouge et Orange puis vers, puis Orange puis Rouge

Créer son composant?

Réfléchir en boite noire

Créer son composant?

Réfléchir en boite noire

Définir les propriétés de son item

Créer son composant?

```
1:  Item {
2:  id: root
3:  property int idState: 0
4:  property string description: 0
5:  property alias  title: titlebox.text
6:  […]
7:
```

Créer son composant?

Réfléchir en boite noire

Définir les propriétés de son item

Séparer les fichiers

Créer son composant?

```
1:
2:    //Square.qml
3:    Item {
4:        id: root
5:        property int side: 0
6:        property Color color: rect.color
7:        Rectangle {
8:            id: rect
9:            width: side
10:           height: side
11:       }
12:   }
13:
```

Créer son composant?

Réfléchir en boite noire

Définir les propriétés de son item

Séparer les fichiers

Utiliser

Créer son composant?

```
1:
2:    Square {
3:        id: square1
4:        side: 200
5:        color: "blue"
6:    }
7:
```

Créer son composant?

Réfléchir en boite noire

Définir les propriétés de son item

Séparer les fichiers

Utiliser

Gérer plusieurs composants perso

Créer son composant?

```
1:  //qmldir
2:  module Rcse
3:
4:  TextInputField 1.0 TextInputField.qml
5:  SelectField 1.0 SelectField.qml
6:  TextAreaField 1.0 TextAreaField.qml
7:  TextFieldField 1.0 TextFieldField.qml
8:  CheckBoxField 1.0 CheckBoxField.qml
9:  DiceButton 1.0 DiceButton.qml
10: ImageField 1.0 ImageField.qml
11:
12:
13: --------------------
14:
15: import "path/to/directory/"
16:
```

Créer son composant?

Réfléchir en boite noire

Définir les propriétés de son item

Séparer les fichiers

Utiliser

Gérer plusieurs composants perso

Exercice: faire un composant lightbox-like

Mauvais exemple

Un bon code ?

Mauvais exemple

Un bon code ?

Une application simple

Button has not been clicked

Click me!

I have been clicked

Click me!

Mauvais exemple

```
1:  Column {
2:      Text{
3:          id: massageBox
4:          text:"Button has not been clicked"
5:      }
6:      Rectangle {
7:          id: button
8:          implicitWidth: buttonText.implicitWidth + 20
9:          implicitHeight: buttonText.implicitHeight + 10
10:         radius: 5
11:         color: "lightblue"
12:         MouseArea {
13:             id: mousearea
14:             anchors.fill: parent
15:             onClicked:  massageBox.text="I have been clicked"
16:
17:         }
```

Mauvais exemple

Un bon code ?

Une application simple

Séparer le bouton dans un composant

Mauvais exemple

```
1:    Column {
2:            Text{
3:               id: massageBox
4:               text:"Button has not been clicked"
5:            }
6:
7:            MyButton {
8:               id: button
9:            }
10:   }
```

Mauvais exemple

```
1:  Rectangle {
2:      id: button
3:      implicitWidth: buttonText.implicitWidth + 20
4:      implicitHeight: buttonText.implicitHeight + 10
5:      radius: 5
6:      color: "lightblue"
7:      Text {
8:          id:buttonText
9:          anchors.fill: parent
10:         text: "Click me!"
11:         color: "white"
12:     }
13:     MouseArea {
14:         id: mousearea
15:         anchors.fill: parent
16:         onClicked:  massageBox.text="I have been clicked"
```

Mauvais exemple

Un bon code ?

Une application simple

Séparer le bouton dans un composant

Corriger la faute d’orthographe

Mauvais exemple

```
1:    Column {
2:            Text{
3:               id: messageBox
4:               text:"Button has not been clicked"
5:            }
6:
7:            MyButton {
8:               id: button
9:            }
10:   }
```

Mauvais exemple

Un bon code ?

Une application simple

Séparer le bouton dans un composant

Corriger la faute d’orthographe

Cassé! Quel modification à causer le problème ?

Mauvais exemple

Un bon code ?

Une application simple

Séparer le bouton dans un composant

Corriger la faute d’orthographe

Cassé! Quel modification à causer le problème ?

Pourquoi c’est un problème ?

Mauvais exemple

- Inter-dépendences entre les composants/fichiers
- Impossible de trouver toutes les instances avec un IDE
- Modifier tous les composants signifie avoir tous les composants en tê
- Cela ralentit le dev et introduit des bugs vicieux

Mauvais exemple

Un bon code ?

Une application simple

Séparer le bouton dans un composant

Corriger la faute d’orthographe

Cassé! Quel modification à causer le problème ?

Pourquoi c’est un problème ?

Bonnes partiques

Mauvais exemple

```
1:    Rectangle {
2:            id: root
3:            signal clicked()
4:            Text {
5:                id:buttonText
6:            }
7:            MouseArea {
8:                id: mousearea
9:                anchors.fill: parent
10:               onClicked:  root.clicked()
11:           }
12:       }
```

Mauvais exemple

```
1:    Rectangle {
2:            id: root
3:            signal clicked()
4:            Text {
5:                id:buttonText
6:            }
7:            MouseArea {
8:                id: mousearea
9:                anchors.fill: parent
10:               onClicked:  root.clicked()
11:           }
12:       }
```

Interaction

Souris

Interaction

Souris

Gesture: PinchArea, MultiPointTouchArea

Interaction

Souris

Gesture: PinchArea, MultiPointTouchArea

Clavier

Interaction

```
1:    Keys.onPressed: {
2:        if (event.key == Qt.Key_Left) {
3:            console.log("move left");
4:            event.accepted = true;
5:        }
6:    }
```

Interaction

Souris

Gesture: PinchArea, MultiPointTouchArea

Clavier

autres: C++

Interaction

Souris

Gesture: PinchArea, MultiPointTouchArea

Clavier

autres: C++

Exercices: Bouger l'image à la souris,
agrandir/réduire par un pinch et molette, Rotation par clavier

Ajouter du code

Le code javascript

Ajouter du code

Le code javascript

Binding

Ajouter du code

```
1:  import QtQuick 2.0
2:
3:  Rectangle {
4:      id: colorbutton
5:      width: 200; height: 80;
6:
7:      color: mousearea.pressed ? "steelblue" : "lightsteelblue"
8:
9:      MouseArea {
10:         id: mousearea
11:         anchors.fill: parent
12:     }
13: }
```

Ajouter du code

Le code javascript

Binding

Répondre à un signal

Ajouter du code

```
1:  import QtQuick 2.0
2:
3:  Rectangle {
4:      id: button
5:      width: 200; height: 80;
6:      color: "lightsteelblue"
7:
8:      MouseArea {
9:          id: mousearea
10:         anchors.fill: parent
11:
12:         onPressed: {
13:             // arbitrary JavaScript expression
14:             button.text = "I am Pressed!"
15:             button.color = "steelblue"
16:         }
17:         onReleased: {
```

Ajouter du code

```
11:
12:         onPressed: {
13:             // arbitrary JavaScript expression
14:             button.text = "I am Pressed!"
15:             button.color = "steelblue"
16:         }
17:         onReleased: {
18:             // arbitrary JavaScript expression
19:             label.text = "Click Me!"
20:         }
21:     }
22:     Text {
23:         id: label
24:         anchors.centerIn: parent
25:         text: "Press Me!"
26:     }
27: }
```

Ajouter du code

Le code javascript

Binding

Répondre à un signal

Fonctions

Ajouter du code

```
1:  import QtQuick 2.0
2:  Item {
3:      function factorial(a) {
4:          a = parseInt(a);
5:          if (a <= 0)
6:              return 1;
7:          else
8:              return a * factorial(a - 1);
9:      }
10:
11:     MouseArea {
12:         anchors.fill: parent
13:         onClicked: console.log(factorial(10))
14:     }
15: }
```

Ajouter du code

Le code javascript

Binding

Répondre à un signal

Fonctions

Fichier .js

Ajouter du code

```
1:  import "factorial.js" as MathFunctions
2:
3:  Item {
4:      MouseArea {
5:          anchors.fill: parent
6:          onClicked: console.log(MathFunctions.factorial(10))
7:      }
8:  }
```

Ajouter du code

Le code javascript

Binding

Répondre à un signal

Fonctions

Fichier .js

plus d’info: http://doc.qt.io/qt-5/qtqml-javascript-expressions.html

Exercice: Générateur de nombres aléatoires

De nouveaux éléments

Des nouveaux modules

De nouveaux éléments

Des nouveaux modules

Controls

De nouveaux éléments

Des nouveaux modules

Controls

Layout

De nouveaux éléments

Des nouveaux modules

Controls

Layout

Window

De nouveaux éléments

Des nouveaux modules

Controls

Layout

Window

Dialog

De nouveaux éléments

Des nouveaux modules

Controls

Layout

Window

Dialog

Multimedia

Exercice: Analyse du code, Gérer le stackview

Multimedia

import QtMultimedia 5.8

Multimedia

import QtMultimedia 5.8

Jouer un son

Multimedia

```
1:  Text {
2:      text: "Click Me!";
3:      font.pointSize: 24;
4:      width: 150; height: 50;
5:
6:      Audio {
7:          id: playMusic
8:          source: "music.wav"
9:      }
10:     MouseArea {
11:         id: playArea
12:         anchors.fill: parent
13:         onPressed:  { playMusic.play() }
14:     }
15: }
```

Multimedia

```
1:  Item {
2:      width: 640
3:      height: 360
4:
5:      SoundEffect {
6:          id: effect
7:          source: "test.wav"
8:      }
9:      MouseArea {
10:         id: playArea
11:         anchors.fill: parent
12:         onPressed:  { effect.play() }
13:     }
14: }
```

Multimedia

import QtMultimedia 5.8

Jouer un son

Jouer une vidéo

Multimedia

```
1:  Video {
2:      id: video
3:      width : 800
4:      height : 600
5:      source: "video.avi"
6:
7:      MouseArea {
8:          anchors.fill: parent
9:          onClicked: {
10:             video.play()
11:         }
12:     }
13:
14:     focus: true
15:     Keys.onSpacePressed: video.paused = !video.paused
16:     Keys.onLeftPressed: video.position -= 5000
17:     Keys.onRightPressed: video.position += 5000
```

Multimedia

import QtMultimedia 5.8

Jouer un son

Jouer une vidéo

MediaPlayer

Multimedia

import QtMultimedia 5.8

Jouer un son

Jouer une vidéo

MediaPlayer

Exercice: Jouer une vidéo sur deux vues

MVC : les models

ListModel

MVC : les models

```
1:  ListModel {
2:      id: listSection
3:      ListElement {
4:          name: "Foo"
5:          firstname:"Paul"
6:          color:"red"
7:      }
8:      ListElement {
9:          name: "Bar"
10:         firstname:"Henry"
11:         color:"blue"
12:     }
13:     ListElement {
14:         name: "Jones"
15:         firstname:"Jessica"
16:         color:"purple"
17:     }
```

MVC : les models

```
7:          }
8:          ListElement {
9:              name: "Bar"
10:             firstname:"Henry"
11:             color:"blue"
12:         }
13:         ListElement {
14:             name: "Jones"
15:             firstname:"Jessica"
16:             color:"purple"
17:         }
18:         ListElement {
19:             name: "Conor"
20:             firstname:"Sarah"
21:             color:"#808000"
22:         }
23:     }
```

MVC : les models

ListModel

XmlModel

MVC : les models

```
1:  XmlListModel{
2:          id: xmlModel
3:          source: "haiku.xml"
4:          query: "/catalog/haiku"
5:          XmlRole {
6:              id: xmlRole; name: "text"; query: "text/string()";    }
7:          onStatusChanged: {
8:              console.log("Count: "+ xmlModel.count);
9:          }
10:     }
11: _____
12:
13:
14: <?xml version="1.0"?>
15: <catalog>
16: <haiku>
17: <text>
```

MVC : les models

```
15:  <catalog>
16:  <haiku>
17:  <text>
18:  Compétence Oubliée
19:  poésie satirique
20:  Personnages joueurs médusés
21:  </text>
22:  </haiku>
23:  <haiku>
24:  <text>
25:  Dans la vieille mare,
26:  une grenouille saute,
27:  le bruit de l'eau.
28:  </text>
29:  </haiku>
30:  </catalog>
31:
```

MVC : les models

ListModel

XmlModel

Model du C++
https://github.com/obiwankennedy/rcm/blob/master/gamemodel.h
https://github.com/obiwankennedy/rcm/blob/master/gamemodel.cpp

MVC : les vues

Listview

MVC : les vues

```
1:  ListView {
2:      id: listOfPerson
3:      model: listperson
4:      height: 80
5:      delegate: Rectangle {
6:          color: colorItem
7:          width: listOfPerson.width/4
8:          height: 20
9:          Text {
10:             text: name
11:         }
12:     }
13: }
```

MVC : les vues

Listview

Gridview

MVC : les vues

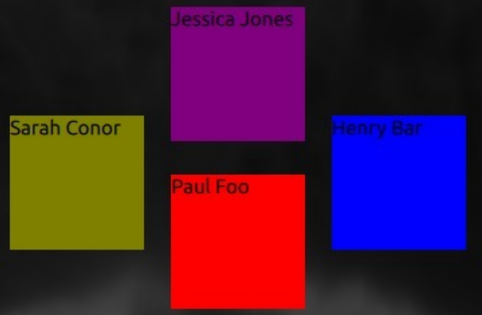
```
1:  GridView {
2:      id: gridOfPerson
3:      model: listperson
4:      width: 300;
5:      height: 200
6:      cellWidth: 100;
7:      cellHeight: 100
8:      delegate: Rectangle {
9:          color: colorItem
10:         width: gridOfPerson.cellWidth;
11:         height: gridOfPerson.cellHeight
12:         Text {
13:             text: name
14:         }
15:     }
16: }
```

MVC : les vues

Listview

Gridview

Pathview



MVC : les vues

```
1:  PathView {
2:      id: pathOfPerson
3:      model: listperson
4:      interactive: true
5:      path: Path {
6:          startX: 200; startY: 150
7:          PathQuad { x: 200; y: 25; controlX: 440; controlY: 125 }
8:          PathQuad { x: 200; y: 150; controlX: -40; controlY: 125 }
9:        }
10:     delegate: Rectangle {
11:       color: colorItem
12:       width: 100
13:       height: 100
14:       Text {
15:         text: name
16:       }
17:       MouseArea {
```

MVC : les vues

```
7:                  PathQuad { x: 200; y: 25; controlX: 440; controlY: 125 }
8:                  PathQuad { x: 200; y: 150; controlX: -40; controlY: 125 }
9:              }
10:         delegate: Rectangle {
11:             color: colorItem
12:             width: 100
13:             height: 100
14:             Text {
15:                 text: name
16:             }
17:             MouseArea {
18:                 anchors.fill: parent
19:                 onClicked:  pathOfPerson.currentIndex = index;
20:             }
21:         }
22:     }
23:
```

MVC : les vues

Listview

Gridview

Pathview

delegate

MVC : les vues

Listview

Gridview

Pathview

delegate

highlight

MVC : les vues

Listview

Gridview

Pathview

delegate

highlight

Exercice: Faire une liste view et une pathview avec le model Players. Gérer le hightlight

Cpp et QML

Définir des éléments dans le monde QML

Cpp et QML

```
 1:  int main(int argc, char *argv[])
 2:  {
 3:      QApplication app(argc, argv);
 4:      QQmlApplicationEngine engine;
 5:      engine.rootContext()->setContextProperty("ScreenW",1920);
 6:      engine.rootContext()->setContextProperty("ScreenH",1080);
 7:      DiceResultModel* model = new DiceResultModel();
 8:      engine.rootContext()->setContextProperty("_diceModel",model);
 9:      engine.load(QUrl(QStringLiteral("qrc:/main.qml")));
10:      QList<QObject*> roots = engine.rootObjects();
11:      return app.exec();
12:  }
13:
```

Cpp et QML

Définir des éléments dans le monde QML

Emettre un signal du QML vers le C++

Cpp et QML

Définir des éléments dans le monde QML

Emettre un signal du QML vers le C++

Appeler une fonction QML

Cpp et QML

```
1:
2:  //MyItem.qml
3:  import QtQuick 2.0
4:
5:  Item {
6:      function myQmlFunction(msg) {
7:          console.log("Got message:", msg)
8:          return "some return value"
9:      }
10: }
11: _______
12:
13: //CPP
14: int main(int argc, char *argv[])
15: {
16:     QApplication app(argc, argv);
17:
```

Cpp et QML

```
13:  //CPP
14:  int main(int argc, char *argv[])
15:  {
16:      QApplication app(argc, argv);
17:
18:      QQmlEngine engine;
19:
20:      QQmlComponent component(&engine, "MyItem.qml");
21:      QObject *object = component.create();
22:
23:      QVariant returnedValue;
24:      QVariant msg = "Hello from C++";
25:      QMetaObject::invokeMethod(object, "myQmlFunction",
26:          Q_RETURN_ARG(QVariant, returnedValue),
27:          Q_ARG(QVariant, msg));
28:
29:      qDebug() << "QML function returned:" << returnedValue.toString();
```

Cpp et QML

```
18:     QQmlEngine engine;
19:
20:     QQmlComponent component(&engine, "MyItem.qml");
21:     QObject *object = component.create();
22:
23:     QVariant returnedValue;
24:     QVariant msg = "Hello from C++";
25:     QMetaObject::invokeMethod(object, "myQmlFunction",
26:         Q_RETURN_ARG(QVariant, returnedValue),
27:         Q_ARG(QVariant, msg));
28:
29:     qDebug() << "QML function returned:" << returnedValue.toString();
30:     delete object;
31:
32:     return app.exec();
33: }
34:
```

Image provider

Définition

Image provider

Définition

Comment cela marche ?

Image provider

Définition

Comment cela marche ?

L'utilisation

Image provider

Définition

Comment cela marche ?

L'utilisation

Le gain ?

Dessiner en QML

Le Canvas

Dessiner en QML

```
1:  Canvas {
2:      id: frise
3:      anchors.left: parent.left
4:      anchors.right: parent.right
5:      anchors.verticalCenter: parent.verticalCenter
6:      anchors.verticalCenterOffset: -200
7:      height: app.height/4
8:  }
9:
```

Dessiner en QML

Le Canvas

Contexte

Dessiner en QML

```
1:    onPaint : {
2:            var ctx = getContext("2d")
3:
4:
```

Dessiner en QML

Le Canvas

Contexte

L’utilisation

Dessiner en QML

```
1:    var heightOfArrow = app.height/20
2:            var lineW = 4
3:            //ctx.fillStyle = Qt.rgba(0, 0, 0, 1);
4:            //ctx.lineWidth = lineW;
5:            ctx.beginPath()
6:            ctx.moveTo(0,heightOfArrow)
7:            ctx.lineTo(frise.width-200,heightOfArrow)
8:            ctx.lineTo(frise.width-200,0)
9:            ctx.lineTo(frise.width,frise.height/2)
10:           ctx.lineTo(frise.width-200,frise.height)
11:           ctx.lineTo(frise.width-200,frise.height-heightOfArrow)
12:           ctx.lineTo(0,frise.height-heightOfArrow)
13:           ctx.closePath()
14:
```

Dessiner en QML

Le Canvas

Contexte

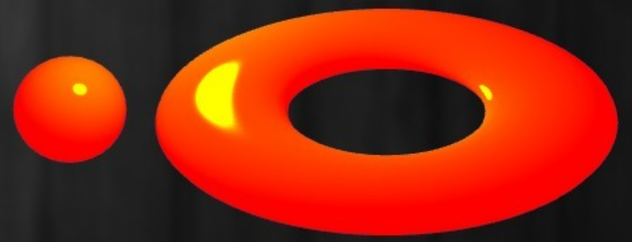
L’utilisation

Exemple

Exercice: Dessiner une progresse bar animée dont la couleur est un dégrade

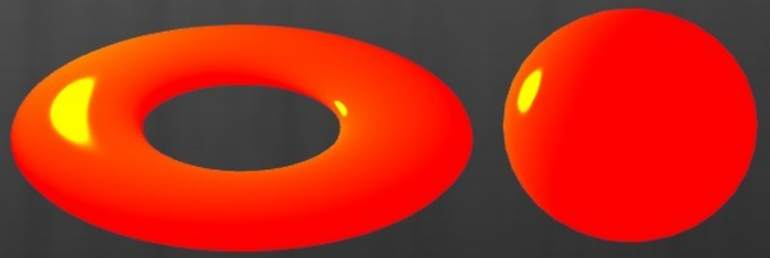
3d en QML

Initialisation du projet



3d en QML

```
1:  #include <Qt3DQuickExtras/qt3dquickwindow.h>
2:  #include <QGuiApplication>
3:
4:  int main(int argc, char* argv[])
5:  {
6:      QGuiApplication app(argc, argv);
7:      Qt3DExtras::Quick::Qt3DQuickWindow view;
8:      view.setSource(QUrl("qrc:/main.qml"));
9:      view.show();
10:
11:     return app.exec();
12: }
```



3d en QML

Initialisation du projet

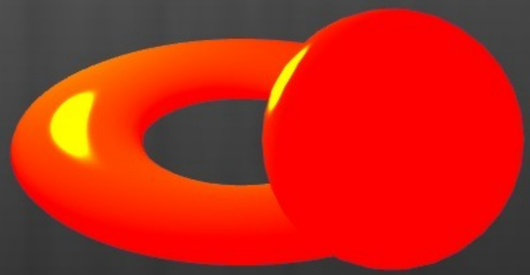
Usage d’une Scene3d

3d en QML

```
1:
2:    import QtQuick.Scene3D 2.0
3:
4:        Scene3D {
5:                id: scene3d
6:                anchors.right: parent.right
7:                anchors.bottom: parent.bottom
8:                width: parent.width*0.5
9:                height: parent.height*0.5
10:               anchors.margins: 10
11:               focus: true
12:               aspects: ["input", "logic"]
13:               cameraAspectRatioMode: Scene3D.AutomaticAspectRatio
14:
```

3d en QML

```
1:
2:    import QtQuick.Scene3D 2.0
3:
4:        Scene3D {
5:                id: scene3d
6:                anchors.right: parent.right
7:                anchors.bottom: parent.bottom
8:                width: parent.width*0.5
9:                height: parent.height*0.5
10:               anchors.margins: 10
11:               focus: true
12:               aspects: ["input", "logic"]
13:               cameraAspectRatioMode: Scene3D.AutomaticAspectRatio
14:
```



3d en QML

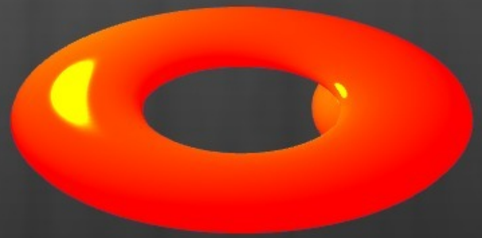
Initialisation du projet

Usage d’une Scene3d

Entity

3d en QML

```
1:
2:    Entity {
3:        id: sceneRoot
4:    }
5:
```



3d en QML

Initialisation du projet

Usage d’une Scene3d

Entity

Mesh

3d en QML

```
1:
2:          TorusMesh {
3:              id: torusMesh
4:              radius: 5
5:              minorRadius: 2
6:              rings: 100
7:              slices: 20
8:          }
9:
```

3d en QML

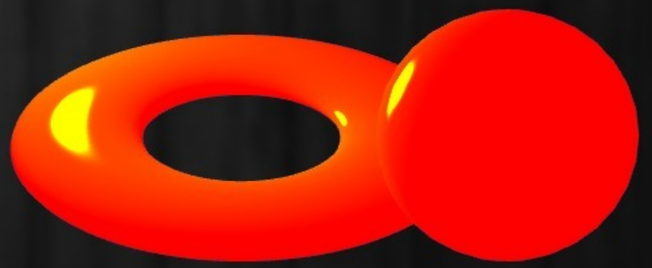
Initialisation du projet

Usage d’une Scene3d

Entity

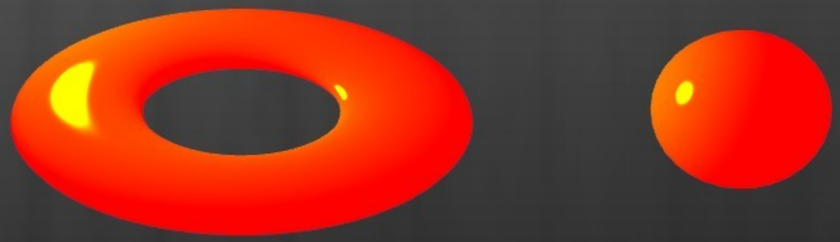
Mesh

Material



3d en QML

```
1:
2:          PhongMaterial {
3:              id: material
4:              ambient: "red"
5:              diffuse: "yellow"
6:              specular: "yellow"
7:          }
8:
```

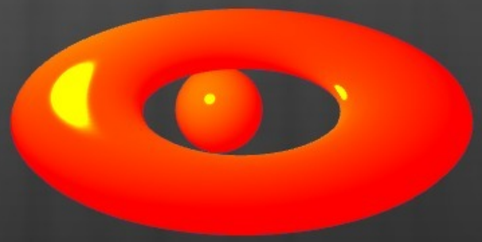


3d en QML

- Initialisation du projet
- Usage d’une Scene3d
- Entity
- Mesh
- Material
- Transform

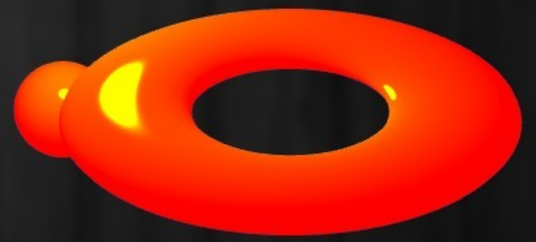
3d en QML

```
1:
2:          Transform {
3:              id: sphereTransform
4:              property real userAngle: 0.0
5:              matrix: {
6:                  var m = Qt.matrix4x4();
7:                  m.rotate(userAngle, Qt.vector3d(0, 1, 0));
8:                  m.translate(Qt.vector3d(20, 0, 0));
9:                  return m;
10:             }
11:         }
12:
```



3d en QML

- Initialisation du projet
- Usage d’une Scene3d
- Entity
- Mesh
- Material
- Transform
- Regroupement



3d en QML

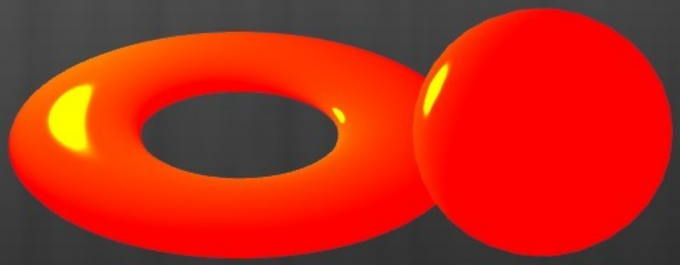
```
1:
2:          Entity {
3:              id: sphereEntity
4:              components: [ sphereMesh, material, sphereTransform ]
5:          }
6:
```

3d en QML

- Initialisation du projet
- Usage d’une Scene3d
- Entity
- Mesh
- Material
- Transform
- Regroupement
- Camera

3d en QML

```
1:
2:          Camera {
3:              id: camera
4:              projectionType: CameraLens.PerspectiveProjection
5:              fieldOfView: 45
6:              aspectRatio: 16/9
7:              nearPlane : 0.1
8:              farPlane : 1000.0
9:              position: Qt.vector3d( 0.0, 0.0, -40.0 )
10:             upVector: Qt.vector3d( 0.0, 1.0, 0.0 )
11:             viewCenter: Qt.vector3d( 0.0, 0.0, 0.0 )
12:         }
13:
14:         OrbitCameraController {
15:             camera: camera
16:         }
```



RCSE

Contact

Un IDE pour du QML dédié

RCSE

Un IDE pour du QML dédié

Petit tour du SDK

RCSE

Un IDE pour du QML dédié

Petit tour du SDK

Petit tour du code généré

RCSE

Un IDE pour du QML dédié

Petit tour du SDK

Petit tour du code généré

Échange

Contact

Site web: **www.rolisteam.org**

Github: **github.com/Rolisteam**

Twitter: **@Rolisteam**

Facebook: **www.facebook.com/rolisteam**

Liberapay: https://liberapay.com/Rolisteam/donate

Youtube: www.youtube.com/channel/UC4uoGZl1nQRXbVs8WjxjKvw

Irc: **#RolisteamOfficial** on freenode.net

Courriel: **renaud@rolisteam.org**

Site Perso: http://renaudguezennec.eu

GitHub Perso: http://github.com/obiwankennedy