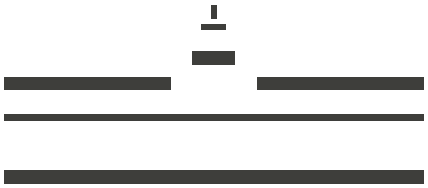
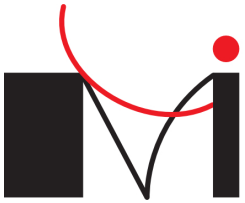


Westfälische
Wilhelms-Universität
Münster



Applied
Mathematics
Münster

Interaktive Simulationen

Lektion 1/3: Event-Driven Design und Signals

living.knowledge
WWU Münster

Prof. Dr. Christian Engwer
& Andreas Buhr

16. April 2014

Normales C++ Programm

Programmstruktur

```
#include <fem_library>

int main(){
  Mesh mesh(creeate_mesh());
  Matrix matrix(create_matrix(mesh));
  Vector rhs(create_rhs());
  Vector solution(solve(matrix,rhs));
  visualize_solution(solution);

}
```

Normales C++ Programm

Programmstruktur

```
#include <fem_library>

int main(){
  Mesh mesh(creeate_mesh());
  Matrix matrix(create_matrix(mesh));
  Vector rhs(create_rhs());
  Vector solution(solve(matrix,rhs));
  visualize_solution(solution);
  add_interactivity(); // funktioniert nicht
}
```

Interaktives C++ Programm

Programmstruktur

```
#include <fem_library>
int main(){
  // definiere:
  // WENN: User bedient Regler DANN: reagiere darauf
  // WENN: User bewegt Telefon DANN: reagiere darauf
  // WENN: User fasst Pendel DANN: reagiere darauf
  // WENN: 50ms vorbei DANN: mache Zeitschritt

  // warte auf WENNs, mache DANNs,
  // sonst warte
}
```

Zentrale Punkte

Interaktive Programme

Die beiden wichtigsten Eigenschaften:

1. Formulierung von wenn-dann Beziehungen
2. Endlosschleife

WENN-DANN in Qt

ein WENN hängt an einem “Signal”

Objekte können Signale haben und auslösen.

WENN-DANN in Qt

ein WENN hängt an einem “Signal”

Objekte können Signale haben und auslösen.

ein DANN ist ein “Slot”

Ein Slot verhält sich ähnlich zu einer Memberfunktion.

WENN-DANN in Qt

ein WENN hängt an einem “Signal”

Objekte können Signale haben und auslösen.

ein DANN ist ein “Slot”

Ein Slot verhält sich ähnlich zu einer Memberfunktion.

eine WENN-DANN Beziehung ist eine Verbindung

... zwischen einem Signal und einem Slot.

Exkurs: Qt

- Bibliothek für Programmierung von GUI-Programmen

[1] Filedatum in ftp://ftp.qt.nokia.com/qt/source/

Exkurs: Qt

- Bibliothek für Programmierung von GUI-Programmen
- Cross-Platform (Windows, Linux, MAC, Android, iOS, Blackberry, ...)

[1] Filedatum in ftp://ftp.qt.nokia.com/qt/source/

Exkurs: Qt

- Bibliothek für Programmierung von GUI-Programmen
- Cross-Platform (Windows, Linux, MAC, Android, iOS, Blackberry, ...)
- mit Tradition[1]:
  - Qt 1.41 Oktober 1998
  - Qt 2.0 Juni 1999
  - Qt 3.0 Oktober 2001
  - Qt 4.0 Juni 2005
  - Qt 5.0 Dezember 2012

[1] Filedatum in ftp://ftp.qt.nokia.com/qt/source/

Exkurs: Qt, C++ Erweiterung

- Klassen, die von QObject ableiten, können “signal”s und “slot”s haben.
- Der “MOC” generiert aus den *.h Dateien des Users (uns) moc_*.cpp Dateien

Signal Declaration

```
class SimulationControl : public QObject
{
    Q_OBJECT
public:
    explicit SimulationControl(QObject *parent = 0);
    void doTimestep(void);

signals:
    void timestepDone(void);
};
```

Slot Declaration

```
class VisualizationControl : public QObject
{
    Q_OBJECT
public:
    explicit VisualizationControl(QObject *parent = 0);

public slots:
    void redraw(void);
};
```

Signal Usage

```
void SimulationControl::doTimestep(void){
  // do a lot of work here
  // finally:
  emit timestepDone();
}
```

Slot Definition

```
void VisualizationControl::redraw(void){
  // do some work here to redraw to screen
}
```

Connect Signal to Slot

```
// anywhere in your code do:
QObject::connect(
                simControl,SIGNAL(timestepDone()),
                visControl,SLOT(redraw()));
```

Was ist ein Signal?

Was ist ein Signal?

- Ein Funktionsaufruf

Was ist ein Signal?

- Ein Funktionsaufruf
- der Aufrufer weiß nicht, ob überhaupt jemand verbunden ist

Was ist ein Signal?

- Ein Funktionsaufruf
- der Aufrufer weiß nicht, ob überhaupt jemand verbunden ist
- der Aufrufer weiß nicht, was er aufruft

Was ist ein Signal?

- Ein Funktionsaufruf
- der Aufrufer weiß nicht, ob überhaupt jemand verbunden ist
- der Aufrufer weiß nicht, was er aufruft

Fragen ergeben sich:

- Was passiert mit Rückgabewerten?

Was ist ein Signal?

- Ein Funktionsaufruf
- der Aufrufer weiß nicht, ob überhaupt jemand verbunden ist
- der Aufrufer weiß nicht, was er aufruft

Fragen ergeben sich:

- Was passiert mit Rückgabewerten?
- Was passiert mit Argumenten?

Was ist ein Signal?

- Ein Funktionsaufruf
- der Aufrufer weiß nicht, ob überhaupt jemand verbunden ist
- der Aufrufer weiß nicht, was er aufruft

Fragen ergeben sich:

- Was passiert mit Rückgabewerten?
- Was passiert mit Argumenten?
- Wann wird die Funktion aufgerufen?

Was ist ein Signal?

- Ein Funktionsaufruf
- der Aufrufer weiß nicht, ob überhaupt jemand verbunden ist
- der Aufrufer weiß nicht, was er aufruft

Fragen ergeben sich:

- Was passiert mit Rückgabewerten?
- Was passiert mit Argumenten?
- Wann wird die Funktion aufgerufen?
- Was passiert, wenn der Empfänger gelöscht wird?

Gelernt

- Signale definieren
- Slots definieren
- Signale und Slots verbinden

Interaktives C++ Programm

Programmstruktur

```
#include <fem_library>
int main(){
  // definiere:
  // WENN: User bedient Regler DANN: reagiere darauf
  // WENN: User bewegt Telefon DANN: reagiere darauf
  // WENN: User fasst Pendel DANN: reagiere darauf
  // WENN: 50ms vorbei DANN: mache Zeitschritt

  // warte auf WENNs, mache DANNs,
  // sonst warte
}
```

Interaktives C++ Programm

Programmstruktur

```
#include <fem_library>
int main(){
  // definiere:

  SimulationControl* mySC = new SimulationControl();
  VisualizationControl* myVC = new VisualizationControl();
  connectSignals(mySC,myVC);

  // warte auf WENNs, mache DANNs,
  // sonst warte
}
```

Interaktives C++ Programm

Programmstruktur

```
#include <fem_library>
int main(int argc, char *argv[]){
  QApplication a(argc, argv);

  SimulationControl* mySC = new SimulationControl();
  VisualizationControl* myVC = new VisualizationControl();
  connectSignals(mySC,myVC);

  // warte auf WENNs, mache DANNs,
  return a.exec();
}
```

Userinteraktion

- Qt hat Signale, wenn der User etwas gemacht hat.
- Qt hat Slots, um Dinge anzuzeigen

QTimer

QTimer als Beispiel für Signale vom System:

QTimer Beispiel

```
QTimer timer;
timer.setInterval(1000);
timer.start();
```

'timer' sendet alle 1000 ms das Signal 'timeout()'.

Beispiel

Beispielempfänger

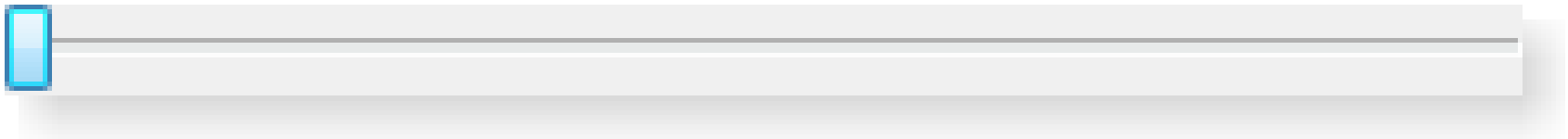
```
class HalloSager : public QObject
{
    Q_OBJECT
public slots:
    void sagHallo(){
        std::cout << "Hallo" << std::endl;
    }
};
```

```
int main(int argc, char** argv)
{
    QApplication app(argc,argv);

    QTimer timer;
    timer.setInterval(1000);
    timer.start();

    HalloSager sager;
    QObject::connect(&timer,SIGNAL(timeout()),
                     &sager,SLOT(sagHallo()));
    app.exec();
}
```

Signals from Slider



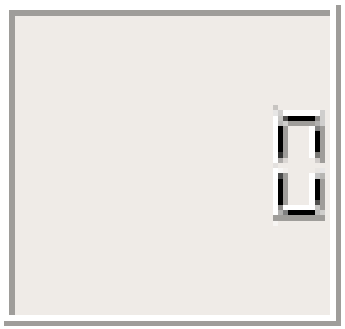
Von `http://qt-project.org/doc/qt-5/qslider.html`:

QSlider inherits a comprehensive set of signals:

Signal	Description
valueChanged()	Emitted when the slider's value has changed. The tracking() determines whether this signal is emitted during user interaction.
sliderPressed()	Emitted when the user starts to drag the slider.
sliderMoved()	Emitted when the user drags the slider.
sliderReleased()	Emitted when the user releases the slider.

Hat Signal “valueChanged(int)”.

Slot in QLCDNumber



Von `http://qt-project.org/doc/qt-5/qlcdnumber.html`:

Public Slots

void	**display**(const QString & *s*)
void	**display**(double *num*)
void	**display**(int *num*)
void	**setBinMode**()

Hat Slot “display(int)”.

FAQ

- Wie erstelle ich QSlider und QLCDNumber?
  → Im Designer anklicken.
- Wie komme ich dann an Slider und Number?
  → Das “Widget” hat ein Member “ui”, das hat sie als Member.
- Wie mache ich die LCDNumber größer?
  → Im Designer unter “minimumSize” die “Height” auf z.B. 100 Pixel stellen.
- Wie mache ich den QSlider größer?
  → Stylesheet von Vorlesungswebseite laden und im Designer beim obersten “Widget” unter “styleSheet” eintragen.