

ÉVALUATION FORMATIVE THÉORIQUE

GIF250 : Interfaces graphiques

SOLUTIONNAIRE

SESSION S2 – UNITÉ 7 informatique

L'évaluation de l'unité consiste en un examen sommatif théorique et un examen final pratique. Vous avez ici une évaluation formative pour la partie théorique.

L'évaluation formative pratique consiste en l'exercice de la formation à la pratique au laboratoire dans le guide étudiant. Cet exercice est représentatif de l'examen pratique en termes de complexité.

QUESTION 1

Sous-question 1.1

À quoi sert qmake et qu'est-ce qu'un fichier *pro* ?

qmake : qmake est un utilitaire qui permet de gérer des projets Qt sous Windows et Unix. Il permet, entre autres, de générer des makefile sous Unix et des projets pour Visual Studio.

fichier *pro* : un fichier *pro* permet de spécifier un projet (fichiers source, d'entête, librairies, etc.) afin que qmake puisse générer le makefile ou le projet pour Visual Studio.

Sous-question 1.2

Qu'est-ce qu'une fente (*slot*) et un *signal* ? À quoi servent-ils ?

Fente : une fente est un mécanisme de Qt qui permet de définir du code qui pourra être exécuté lorsqu'une action de l'interface graphique survient.

Signal : un signal est un mécanisme de Qt qui survient lorsqu'une action est posée dans l'interface graphique. Dans ce cas, un signal est émis, comme dans le cas d'un bouton qui est cliqué.

Servent à : associer du code C++ avec des actions dans l'interface graphique.

Sous-question 1.3

Qu'est-ce qui permet d'associer une action posée sur un objet graphique, comme cliquer un bouton, avec du code C++ ?

Cochez ou noircir le seul énoncé qui est vrai parmi les énoncés suivants.

ÉNONCÉS

La macro Q_OBJECT ☐

La méthode connect ☒

Un signal ☐

Une fente (*slot*) ☐

Sous-question 1.4

Qu'est-ce que le moc ? À quoi sert-il ?

moc : Le moc est le *meta object compiler*. C'est un outil qui fait partie de la distribution de Qt.

Sert à : Il permet de faire un traitement sur certains fichiers afin de gérer les extensions faites par Qt au C++, comme les fentes et les signaux.

Sous-question 1.5

Qu'est-ce que Q_OBJECT et quand doit-on s'en servir ?

Q_OBJECT : C'est une macro fournie par Qt.

Quand on s'en sert : Il faut ajouter dans la définition d'une classe qui utilise les extensions faites au C++ par Qt, comme dans le cas de fentes et de signaux.

QUESTION 2

Soit le code suivant :

Fichier d'entête

```
#ifndef EXAMENGUI_H
#define EXAMENGUI_H

#include <QMainWindow>
#include <QLabel>
#include <QPushButton>
#include <QList>
#include <QString>

class MonInterface : public QWidget {
public:
    MonInterface(QWidget *);
private:
    QPushButton *monBouton;
};

class MonGui : public QMainWindow
{
    Q_OBJECT
public:
    MonGui();
protected slots:
    void a1();
    void a2();
    void a3();
private:
    MonInterface * gui;
    QList< QString *> maListe;
};

#endif // EXAMENGUI_H
```

Fichier de code

```
#include <QMenu>
#include <QMenuBar>
#include <QStatusbar>
#include <QVBoxLayout>

#include "examengui.h"

MonInterface::MonInterface(QWidget * parent):QWidget(parent) {
    QVBoxLayout *tmp = new QVBoxLayout;
    setLayout(tmp);
    monBouton = new QPushButton("OK!", this);
    tmp->addWidget(monBouton);
    connect(monBouton,SIGNAL(clicked()),parent,SLOT(a3()));
}

MonGui::MonGui() {
    QMenu *tmpmenu;
    setWindowTitle("Examen v1.0");
    tmpmenu = menuBar()->addMenu( tr("Menu") );
    tmpmenu->addAction(tr("menu 1"),this, SLOT(a1()));
    tmpmenu->addAction(tr("menu 2"),this, SLOT(a2()));
    gui = new MonInterface(this);
    setCentralWidget(gui);
    statusBar()->showMessage("Pret!");
    show();
}

void MonGui::a1() {
    statusBar()->showMessage("a1");
    maListe.append(new QString("a1"));
}

void MonGui::a2() {
    statusBar()->showMessage("a2");
    maListe.append(new QString("a2"));
}

void MonGui::a3() {
    statusBar()->showMessage("a3");
    maListe.append(new QString("a3"));
}
```

Décrivez ce qui se passe à l'écran pour les actions suivantes.

L'utilisateur clique sur le bouton :

La chaîne "a3" est affichée dans la barre de statut de la fenêtre. Elle est aussi ajoutée dans la liste de chaînes maListe.

L'utilisateur sélectionne le menu menu 1 :

La chaîne "a1" est affichée dans la barre de statut de la fenêtre. Elle est aussi ajoutée dans la liste de chaînes maListe.

L'utilisateur sélectionne le menu menu 2 :

La chaîne "a2" est affichée dans la barre de statut de la fenêtre. Elle est aussi ajoutée dans la liste de chaînes maListe.

Dessinez la fenêtre principale de l'application :

Examen v1.0	
Menu	<div>OK!</div>
Prêt!	