

```

#include "fenetre.h"
#include "iostream"

fenetre::fenetre():flag_dock(false)
{
    int largeur= QApplication::desktop()->width()-100;
    int hauteur = QApplication::desktop()->height()-100;
    this->setMinimumSize(largeur,hauteur);
    //widget central
    QWidget * w = new QWidget;
    setCentralWidget(w);

    /*****
                                //CARTE
    image = new carte();

    *****/

    /*****
                                //MENUBAR
    //FICHIER

    menuFichier = menuBar()->addMenu("&Fichier");

    //ouvrir
    QAction *ouvrir = menuFichier->addAction("Ouvrir un projet");
    ouvrir->setShortcut(QKeySequence("Ctrl+O"));
    ouvrir->setIcon(QIcon("ouvrir1.png"));

    //Charger
    charge = menuFichier->addAction("Charger un projet");
    charge->setShortcut(QKeySequence("Ctrl+C"));

    //Enregistrer sous
    save_as = menuFichier->addAction("Sauvegarder le projet sous");
    save_as->setEnabled(false);

    //Enregistrer
    save = menuFichier->addAction("Sauvegarder le projet");
    save->setShortcut(QKeySequence("Ctrl+S"));
    save->setEnabled(false);
    save->setIcon(QIcon("svg.png"));
    //quitter
    QAction *quitter = menuFichier->addAction("&Quitter");
    quitter->setShortcut(QKeySequence("Ctrl+Q"));

    //EDITION

    menuEdition = menuBar()->addMenu("&Edition");

    //export
    exp = menuEdition->addAction("Export aux format .gpx");
    exp->setShortcut(QKeySequence("Ctrl+E"));
    exp->setEnabled(false);

    //AFFICHAGE

    menuAffichage = menuBar()->addMenu("&Affichage");

    //zoom avant

```

```

        zoom_in = menuAffichage->addAction("Zoom avant");
        zoom_in->setShortcut(QKeySequence("Ctrl+W"));
        zoom_in->setToolTip(trUtf8("La molette de la souris peut-être aussi
utilisée"));
        zoom_in->setEnabled(false);
        zoom_in->setIcon(QIcon("zoom+.png"));
        //zoom arriere
        zoom_out = menuAffichage->addAction("Zoom arriere");
        zoom_out->setShortcut(QKeySequence("Ctrl+alt+W"));
        zoom_out->setToolTip(trUtf8("La molette de la souris peut-être aussi
utilisée"));
        zoom_out->setEnabled(false);
        zoom_out->setIcon(QIcon("zoom-.png"));
        //afficher le dock
        affich_dock = menuAffichage->addAction("Afficher le gestionnaire GPS");
        affich_dock->setShortcut(QKeySequence("Ctrl+D"));
        affich_dock->setEnabled(false);

/*****
//TOOLBAR

toolbar = addToolBar("Barre d'outils");

//ouvrir //preparer a mettre les icones
toolbar->addAction(ouvrir);
toolbar->addSeparator();

//enregistrer
toolbar->addAction(save);
toolbar->addSeparator();

//Effacer
effacer = toolbar->addAction("Fermer le projet");
toolbar->addAction(effacer);
effacer->setEnabled(false);
effacer->setIcon(QIcon("fermer2.png"));
toolbar->addSeparator();

//Zoom
toolbar->addAction(zoom_in);
toolbar->addSeparator();
toolbar->addAction(zoom_out);

//zone de sélection de couleur

toolbar->addSeparator();
label = new QLabel (toolbar);
couleur = "background-color: rgb(255,255,255);";
label->setStyleSheet(couleur);
label->setFixedWidth(15);
label->setFixedHeight(15);
label->setToolTip(trUtf8("Couleur du chemin sélectionné"));
toolbar->addWidget(label);
toolbar->addSeparator();

//gestionnaire des coordonnées
gestionnaire = new QPushButton("Gestionnaire GPS");
gestionnaire->setToolTip("Afficher ou non le gestionnaire GPS");
gestionnaire->setIcon(QIcon("gps.png"));
gestionnaire->setEnabled(false);

```

```

toolbar->addWidget(gestionnaire);
toolbar->addSeparator();

// penser a faire le connecte et les fonctions de modifications ...

/*****
//DOCK

dock = new QDockWidget(trUtf8("Gestion des coordonnées :"), this);
dock->setFeatures(QDockWidget::DockWidgetMovable);
addDockWidget(Qt::RightDockWidgetArea, dock);

QWidget * contenuDock = new QWidget;
dock->setWidget(contenuDock);

QVBoxLayout* VdockLayout1 = new QVBoxLayout();
QVBoxLayout* VdockLayout2 = new QVBoxLayout();
QVBoxLayout* VdockLayout3 = new QVBoxLayout();

QHBoxLayout* HdockLayout1 = new QHBoxLayout();
QHBoxLayout* HdockLayout2 = new QHBoxLayout();
QHBoxLayout* HdockLayout3 = new QHBoxLayout();
QHBoxLayout* HdockLayout4 = new QHBoxLayout();

//zone coordonnées décimales

QLabel *titre = new QLabel (trUtf8("<font color=\"red\">En décimales
:</font>"));
VdockLayout1->addWidget(titre);
QLabel *dd = new QLabel (trUtf8("<font color=\"green\">Premier point
:</font>"));
VdockLayout1->addWidget(dd);
//Latitude
QLabel *lat = new QLabel ("Latitude : ");
VdockLayout1->addWidget(lat);
la = new QDoubleSpinBox (dock);
la->setMaximum((90.0));
la->setMinimum((-90.0));
la->setDecimals(7);
VdockLayout1->addWidget(la);
la->setFixedHeight(20);
la->setFixedWidth(150);

QLabel *lon = new QLabel ("Longitude : ");
VdockLayout1->addWidget(lon);
lo = new QDoubleSpinBox (dock);
lo->setMaximum((180.0));
lo->setMinimum((-180.0));
lo->setDecimals(7);
VdockLayout1->addWidget(lo);
lo->setFixedHeight(20);
lo->setFixedWidth(150);

QLabel *ddd1 = new QLabel (trUtf8("<font color=\"green\">Deuxième point
:</font>"));
VdockLayout1->addWidget(ddd1);
QLabel *lat1 = new QLabel ("Latitude : ");
VdockLayout1->addWidget(lat1);
la1 = new QDoubleSpinBox (dock);
la1->setMaximum((90.0));

```

```

la1->setMinimum((-90.0));
la1->setDecimals(7);
VdockLayout1->addWidget(la1);
la1->setFixedHeight(20);
la1->setFixedWidth(150);

QLabel *lon1 = new QLabel ("Longitude : ");
VdockLayout1->addWidget(lon1);
lo1 = new QDoubleSpinBox (dock);
lo1->setMaximum((180.0));
lo1->setMinimum((-180.0));
lo1->setDecimals(7);
VdockLayout1->addWidget(lo1);
lo1->setFixedHeight(20);
lo1->setFixedWidth(150);

QPushButton *valider1 = new QPushButton("Valider");
valider1->setFixedSize(100,30);
VdockLayout1->addWidget(valider1);

//zone coordonnées sexagésimales
QLabel *titre2 = new QLabel (trUtf8("<font color=\"red\">En sexagésimales
:</font>"));
VdockLayout2->addWidget(titre2);

QLabel *DMS1 = new QLabel (trUtf8("<font color=\"green\">Premier point en
degrés minutes secondes :</font>"));
VdockLayout2->addWidget(DMS1);

d1 = new QSpinBox(dock);
d1->setFixedHeight(20);
d1->setFixedWidth(75);

m1 = new QSpinBox(dock);
m1->setFixedHeight(20);
m1->setFixedWidth(60);

s1 = new QDoubleSpinBox(dock);
s1->setFixedHeight(20);
s1->setFixedWidth(100);
s1->setDecimals(5);

dd1 = new QSpinBox(dock);
dd1->setFixedHeight(20);
dd1->setFixedWidth(75);

mm1 = new QSpinBox(dock);
mm1->setFixedHeight(20);
mm1->setFixedWidth(60);

ss1 = new QDoubleSpinBox(dock);
ss1->setFixedHeight(20);
ss1->setFixedWidth(100);
ss1->setDecimals(5);

HdockLayout1->addWidget(d1);
HdockLayout1->addWidget(m1);
HdockLayout1->addWidget(s1);

```

```

HdockLayout2->addWidget(dd1);
HdockLayout2->addWidget(mm1);
HdockLayout2->addWidget(ss1);

QLabel *lat2 = new QLabel ("Latitude : ");
VdockLayout2->addWidget(lat2);
VdockLayout2->addLayout(HdockLayout1);
QLabel *lon2 = new QLabel ("Longitude : ");
VdockLayout2->addWidget(lon2);
VdockLayout2->addLayout(HdockLayout2);

QLabel *DMS2 = new QLabel (trUtf8("<font color=\"green\">Deuxième point en
degrés minutes secondes :</font>"));
VdockLayout2->addWidget(DMS2);

d2 = new QSpinBox(dock);
d2->setFixedHeight(20);
d2->setFixedWidth(75);

m2 = new QSpinBox(dock);
m2->setFixedHeight(20);
m2->setFixedWidth(60);

s2 = new QDoubleSpinBox(dock);
s2->setFixedHeight(20);
s2->setFixedWidth(100);
s2->setDecimals(5);

dd2 = new QSpinBox(dock);
dd2->setFixedHeight(20);
dd2->setFixedWidth(75);

mm2 = new QSpinBox(dock);
mm2->setFixedHeight(20);
mm2->setFixedWidth(60);

ss2 = new QDoubleSpinBox(dock);
ss2->setFixedHeight(20);
ss2->setFixedWidth(100);
ss2->setDecimals(5);

HdockLayout3->addWidget(d2);
HdockLayout3->addWidget(m2);
HdockLayout3->addWidget(s2);

HdockLayout4->addWidget(dd2);
HdockLayout4->addWidget(mm2);
HdockLayout4->addWidget(ss2);

QLabel *lat3 = new QLabel ("Latitude : ");
VdockLayout2->addWidget(lat3);
VdockLayout2->addLayout(HdockLayout3);
QLabel *lon3 = new QLabel ("Longitude : ");
VdockLayout2->addWidget(lon3);
VdockLayout2->addLayout(HdockLayout4);

QPushButton *valider2 = new QPushButton("Valider");
valider2->setFixedSize(100,30);
VdockLayout2->addWidget(valider2);

```

```

VdockLayout3->addStretch();
VdockLayout3->addLayout(VdockLayout1);
VdockLayout3->addStretch();
VdockLayout3->addLayout(VdockLayout2);
VdockLayout3->addStretch();

expo = new QPushButton(trUtf8("Export en format .gpx"));
expo->setEnabled(false);
VdockLayout3->addWidget(expo);
QPushButton *reinit = new QPushButton(trUtf8("réinitialiser"));
VdockLayout3->addWidget(reinit);
contenuDock->setLayout(VdockLayout3);
dock->setVisible(false);
/*****
//STATUSBAR

SB = statusBar();
SB->setVisible(false);

/*****
//LAYOUTS
mainLayout= new QHBoxLayout(w);
//scroll= new QScrollArea;
//scroll->setWidget(image);
mainLayout->addWidget(image);
//mainLayout->addWidget(scroll);

/*****
//CONNECT

QObject::connect(quitte, SIGNAL(triggered()), qApp, SLOT(quit()));
QObject::connect(ouvrir, SIGNAL(triggered()),this,
SLOT(telechargerImage()));
QObject::connect(affich_dock, SIGNAL(triggered()),this,
SLOT(afficher_dock()));
QObject::connect(save_as,SIGNAL(triggered()),this,SLOT(svg_as()));
QObject::connect(save,SIGNAL(triggered()),this,SLOT(svg()));
QObject::connect(charge, SIGNAL(triggered()),this, SLOT(charger()));

QObject::connect(valider1, SIGNAL(clicked()),this, SLOT(valider_dec()));
QObject::connect(valider2, SIGNAL(clicked()),this, SLOT(valider_sexa()));
QObject::connect(gestionnaire, SIGNAL(clicked()),this,
SLOT(afficher_dock()));

QObject::connect(effacer, SIGNAL(triggered()),this, SLOT(fermerProjet()));
QObject::connect(zoom_in, SIGNAL(triggered()),image,
SLOT(augmenter_zoom()));
QObject::connect(zoom_out, SIGNAL(triggered()),image,
SLOT(diminuer_zoom()));

QObject::connect(reinit, SIGNAL(clicked()),image, SLOT(setNbpoint()));
QObject::connect(exp,SIGNAL(triggered()),image,SLOT(exporter_gpx()));
QObject::connect(expo,SIGNAL(clicked()),image,SLOT(exporter_gpx()));

QObject::connect(image, SIGNAL(ChangeRes()),this, SLOT(setCouleur()));
}

/*****
// SLOTS

void fenetre::telechargerImage()

```

```

{
    QString fichier = QFileDialog::getOpenFileName(this, "Ouvrir un
fichier", QString(), "Images (*.png *.gif *.jpg *.jpeg)");

    //MessageBox::information(this, "Fichier", "Vous avez selectionne
:\n" + fichier);

    // QMessageBox::information(this, "Fichier", "Vous avez selectionne
:\n" + fichier);
    image->afficherCarte(fichier);
    if (image->test_carte()==true){
        save_as->setEnabled(true);
        save->setEnabled(true);
        effacer->setEnabled(true);
        zoom_in->setEnabled(true);
        zoom_out->setEnabled(true);
        affich_dock->setEnabled(true);
        dock->setVisible(false);
        questionnaire->setEnabled(true);
        image->setFlags(1);
    }

}

void fenetre::setCouleur()
{
    QString r,g,b;
    r= QString::number(qRed(image-
>getCouleur()));g=QString::number(qGreen(image->getCouleur()));
b=QString::number(qBlue(image->getCouleur()));
    couleur="background-color: rgb("+r+", "+g+", "+b+)";
    label->setStyleSheet(couleur);
    update();

}

void fenetre::valider_dec()
{
    std::cout<<la->value()<<" "<<lo->value()<<std::endl;
    double a = la->value();
    double b = lo->value();
    double c = la1->value();
    double d = lo1->value();
    //std::cout<<"latitude " <<a<<" "<<b<<std::endl;
    //std::cout<<"longitude " <<c<<" "<<d<<std::endl;
    if ((a!=0)|| (a!=0)|| (b!=0)|| (c!=0)) {
        exp->setEnabled(true);
        expo->setEnabled(true);
        image->setCoordDec(a,b,c,d);
    } else QMessageBox::critical(this, "Attention", trUtf8("Vous devez entrer
des coordonnées"));
    /*SB->setVisible(true);
    QLabel *lat = new QLabel (SB);
    lat->
    QLabel *lon = new QLabel (trUtf8("Latitude du point 1 :"+image-
>getCoordDec().toSexaLa(b));
    QLabel *lat1 = new QLabel (trUtf8("Latitude du point 1 :"+image-
>getCoordDec1().toSexaLa(c));
    QLabel *lon1 = new QLabel (trUtf8("Latitude du point 1 :"+image-
>getCoordDec1().toSexaLa(d));

```

```

        SB->addWidget(lat);
        SB->addWidget(lon);
        SB->addWidget(lat1);
        SB->addWidget(lon1);*/

    }

    void fenetre::valider_sexa()
    {
        int a = d1->value();
        int b = m1->value();
        int c = s1->value();
        int d = dd1->value();
        int e = mm1->value();
        int f = ss1->value();
        int g = d2->value();
        int h = m2->value();
        int i = s2->value();
        int j = dd2->value();
        int k = mm2->value();
        int l = ss2->value();
        if ((a!=0) || (b!=0) || (c!=0) || (d!=0) || (e!=0) || (f!=0) || (g!=0) || (h!=0) || (i!
=0) || (j!=0) || (k!=0) || (l!=0)) {
            exp->setEnabled(true);
            expo->setEnabled(true);
            image->setCoordSeg(a,b,c,d,e,f,g,h,i,j,k,l);
        } else QMessageBox::critical(this, "Attention", trUtf8("Vous devez
entrer des coordonnées"));
    }

    void fenetre::ignorer()
    {
        affich_dock->setEnabled(true);
        dock->setVisible(false);
        image->setFlags(1);
    }

    void fenetre::afficher_dock()
    {
        if (flag_dock==false) {
            dock->setVisible(true);
            if (image->getFlags()==1)
                image->setFlags(2);

            flag_dock=true;
        } else if (flag_dock==true) {
            dock->setVisible(false);
            if (image->getFlags()==2)
                image->setFlags(1);

            flag_dock=false;
        }
    }

    void fenetre::svg()
    {
        image->sauvegarde_sous();
    }

```



```

}

void fenetre::svg_as()
{
    image->setTest_enregistrer(false);
    image->sauvegarde_sous();
}

void fenetre::charger()
{
    image->charger();
    if (image->test_carte()==true){
        save_as->setEnabled(true);
        save->setEnabled(true);
        effacer->setEnabled(true);
        zoom_in->setEnabled(true);
        zoom_out->setEnabled(true);
        affich_dock->setEnabled(true);
        dock->setVisible(false);
        gestionnaire->setEnabled(true);
        la->setValue(image->getCoordDec().getLatitude());
        lo->setValue(image->getCoordDec().getLongitude());
        la1->setValue(image->getCoordDec1().getLatitude());
        lo1->setValue(image->getCoordDec1().getLongitude());
    }
}

void fenetre::fermerProjet()
{
    image->fermerProjet();
    image->setTest_carte(false);
    save_as->setEnabled(false);
    save->setEnabled(false);
    effacer->setEnabled(false);
    zoom_in->setEnabled(false);
    zoom_out->setEnabled(false);
    affich_dock->setEnabled(false);
    dock->setVisible(false);
    gestionnaire->setEnabled(false);
    couleur= "background-color: rgb(255,255,255);";
    label->setStyleSheet(couleur);

    la->setValue(0);
    lo->setValue(0);
    la1->setValue(0);
    lo1->setValue(0);

    d1->setValue(0);
    m1->setValue(0);
    s1->setValue(0);
    dd1->setValue(0);
    mm1->setValue(0);
    ss1->setValue(0);
    d2->setValue(0);
    m2->setValue(0);
    s2->setValue(0);
    dd2->setValue(0);
    mm2->setValue(0);
    ss2->setValue(0);

    image->setFlags(0);

```

```
    update();  
}
```