```cpp
#ifndef CARTE_H
#define CARTE_H

#include <QtGui/QtGui>
#include <QtCore/QtCore>
#include "coord_decimal.h"
#include "point_gps.h"

class carte: public QWidget
{
    Q_OBJECT

private:

    QImage* imageCarte;
    QImage* tracerChemin;
    QImage* tracerChemin2;
    QImage* imageAffichage;
    QImage* copieTailleNormale;
    QImage* p1;
    QImage* p2;

    QMouseEvent *event;

    QPoint point_click;
    QPoint point_depart;
    QPoint point_release;
    QPoint point1_gps;
    QPoint point1;
    QPoint point2_gps;
    QPoint point2;

    QRgb coul;

    QLabel* valeurZoom;

    coord_decimal dec;
    coord_decimal dec1;

    QStack<QPoint>pile;
    QStack<QPoint>pile_release;

    QString md5;
    QString source;
    QString source_chemin;

    bool carteDessiner;
    bool coord_gps;
    bool enregistrer;
    bool tracer;

    float echelle;

    int nbpoint;
    int hauteur;
    int largeur;
    int flags;
    int etendueZone;
```

```
public:
    carte();
    ~carte();
    bool getCarteDessiner();
    void setCartedessiner(bool choix);
    void afficherCarte(QString chemin);
    void zoom(float valeur);

    QRgb getCouleur();
    void setCouleur(QRgb c);

    QPoint getPoint1();
    QPoint getPoint2();
    void setPoint1(QPoint p);
    void setPoint2(QPoint p);

    QPoint getPoint();
    void setPoint(QPoint p);

    coord_decimal getCoordDec();
    coord_decimal getCoordDec1();
    void setCoordDec(double la, double lo, double la1, double lo1);
    void setCoordSeg(int d1,int m1, double s1, int dd1, int mm1,double ss1, int
d2, int m2, double s2, int dd2, int mm2, double ss2);

    bool comparerCouleurAvecMarge(QRgb p1, QRgb p2);
    int maximum(int a, int b);
    int minimum(int a, int b);
    int tracerZone(const QPoint &p, const QRgb &color);
    void parcoursImageAffichage();
    QPoint directionChemin();

    int getFlags();
    void setFlags(int f);

    point_gps pt_gps (QPoint a, QPoint b,QPoint c);

    void sauvegarde_sous();

    bool test_carte();
    void setTest_carte(bool b);
    bool test_enregistrer();
    void setTest_enregistrer(bool b);

    void calcul_md5(QString src);
    void charger();

signals:
    void ChangeRes();
    void changeRes2(const QPoint &p);
    void ChangeZoom();
    void ChangeZoomIn();
    void signalDessinerChemin(const QPoint &p);
    void SignalFlag(const QPoint &p);

public slots:
    void augmenter_zoom();
    void diminuer_zoom();
    void fermerProjet();
    void setNbpoint ();
    void dessinerChemin(const QPoint &p);
    void attributCouleur(const QPoint &p);
    void sauvegardeItineraire(const QPoint &p);
    void placerFlag1(const QPoint &p);
    void placerFlag2(const QPoint &p);
    void mousePressEvent(QMouseEvent *event);
    void mouseReleaseEvent(QMouseEvent *event);
    void wheelEvent(QWheelEvent *event);
    void paintEvent(QPaintEvent *event);
    void exporter_gpx();

};

#endif // CARTE_H
```