

```

/*
 * File: tracker.h
 * Author: Mike
 *
 * Created on June 22, 2014, 11:08 AM
 */

#include <stdlib.h>
#include <stdio.h>
#include "opencv2/opencv.hpp"
//#include </usr/local/include/opencv2/opencv.hpp>
//#include "opencv2/nonfree/nonfree.hpp"
#include <cmath>
#include <cstdio>
#include <list>

#include "PotentialBug.h"
#include "LineDesc.h"
#include "GlobalDefinitions.h"
using namespace cv;
using namespace std;

#ifndef TRACKER_H
#define TRACKER_H

class tracker {
public:
    tracker();
    tracker(const tracker& orig);
    virtual ~tracker();
    tracker(Mat *imgOut);

    void evaluatePoint(PotentialBug *pb, unsigned int frameCount);
    void cleanupPotentialBugs(unsigned int currentFrame);
    virtual void writeATrack(int index, int mAvr, double fps, char * baseFileName, char *trackname);
    unsigned int getNumTracks();
    unsigned int getTrackSize(unsigned int j);
    unsigned int getInitialTrackFrame(unsigned int j);
    unsigned int getFinalTrackFrame(unsigned int j);
    double getTrackDistance(unsigned int j, int mAvr);
    double getTrackDegreesTurned(unsigned int j, int mAvr);
    double getTrackAverageDegreesTurned(unsigned int j, int mAvr);
    double getTrackAverageDistance(unsigned int j, int mAvr);
    unsigned int findNearestTrackToPoint(Point2f p);
    double getPointDegreesTurned(unsigned int j, unsigned int k, int mAvr);
    double getPointSignedDegreesTurned(unsigned int j, unsigned int j1, int mAvr);
    double getTrackUnweightedDegreesTurned(unsigned int j, int mAvr);
    double getTrackAverageUnweighterDegreesTurned(unsigned int j, int mAvr);

```

```
protected:
    Mat *img;
    vector<vector<PotentialBug> > tracks;

private:
    list<PotentialBug> potentialBugs;
    unsigned int nTracks;
    unsigned int nPointsInTrack;
    int randomInt(int min, int max);
    list<PotentialBug>::iterator findNearestNeighbor(PotentialBug& pb, double *MinDis2, double
current, bool newPoint);
};

#endif /* TRACKER_H */
```