

9/1/2015

ایده ی کلی برنامه به این صورت است که تعدادی نقطه با مختصاتشان در واقعیت را در حافظه نگه می داریم. با توجه به مکان دوربین آنها جا به جا می کنیم یعنی اگر دوربین ۵ متر به سمت چپ برود همه ی نقطه ها را ۵ متر به سمت راست می بریم. همچنین اگر دوربین بچرخد نقطه ها را خلاف جهت چرخش دوربین می چرخانیم. سپس با توجه به پارامتر های دوربین آنها را روی صفحه ی نمایش دوربین تصویر می کنیم.

`dots[]`: آرایه ای از DOT هاست که مختصات واقعی نقطه ها را نگه می دارد.

`dots0[]`: آرایه ای از DOT هاست که مختصات نقطه ها نسبت به وضعیت دوربین را نگه می دارد.

`imdots[]`: آرایه ای از DOT هاست که مختصات نقطه ها روی صفحه ی تصویر دوربین را نگه می دارد.

`class MyGLWidget:`

این کلاس مربوط به یک شی از نوع `QGLWidget` است. این ویجت تصویر دوربین را به صورت گرافیکی نشان می دهد.

`MyGLWidget::camTrans():`

با توجه به چرخش دوربین و مکان آن نقطه های آرایه ی `dots[]` را تغییر داده و در `dots0[]` می ریزد.

`MyGLWidget::Intrinsics():`

نقطه های آرایه ی `dots0[]` را با توجه به پارامتر های دوربین روی صفحه ی نمایش تصویر می کند و در آرایه ی `imdots` می ریزد.

`MyGLWidget::makeTherCar():`

با توجه به پارامتر هایی که در UI برنامه وارد می کنیم ۸ نقطه ی مربوط به مکعب نشان دهنده ی ماشین را می سازد و در `dots` می ریزد.

`MyGLWidget::makeTheRec():`

با توجه به پارامتر هایی که در UI برنامه وارد می کنیم ۴ نقطه ی مربوط به مستطیل روی جاده را می سازد و در `dots` می ریزد.

`MyGLWidget::makeThePLK():`

با توجه به پارامتر هایی که در UI برنامه وارد می کنیم ۴ نقطه ی مربوط به مستطیل پلاک را می سازد و در `dots` می ریزد.

`MyGLWidget::makeThePLK():`

با توجه به پارامتر هایی که در UI برنامه وارد می کنیم ۴ نقطه ی مربوط به مستطیل پلاک را می سازد و در `dots` می ریزد.

`MyGLWidget::makeTheContours():`

فایل مربوط به مختصات نقاط کانتور پلاک را می خواند و آن را `parse` می کند. آن ها را در `counters[][]` می ریزد که بعد نخست آن مربوط به شماره ی هر شی درون کانتور است.

MyGLWidget::paintGL():

تابعی است که هر موقع نیاز بود تصویر glwidget به روز شود فراخوانی می شود. در این تابع با کمک توابعی از OpenGL می توانیم تصاویر را نشان دهیم برای مثال برای نشان دادن خطوط روی جاده دو نقطه ی انتهایی آنها را که در dots ذخیره کرده بودیم و به کمک camtrans و intrinsics آنها را روی imdots[] تصویر کرده بودیم را به تابع glVertex3f می دهیم و در صفحه ی تصویر می کشیم. مثال:

```
glBegin(GL_LINES);

glColor3f(0.0f,1.0f,0.0f);

for(int i=0; i<1800; i=i+2){

    glVertex3f(-(imdots[i].x/w)*2, -(imdots[i].y/h)*2, 0);

    glVertex3f(-(imdots[i+1].x/w)*2, -(imdots[i+1].y/h)*2, 0);

}

glEnd();
```

MyGLWidget::resizeaa(int height, int width):

این تابع اندازه ی قابی که glwidget درونش قرار گرفته را می گیرد و سایز آن را با توجه به قاب مقیاس بندی می کند.

MyGLWidget::mousePressEvent(QMouseEvent *event):

در صورت کلیک بر روی هر glwidget فراخوانی می شود و entry های ورودی در UI را برای آن glwidget نشان می دهد.

MyGLWidget::doubleClicked():

در صورت دابل کلیک بر روی هر glwidget فراخوانی می شود و اندازه آن widget را تمام صفحه می کند و سایر widget ها را پنهان می کند. اگر دوباره روی آن دابل کلیک کنیم شرایط را به حالت قبل بر می گرداند.

class MainWindow:

کلاس اصلی مربوط به UI است. در این کلاس یک vector از glwidget های که قرار است در آن نمایش بدهیم داریم.

MainWindow::createRoad():

با توجه به مشخصات وارد شده در UI، مختصات خطوط روی جاده را مقدار می دهد و در glwidget مربوطه وارد می کند.

MainWindow::createRoad():

با توجه به مشخصات وارد شده در UI، مختصات خطوط روی جاده را مقدار می دهد و در glwidget مربوطه وارد می کند.

MainWindow::on_pushButton_clicked():

هر زمانی که نیاز شد glwidget به روز شود (مثلاً زمانی که دکمه ی enter را فشار می دهیم) این تابع اطلاعات وارد شده در entry های UI را می خواند و در glwidget آنها را وارد می کند و در انتها تابع MyGLWidget::paintGL فراخوانی می شود.

MainWindow::on_pushButton_2_clicked():

با توجه به اطلاعات وارد شده در tab ساخت زمانی که دکمه ی "بساز" را فشار می دهیم این تابع فراخوانی می شود و glwidget را مقدار دهی اولیه می کند و نشان می دهد.

MainWindow::on_openfile_clicked():

مربوط به load فایل اطلاعات برنامه است. در این فایل تعداد دوربین ها و شرایط هر دوربین نوشته شده. با بار گذاری این فایل می توانیم تصاویر هر دوربین را ببینیم.

MainWindow::on_savefile_clicked():

اطلاعات مربوط به همه ی دوربین ها را با فرمت مخصوصی که توسط load قابل خواندن است در فایلی ذخیره می کند.

**** برای کامپایل برنامه پوشه ی Files باید در کنار فایل اجرایی exe. در پوشه ی debug قرار داده شود.