Contents

Introduction

Before using this package, make sure, that you have this settings:

```
settings.outformat = "pdf";
settings.render = 0;
settings.prc = false;
```

and specified size of picture by `size3`. Also, you have to wrap your code into function (say `main`) and put `with_geometry3d(main);` after `main` function ends.

Objects types list

The package *geometry3d.asy* is the extension of the module *geometry.asy*. Basically, this package provides you a tools to creare a really nice 3D pictures in solid geometry.

Here is all types, defined in this module
`basis3` – a 3D ray
`curve3` – a 3D ray
`ray3` – a 3D ray
`vector3` – a 3D vector
`line3` – a 3D line
`plane3` – a plane
`sphere3` – a sphere

Temp: all functions

```
void drawAllObjects();
```

this function draws all objects on the scene with front-back feature and is called by default in function `with_geometry3d`.

```
void withGeometry3d(void main());
```

this function is meant to be ending of your programm, executing essential function for drawing figures properly.

```
void add2dFrame();
```

add 2D frame in order to be able to draw a 2D figures

```
void drawCurve(picture pic=currentpicture, curve3 curve, pen
frontpen=currentpen, pen backpen=currentpen+dashed);
```

draw `curve` with pens `frontpen` and `backpen` respectively.

`circle3 circle3(triple A, triple B, triple C);`

returns circumcircle of triangle ABC.

`circle3 incircle3(triple A, triple B, triple C);`

returns incircle of triangle ABC.

`transform3 orthogonalproject(plane3 p);`

returns `transform3`, which projects in direction of normal to the `plane p`.

`triple foot3(triple A, line3 l);`

return the foot of the perpendicular dropped from `triple A` onto the `line3 l`.

`triple foot3(triple A, plane3 p);`

return the foot of the perpendicular dropped from `triple A` to the `plane3 p`.

```
void markrightangle3(triple A, triple B, triple C, real n=5, pen
p=currentpen);
```

marks right angle $\angle ABC$ with `pen p`, size of `real n`.

The type `line3`

The type `sphere3`

Represent sphere `sphere(C,r);` as a circle `Circle(project3(C),r);` from package `graph`.