com._604robotics.robot2012.vision

Class DistanceCalculations

java.lang.Object
    com._604robotics.robot2012.vision.DistanceCalculations

```
public class DistanceCalculations
extends java.lang.Object
```

This code does the 2D-to-3D calculations

Field Summary

Fields

Modifier and Type	Field and Description
static double	**cameraPixelHeight** The size of the Axis camera, in pixels
static double	**cameraPixelWidth** The size of the Axis camera, in pixels

Constructor Summary

Constructors

Constructor and Description
DistanceCalculations()

Method Summary

Methods

Modifier and Type	Method and Description
double	**getAngleOfTarget**(**Quad** q, double z) This function gets the direction the target is facing, relative to the camera.
Target	**getApproximationOfTarget**(**Quad** quad) A method that tries to find the most likely location for the vision target to lie in 3D space
Point3d	**getRelXYZOfTarget**(**Quad** q) Remember that this requires the camera to be "perfectly" flat, and the targets to be "perfectly" vertical.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

cameraPixelHeight

```
public static final double cameraPixelHeight
```

The size of the Axis camera, in pixels

See Also:

Constant Field Values

cameraPixelWidth

```
public static final double cameraPixelWidth
```

The size of the Axis camera, in pixels

See Also:

Constant Field Values

Constructor Detail

DistanceCalculations

```
public DistanceCalculations()
```

Method Detail

getAngleOfTarget

```
public double getAngleOfTarget(Quad q,
                               double z)
```

This function gets the direction the target is facing, relative to the camera. It is imperfect, and half-assumes a simple orthographic projection (which is not quite like real life). If it causes issues (which the accuracy of this function doesn't need to be very high), we can fix it later.

Returns:

the resulting angle in radians.

getApproximationOfTarget

```
public Target getApproximationOfTarget(Quad quad)
```

A method that tries to find the most likely location for the vision target to lie in 3D space

Parameters:

`quad` - a quadrilateral with corners indicating the corners of the target

Returns:

a Target as an estimation of

getRelXYZOfTarget

```
public Point3d getRelXYZOfTarget(Quad q)
```

Remember that this requires the camera to be "perfectly" flat, and the targets to be "perfectly" vertical. A new function will probably need to be created for use on the robot. That, or we'll need to manipulate the points based on camera angle.

Returns:

a Point3d holding the X, Y, and Z of the target, relative to the camera.