Overview Package Class Tree Deprecated Index Help

Prev Class Next Class Frames No Frames All Classes

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

com._604robotics.robot2012.physics

Class Physics

java.lang.Object

com._604robotics.robot2012.physics.Physics

public class Physics
extends Object

Used for determining launch velocities of the ball. It gives velocity as a function of displacement and final vertical velocity

Author:

Kevin Parker

Constructor Summary

Constructors

Constructor and Description

Physics()

Method Summary

Methods	
Modifier and Type	Method and Description
Point2d	<pre>betterVersionOfgetFiringVelocity(double distH, double distV)</pre>
	This function guesses a good vertical velocity to enter the hoop, then determines the firing velocities (and time) for a given distance (horizontally, and vertically).
Point2d	<pre>betterVersionOfgetFiringVelocity(double distH, double distV, double verticalVel)</pre>
	This function determines the firing velocities (and time) for a given distance (horizontally, and vertically) and a vertical velocity at which the ball should enter the hoop.
BallFireInfo	<pre>GetBallFiringInfo(double xDist, double yDist, double zDist, double robotVelX, double robotVelZ)</pre>
	This function will determine how to fire the ball if the shooter only has 2 vertical angles.
double	<pre>getSubparFiringVelocity(double distH, double distV, double slope)</pre>
	This untested function might determine the firing velocity for a given distance (horizontally, and vertically) and the angle of the shooter.
static double	velToPow(double vel)
	Returns an approximation of the power the shooter should be spun at

Methods inherited from class java.lang.Object

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

Constructor Detail

Physics

public Physics()

Method Detail

velToPow

public static double velToPow(double vel)

Returns an approximation of the power the shooter should be spun at

Parameters:

ı arameterə.

vel - - velocity, in inches/second

Returns:

the power to spin the shooter wheel at

getSubparFiringVelocity

This untested function might determine the firing velocity for a given distance (horizontally, and vertically) and the angle of the shooter.

Parameters:

```
distH - Horizontal distance the ball must travel.
```

distV - Vertical distance the ball must travel.

slope - What slope the launcher is at.

Returns:

The firing velocity

betterVersionOfgetFiringVelocity

This function determines the firing velocities (and time) for a given distance (horizontally, and vertically) and a vertical velocity at which the ball should enter the hoop.

Parameters:

distH - Horizontal distance the ball must travel.

distV - Vertical distance the ball must travel.

verticalVel - Velocity at which the ball should enter the hoop.

Returns:

A Point2d with the x and y velocities does not return the time.

betterVersionOfgetFiringVelocity

This function guesses a good vertical velocity to enter the hoop, then determines the firing velocities (and time) for a given distance (horizontally, and vertically).

Parameters:

distH - Horizontal distance the ball must travel.

 $\verb"distV-Vertical distance" the ball must travel$

Returns:

A Point2d with the x and y velocities does not return the time.

GetBallFiringInfo

This function will determine how to fire the ball if the shooter only has 2 vertical angles.

Parameters:

```
{\tt xDist} - Left-right distance of the target.
```

yDist - Vertical distance of the target.

 ${\tt zDist}$ - Depth distance of the target.

robotVelX - Current velocity (x axis) of the robot.

robotVelz - Current velocity (z axis) of the robot

Returns:

A BallFireInfo with the velocity, angle, and horizontalAngle to fire the ball at (eventually)

