Team Godspeed

Generated by Doxygen 1.8.20

1 Namespace Index 1

1.1 Namespace List 1

2 Class Index 3

2.1 Class List 3

3 Namespace Documentation 5

3.1 godspeed Namespace Reference 5

3.1.1 Detailed Description 5

3.2 godspeed::BehaviorManager Namespace Reference 6

3.2.1 Detailed Description 6

3.3 godspeed::behaviors Namespace Reference 6

3.3.1 Detailed Description 7

3.4 godspeed::Binder Namespace Reference 7

3.4.1 Detailed Description 8

3.5 godspeed::framework Namespace Reference 8

3.5.1 Detailed Description 8

3.6 godspeed::inputs Namespace Reference 8

3.6.1 Detailed Description 9

3.7 godspeed::inputs::BallStorage Namespace Reference 9

3.7.1 Detailed Description 9

3.8 godspeed::inputs::RangeFinderSuite Namespace Reference 9

3.8.1 Detailed Description 10

3.9 godspeed::inputs::RemoteController Namespace Reference 10

3.9.1 Detailed Description 10

3.10 godspeed::inputs::VisionSensor Namespace Reference 10

3.10.1 Detailed Description 11

3.10.2 Function Documentation 12

3.10.2.1 BallCount() 12

3.10.2.2 BallDistance() 12

3.10.2.3 BallSize() 12

3.10.2.4 BallXOffset() 12

3.10.2.5 BallYOffset() 12

3.10.2.6 GoalCount() 13

3.10.2.7 GoalDistance() 13

3.10.2.8 GoalSize() 13

3.10.2.9 GoalXOffset() 13

3.10.2.10 GoalYOffset() 13

3.10.2.11 Snapshot() 13

3.11 godspeed::outputs Namespace Reference 14

3.11.1 Detailed Description 14

3.12 godspeed::outputs::BallCollector Namespace Reference 14

3.12.1 Detailed Description 14

3.13 godspeed::outputs::BallScorer Namespace Reference 14
3.13.1 Detailed Description 15
3.14 godspeed::outputs::OmniDrive3Wheel Namespace Reference 15
3.14.1 Detailed Description 16
3.15 godspeed::outputs::OutputUtilities Namespace Reference 16
3.15.1 Detailed Description 16
3.15.2 Function Documentation 16
3.15.2.1 setMotorSpeed() 16

4 Class Documentation **17**
4.1 godspeed::inputs::PathScript Class Reference 17
4.1.1 Detailed Description 18

Index **19**

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

godspeed
The primary namespace for the project 5
godspeed::BehaviorManager
A namespace containing functions and logic for managing autonomous behaviors 6
godspeed::behaviors
A namespace containing functions for creating autonomous mode behavior bindings 6
godspeed::Binder
A namespace containing functions for binding value producing functions to value consuming functions 7
godspeed::framework
Contains core classes and functions used as building blocks for the rest of the project 8
godspeed::inputs
Contains all classes and namespaces for input devices 8
godspeed::inputs::BallStorage
A namespace with a function for accessing the current number of balls stored in the robot . . . 9
godspeed::inputs::RangeFinderSuite
A namespace containing functions to access the distance values from all the rangefinders . . . 9
godspeed::inputs::RemoteController
A namespace containing functions for accessing all remote controller inputs 10
godspeed::inputs::VisionSensor
A namespace containing functions for accessing data from the Vision Sensor 10
godspeed::outputs
Contains all classes and namespaces for output devices 14
godspeed::outputs::BallCollector
A namespace containing functions for controlling the ball collector 14
godspeed::outputs::BallScorer
A namespace containing functions for controlling the ball scorer 14
godspeed::outputs::OmniDrive3Wheel
A namespace with functions for controlling the drive train 15
godspeed::outputs::OutputUtilities
A namespace for utility functions used by output classes 16

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

godspeed::inputs::PathScript
A class for creating scripted paths for the robot to take 17

Chapter 3

Namespace Documentation

3.1 godspeed Namespace Reference

The primary namespace for the project.

Namespaces

- BehaviorManager

  A namespace containing functions and logic for managing autonomous behaviors.
- behaviors

  A namespace containing functions for creating autonomous mode behavior bindings.
- Binder

  A namespace containing functions for binding value producing functions to value consuming functions.
- framework

  Contains core classes and functions used as building blocks for the rest of the project.
- inputs

  Contains all classes and namespaces for input devices.
- outputs

  Contains all classes and namespaces for output devices.

Functions

- void LoadBehaviorTest ()

  Loads the tiers, behaviors, and conditions for an autonomous test [FOR TESTING AND DEBUGGING].
- void LoadBehaviorStack ()

  Loads the tiers, behaviors, and conditions for autonomous mode [UNFINISHED].

3.1.1 Detailed Description

The primary namespace for the project.

3.2 godspeed::BehaviorManager Namespace Reference

A namespace containing functions and logic for managing autonomous behaviors.

Functions

- int AddTier ()

  Add a tier and return the ID of the tier created.
- void AddTiers (int num)

  Add a number of tiers.
- void AddBehavior (int tier, int binding_id)

  Add a behavior.
- void AddCondition (int tier, bool(∗condition)(void))

  Add a condition.
- void Update ()

  Update function called by the Behavior Manager thread.
- void Init ()

  Run the Behavior Manager on it's own thread.

Variables

- std::list< std::list< bool(∗)(void)> > **conditions** = std::list<std::list<bool(∗)(void)>>()
- std::list< BEHAVIOR_TUPLE > **behaviors** = std::list<BEHAVIOR_TUPLE>()

3.2.1 Detailed Description

A namespace containing functions and logic for managing autonomous behaviors.

3.3 godspeed::behaviors Namespace Reference

A namespace containing functions for creating autonomous mode behavior bindings.

Functions

- double AlignPipe (double d)

  A pipe function used in the aligning behaviors.
- double Stop ()

  A source function used in some behaviors.
- double LocateSpeed ()

  A source function used in locate object behaviors.
- double BallScorerSpeed ()

  A source function used in the score ball behavior.
- double BallPickupSpeed ()

  A source function used in the ball pickup behavior.
- double ForwardSpeed ()

A source function used in the move forward behavior.

- int AlignWithBall ()

  A behavior that turns the robot to face the largest ball found.
- int AlignWithGoal ()

  A behavior that turns the robot to face the largest goal backboard icon found.
- int MoveForward ()

  A behavior that moves the robot forward.
- int LocateObject ()

  A behavior that turns the robot in place in order to locate an object.
- int ScoreBall ()

  A behavior that turns the center tread to move a ball into the goal.
- int PickUpBall ()

  A behavior that turns the ball collector treads in order to pickup a ball.
- int AvoidObstacle ()

  A behavior to move the robot away from a detected obstacle [NOT IMPLEMENTED].
- int Wander ()

  A behavior to move the robot around in a random path [NOT IMPLEMENTED].
- int StopX ()

  A behavior that sets the robots X-speed to zero.
- int StopY ()

  A behavior that sets the robots Y-speed to zero.
- int StopTurn ()

  A behavior that sets the robots angular speed to zero.
- int StopCollectors ()

  A behavior that sets the ball collector treads speeds to zero.
- int StopScorer ()

  A behavior that sets the center tread speed to zero.

3.3.1 Detailed Description

A namespace containing functions for creating autonomous mode behavior bindings.

When called, behavior functions create a binding with the Binder and then return the ID of the binding created.

3.4 godspeed::Binder Namespace Reference

A namespace containing functions for binding value producing functions to value consuming functions.

Functions

- int Bind (double(∗source)(void), void(∗sink)(double))

  Bind a source function to a sink function.
- int Bind (double(∗source)(void), double(∗pipe)(double), void(∗sink)(double))

  Bind a source function to a sink function through a pipe function.
- void Disable (int id)

  Disable a binding using the bindings ID.
- void Enable (int id)

Re-enable a disabled binding using the bindings ID.
- bool IsDisabled (int id)

  Check if a binding is disabled using the bindings ID.
- void Update ()

  calls all bindings that are not disabled. This does NOT need to be called manually.
- void Init ()

  Runs the binder update function on it's own thread.

Variables

- std::list< BINDING_TUPLE > bindings = std::list<BINDING_TUPLE>()

  A list of tuples. Each tuple represents a binding.
- std::list< int > disabled = std::list<int>()

  A list of disabled binding IDs.

3.4.1 Detailed Description

A namespace containing functions for binding value producing functions to value consuming functions.

There are 3 types of functions considered here, source functions which take no values and return a double, pipe functions which take a double and return a double, and sink functions which take a double and return nothing. The Bind function allows you to bind the output of a source function to a sink function (optionally through a pipe function). Then, once the Binder is initialized, it will go through and repeatedly call the sink functions with the value produced by the source functions as the input.

3.5 godspeed::framework Namespace Reference

Contains core classes and functions used as building blocks for the rest of the project.

3.5.1 Detailed Description

Contains core classes and functions used as building blocks for the rest of the project.

3.6 godspeed::inputs Namespace Reference

Contains all classes and namespaces for input devices.

Namespaces

- BallStorage

  A namespace with a function for accessing the current number of balls stored in the robot.
- RangeFinderSuite

  A namespace containing functions to access the distance values from all the rangefinders.
- RemoteController

  A namespace containing functions for accessing all remote controller inputs.
- VisionSensor

  A namespace containing functions for accessing data from the Vision Sensor.

Classes

- class PathScript

  A class for creating scripted paths for the robot to take.

3.6.1 Detailed Description

Contains all classes and namespaces for input devices.

3.7 godspeed::inputs::BallStorage Namespace Reference

A namespace with a function for accessing the current number of balls stored in the robot.

Functions

- void inc ()

  Increment the ball count. This does NOT need to be called manually.
- void dec ()

  Decrement the ball count. This does NOT need to be called manually.
- double BallCount ()

  Returns the current ball count.
- void Init ()

  Setup the ball storage counter to track balls.

Variables

- int BallCounter = 0

  A variable for tracking the ball count.

3.7.1 Detailed Description

A namespace with a function for accessing the current number of balls stored in the robot.

The Init() function must be called before the ball storage tracking can begin. That function attaches the inc and dec function as callbacks for when bumper A and bumper B are pressed (respectively).

3.8 godspeed::inputs::RangeFinderSuite Namespace Reference

A namespace containing functions to access the distance values from all the rangefinders.

Functions

- double RangeFinder ()

  returns the distance value of Range Finder 1

3.8.1 Detailed Description

A namespace containing functions to access the distance values from all the rangefinders.

3.9 godspeed::inputs::RemoteController Namespace Reference

A namespace containing functions for accessing all remote controller inputs.

Functions

- double **UpButton** ()
- double **DownButton** ()
- double **RightButton** ()
- double **LeftButton** ()
- double **XButton** ()
- double **YButton** ()
- double **AButton** ()
- double **BButton** ()
- double **RightTrigger** ()
- double **RightBumper** ()
- double **LeftTrigger** ()
- double **LeftBumper** ()
- double **LeftStickX** ()
- double **LeftStickY** ()
- double **RightStickX** ()
- double **RightStickY** ()

3.9.1 Detailed Description

A namespace containing functions for accessing all remote controller inputs.

3.10 godspeed::inputs::VisionSensor Namespace Reference

A namespace containing functions for accessing data from the Vision Sensor.

Functions

- int Snapshot (signature sig)

  Take a snapshot, looking for the given signature.
- double XOffset ()

  Get the X offset of the largest object from the center of the screen, normalize to between -1 and 1.
- double YOffset ()

  Get the Y offset of the largest object from the center of the screen, normalize to between -1 and 1.
- double Size ()

  Returns the width, in pixels, of the largest object.
- double Distance ()

  Returns the distance, in inches, to the largest object [NOT IMPLEMENTED].
- double BallDistance ()

  Returns the distance, in inches, to the largest ball found [NOT IMPLEMENTED].
- double BallXOffset ()

  Returns the X offset of the largest ball from the center of the screen, normalize to between -1 and 1.
- double BallYOffset ()

  Returns the Y offset of the largest ball from the center of the screen, normalize to between -1 and 1.
- double BallSize ()

  Returns the width, in pixels, of the largest ball found.
- double GoalSize ()

  Returns width, in pixels, of the largest goal backboard icon found.
- double BallCount ()

  Returns the number of balls found.
- double GoalCount ()

  Returns the number of goals found.
- double GoalDistance ()

  Returns the distance, in inches, to the largest goal backboard icon found [NOT IMPLEMENTED].
- double GoalXOffset ()

  Returns the X offset of the largest goal backboard icon from the center of the screen, normalize to between -1 and 1.
- double GoalYOffset ()

  Returns the Y offset of the largest goal backboard icon from the center of the screen, normalize to between -1 and 1.

Variables

- double ScreenWidth = 315

  Width of the screen in pixels.
- double ScreenHeight = 210

  Height of the screen in pixels.
- double **ballCount** = 0
- double **goalCount** = 0
- double CountSmoothing = 0.125

  Smoothing applied to ball counting.
- double **ballSize** = 0
- double **goalSize** = 0
- double SizeSmoothing = 0.5

  Smoothing applied to size of largest object.

3.10.1 Detailed Description

A namespace containing functions for accessing data from the Vision Sensor.

3.10.2 Function Documentation

3.10.2.1 BallCount()

```
double godspeed::inputs::VisionSensor::BallCount ( )
```

Returns the number of balls found.

Take a snapshot looking for the ball, take the number of objects found and apply smoothing based on the last count

3.10.2.2 BallDistance()

```
double godspeed::inputs::VisionSensor::BallDistance ( )
```

Returns the distance, in inches, to the largest ball found [NOT IMPLEMENTED].

Take a snapshot looking for the ball, and then return Distance()

3.10.2.3 BallSize()

```
double godspeed::inputs::VisionSensor::BallSize ( )
```

Returns the width, in pixels, of the largest ball found.

Take a snapshot looking for the ball, then call Size() and apply smoothing based on the last value found

3.10.2.4 BallXOffset()

```
double godspeed::inputs::VisionSensor::BallXOffset ( )
```

Returns the X offset of the largest ball from the center of the screen, normalize to between -1 and 1.

Take a snapshot looking for the ball, and then return XOffset()

3.10.2.5 BallYOffset()

```
double godspeed::inputs::VisionSensor::BallYOffset ( )
```

Returns the Y offset of the largest ball from the center of the screen, normalize to between -1 and 1.

Take a snapshot looking for the ball, and then return YOffset()

3.10.2.6 GoalCount()

```
double godspeed::inputs::VisionSensor::GoalCount ( )
```

Returns the number of goals found.

Take a snapshot looking for the goal, take the number of objects found and apply smoothing based on the last count

3.10.2.7 GoalDistance()

```
double godspeed::inputs::VisionSensor::GoalDistance ( )
```

Returns the distance, in inches, to the largest goal backboard icon found [NOT IMPLEMENTED].

Take a snapshot looking for the goal, and then return Distance()

3.10.2.8 GoalSize()

```
double godspeed::inputs::VisionSensor::GoalSize ( )
```

Returns width, in pixels, of the largest goal backboard icon found.

Take a snapshot looking for the goal, then call Size() and apply smoothing based on the last value found

3.10.2.9 GoalXOffset()

```
double godspeed::inputs::VisionSensor::GoalXOffset ( )
```

Returns the X offset of the largest goal backboard icon from the center of the screen, normalize to between -1 and 1.

Take a snapshot looking for the goal, and then return XOffset()

3.10.2.10 GoalYOffset()

```
double godspeed::inputs::VisionSensor::GoalYOffset ( )
```

Returns the Y offset of the largest goal backboard icon from the center of the screen, normalize to between -1 and 1.

Take a snapshot looking for the goal, and then return YOffset()

3.10.2.11 Snapshot()

```
int godspeed::inputs::VisionSensor::Snapshot (
             signature sig )
```

Take a snapshot, looking for the given signature.

This function also prints an 'x' on the controller screen at the location of the center of the largest object recognized in the snapshot.

3.11 godspeed::outputs Namespace Reference

Contains all classes and namespaces for output devices.

Namespaces

- BallCollector

  A namespace containing functions for controlling the ball collector.
- BallScorer

  A namespace containing functions for controlling the ball scorer.
- OmniDrive3Wheel

  A namespace with functions for controlling the drive train.
- OutputUtilities

  A namespace for utility functions used by output classes.

3.11.1 Detailed Description

Contains all classes and namespaces for output devices.

3.12 godspeed::outputs::BallCollector Namespace Reference

A namespace containing functions for controlling the ball collector.

Functions

- void TreadSpeed (double speed)

  Sets the speed of the two collector treads.

3.12.1 Detailed Description

A namespace containing functions for controlling the ball collector.

This namespace contains a single function that corresponds to both collector arm treads.

3.13 godspeed::outputs::BallScorer Namespace Reference

A namespace containing functions for controlling the ball scorer.

Functions

- void TreadSpeed (double speed)

  Sets the speed of the center tread.
- void SpinLeftExpander ()

  Spins the left expander motor to the value stored in expanderVar.
- void SpinRightExpander ()

  Spins the right expander motor to the value stored in expanderVar.
- void ExpanderPosition (double angleDeg)

  Sets the angular position of the motor that extends the ball guide.

Variables

- double **expanderVar**

3.13.1 Detailed Description

A namespace containing functions for controlling the ball scorer.

This class contains two functions, one for the center tread and one for the ball guide expander.

3.14 godspeed::outputs::OmniDrive3Wheel Namespace Reference

A namespace with functions for controlling the drive train.

Functions

- void SetVelocity (double x, double y, double a)

  Set the X, Y, and angular velocities of the drivetrain.
- void SetOrthogonalDirection (int direction)

  Set the orthogonal direction of the robot.
- void XSpeed (double x)

  Sets the x-speed of the drivetrain.
- void YSpeed (double y)

  Sets the y-speed of the drivetrain.
- void AngleSpeed (double a)

  Sets the angular speed of the drivetrain.
- void Forward (double d)

  Set the forward speed of the drivetrain.
- void Backward (double d)

  Set the backward speed of the drivetrain.
- void Right (double d)

  Set the right speed of the drivetrain.
- void Left (double d)

  Set the left speed of the drivetrain.

Variables

- double XSpeedVar

  A variable to track the current desired x-speed.
- double YSpeedVar

  A variable to track the current desired y-speed.
- double AngleSpeedVar

  A variable to track the current desired angular speed.

3.14.1 Detailed Description

A namespace with functions for controlling the drive train.

3.15 godspeed::outputs::OutputUtilities Namespace Reference

A namespace for utility functions used by output classes.

Functions

- void setMotorSpeed (double motorSpeed, motor &m)

  Sets the speed of a motor.

3.15.1 Detailed Description

A namespace for utility functions used by output classes.

3.15.2 Function Documentation

3.15.2.1 setMotorSpeed()

```
void godspeed::outputs::OutputUtilities::setMotorSpeed (
            double motorSpeed,
            motor & m )
```

Sets the speed of a motor.

Takes a value -1 to 1 and sets the velocity appropriately. Has a "dead zone" of -0.1 to 0.1 which it treats as equal to zero (no movement)

Chapter 4

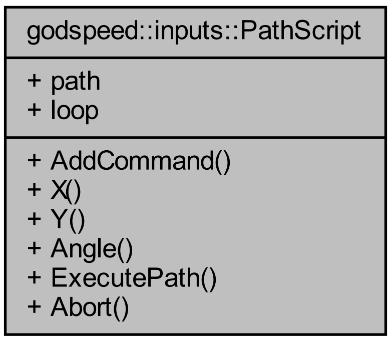
Class Documentation

4.1 godspeed::inputs::PathScript Class Reference

A class for creating scripted paths for the robot to take.

`#include <path-script.h>`

Collaboration diagram for godspeed::inputs::PathScript:



Public Member Functions

- void AddCommand (double x, double y, double a, double duration)

  Adds a command to the path script.

Static Public Member Functions

- static double **X** ()
- static double **Y** ()
- static double **Angle** ()
- static void ExecutePath (PathScript &path)

  Starts execution of the given path.
- static void Abort ()

  Flags the current path to stop execution once the current update cycle is finished.

Public Attributes

- std::list< COMMAND_TUPLE > path

  List of commands of form x-speed, y-speed, angle-speed, duration.
- bool loop

  Set this to true if you wish the path script to repeat after finishing.

4.1.1 Detailed Description

A class for creating scripted paths for the robot to take.

The paths are scripted using "commands" which are tuples consisting of an x-speed, a y-speed, an angular speed, and a duration. First create an instance of PathScript, then add all commands you want, then call PathScript::ExecutePath() passing in the PathScript you just created as the argument. This will immediately begin execution of the path. Path execution proceeds by recursive time-delayed updates which update variables, then whatever is bound the X(), Y(), and Angle() will have access to the updated variables. So the udpates are asyncronous from the actual movement changes and may not be deterministic.

The documentation for this class was generated from the following files:

- C:/Users/anzel/source/repos/team-godspeed/include/inputs/path-script.h
- C:/Users/anzel/source/repos/team-godspeed/src/inputs/path-script.cpp

Index

BallCount
    godspeed::inputs::VisionSensor, 12
BallDistance
    godspeed::inputs::VisionSensor, 12
BallSize
    godspeed::inputs::VisionSensor, 12
BallXOffset
    godspeed::inputs::VisionSensor, 12
BallYOffset
    godspeed::inputs::VisionSensor, 12

GoalCount
    godspeed::inputs::VisionSensor, 12
GoalDistance
    godspeed::inputs::VisionSensor, 13
GoalSize
    godspeed::inputs::VisionSensor, 13
GoalXOffset
    godspeed::inputs::VisionSensor, 13
GoalYOffset
    godspeed::inputs::VisionSensor, 13
godspeed, 5
godspeed::BehaviorManager, 6
godspeed::behaviors, 6
godspeed::Binder, 7
godspeed::framework, 8
godspeed::inputs, 8
godspeed::inputs::BallStorage, 9
godspeed::inputs::PathScript, 17
godspeed::inputs::RangeFinderSuite, 9
godspeed::inputs::RemoteController, 10
godspeed::inputs::VisionSensor, 10
    BallCount, 12
    BallDistance, 12
    BallSize, 12
    BallXOffset, 12
    BallYOffset, 12
    GoalCount, 12
    GoalDistance, 13
    GoalSize, 13
    GoalXOffset, 13
    GoalYOffset, 13
    Snapshot, 13
godspeed::outputs, 14
godspeed::outputs::BallCollector, 14
godspeed::outputs::BallScorer, 14
godspeed::outputs::OmniDrive3Wheel, 15
godspeed::outputs::OutputUtilities, 16
    setMotorSpeed, 16

setMotorSpeed
    godspeed::outputs::OutputUtilities, 16
Snapshot
    godspeed::inputs::VisionSensor, 13