RobotClient API
By Mark Monnin

RobotClient.vb is used to control a robot soccer player. It will make the connection to the server, and send movement commands to the server. It will also keep track of all players and all soccer balls on the field.

Creating the Variable:

For most operations, you will want to create a single variable and use it throughout your program. Example:

```
        Dim robo As New RobotClient
```

Accessing the Remote Server:

Use the **.Connect** method to access the server For the connect, you will need to at least specify (as a string) the system running **RobotSoccer.exe**. Use the string "127.0.0.1" to connect to a server on your own system. Optionally you may also specify a TCP port (but often it is not necessary).

```
        Robo.Connect("127.0.0.1")
```

Use the **.Disconnect** method when you are done with the server (such as when your program wants to exit). Any players that you have on the server will be removed from the field when you do this.

Creating a Robot:

When you first connect to the server, your client will get a list of all existing robots and soccer balls that are on the field. You can use RobotClient to create a viewer, but it's more fun to have something to control. Use **.CreatePlayer** to create a new robot player.

```
        Robo.CreatePlayer()
```

CreatePlayer has two optional arguments to it, but none are necessary. By default, it will create your first player to be on the side that needs another player. Future calls to **CreatePlayer** with no arguments are then created on the same team. If you want to give your player a specific name, pass a string as the first argument. If you want to be on a specific side, pass it that as a second argument.

You may create as many players as you wish. All players will be removed when you disconnect from the server. If you wish to remove a specific player, you may do so by using the **RemovePlayer** method. If you are only controlling one player, you do not need any arguments. If you are controlling more than one, you will have to pass the playerId to this routine.

Moving the Robot:

Your robot has two motors. As you are looking at the board from above, one moves the robot left and right, the other moves the robot up and down. Use the **.SetSpeed** method to change the speeds of the two motors. Speeds can be negative for moving the robot left or up.

```
        Robo.SetSpeed(10,10)     ' Move diagonally
```

Your robot has a maximum speed (controlled by the server) otherwise your motors would explode, so setting a speed above the maximum speed won't be any faster than setting the robot at the maximum speed.

BTW. Your robot obeys (many of) the laws of physics.   If you are going fast and hit something (a soccer ball or other robot), the amount of energy transferred is proportial to the speeds the two objects are going at.

SetSpeed will immediately change the speed of the robot to the new value.   If this is not what you want, you can is a **.ChangeSpeed** method that will allow you to add or subtract the speeds of each of the two motors.

Examining the Field:

When you are ready to have the robots do things on their own, you may use the **Get_*item*_Loc** routines, or access the **allPlayers** array directly.

Simple Example:

```
Dim robo As New RobotClient

Robo.Connect(server)
Robo.CreatePlayer()
Robo.SetSpeed(10,10)
Robo.Disconnect()
```

Definitions:

```
    '
    '       objType
    '
    '       What type of thing is each entry in the array?
    '
Enum objType
        Empty = 0
        RedPlayer = 1
        BluePlayer = 2
        Ball = 3
        RedGoal = 4
        BlueGoal = 5
        Other = 6
End Enum

Enum teamSides
        thedefault = 0   ' The default is first either side, then keep same
        red = 1
        blue = 2
End Enum



    '
    '   Structure to be used to hold a single player
    '   and the relevant graphics parts
    '
Structure onePlayer
        X As Integer                ' Location on board (X)
        Y As Integer                ' Location on board (Y)
        Size As Integer             ' Size of object
        name As String              ' Description (given by person)
        type As objType             ' What is it?
        isMine As Boolean           ' Can I control it?
End Structure

allPlayers(maxArray) As onePlayer        ' All of the things on the field
maxPlayers As Integer = 0                ' What is the max # for the array?
myPlayerNum As Integer = -1              ' Which one of these is mine?
theBall As Integer = -1                  ' Shortcut to finding the ball
ourGoal As Integer = -1                  ' Where I don't want the ball to go
theirGoal As Integer = -1                ' Where I want the ball to go
redScore As Integer                      ' Current score Red side
blueScore As Integer                     ' Current score Blue side
fieldX As Integer                        ' Size of the field (X/width)
fieldY As Integer                        ' Size of the field (Y/height)
tickNum As Integer                       ' What tick # are we on?
secondsLeft As Integer                   ' Time left in the game (not active)
newTick As Boolean = False               ' Is a tick since last SetSpeed?

    '
    '       canMove()
    '
    '       Is it time for a new move?   (and do I exist?)
    '
```

```
ReadOnly Property canMove() As Boolean

    '
    '       HavePlayer()
    '
    '       Do I exist?
    '
ReadOnly Property havePlayer() As Boolean


    '
    '   myTeam()
    '
    '       Look or change the team that I am on
    '       (does not send change to the server, though)
    '
Property myTeam() As teamSides


    '
    '   IsConnected() = Are we connected to the server right now?
    '
ReadOnly Property IsConnected() As Boolean


    '
    '   Get_Score(red,blue)
    '
    '   Return the scores of the two sides
    '
Sub Get_Score(ByRef red As Integer, ByRef blue As Integer)

    '
    '   Connect = start the connection up
    '
Sub Connect(ByVal server As String, Optional ByVal Port As Integer = 8080)

    '
    '   Disconnect = stop playing
    '
Sub Disconnect()


    '
    '   CreatePlayer
    '
    '   Called to create a new player on the playing field
    '
    '   Once a team side is decided, defaults to "same" side for subsequent
    '   players (still can be requested to server to have players on both
    ' teams)
    '
Sub CreatePlayer(Optional ByVal name As String = "",
                  Optional ByVal desiredTeam As teamSides =
                  teamSides.thedefault)
    '
    '   RemovePlayer()
    '
    '   Removes one of my players from the field. If no id specified, then
    '   removes the "known" player
    '
Sub RemovePlayer(Optional ByVal who As Integer = -1)

    '
    '       SetSpeed
    '
    '       Used to control the robot.   Speed will deteriorate if no new
    '       calls to SetSpeed are sent.  Also, only one SetSpeed is observed
    '       for a time tick (the last one).
    '

Sub SetSpeed(ByVal Xspeed As Double, ByVal Yspeed As Double,
            Optional ByVal who As Integer = -1)


    '
    '       ChangeSpeed
    '
    '       Used to control the robot.   Add or remove energy to the speed.
    '
    '

Sub ChangeSpeed(ByVal Xspeed As Double, ByVal Yspeed As Double,
                Optional ByVal who As Integer = -1)



    '
    '       Get_Obj_Loc
    '
    '       Given an id, return the X and Y location of that object
    '
Function Get_Obj_Loc(ByVal who As Integer,
                     ByRef X As Integer, ByRef Y As Integer)

    '
    '       Get_Ball_Loc
    '
    '       Return the X & Y location of the ball
    '
Function Get_Ball_Loc(ByRef X As Integer, ByRef Y As Integer)


    '
    '       Get_Our_Goal_Loc
    '
    '       Return the X & Y location of our goal (where we want to defend)
    '
Function Get_Our_Goal_Loc(ByRef X As Integer, ByRef Y As Integer)


    '
    '       Get_Their_Goal_Loc
    '
    '       Return the X & Y location of their goal (where we want to send
    '       the ball)
    '

Function Get_Their_Goal_Loc(ByRef X As Integer, ByRef Y As Integer)


    '
    '       Get_Ball_Loc
    '
    '       Return the X & Y location of me
    '
Function Get_My_Loc(ByRef X As Integer, ByRef Y As Integer)
```