

The Java Player Client

Esben Østergård

esben@robotics.usc.edu

July 24, 2001

The Java client is similar to the C++ client in architecture.

The client is meant to be used in the following way:

```
1:  Connect to robot by constructing a ‘‘PlayerClient’’ object.
2:  Request devices, generating ‘‘PlayerDevice’’ objects.
3:  do forever {
4:      perform blocking read of data from robot
5-?:  decide action based on data recieved from robot
?+1: }
```

Lines 5 to ? will be repeated forever 10 times a second, since line 4 performs a blocking read of data, and data is sent from the robot 10 times a second.

The following JAVA program makes the robot move forwards in circles forever.

```
1: public class simple_demo {
2:     public static void main(String args[]) {
3:
4:         PlayerClient robot1=new PlayerClient("bug",6665);
5:         PositionPlayerDevice ppd=robot1.requestPosition('a');
6:
7:         while (true) {
8:             robot1.readAll();
9:             ppd.setSpeed( 50 , 10 );
10:        }
11:    }
12: }
```

The object `robot1` of type `PlayerClient` acts as a handle for the specific connection to the robot called “bug”. In line 5 position information requested from the player device. The object `ppd` of type `PositionPlayerDevice` holds methods for setting robot speed commands, and reading odometry. The ‘a’ gives both read and write access to the `PositionPlayerDevice`. Line 7 to 9 are repeated forever. Line 8 performs a blocking read of data from the robot, which returns data every 100 ms, so line 9 is executed 10 times a second. Line 9 sets translation speed to 50 and rotation speed to 10.

1 The PlayerClient class

1.1 The constructor

```
public PlayerClient(String serverName, int portNumber)
```

Where `serverName` is the URL to the server on which player runs (“localhost” if the Java program runs on the robot), and port number is 6665. If *Stage* is used to simulate pioneer robots, then port number should be 6665 plus the ID number of the robot.

1.2 Methods

```
public MiscPlayerDevice requestMisc(char r)
public GripperPlayerDevice requestGripper(char r)
public PositionPlayerDevice requestPosition(char r)
public SonarPlayerDevice requestSonar(char r)
public LaserPlayerDevice requestLaser(char r)
public VisionPlayerDevice requestVision(char r)
public PtzPlayerDevice requestPtz(char r)
public AudioPlayerDevice requestAudio(char r)
```

These methods are used to request access to specific devices. Each of these methods returns an specialized object of type `PlayerDevice`. This object has special methods for communicating with the device.

```
public void readAll()
```

This method performs a blocking read to update all the requested devices. This method should be called once per program cycle.

2 The abstract PlayerDevice class

This class is the parent of all classes for player devices.

The following methods;

```
public int getTimeForDataSampled_sec()
public int getTimeForDataSampled_usec()
public int getTimeForDataSent_sec()
public int getTimeForDataSent_usec()
```

can be used to retrieve information about time when data was sampled, and time when data was sent to the player client.

3 The Device Classes

The following sections describe each of the implemented player devices.

3.1 The AudioPlayerDevice

```
public int[] getFiveHighestFrequencies()
public int[] getFiveHighestAmplitudes()
```

Each returns a list of length 5 which contains the frequency/amplitude respectively of the five highest peaks in frequency domain heard by the robot.

3.2 The GripperPlayerDevice

```
public byte getState()
public byte getBeams()
```

Returns the data bits from the gripper.

```
public void setGripper(byte cmd, byte arg)
```

Can be used to send commands to the gripper

3.3 The LaserPlayerDevice

```
public int[] getRange()
public int[] getReflection()
```

return respectively range data and reflection data for each of the 361 laser scans.

3.4 The MiscPlayerDevice

```
public byte getFrontBumpers()
public byte getReadBumpers()
public byte getVoltage()
```

Returns various information about the robot.

3.5 The PositionPlayerDevice

```
public int getX()
public int getY()
public short getHeading()
public short getSpeed()
public short getTurnrate()
public short getCompass()
public byte stall()
```

Returns various information about the odometry and orientation of the robot.

```
public void setSpeed(int speed, int turnrate)
```

Can be used to set the translational and rotational speed of the robot

3.6 The PtzPlayerDevice

```
public short getPan()  
public short getTilt()  
public short getZoom()
```

Returns either the pan, tilt or zoom of the SONY Ptz camera.

```
public void setPTZ(short pan, short tilt, short zoom)
```

Can be used to command the PTZ camera to a certain position.

3.7 The SonarPlayerDevice

```
public int[] getSonar()
```

Returns 16 int's determining the range reading for each sonar.