Programming Reference

CLARIFICATIONS:

Info objects produced by **getProjectileInfo()** will contain:
- ID
- position (x,y) at time of impact with projectile
- health, speed and direction are **not** set

Info objects contained within your bot's **hitlist** will contain:

- ID
- position (x,y) of the bot who hit your bot at the time it fired its projectile
- health, speed, and direction are **not** set

Info objects contained within your bot's **collidelist** will contain:

- ID
- position (x,y) at time of collision
- health at time of collision
- speed and direction at time of collision
- **Special note:** health, speed and direction for obstacles (inner walls, powerups, and mines) are **not** set.

• Info class Reference

- **arctan**

```
public final double arctan(double x,
                           double y)
```

This method accepts a vector (x,y) and returns an angle from 0 - 2PI. arctan() is a convenience method that calls the java.lang.Math.atan() function and converts the -PI/2 - PI/2 output of atan() to 0 - 2PI using the signs on the components of the vector.

- **ai**

```
public void ai()
```

This is the brains of the bot. ai() should be over-ridden (redefined) to contain your own code.

- **Scan**

```
protected final Vector Scan(double dir)
```

Scanning takes a radian value specifying the direction in which a scan should be made. If values greater than 0 - 2PI are specified, they will be converted to equivalent forms within the above range. The sweep of the

scan will be centered around this value. A Vector is returned containing all *Info objects* of all objects scanned in that region.
The first *Info object* -- elementAt(0) -- will contain information on your own bot. The proceeding *Info objects* will contain, in order, bots, projectiles, and obstacles (powerups/mines/inner-walls).

- **getProjectileInfo**

```
protected final Info getProjectileInfo(int id)
```

This function returns a single *Info object* containing information for the specified projectile id. Projectile id's are returned by successful calls to Fire().

- **Cloak**

```
protected final boolean Cloak(boolean engagingcloaking)
```

Cloak will enable or disable invisibility for a bot. Specify ***true*** to enable and ***false*** to disable. *Cloak()* will return true when the command is successful and false when it fails.

- **Drive**

```
protected final boolean Drive(double dir)
protected final boolean Drive(double x,
                              double y)
protected final boolean Drive(double x,
                              double y,
                              double speed)
```

The first two *Drive()* methods take either a radian direction or a (x,y) vector specifying the direction in which to move. These methods will use a default speed of 5 units per turn. The third *Drive()* method takes a (x,y) vector and then a speed which is from 0-20 units per turn.
Drive() will return ***true*** when successful and ***false*** when it fails.

- **Fire**

```
protected final int Fire(double direction)
protected final int Fire(double x,
                         double y)
```

Fire() accepts either a direction specified in radians or a (x,y) vector. The method returns -1 when it fails and an integer projectile ID which can be used with *getProjectileInfo()* to obtain information on the projectile.

Back to the main JavaBots page.