BoS and BlueJ

Stephan Euler

August 14, 2018

The Board of Symbols (BoS) application comes with a simple integrated development environment (IDE) for writing and executing code snippets. This IDE provides some specific help such as a list of methods and a selector for color names. The IDE should give beginners a quick start. The functionally, however, is limited and sooner or later users will move to a more powerful IDE. As an alternative we can start immediately with another IDE and use BoS only as output device. in this document we describe this approach with BlueJ.

1 Installation

Download the archive `jserver.jar` from github and add it in BlueJ as user library (menu preferences – libraries).

2 Use

Commands can be send to a `Board` with sender classes. These classes provide different languages for the method names. Currently English and German are implemented. The following code shows a minimal example using the English method names.

```
import jserver.*;

public class BoSENApplication
{
    public void draw() {
        XSendAdapterEN sender = new XSendAdapterEN();
        sender.color2( 2, 3, XSend.RED );
    }
}
```

The default class `XSendAdapter` contains the methods with German names. If necessary we can get the board object from `sender` and then send low level commands in BoS language (BoSL) directly to the board:

```
Board board = sender.getBoard();
// change the font size for symbol text
board.receiveMessage( ">>fontsize 32" );
```

BoS beginner can start with small code snippets. These snippets are internally embedded in boilerplate code to build complete classes. In this way the sender mechanism is hidden. We can simulate this approach as follows:

- write a class that extends `XSendEN` (or `XSendDE` if you prefer German method names).
- call the method `newBoard()` to create a board
- implement the abstract method `send()` from `XSendEN`
- place the snippet code in the body of this method

In the example class below we use the constructor to create the board and then change the color of a symbol in `send()`. While it is necessary to implement `send()` we are free to add more methods such as the `hline()` in the example.

```
import jserver.XSendEN;

public class BoSEN extends XSendEN
{
    public BoSEN() {
        newBoard();
    }

    public void send() {
        color2(2,3,BLUE);
    }

    public void hline( int x, int y, int n ) {
        for( int i=0; i<n; i++ ) {
            color2( x+i, y, RED );
        }
    }
}
```