ProkScreen v1

Roman A. Prokopenko (support@prokopenko.org)

Prokopenko.Screen is COM .NET wrapper of some Windows API services to use from MATLAB. Direct graphical output to any monitor attached to the system can be combined in real time with joystick and keyboard input, as well as, with other input devices. This component is designed for human behavioral experiments. MATLAB scripting allows elegant development and modification of experimental protocols.

REQUIREMENTS

- Mathworks MATLAB 6.5 or higher. **No toolboxes required**
- Microsoft Net Framework 2.0 or higher

USAGE

```
h = actxserver('Prokopenko.Screen');       % Activate this component
h.get                                      % List all properties
h.invoke                                   % List all methods
```

After activation of this component you can use its handle H to read and modify properties and invoke methods described below. If you are not familiar with COM programming in MATLAB I recommend MATLAB help (MATLAB/External Interfaces/COM Support in MATLAB).

METHODS

```
[ POSITION, BUTTONS ] = h.joystick(ID);
```

Requests current position and state of buttons of a joystick with the given ID (1,2,..). POSITION is a 3D vector. The vector BUTTONS has 16 binary values for each button.

```
[ PRESSED, STATE ] = h.key(ID);
```

Requests state of a key with the key code ID. The output parameter PRESSED has a non zero value (1) if the key has been pressed since the last call of this method. The output parameter STATE has a non zero value (1) if the key is pressed at current instant. The list of the virtual key codes can be found on the MDSN website http://msdn.microsoft.com/en-us/library/ms927178.aspx

```
h.sleep(SECONDS);
```

Delay the script to release CPU time. The actual delay time is multiple of Windows timer period (1/64 of a sec in my computer).

GRAPHICAL OUTPUT

The methods below use SCREEN argument (1, 2, ..) to select a monitor attached to the computer. COLOR is 3 element RGB vector with values from 0 to 1. All outputs are postponed until call to the DRAW method. Coordinate units have values from 0 to 1 for both X (left to right) and Y (top to bottom) axes. Please take into account that the actual monitor screen is not rectangular and have different vertical and horizontal dimensions.

```
h.pen(THICKNESS,COLOR)
```

Set THICKNESS and COLOR of the current pen.

```
h.brush(COLOR)
```

Set COLOR of the current brush.

```
h.clear(SCREEN,COLOR)
```

Clear the SCREEN with the given COLOR.

```
h.ellipse(SCREEN,POSITION,SIZE)
```

Draw an ellipse of the given SIZE (2D vector) at the given POSITION of its center (2D vector) using current brush and pen.

```
h.rectangle(SCREEN,POSITION,SIZE)
```

Draw a rectangle of the given SIZE (2D vector) at the given POSITION of its center (2D vector) using current brush and pen.

```
h.line(SCREEN,POINTS)
```

Draw a sequence of connected lines with vertices specified in the POINTS argument (N by 2 matrix).

```
h.images(SCREEN,FILENAMES)
```

Load several bitmap images. The FILENAMES argument is a cell of their filenames.

```
h.image(SCREEN,ID,POSITION)
```

Draw an image with the given ID (1,2, ..) at the given POSITION of its center. Image is not scaled and outputs as is.

```
h.draw(SCREEN)
```

Perform the output of the previous drawings to the SCREEN.

LICENSE