mmsv2gui (version 0.3, 24 june 2006)

Classes

- Control
  - ButtonControl
  - LabelControl
  - ImageControl
  - ListControl
  - TextBoxControl
  - RectangleControl

- ListItem
- Window
- Dialog
- DialogProgress

class **ButtonControl(**Control**)**

```
ButtonControl(int x, int y, int width, int height [, text, font, rgb,
              rgbfocus, focus, nofocus, textoffset, alignment])

text           : string (button text)
font           : string (example, 'Vera')
rgb            : hex string (example, '0xffffff')
rgbfocus       : hex string (example, '0xffffff')
focus          : string (filename for focus texture)
nofocus        : string (filename for no focus texture)
textoffset     : integer (text offset in pixels counting from left)
alignment      : string (text alignment, 'left', 'center' or 'right')

Image path for focus and nofocus texture has to be relative script folder
or mmsv2 theme folder.
```

Default values:

```
font='Vera', rgb='0xffffff' (white), rgbfocus='0xffffff' (white),
textoffset=0, alignment='left'
```

Methods defined here:

```
set_label([text, font, rgb, rgbfocus])

text           : string
font           : string (example, 'Vera')
rgb            : hex string (example, '0xffffff')
rgbfocus       : hex string (example, '0xffffff')
```

Methods inherited from Control:

control_next`( Control ) -- Set next control.`

```
Throws:       TypeError, if the supplied argument is not a Control type
              ReferenceError, if the control is not added to a window
```

When this control is active and the user presses next, the supplied Control will receive focus.

control_left(Control) -- Set left control.

```
Throws:       TypeError, if the supplied argument is not a Control type
              ReferenceError, if the control is not added to a window
```

When this control is active and the user presses left, the supplied Control will receive focus.

control_right(Control) -- Set right control.

```
Throws:       TypeError, if the supplied argument is not a Control type
              ReferenceError, if the control is not added to a window
```

When this control is active and the user presses right, the supplied Control will receive focus.

control_prev(Control) -- Set prev control.

```
Throws:       TypeError, if the supplied argument is not a Control type
              ReferenceError, if the control is not added to a window
```

When this control is active and the user presses prev, the supplied Control will receive focus.

set_height(int height) -- Set's the height of this control.

set_position(int x, int y) -- Set's the position of this control.

set_visible(bool visible) -- Hide's or Show's this control.

get_visible() -- Return 0 or 1 (not visible, visible).

set_width(int width) -- Set's the width of this control.

class **LabelControl(**Control**)**

LabelControl(int x, int y, int width, int height [, text, font, rgb, textoffset, alignment])

```
text           : string (label text)
font           : string (example, 'Vera')
rgb            : hex string (example, '0xffffff')
textoffset     : integer (text offset in pixels counting from left)
alignment      : string (text alignment, 'left', 'center' or 'right')
```

Default values:

font='Vera', rgb='0xffffff' (white), textoffset=0, alignment='left'

Methods defined here:

set_label([text, font, rgb])

text : string
font : string (example, 'Vera')
rgb : hex string (example, '0xffffff')

Methods inherited from Control:

control_next(Control) -- Set next control.

Throws: TypeError, if the supplied argument is not a Control type
ReferenceError, if the control is not added to a window

When this control is active and the user presses next, the supplied Control will receive focus.

control_left(Control) -- Set left control.

Throws: TypeError, if the supplied argument is not a Control type
ReferenceError, if the control is not added to a window

When this control is active and the user presses left, the supplied Control will receive focus.

control_right(Control) -- Set right control.

Throws: TypeError, if the supplied argument is not a Control type
ReferenceError, if the control is not added to a window

When this control is active and the user presses right, the supplied Control will receive focus.

control_prev(Control) -- Set prev control.

Throws: TypeError, if the supplied argument is not a Control type
ReferenceError, if the control is not added to a window

When this control is active and the user presses prev, the supplied Control will receive focus.

set_height(int height) -- Set's the height of this control.

set_position(int x, int y) -- Set's the position of this control.

set_visible(bool visible) -- Hide's or Show's this control.

get_visible() -- Return 0 or 1 (not visible, visible).

set_width(int width) -- Set's the width of this control.

class **ImageControl(**Control**)**

ImageControl(int x, int y, int width, int height, path])

path : string (filename for image)

Image path has to be relative script folder or mmsv2 theme folder.

Default values:

Methods defined here:

Methods inherited from Control:

control_next(Control) -- Set next control.

```
Throws:     TypeError, if the supplied argument is not a Control type
            ReferenceError, if the control is not added to a window
```

When this control is active and the user presses next, the supplied Control will receive focus.

control_left(Control) -- Set left control.

```
Throws:     TypeError, if the supplied argument is not a Control type
            ReferenceError, if the control is not added to a window
```

When this control is active and the user presses left, the supplied Control will receive focus.

control_right(Control) -- Set right control.

```
Throws:     TypeError, if the supplied argument is not a Control type
            ReferenceError, if the control is not added to a window
```

When this control is active and the user presses right, the supplied Control will receive focus.

control_prev(Control) -- Set prev control.

```
Throws:     TypeError, if the supplied argument is not a Control type
            ReferenceError, if the control is not added to a window
```

When this control is active and the user presses prev, the supplied Control will receive focus.

set_height(int height) -- Set's the height of this control.

set_position(int x, int y) -- Set's the position of this control.

set_visible(bool visible) -- Hide's or Show's this control.

get_visible() –- Return 0 or 1 (not visible, visible).

set_width(int width) -- Set's the width of this control.

class **ListControl(**Control**)**

ListControl(int x, int y, int width, int height [, font, rgb, rgbfocus,
                focus, nofocus, textoffset, alignment, itemheight,
                itemspace])

font : string (example, 'Vera')
rgb : hex string (example, '0xffffff')
rgbfocus : hex string (example, '0xffffff')
focus : string (filename for focus texture)
nofocus : string (filename for no focus texture)
textoffset : integer (text offset in pixels counting from left)
alignment : string (text alignment, 'left', 'center' or 'right')
itemheight : integer (height of items)
itemspace : integer (space between items)

Image path for focus and nofocus texture has to be relative script folder or mmsv2 theme folder.

Default values:

font='Vera', rgb='0xffffff' (white), rgbfocus='0xffffff' (white), textoffset=0, alignment='left', itemheight=30, itemspace=0

Methods defined here:

set_label([font, rgb, rgbfocus])

font : string (example, 'Vera')
rgb : hex string (example, '0xffffff')
rgbfocus : hex string (example, '0xffffff')

Will change the font and/or rgb of all list items.

add_item(ListItem item) -- Add a new listitem to this list.

Throws: TypeError, if supplied argument is not a ListItem type
ReferenceError, if item is already in list

get_selected_item()-- Returns the selected listitem.

Throws: RuntimeError, if there are no items in list

get_selected_position()-- Returns the current position (integer)

clear()-- Clear all listitems in this list.

set_itemheight(int height) -- Set height of items in this list.

set_itemspace(int space) -- Set space between items in this list.

Methods inherited from Control:

control_next(Control) -- Set next control.

Throws: TypeError, if the supplied argument is not a Control type
ReferenceError, if the control is not added to a window

When this control is active and the user presses next, the supplied Control will receive focus.

control_left(Control) -- Set left control.

Throws: TypeError, if the supplied argument is not a Control type
ReferenceError, if the control is not added to a window

When this control is active and the user presses left, the supplied Control will receive focus.

control_right(Control) -- Set right control.

Throws: TypeError, if the supplied argument is not a Control type
ReferenceError, if the control is not added to a window

When this control is active and the user presses right, the supplied Control will receive focus.

control_prev(Control) -- Set prev control.

Throws: TypeError, if the supplied argument is not a Control type
ReferenceError, if the control is not added to a window

When this control is active and the user presses prev, the supplied Control will receive focus.

set_height(int height) -- Set's the height of this control.

set_position(int x, int y) -- Set's the position of this control.

set_visible(bool visible) -- Hide's or Show's this control.

get_visible() -- Return 0 or 1 (not visible, visible).

set_width(int width) -- Set's the width of this control.

class **TextBoxControl(**Control**)**

TextBoxControl(int x, int y, int width, int height [, font, rgb, textheight, textspace])

font : string (example, 'Vera')
rgb : hex string (example, '0xffffff')
textheight : integer (height of text lines)
textspace : integer (space between text lines)

Default values:

font='Vera', rgb='0xffffff' (white), textheight=30, textspace=0

Methods defined here:

set_text([text, font, rgb])

text : string (text to show)
font : string (example, 'Vera')
rgb : hex string (example, '0xffffff')

clear()-- Clear all text.

set_textheight(int height) -- Set height of text lines.

set_textspace(int space) -- Set space between text lines.

Methods inherited from Control:

control_next(Control) -- Set next control.

```
Throws:     TypeError, if the supplied argument is not a Control type
            ReferenceError, if the control is not added to a window
```

When this control is active and the user presses next, the supplied Control will receive focus.

control_left(Control) -- Set left control.

```
Throws:     TypeError, if the supplied argument is not a Control type
            ReferenceError, if the control is not added to a window
```

When this control is active and the user presses left, the supplied Control will receive focus.

control_right(Control) -- Set right control.

```
Throws:     TypeError, if the supplied argument is not a Control type
            ReferenceError, if the control is not added to a window
```

When this control is active and the user presses right, the supplied Control will receive focus.

control_prev(Control) -- Set prev control.

```
Throws:     TypeError, if the supplied argument is not a Control type
            ReferenceError, if the control is not added to a window
```

When this control is active and the user presses prev, the supplied Control will receive focus.

set_height(int height) -- Set's the height of this control.

set_position(int x, int y) -- Set's the position of this control.

set_visible(bool visible) -- Hide's or Show's this control.

get_visible() -- Return 0 or 1 (not visible, visible).

set_width(int width) -- Set's the width of this control.

class **RectangleControl(**Control**)**

RectangleControl(int x, int y, int width, int height [, alpha, rgb])

```
alpha          : integer
rgb            : hex string (example, '0xffffff')
```

Default values:

alpha=100, rgb='0xffffff' (white)

Methods defined here:

set_color([alpha, rgb])

```
alpha          : integer
rgb            : hex string (example, '0xffffff')
```

Methods inherited from Control:

control_next(Control) -- Set next control.

```
Throws:     TypeError, if the supplied argument is not a Control type
            ReferenceError, if the control is not added to a window
```

When this control is active and the user presses next, the supplied Control will receive focus.

control_left(Control) -- Set left control.

```
Throws:     TypeError, if the supplied argument is not a Control type
            ReferenceError, if the control is not added to a window
```

When this control is active and the user presses left, the supplied Control will receive focus.

control_right(Control) -- Set right control.

```
Throws:     TypeError, if the supplied argument is not a Control type
            ReferenceError, if the control is not added to a window
```

When this control is active and the user presses right, the supplied Control will receive focus.

control_prev(Control) -- Set prev control.

```
Throws:     TypeError, if the supplied argument is not a Control type
            ReferenceError, if the control is not added to a window
```

When this control is active and the user presses prev, the supplied Control will receive focus.

set_height(int height) -- Set's the height of this control.

set_position(int x, int y) -- Set's the position of this control.

set_visible(bool visible) -- Hide's or Show's this control.

get_visible() -- Return 0 or 1 (not visible, visible).

set_width(int width) -- Set's the width of this control.

class **ListItem()**

ListItem() -- Creates a new listitem.

Inherit from this class to create a custom listitem. Just remember to set the label.

Methods defined here:

set_label(string label) -- Set's the listitem label.

get_label() -- Return's the listitem label.

set_label2(string label) -- Set's the listitem label2.

get_label2() -- Return's the listitem label2.

set_thumbimage(string thumbimage) -- Set's the listitem thumbimage.

Image path for thumb texture has to be relative script folder or mmsv2 theme folder.

class **Window()**

Window()

Deleting this window will activate the old window that was active, remove and reset (not delete) all controls that are associated with this window

Methods defined here:

add_control(Control) -- Add a Control to this window.

```
Throws:     TypeError, if supplied argument is not a Control type
            ReferenceError, if control is already added
            RuntimeError, should not happen
```

The controls below can be added to a window atm

- -LabelControl
- -TextBoxControl
- -ButtonControl
- -ListControl
- -ImageControl
- -RectangleControl

on_action(string action) -- on action method.

This method will receive all actions (key presses) that the main program will send to this window.
By default only the 'back' action is handled.
Overwrite this method to let your script handle all actions. Don't forget to capture some action (key press), else the user can't close this window.

do_modal() -- Display this window until close() is called.

This method will block and wait for key presses by user. Will in turn call the on_action method.

show() -- Shows/activates the calling window.

This method will **not** block. No key presses can be handled by the window.

close() -- Closes this window.

get_focus() -- Returns the control which is in focus.

Throws: RuntimeError, if no control has focus

set_focus(Control) -- Give the supplied control focus.

Throws: TypeError, if supplied argument is not a Control type
RuntimeError, if control is not added to this window

get_width() -- Returns the width of the screen.

get_height() -- Returns the height of the screen.

remove_control(Control) -- Removes the control from this window.

Throws: TypeError, if supplied argument is not a Control type
RuntimeError, if control is not added to this window

This will not delete the control. It's only removed from the window.

class **Dialog()**

Dialog() -- Creates a new Dialog.

Methods defined here:

yesno(header, line1 [, line2, line3]) -- Show a dialog 'Yes/No'.

header : string
line1 : string
line2 : string
line3 : string

Returns True when user press 'Yes', else False

ok(header, line1 [, line2, line3]) -- Show a dialog 'Ok'.

header : string
line1 : string
line2 : string
line3 : string

Returns True when user press 'Ok'

class **DialogProgress()**

DialogProgress() -- Creates a new DialogProgress.

Methods defined here:

close() -- Close the progress dialog.

create(header [, line1, line2, line3])

```
header         : string
line1          : string
line2          : string
line3          : string
```

Create and show a progress dialog.

is_canceled() -- Returns True if user pressed 'Cancel'.

update(int value) -- Updates the progress dialog.

```
value          : int (0-100)
```

If value is 0, the progress bar will be hidden.

set_text([header, line1, line2, line3])

```
header         : string
line1          : string
line2          : string
line3          : string
```

Author

Fredrik