4 API

The GLUI library consists of 3 main classes: `GLUI_Master_Object`, `GLUI`, and `GLUI_Control`. There is a single global `GLUI_Master_Object` object, named `GLUI_Master`. *All GLUI window creation must be done through this object.* This lets the GLUI library track all the windows with a single global object. The `GLUI_Master` is also used to set the GLUT Idle function, and to retrieve the current version of GLUI.

4.1 Windows

This section describes the functions related to window creation and manipulation. The functions listed here belong to two classes: `GLUI_Master_Object` and `GLUI`. Keep in mind that any member function of the `GLUI_Master_Object` class should be invoked from the global object, named `GLUI_Master`, while any function of the `GLUI` class should be invoked via a `GLUI` pointer returned from `GLUI_Master.create_glui()`. For example:

```
float version = GLUI_Master.get_version();
GLUI *glui_window = GLUI_Master.create_glui ( "GLUI" );
glui_window->add_StaticText( "Hello World!" );
```

4.1.1 Initialization

get_version

Returns the current GLUI version.

Usage

```
float GLUI_Master_Object::get_version( void );
```

Returns: Current GLUI version

create_glui

Creates a new user interface window

Usage

```
GLUI  *GLUI_Master_Object::create_glui(    char *name, int flags=0,
                                           int x=-1, int y=-1 );
```

`name`	-	Name of new GLUI window
`flags`	-	Initialization flags. `GLUI_DOUBLE` is the only flag defined in the current version. If used, most of the drawing of the GLUI controls will be done in the back buffer, which is faster and avoids display flickering. `GLUI_DOUBLE` is recommended if double buffering is supported, which can be checked by calling `glutGet(GLUT_DISPLAY_MODE_POSSIBLE)`. If `GLUI_DOUBLE` is not specified, drawing is done in the front buffer of a single buffer window.
`x,y`	-	Initial location of window. Note that no initial size can be specified, because GLUI automatically resizes windows to fit all controls.

Returns: Pointer to a new GLUI window.

create_glui_subwindow

Creates a new user interface subwindow, inside an existing GLUT graphics window.

Usage

```
GLUI  *GLUI_Master_Object::create_glui_subwindow(   int window,
                                                    int position );
```

`window` - ID of existing GLUT graphics window

`position` - Position of new sub window, relative to the GLUT graphics window it is embedded in. This argument can take one of the following values:

```
GLUI_SUBWINDOW_RIGHT
GLUI_SUBWINDOW_LEFT
GLUI_SUBWINDOW_TOP
GLUI_SUBWINDOW_BOTTOM
```

Theses values may be ORed with `GLUI_DOUBLE`; see `create_glui`. You can place any number of subw indows at the same relative position; in this case, multiple subwindo ws will simply be stacked on top of one another. For example, if two subwindows are created inside the same GLUT window, and both use `GLUI_SUBWINDOW_TOP`, then the two are placed at the top of the window, although the first subwindow will be above the second.

Returns: Pointer to a new GLUI subwindow

set_glutIdleFunc

Registers a standard GLUT Idle callback `f()` with GLUI. GLUI registers its own Idle callback w ith GLUT, but calls this user function `f()` after each idle event. Thus every idle event is received by the callback `f()`, but only after GLUI has done its ow n idle processing. This is mostly transparent to th e GLUT application: sim ply register the idle callback with this function rather than the standard GLUT function `glutIdleFunc()`, and the GLUT application will work as usual. The only caveat is that under the G LUT specification, the current wind ow is undefined in an idle callback. Therefore, your application will need to explicitly set the current window before rendering or posting any GLUT redisplay events:

```
int main_window;

void myGlutIdle( void )
{
   /* ... */

   if ( glutGetWindow() != main_window )
      glutSetWindow(main_window);

   glutPostRedisplay();
}
```

This ensures that the redisplay message is properly sent to the graphics window rather than to a GLUI window.

Usage

```
void  GLUI_Master_Object::set_glutIdleFunc( void (*f)(void) );
```

`f` - GLUT Idle event callback function