

# INSTINCT ENGINE GRAPHICS PACKAGE

## REFERENCE SHEET

instinct.h



### COLORS

BLACK	
WHITE	
RED	
GREEN	
BLUE	
YELLOW	
ORANGE	
DARK_RED	
DARK_GREEN	
DARK_BLUE	
GRAY	
PINK	
CYAN	
BROWN	
PURPLE	

(Example: Color c = RED;)

### GENERAL INPUT FUNCTIONS

<code>win.get_mouse_loc()</code>	Returns a Point with the current location of the mouse in the coordinate system.
<code>win.get_mouse_x()</code>	Returns a double with the current x location of the mouse in the coordinate system.
<code>win.get_mouse_y()</code>	Returns a double with the current y location of the mouse in the coordinate system.
<code>wait_for(milliseconds)</code>	Pauses the program for a given number of milliseconds. Milliseconds is an integer.
<code>win.left_mouse_down()</code>	Returns true if left mouse button is down.
<code>win.right_mouse_down()</code>	Returns true if the right mouse button is down.
<code>win.is_key_down(key)</code>	Returns true if the string key is being pressed on the keyboard. Otherwise returns false.

#### Possible key values for `is_key_down(key)`

“a” “b” “c” ... or “A” “B” “C” ... “1” “2” “3” ... or “!” “@” “#” ...  
“.” “,” “/” “;” “'” “” “” “\” “\_” “=” “~” or “<” “>” “?” “.” “\” “{” “}” “|” “\_” “+” “~”  
“up” “down” “left” “right” “backspace” “enter” “insert” “end” “pageup”  
“pagedown” “tab” “esc” “home” “lalt” “lctrl” “ralt” “rctrl” “delete” “space”  
“lshift” “rshift” “printscreen” “F1” “F2” “F3” ... “F12”

### WIN GRAPHICS FUNCTIONS

<code>win.clear_buffer()</code>	Clears the shapes & messages from the buffer.
<code>win.output_buffer()</code>	Outputs everything in the window buffer to the screen.
<code>win.coord(ll_x, ll_y, ur_x, ur_y)</code>	Sets the coordinate system of the window to the lower left Point (ll_x, ll_y) and upper right Point(ur_x, ur_y).
<code>win.get_int(s)</code>	Pauses the program and prompts the user with the string s to get an integer. Returns that integer. The color of the string s will be BLACK.
<code>win.get_int(s, color)</code>	Same as <code>get_int(s)</code> except the prompt s will be the color passed in.
<code>win.get_double(s)</code>	Pauses the program and prompts the user with the string s to get a double. Returns that double. The color of the string s will be BLACK.
<code>win.get_double(s, color)</code>	Same as <code>get_double(s)</code> except the prompt s will be the color passed in.
<code>win.get_string(s)</code>	Pauses the program and prompts the user with the string s to get a string. Returns that string. The color of the string s will be BLACK.
<code>win.get_string(s, color)</code>	Same as <code>get_string(s)</code> except the prompt will be the color passed in.
<code>win.get_click(s)</code>	Pauses the program and prompts the user with the string s to click somewhere on the screen. Returns a Point of the click location. The color of the string s will be BLACK.
<code>win.get_click(s, color)</code>	Same as <code>get_mouse(s)</code> except the prompt will be the color passed in.
<code>win.play_sound(file)</code>	Plays the sound specified by the filename passed in.
<code>win.play_music(file, loop_num)</code>	Plays the music specified by the filename and loops it loop_num times. -1 will make it loop forever. If no value is passed in, it defaults to 0.
<code>win.stop_music()</code>	Stops any music that is currently playing.

<code>win.get_coord_width()</code>	Returns the width of the coordinate system.
<code>win.get_coord_height()</code>	Returns the height of the coordinate system.
<code>win.set_background_color(color)</code>	Sets the background color of the window to the color passed in.
<code>win.set_title(s)</code>	Sets the title of the window to the string s.
<code>win.set_window_size(int w, int h)</code>	Changes the size of the physical window to the width and height passed in. The default size of the the window 500 x 500.
<code>win.get_window_width()</code>	Returns the width in pixels of the physical window.
<code>win.get_window_height()</code>	Returns the height in pixels of the physical window.

<b>Point</b>	<b>INSTINCT OBJECTS</b>	<b>Circle</b>
--------------	-------------------------	---------------

<code>Point(x, y)</code>	Constructs a black Point at location x, y.	<code>Circle(p, r)</code>	Constructs a black circle at center p and radius r.
<code>Point(x, y, color)</code>	Constucts a Point at location x, y. Will be the color passed in.	<code>Circle(p, r, color)</code>	Constructs a circle at center p and radius r. Will be the color passed in.
<code>p.get_x()</code>	Returns the x location of p.	<code>c.get_center()</code>	Returns the center Point of c.
<code>p.get_y()</code>	Returns the y location of p.	<code>c.get_radius()</code>	Returns the radius of c.
<code>p.set_x(new_x)</code>	Sets the x location of p to the new x value passed in.	<code>c.set_radius(new_r)</code>	Sets the radius of c to the new radius value passed in.
<code>p.set_y(new_y)</code>	Sets the y location of p to the new y value passed in.	<code>c.set_center(new_cntr)</code>	Sets the center of c to the new Point passed in.

<b>Line</b>	<b>Triangle</b>
-------------	-----------------

<code>Line(st, e)</code>	Constructs a black Line with start Point st and end Point e.	<code>Triangle(p1, p2, p3)</code>	Constructs a black Triangle with the Points p1, p2, and p3.
<code>Line(st, e, color)</code>	Constructs a Line with start Point st and end Point e. Will be the color passed in.	<code>Triangle(p1, p2, p3, color)</code>	Constructs a Triangle with the Points p1, p2, and p3. Will be the color passed in.
<code>l.get_start()</code>	Returns the start of l.	<code>t.get_p1()</code>	Returns the Point p1 of t.
<code>l.get_end()</code>	Returns the end of l.	<code>t.get_p2()</code>	Returns the Point p2, of t.
<code>l.set_start(new_st)</code>	Sets the start Point to new_st.	<code>t.get_p3()</code>	Returns the Point p3 of t.
<code>l.set_end(new_e)</code>	Sets the end Point to new_e.	<code>t.set_p1(new_p1)</code>	Sets p1 of t to new_p1.
		<code>t.set_p2(new_p2)</code>	Sets p2 of t to new_p2.
		<code>t.set_p3(new_p3)</code>	Sets p3 of t to new_p3.

<b>Rect</b>
-------------

<code>Rect(p, w, h)</code>	Constructs a black Rect with the lower left Point p, a width w, and height h.
<code>Rect(p, w, h, color)</code>	Constructs a Rect with lower left Point p, width w, height h, and color C.
<code>r.get_width()</code>	Returns the width of the Rect r.
<code>r.get_height()</code>	Returns the height of the Rect r.
<code>r.get_lower_left()</code>	Returns the lower left Point of the Rect r.
<code>r.set_width(w)</code>	Sets the width of the Rect r to w.
<code>r.set_height(h)</code>	Sets the height of the Rect r to h.

## Image

Image(p, w, h, f)	Constructs an Image at lower left Point p, width w, and height h. The filename of the image is f. Example = “./spaceship.png”.	i.set_height(h)	Sets the height of i to h.
i.get_lower_left()	Returns the ll Point of i.	i.get_width()	Returns the width of i.
i.add_angle(ang)	Adds ang to the angle of i.	i.get_height()	Returns the height of i.
i.get_angle()	Returns the radian angle of i.	i.get_file_name()	Returns the image file of i.
i.set_width(w)	Sets the width of i to w.	i.set_angle(ang)	Sets the angle to draw the image to ang. Ang is in radians.

## Message



Message(lower_left, text)	Constructs a black message starting at the Point lower_left and displays the value of text. The variable text can be a string, integer, or double.
Message(lower_left, text, color)	Constructs a message starting at the Point lower_left and displays the value of text. The variable text can be a string, integer, or double. Will be the color passed in.
m.get_text()	Returns the text value of m.
m.get_lower_left()	Returns the lower left Point of m.
m.set_text(new_text)	Sets the text of m to new_text which can be a string, int, or double.

## Extra values used for Messages

LETTER_WIDTH	The length of a letter. This value works for any coordinate system.
LETTER_HEIGHT	The height of a letter. This value works for any coordinate system.

## Functions usable by all Instinct Objects (Point, Circle, Rect, Line, Image, Triangle, Message)

obj.move(dx, dy)	Moves the object dx in the x direction and dy in the y direction.
obj.get_color()	Returns the color of the object.
obj.is_filled_in()	Returns if the object is filled in with color or not.
obj.set_color(color)	Sets the color of the obj with the value passed in.
obj.set_filled_in(is_filled)	Makes the obj filled in with color if is_filled is true. Otherwise is makes the obj only outlined with color.

**Filled in**  **Not Filled in** 

## Timer

Timer()	Creates a timer object.	tmr.get_time()	Returns the time on tmr in milliseconds.
tmr.start()	Starts the timer tmr.	tmr.add_time(millisecs)	Adds millisecs to the time on tmr.
tmr.stop()	Stops the timer tmr.	tmr.has_passed(millisecs)	Returns true if the time on tmr has passed millisecs.
tmr.reset()	Resets tmr back to zero.		

## BASIC STARTING CODE FOR INSTINCT PROGRAMS

### Console Program

```
#include <iostream>
int main()
{
    //main code goes here.
    return 0;
}
```

### Instinct Graphics Package Program

```
#include "instinct.h"
int instinct_main()
{
    //main code goes here.
    return 0;
}
```