| | [**Overview**](http://docs.google.com/index-overview-summary.html) | [**Project**](http://docs.google.com/project-summary.html) | **Class** | [**Tree**](http://docs.google.com/project-tree.html) | [**Deprecated**](http://docs.google.com/index-deprecated-list.html) | [**Index**](http://docs.google.com/index-all.html) | | --- | --- | --- | --- | --- | --- | | | ***CarnegieMellonGraphics*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**FRAMES**](http://docs.google.com/index.html)    [**NO FRAMES**](http://docs.google.com/CarnegieMellonGraphics2/MouseEvent.html) |
| SUMMARY:  INNER | FIELD | [CONSTR](#1fob9te) | [METHOD](#3znysh7) | DETAIL:  FIELD | [CONSTR](#2et92p0) | [METHOD](#2s8eyo1) |  |

## **CarnegieMellonGraphics2**

Class MouseEvent

   in [CarnegieMellonGraphics.h](http://docs.google.com/CarnegieMellonGraphics.h.html)

class **MouseEvent**

MouseEvents are generated whenever the user performs some action with the mouse. The types of MouseEvents that can occur are listed below as part of the MouseEvent::Event enumeration.

MouseEvents are designed to work in a manner so that a user can easily "match" and filter out events in which they have interest. Basically the user constructs a new event to match against the event they currently have. The MouseEvent constructor has several default parameters so the user doesn't need to specify more detail about the event than they need.

It is also possible to simply obtain pertinent information about an event through the accessor member functions.

For example, if the user has just removed a mouse event from the mouse input queue, and they want to see if the mouse was clicked they can do the following:

if(MouseEvent(MouseEvent::BUTTON\_CLICK\_EVENT) == an\_event)   
 // do something

Or a more complicated example would be if the user wanted to check to see if the mouse has was moved over position (x, y) while holding down the left button they would do this:

if(MouseEvent(MouseEvent::MOUSE\_MOVE\_EVENT, MouseEvent::LEFT\_BUTTON, x, y) == an\_event)   
 // do something else

Finally, if you wanted to check to see if the right mouse buton has been pressed down while holding down the shift key, you would do the following:

if(MouseEvent(MouseEvent::ButtonDown, MouseEvent::RIGHT\_BUTTON, -1, -1,   
 KeyModifiers::SHIFT) == an\_event)   
 // do more of something else

In this case the coordinate of (-1, -1) indicates the location at which the button press occurred does not matter.

Note that on some platforms, certain mouse and modifier combinations will be trapped by the operating system/window manager and will not be reported as having taken place.

| **Inner Classes, Typedefs, and Enums** | |
| --- | --- |
| typedef | [**MouseEvent::Button**](http://docs.google.com/CarnegieMellonGraphics2/MouseEvent..Button.html)            These are the buttons that CMG recognizes. |
| typedef | [**MouseEvent::Event**](http://docs.google.com/CarnegieMellonGraphics2/MouseEvent..Event.html)            These are the events that CMG reports. |

| **Constructor Summary** | |
| --- | --- |
| [**MouseEvent**](#tyjcwt)( const [MouseEvent::Event](http://docs.google.com/CarnegieMellonGraphics2/MouseEvent..Event.html) e, const [MouseEvent::Button](http://docs.google.com/CarnegieMellonGraphics2/MouseEvent..Button.html) b = ANY\_BUTTON, const int x =-1, const int y =-1, const [KeyModifiers](http://docs.google.com/CarnegieMellonGraphics2/KeyModifiers.html) km = KeyModifiers::ANY\_MODIFIERS )            Constructor for a mouse event from a set of parameters. |
| [**MouseEvent**](#1t3h5sf)( const [**MouseEvent**](#1t3h5sf)& me )            Copy constructor |
| [**MouseEvent**](#4d34og8)()            Default constructor |

| **Method Summary** | |
| --- | --- |
| const [MouseEvent::Button](http://docs.google.com/CarnegieMellonGraphics2/MouseEvent..Button.html) | [**getButton**](#17dp8vu)() const |
| const [MouseEvent::Event](http://docs.google.com/CarnegieMellonGraphics2/MouseEvent..Event.html) | [**getEvent**](#26in1rg)() const |
| const [KeyModifiers](http://docs.google.com/CarnegieMellonGraphics2/KeyModifiers.html) | [**getModifiers**](#35nkun2)() const |
| const int | [**getX**](#44sinio)() const |
| const int | [**getY**](#z337ya)() const |
| bool | [**operator!=**](#1y810tw)( const MouseEvent& rhs ) const            Comparison operator for inequality on events. |
| bool | [**operator==**](#2xcytpi)( const MouseEvent& rhs ) const            Comparison operator for equality on events. |

| **Constructor Detail** |
| --- |

### MouseEvent

public **MouseEvent**( const [MouseEvent::Event](http://docs.google.com/CarnegieMellonGraphics2/MouseEvent..Event.html) e, const [MouseEvent::Button](http://docs.google.com/CarnegieMellonGraphics2/MouseEvent..Button.html) b = ANY\_BUTTON, const int x =-1, const int y =-1, const [KeyModifiers](http://docs.google.com/CarnegieMellonGraphics2/KeyModifiers.html) km = KeyModifiers::ANY\_MODIFIERS );

Constructor for a mouse event from a set of parameters.

### MouseEvent

public **MouseEvent**( const **MouseEvent**& me );

Copy constructor

### MouseEvent

public **MouseEvent**();

Default constructor

| **Method Detail** |
| --- |

### getButton

public const [MouseEvent::Button](http://docs.google.com/CarnegieMellonGraphics2/MouseEvent..Button.html) **getButton**() const;

**Returns:** Which button(s), if any, were involved with this event.

### getEvent

public const [MouseEvent::Event](http://docs.google.com/CarnegieMellonGraphics2/MouseEvent..Event.html) **getEvent**() const;

**Returns:** The actual event information for the mouse event.

### getModifiers

public const [KeyModifiers](http://docs.google.com/CarnegieMellonGraphics2/KeyModifiers.html) **getModifiers**() const;

**Returns:** set of modifiers active at the time of the event, if relevant.

### getX

public const int **getX**() const;

**Returns:** The X coordinate that this event occurred, if relevant, otherwise returns -1.

### getY

public const int **getY**() const;

**Returns:** The Y coordinate that this event occurred, if relevant, otherwise returns -1

### operator!=

public bool **operator!=**( const MouseEvent& rhs ) const;

Comparison operator for inequality on events.

### operator==

public bool **operator==**( const MouseEvent& rhs ) const;

Comparison operator for equality on events.

| | [**Overview**](http://docs.google.com/index-overview-summary.html) | [**Project**](http://docs.google.com/project-summary.html) | **Class** | [**Tree**](http://docs.google.com/project-tree.html) | [**Deprecated**](http://docs.google.com/index-deprecated-list.html) | [**Index**](http://docs.google.com/index-all.html) | | --- | --- | --- | --- | --- | --- | | | ***CarnegieMellonGraphics*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**FRAMES**](http://docs.google.com/index.html)    [**NO FRAMES**](http://docs.google.com/CarnegieMellonGraphics2/MouseEvent.html) |
| SUMMARY:  INNER | FIELD | [CONSTR](#1fob9te) | [METHOD](#3znysh7) | DETAIL:  FIELD | [CONSTR](#2et92p0) | [METHOD](#2s8eyo1) |  |