zCViewCursor : zCViewAnimated

enum zEMouseButton

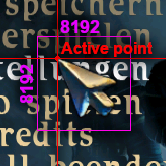
zBUTTON\_LEFT

zBUTTON\_MIDDLE

zBUTTON\_RIGHT

Перечислитель основных кнопок мыши.

В качестве текстуры курсор использует объекты класса zCViewCursorVisual, которые помимо изображения хранят информацию об активной точке. Сам курсор может располагаться на изображении в любом месте, следовательно и его активная точка может быть где угодно. Текстура курсора находится в стандартном для zCView диапазоне – 8192х8192. Активная точка также определяется в пределах данного диапазона.

Как выглядит курсор в игре Реальная текстура курсора

Переключение между текстурами курсора происходит посредством индексов, которые автоматически назначаются всем добавляемым в курсор текстурам. Например назначенный индекс можно передать в интерактивные объекты и при наведении на них курсор самостоятельно поменяет изображение.

Защищенные члены:

Свойства:

Array<zCViewCursorVisual\*> Visuals;

Список изображенийкурсора.

Array<zCView\*> SelectedCollection;

Список объектов в области которых находится курсор.

zCView\* TopSelectedView;

Верхний объект в области которого находится курсор.

zCViewInteractive\* TopSelectedViewInteractive;

Принимает значение, если TopSelectedView является интерактивным объектом.

zCViewNote\* TopSelectedViewNote;

Принимает значение, если TopSelectedView является редактором текста.

int PosX;

Положение курсора в пикселях по X.

int PosY;

Положение курсора в пикселях по Y.

uint References;

Количество функций, использующих курсор. Курсор отображается на экране, пока значение больше нуля.

bool\_t LeftPressed;

Свойство принимает True, если зажата левая кнопка мыши.

bool\_t LeftToggled;

Свойство принимает True, если левая кнопка мыши была нажата в текущем кадре.

bool\_t MiddlePressed;

Свойство принимает True, если зажата средняя кнопка мыши.

bool\_t MiddleToggled;

Свойство принимает True, если средняя кнопка мыши была нажата в текущем кадре.

bool\_t RightPressed;

Свойство принимает True, если зажата правая кнопка мыши.

bool\_t RightToggled;

Свойство принимает True, если правая кнопка мыши была нажата в текущем кадре.

zCView\*\* HandledView;

Указатель на объект, с которого берется самый верхний, в область которого входит курсор.

zCViewInteractive\* DragObject;

Перетаскиваемый в данный момент объект, если существует. Иначе Null.

Методы:

void OnEnter();

Вызывается, если курсор входит в область какого-то объекта.

void OnLeave();

Вызывается, если курсор выходит из области какого-то объекта.

void OnDown( zEMouseButton button );

Вызывается, если курсор нажимает кнопку.

void OnDrag( int oldx, int oldy );

Вызывается, если курсор начинает перетаскивать какой-то объект.

void OnUp( zEMouseButton button );

Вызывается, если курсор отжимает кнопку.

void Collect( zCView\* view );

Собирает список всех объектов, в области которых находится курсор. Параллельно определяет TopSelectedView.

void FrameBegin();

Подготавливает курсор к работе.

void FrameEnd();

Позиционирует курсор, строит список объектов и вызывает их события.

void UpdateVisual();

Обновляет текстуру курсора с учетом его нахождения над объектами.

void UpdateRect();

Обновляет регион по которому может перемещаться курсор. Регион равен окну.

void Render( bool blit = true );

Помещает курсор на экран.

* blit – если true, то принудительно рисует курсор на экран

Публичные члены:

Свойства:

uint DefaultVisualIndex;

Индекс изображения курсора по умолчанию.

uint VisualIndex;

Индекс изображения курсора в нормальном состоянии.

Методы:

void UpdateKeyStates();

Обновляет состояния нажатия кнопок мыши.

void UpdateInput();

Определяет какие кнопки мыши нажаты в данный момент и вызывает обновление состояний.

void ClearInput();

Сбрасывает информацию о нажатых кнопках мыши.

void ClearSelection();

Сбрасывает информацию о верхнем объекте в области которого находится курсор.

void GetActivePoint( int& px, int& py );

Возвращает информацию об активной точке текущей текстуры курсора.

* px – координата X в виртуальных координатах
* py – координата Y в виртуальных координатах

void GetActivePointPosition( int& px, int& py );

Возвращает положение активной точки текущей текстуры курсора на экране.

* px – координата X в виртуальных координатах
* py – координата Y в виртуальных координатах

void GetCursorPosition( int& px, int& py );

Возвращает положение курсора без учета активной точки текстуры.

* px – координата X в виртуальных координатах
* py – координата Y в виртуальных координатах

void Show();

Отправляет запрос на отображение курсора, пока не будет вызван метод Hide.

void ShowAt( int px, int py );

Отправляет запрос на отображение курсора в заданных координатах, пока не будет вызван метод Hide.

* px – координата X в виртуальных координатах
* py – координата Y в виртуальных координатах

void ShowAtCenter();

Отправляет запрос на отображение курсора в центре экрана, пока не будет вызван метод Hide.

void Hide();

Сообщает программе о том, что одной из операций курсор больше не требуется.

uint CreateVisual( const zSTRING& texName, const zVEC2& activePoint, const int& aniFps = 0 );

Создает новую текстуру курсора и возвращает ее индекс.

* texName – имя текстуры
* activePoint – активная точка текстуры в виртуальных координатах
* aniFps – скорость анимации, если текстура должна быть анимирована

zCViewCursorVisual\* GetActiveVisual();

Возвращает информацию о текущей текстуре курсора.

zCViewCursorVisual\* GetDefaultVisual();

Возвращает информацию о текстуре курсора по умолчанию.

void SetHandledView( zCView\*& view );

Устанавливает объект, в котором будет искаться верхний объект, над которым расположится курсор.

zCView\*& GetHandledView();

Возвращает объект, в котором производится поиск верхнего объекта, над которым расположится.

void ClearHandledView();

Очищает объект, в котором производится поиск верхнего объекта, над которым расположится.

zCView\* GetTopSelectedView();

Возвращает верхний объект, над которым расположен курсор. Либо Null, если он не существует.

Array<zCView\*>& GetTopSelectedCollection();

Возвращает весь список объектов, над которыми расположен курсор от самого нижнего к самому верхнему.

zCViewInteractive\* GetDragObject();

Возвращает текущий перетаскиваемый объект. Либо Null, если он не существует.

bool\_t IsLeftPressed();

Определяет, нажата ли левая кнопка мыши.

bool\_t IsLeftToggled();

Определяет, была ли нажата левая кнопка мыши в текущем кадре.

bool\_t IsMiddlePressed();

Определяет, нажата ли средняя кнопка мыши.

bool\_t IsMiddleToggled();

Определяет, была ли нажата средняя кнопка мыши в текущем кадре.

bool\_t IsRightPressed();

Определяет, нажата ли правая кнопка мыши.

bool\_t IsRightToggled();

Определяет, была ли нажата правая кнопка мыши в текущем кадре.

static zCViewCursor\* GetCursor();

Возвращает курсор.