MChSprites Components

Real Time Scaleable Sprites
Components
for
Borland Delphi

Description

Real Time Scaleable Sprites for Borland's Delphi consists of two interacting components:

Component: **Animation Background**
- class TMChSpriteBgr
- Unit MChSpBg
- Properties
- Methods
- Types
- Const
- Functions

Component: **Sprite**
- class TMChSprite
- Unit MChSprt
- Properties
- Methods
- Types

Animation background and transparent sprites (up to 100) are created as instances of corresponding components by double clicking on Delphi component palette and by selecting proper image (e.g. bitmap) with Object Inspector property editor (Picture Editor). All initialisation is handled by components themselves. Instance of TMChSprite can be moved on instance of TMChSpriteBgr with single method call e.g. MChSprite1.SprGo(From, Dest, AtTime). Sprite position is determined by real time clock and is independent on computer and graphic cart performance (slow cart results in jetted movements). Sprites are implemented as autonomous objects and support transparent colour, collision, Z-order (overlapping), Z-order dynamic change, sprite scaling, dynamic scale change, image flipping, linear movement and movement along curve. Sprites can collide with borders and with other sprites, respond to mouse pointer and can be drag with mouse to new position. Sprite movement can be synchronised with sound effects.

My (Marek Chmielowski) e-mail:
on Compuserve: 76360,2775
on Internet: 76360,2775@compuserve.com
Mail:
Marek Chmielowski
5/56 Kozia St.
Warsaw 00-070
Poland
(48) (22) 226-01-85
or
Marek Chmielowski
10005 Broad St.
Bethesda, MD 20814
USA
(301) 897-5266

TMChSpriteBgr component (class)

Unit MChSpBg
Properties
Methods
Types
Const
Functions

Description:

The TMChSpriteBgr component descended from TImage provides static image background on which sprites can be moved.   Additionally to functionality TImage it provides arbitration in between Sprites (TMChSprite instances) requests and synchronise screen update in flicker free fashion.

Usage - Design stage:

Double click on component to place it on Form.   With Object Inspector edit picture property - just like for TImage picture property.   Three dots button will open Picture Editor and let you select background for sprites.   Than, create Sprites i.e. instances of class TMChSprite component.

Usage - Run time stage:

All initialisation is performed automatically and in most cases no interaction with TMChSpriteBgr instance is required.

TMChSprite component (class)

Unit MChSprt
Properties
Methods
Event Handler Types

Description:

The TMChSprite component provides movable in real time (i.e. position is determined by real time clock and not by graphic cart performance), scaleable, and flicker free sprites.   Sprites are implemented as autonomous objects and support transparent colour, collision, Z-order (overlapping), Z-order dynamic change, sprite scaling, dynamic scale change, image flipping, linear movement and movement along curve. Sprites can respond to mouse pointer and can be drag with mouse to new position. Sprite movement can be synchronised with sound effects.   Sprite Background component (TMChSpriteBgr) provides static background and arbitration in between Sprites requests and synchronise screen update in flicker free fashion.

Usage - Design stage:

It is recommended to create background for Sprites first - see TMChSpriteBgr Double click on component to place it on Form.   With Object Inspector edit Bitmap property - just like for any TBitmap field.   Three dots button will open Picture Editor and let you select and preview sprite image.   Than, place sprite outline in initial position on background image.   Repeat the above described procedure for each Sprite required - the last on will be the top most one.   During design sprites are visible on Form only if selected and as outlines only - of correct size and at initial positions.

Usage - run time stage:

Sprites are initially invisible on running application background.   Sprite will show up in response to one of the method call (ShowOn, ShowAt, SprGoTo, SprGo, SprRun, SprCruise).   To stop moving sprite call SprStop.   To hide call SprHide.

Unit MChSpBg

File:

mchspbg.pas

Description:

Unit MChSpBg contain source code for TMChSpriteBgr component.   To install select file mchspbg.pas from Delphi IDE Option|Install_Components|Add|Browse and click OK.   Component will be available on Samples palette.

Const in unit MChSpBg

Constant and typed constant defined in unit MChSpBg are also used by unit MChSprt and can be used by user code.

const:

BgrMaxSpriteNum = 100;
Maximum number of sprites managed by background - can be increase if needed

const typed:

NulPoint: TPoin;
Point 0,0

NulRect: TRect;
Empty rectangle at 0,0

Types in Unit MChSpBg

Types defined in unit MChSpBg are also used by unit MChSprt and can be used by user code.

TBgrOnInit = procedure;
User procedure which may be called after Background initialisation.

TBgrSpriteList = array[1..BgrMaxSpriteNum] of TGraphicControl;
List of sprites in reverse Z-Order i.e. the highest number correspond to the top most sprite.

TDirtyReg = record
```
  Old: TRect;
  New: TRect;
  end;
```
Areas to be updated on canvas.

Functions in Unit MChSpBg

Functions defined in unit MChSpBg are also used by unit MChSprt and can be used by user code.

InRect(Tp: TPoint; Tr: TRect): Boolean;
True if Test Point is in rectangle   works fine with reversed rectangles i.e. ones with Left>Right and/or Top>Bottom;

CheckNotNulRect(Rt: TRect): Boolean;
True if Rt is NOT NulRect

DirtyReg(Old,New: TRect): TDirtyreg;
Type conversion two rectangles into TDirtyReg

Methods of Class TMChSpriteBgr

Most public methods of TMChSpriteBgr component is intended to interact with TMChSprite components but they have to be declared public as TMChSpriteBgr and TMChSprite components are descendent from different ancestors.

Methods useful for Component Users:

BgrSprExchangeToTop(Sprite: TGraphicControl);
Exchange Sprite with Topmost one

BgrSprExchangeZ(Sprite1, Sprite2: TGraphicControl);
Exchange Z order of sprites

BgrSprShiftToTop(Sprite: TGraphicControl);
Place Sprite as topmost and shift all to fill the place

BgrSprShiftZ(Sprite1, Sprite2: TGraphicControl);
Place Sprite1 in Z order of Sprite2 and shift all (including Sprite2)

BgrAddTopSpr(Spr: TGraphicControl): Boolean;
Dynamicly add Sprite - will be topmost one

BgrDeleteTopSpr;
Dynamically deletes topmost Sprite

Instance of TMChSprite can be used in place of formal parameters of TGraphicControl type as TMChSprite is descendant of TGraphicControl type and therefore is assgment compatible.

Properties of Class TMChSpriteBgr

Most public properties of TMChSpriteBgr component is intended to interact with TMChSprite components but they have to be declared public as TMChSpriteBgr and TMChSprite components are descendent from different ancestors.

Properties useful for Component Users:

Must be defined:

Picture: TPicture;
Image for Sprites background - use Picture Editor to select

Optional:

BgrSearchSprts: Boolean;
Control automatic search for Sprites upon initialisation

BgrRespondToMouse: Boolean;
To Switch of mouse sensing for all sprites

OnMouseDown;
Points to user own handler if needed, otherwise assigned to MChSpriteBgrMouseDown

OnMouseMove;
Points to user own handler if needed, otherwise assigned to MChSpriteBgrMouseMove

OnMouseUp;
Points to user own handler if needed, otherwise assigned to MChSpriteBgrMouseUp

Public:

BgrNumOfSprites: Cardinal;
Number of Sprites = Z Index of the top most one

BgrOnInit: TBgrOnInit;
Points to user initialisation routine if needed

BgrSprHitted: TGraphicControl;
Points to clicked Sprite - valid (not **nil**) in between MouseDown and MouseUp

BgrSprHittedWas: TGraphicControl;
Points to last clicked Sprite - valid also after MouseUp

BgrSprWasHitted: Boolean;
Flags hit inside unmasked portion of Sprite - True in between MouseDown and MouseUp

BgrSprCaptured: TGraphicControl;
Points to currently dragged sprite

BgrSprCapturedIndexWas: Cardinal;
Captured Sprite is automatically exchanged with topmost one on MouseDown and exchange back on MouseUp events. BgrSprCapturedIndexWas stores Z-order index of captured Sprite as was before MouseDown event.

BgrPause: Boolean;
True suspend temporary updating Sprites position and Image Canvas - use when changing Sprite parameters which may cause flicker

BgrInAppIdle: Boolean;
Flag indicating that Background manager is now active

BgrCntsPerSec: double;
Performance measure - for first few second is affected by initialisation

Picture property of TMChSpriteBgr

Must be defined:

Picture: TPicture;
Image for Sprites background - use Picture Editor to select

Summary of TMChSpriteBgr Properties

BgrSearchSprts Property of TMChSpriteBgr

Optional:

BgrSearchSprts: Boolean;
Control automatic search for Sprites upon initialisation

Summary of TMChSpriteBgr Properties

BgrRespondToMouse Property of TMChSpriteBgr

Optional:

BgrRespondToMouse: Boolean;
To Switch of mouse sensing for all sprites

Summary of TMChSpriteBgr Properties

OnMouseDown Property of TMChSpriteBgr

Optional:

OnMouseDown;

Points to user own handler if needed, otherwise assigned to MChSpriteBgrMouseDown

Summary of TMChSpriteBgr Properties

OnMouseMove Property of TMChSpriteBgr

OnMouseMove;

Points to user own handler if needed, otherwise assigned to MChSpriteBgrMouseMove

Summary of TMChSpriteBgr Properties

OnMouseUp Property of TMChSpriteBgr

OnMouseUp;

Points to user own handler if needed, otherwise assigned to MChSpriteBgrMouseUp

Summary of TMChSpriteBgr Properties

Unit MChSprt

File:

mchsprt.pas

Description:

Unit MChSprt contain source code for TMChSprite component.   To install select file mchsprt.pas from Delphi IDE Option|Install_Components|Add|Browse and click OK.   Component will be available on Samples palette.

Types in Unit MChSprt

Types defined in unit MChSprt can be used by user code.

TSprPosFunc = function(AtTime: TDateTime): TPoint;
User defined function returning Sprite position at given time (Time is in Days !!!!)

TSprOnBorder = procedure(AtTime: TDateTime);
User defined procedure called when Sprite collided with Sprites Background border

TSprOnCollide = procedure(SprCollidedWith: TMChSprite; AtTime: TDateTime);
User defined procedure called when Sprite collided with other Sprite

TSprNoCollide = procedure(AtTime: TDateTime);
User defined procedure called when Sprite moved without colliding with other Sprites

Methods of Class TMChSprite

Most public methods of TMChSprite component is intended to interact with TMChSpriteBgr components but they have to be declared public as TMChSpriteBgr and TMChSprite components are descendent from different ancestors.

Methods useful for Component Users:

SprShowOn;
Makes Sprite visible at current position
SprShowAt(Dest: TPoint);
Makes Sprite visible at Destination point
SprHide;
As you guessed makes Sprite invisible
SprStop;
As you probably also guessed stops Sprite
SprGoTo(Dest: TPoint; TimeToRunSec: TDateTime);
Makes Sprite visible at current position and moves to Destination during the next TimeToRunSec seconds
SprGo(From, Dest: TPoint; TimeToRunSec: TDateTime);
Makes Sprite visible at From position and moves to Destination during the next TimeToRunSec seconds
SprRun(From, Dest: TPoint; TimeToRunSec: TDateTime);
Makes Sprite visible at From position and moves to Destination during the next TimeToRunSec seconds just like SprGo but blocks other sprites and user interaction to achive better performance.
SprCruise(TimeToRunSec: TDateTime);
Makes Sprite visible and calls used defined SprPosFunc of TSprPosFunc type tu display Sprite at user defined positions, after TimeToRunSec stops Sprite at then current position.   If TimeToRunSec is negetive runs Sprite forever (You can of cource switch computer off or just call SprStop)
SprMoveToDest: TPoint);
Makes Sprite invisible and place (hidden) at Destination position

Properties of Class TMChSprite

Most public properties of TMChSprite component is intendend to interact with TMChSpriteBgr components but they have to be declared public as TMChSpriteBgr and TMChSprite components are descendent from different ancestors.

Properties usefull for Component Users:

Must be defined:

SprSpriteBitmap: TBitmap;
Bitmap for Sprite - use Picture Editor to select
SprTrColor: TColor;
Specifies transparet color on bitmap

Optional:

SprDragable: Boolean;
Set True to be able drag Sprite with mouse
SprColliding: Boolean;
Set True if Sprite should collide i.e. call user def. function pointed by:
SprHideAfter: Boolean;
Set to True to hide Sprite after completion movement
SprScaleX: double;
Sprite scale in horizontal direction - negetive values results in horizontally flipped image
SprScaleY: double;
Sprite scale in vertical direction - negetive values results in vertically flipped image
SprRefX: integer;
Reference point (x) on Sprite bitmap
SprRefY: integer;
Reference point (y) on Sprite bitmap
SprRadiusX: integer;
Collision radius (horizontal) from SprRef point - elipsoid aproximation
SprRadiusY: integer;
Collision radius (vertical) from SprRef point - elipsoid aproximation
SprGuessBgr: Boolean;
Set to False to switch off automatic initialisation

Public:

SprOnCollide: TSprOnCollide;
Pointer to user function of TSprOnCollide type
SprNoCollide: TSprNoCollide;
Pointer to user function of TSprNoCollide type
SprOnBorder: TSprOnBorder;
Pointer to user function of TSprOnBorder type
SprPosFunc: TSprPosFunc;
Pointer to user function of TSprPosFunc type
SprCollisionMask: Boolean;
Set it (temporary) to True if as a result of collision you SprOnCollide function mess with SpriteList e.g. change Z-order of Sprites with collision responce function.   If both Sprites will respond it is possibility of endless loop.

Property SprSpriteBitmap of TMChSprite

Must be defined:

SpriteBitmap: TBitmap;
Bitmap for Sprite - use Picture Editor to select

Summary of TMChSprite Properties

Property SprTrColor of TMChSprite

Must be defined:

SprTrColor: TColor;
Specifies transparet color on bitmap

Summary of TMChSprite Properties

Property SprDragable of TMChSprite

Optional:

SprDragable: Boolean;

Set True to be able drag Sprite with mouse

Summary of TMChSprite Properties

Property SprColliding of TMChSprite

Optional:

SprColliding: Boolean;

Set True if Sprite should collide i.e. call user def. function pointed by:

Summary of TMChSprite Properties

Property SprHideAfter of TMChSprite

Optional:

SprHideAfter: Boolean;

Set to True to hide Sprite after completion movement

Summary of TMChSprite Properties

Property SprScaleX of TMChSprite

Optional:

SprScaleX: double;
Sprite scale in horizontal direction - negetive values results in horizontally flipped image

Summary of TMChSprite Properties

Property SprScaleY of TMChSprite

Optional:

SprScaleY: double;
Sprite scale in vertical direction - negetive values results in vertically flipped image

Summary of TMChSprite Properties

Property SprRefX of TMChSprite

Optional:

SprRefX: integer;
Reference point (x) on Sprite bitmap

Summary of TMChSprite Properties

Property SprRefY of TMChSprite

Optional:

SprRefY: integer;
Reference point (y) on Sprite bitmap

Summary of TMChSprite Properties

Property SprRadiusX of TMChSprite

Optional:

SprRadiusX: integer;
Collision radius (horizontal) from SprRef point - elipsoid aproximation

Summary of TMChSprite Properties

Property SprRadiusY of TMChSprite

Optional:

SprRadiusY: integer;
Collision radius (vertical) from SprRef point - elipsoid aproximation
Summary of TMChSprite Properties

Property SprGuessBgr of TMChSprite

Optional:

SprGuessBgr: Boolean;
Set to False to switch off automatic initialisation
Summary of TMChSprite Properties