```
-- Rectangledefs.mesa  Edited by Sandman on May 24, 1977  9:13 AM

DIRECTORY
  Mopcodes: FROM "mopcodes",
  SegmentDefs: FROM "segmentdefs";

RectangleDefs: DEFINITIONS =
  BEGIN

-- Magic memory locations and assoc. constants

  mousebuttonsup: CARDINAL = 7;

-- Constants for Display stuff

  leftmargin: CARDINAL = 10;
  blanklines: CARDINAL = 6; -- # of blank lines at the top
  maxwordsperline: CARDINAL = 38;
  maxbitsperline: CARDINAL = maxwordsperline*16;
  minwidth: CARDINAL = 32;
  minheight: CARDINAL = 32;

-- some TYPE's and null POINTERS

  restype: TYPE = {high, low};
  backgtype: TYPE = {white, black};
  xCoord: TYPE = [0..606];
  yCoord: TYPE = [0..808];

-- Display Hardware Stuff

  DCBchainHead: DCBptr = LOOPHOLE[420B];
  DCBnil: DCBptr = LOOPHOLE[0];
  DCBptr: TYPE = POINTER TO DCB;
  DCB: TYPE = MACHINE DEPENDENT RECORD
    [
    next: DCBptr,
    resolution: restype,
    background: backgtype,
    indenting: [0..77B],        -- in units of 16 bits
    width: [0..377B],   -- likewise; must be even
    bitmap: BMptr,              -- must be even
    height: CARDINAL    -- in double scan lines
    ];

-- Font Stuff

  FHptr: TYPE = POINTER TO FontHeader;
  Fptr: TYPE = POINTER TO FONT;
  FCDptr: TYPE = POINTER TO FCD;
  FAptr: TYPE = POINTER TO fontarray;
  fontarray: TYPE = ARRAY [0..255] OF FCDptr;

  FONT: TYPE = MACHINE DEPENDENT  RECORD
    [
    FHeader: FontHeader,
    FCDptrs: fontarray,         -- array of self-relative pointers to
      -- FCD's.  Indexed by char value.
      -- font pointer points hear!
    ExtFCDptrs: fontarray       --  array of self-relative pointers to
      -- FCD's for extentions.  As large an
      -- array as needed.
    ];

  FontHeader: TYPE = MACHINE DEPENDENT RECORD
    [
    MaxHeight: CARDINAL,        -- height of tallest char in font (scan lines)
    VariableWidth: [0..1],      -- IF TRUE, proportionally spaced font
    blank:      [0..177B],      -- not used
    MaxWidth: [0..377B] -- width of widest char in font (raster units).
    ];

  FCD: TYPE = MACHINE DEPENDENT  RECORD
    [
    widthORext: [0..77777B],    -- width or extention index
    HasNoExtension: BOOLEAN,    -- TRUE=> no ext.;prevfield=width
```

```
    height: [0..377B],          -- # scan lines to skip for char
    displacement: [0..377B]     -- displacement back to char bitmap
    ];

-- Bit Map Stuff

  BitmapError: SIGNAL [bitmap: BMHandle, error: BitmapErrorCode];
  BitmapErrorCode: TYPE = {BitmapOperation};
  BITMAP: TYPE = WORD; -- array [0..15] of word
  BMptr: TYPE = POINTER TO BITMAP;
  BMHandle: TYPE = POINTER TO BitmapObject;

  BitmapObject: TYPE = RECORD -- all about a Bitmap

    [
    link: BMHandle,     -- # NIL iff being displayed
    rectangles: Rptr, -- list of display streams for this map
    dcb: DCBptr,
    addr: BMptr,        -- it's address
    words: CARDINAL,    -- size of map (in words)
    wordsperline: [0..maxwordsperline],
    x0: xCoord,         -- x,y of upper left corner
    y0: yCoord,
    width: xCoord,
    height: yCoord,
    indenting: [0..77B],        -- in units of 16 bits
    resolution: restype,
    background: backgtype
    ];

-- stuff for Bitmap Rectangles

  RectangleError: SIGNAL [rectangle:Rptr, error:RectangleErrorCode];
  RectangleErrorCode: TYPE = {RightOverflow, BottomOverflow, NotVisible};
  Rptr: TYPE = POINTER TO Rectangle;
  Rectangle: TYPE = RECORD
    [
    link: Rptr,
    visible: BOOLEAN,
    options: ROptions,
    bitmap: BMHandle,
    x0, width, cw: xCoord,
    y0, height, ch: yCoord
    ];


  -- Rectangle options field definitions

    ROptions: TYPE = RECORD     -- Rectangle options
      [
      NoteInvisible: BOOLEAN,   -- SIGNAL if rectangle off bitmap
      NoteOverflow: BOOLEAN     -- SIGNAL if storing outside
      ];

-- BITBLT stuff

  BBptr: TYPE = POINTER TO BBTable;
  BBoperation: TYPE = {replace, paint, invert, erase};
  BBsourcetype: TYPE = {block, compliment, andgray, gray};
  GrayArray: TYPE = ARRAY [0..3] OF WORD;
  GrayPtr: TYPE = POINTER TO GrayArray;

  BBTable: TYPE = MACHINE DEPENDENT RECORD
    [
    pad: [0..7777B],
    sourcetype: BBsourcetype,
    function: BBoperation,
    unused: CARDINAL,
    dbca: BMptr,        -- destination BCA
    dbmr: CARDINAL,     -- destination width(in words)
    dlx: CARDINAL,      -- destination left x
    dty: CARDINAL,      -- destination top y
    dw: CARDINAL,
    dh: CARDINAL,
    sbca: BMptr,        -- source BCA
    sbmr: CARDINAL,     -- source width(in words)
```

```
    slx: CARDINAL,      -- source left x
    sty: CARDINAL,      -- source top y
    gray0: CARDINAL,    -- four words of "gray"
    gray1: CARDINAL,
    gray2: CARDINAL,
    gray3: CARDINAL
    ];

-- Procedures Implementing RECTANGLES

  -- Bitmap Rectangle Routines

    CreateRectangle: PUBLIC PROCEDURE
      [bitmap: BMHandle, x0, width: xCoord, y0, height: yCoord]
      RETURNS[Rptr];
    DestroyRectangle: PUBLIC PROCEDURE[rectangle: Rptr];
    MoveRectangle: PUBLIC PROCEDURE
      [rectangle: Rptr, x: xCoord, y: yCoord];
    GrowRectangle: PUBLIC PROCEDURE
      [rectangle: Rptr, width: xCoord, height: yCoord];
    ClearBoxInRectangle: PUBLIC PROCEDURE
      [rectangle: Rptr, x0, width: xCoord, y0, height: yCoord, gray: GrayPtr];
    DrawBoxInRectangle: PUBLIC PROCEDURE
      [rectangle: Rptr, x0, width: xCoord, y0, height: yCoord];
    ScrollBoxInRectangle: PUBLIC PROCEDURE
      [rectangle: Rptr, x0, width: xCoord, y0, height: yCoord, incr: INTEGER];
    InvertBoxInRectangle: PUBLIC PROCEDURE
      [rectangle: Rptr, x0, width: xCoord, y0, height: yCoord];
    IsRectangleVisible: PUBLIC PROCEDURE[rectangle: Rptr]
      RETURNS[BOOLEAN];
    WriteRectangleChar: PUBLIC PROCEDURE
      [rectangle: Rptr, x: xCoord, y: yCoord, char: CHARACTER, pfont: FAptr]
      RETURNS[xCoord, yCoord];
    WriteRectangleString: PUBLIC PROCEDURE
      [rectangle: Rptr, x: xCoord, y: yCoord, str: STRING, pfont: FAptr]
      RETURNS[xCoord, yCoord];
    SaveRectangle:PUBLIC PROCEDURE [rectangle: Rptr] RETURNS [POINTER];
    RestoreRectangle:PUBLIC PROCEDURE [rectangle: Rptr, SegPtr: POINTER];

  -- Bitmap Routines

    GetDefaultBitmap: PUBLIC PROCEDURE
      RETURNS[BMHandle];
    CreateBitmap: PUBLIC PROCEDURE
      [pagesformap, wordsperline: CARDINAL]
      RETURNS[BMHandle];
    DestroyBitmap: PUBLIC PROCEDURE[mapdata: BMHandle]
      RETURNS[POINTER];
    UpdateBitmap: PUBLIC PROCEDURE [mapdata: BMHandle]
      RETURNS[DCBptr];
    ReallocateBitmap: PUBLIC PROCEDURE
      [mapdata: BMHandle, pagesformap, wordsperline: CARDINAL];
    DisplayBitmap: PUBLIC PROCEDURE
      [mapdata: BMHandle];
    UnDisplayBitmap: PUBLIC PROCEDURE[mapdata: BMHandle];
    CursorToMapCoords: PUBLIC PROCEDURE
      [mapdata: BMHandle, x: xCoord, y: yCoord]
      RETURNS[xCoord, yCoord];
    RectangleToMapCoords: PUBLIC PROCEDURE
      [rectangle: Rptr, x: xCoord, y: yCoord]
      RETURNS[xCoord, yCoord];
    CursorToRecCoords: PUBLIC PROCEDURE
      [rectangle: Rptr, x: xCoord, y: yCoord]
      RETURNS[xCoord, yCoord];


  -- Display Stuff

    DisplayOff: PUBLIC PROCEDURE [backgtype];
    DisplayOn: PUBLIC PROCEDURE;


  -- Font Stuff

    ComputeCharWidth: PUBLIC PROCEDURE
```

```
      [char: CHARACTER, font: POINTER]
      RETURNS [CARDINAL];
    GetDefaultFont: PUBLIC PROCEDURE
      RETURNS [FAptr, CARDINAL];
    GetFont: PUBLIC PROCEDURE [filename: STRING]
      RETURNS [SegmentDefs.FileSegmentHandle];
    LoadFont: PUBLIC PROCEDURE [segment: SegmentDefs.FileSegmentHandle]
      RETURNS [p: Fptr];

  -- Hard/Soft BitBlt Interface

    BitBlt: PUBLIC PROCEDURE [ptr: BBptr];

-- Machine Implimented Procedures

  CONVERT: MACHINE CODE
    [char: CHARACTER, font: POINTER, DestWord: BMptr,
    ScanLineLength: CARDINAL, DestBit: [0..15]]
    RETURNS [CARDINAL, [0..15], BMptr] =
     -- Scan Convert the char into the bitmap--
      INLINE [Mopcodes.zCONVERT];

  HardwareBitBlt: MACHINE CODE
    [ptr: BBptr] = INLINE [Mopcodes.zBITBLT];

  END. of RectangleDef
```