CreateMandelbrotImageThreads

Spawn a series of threads up to maxThreads, each of which will calculate the points in successive sections of the image, defined by a start and stop row in threadInfo.

Check that noRowsCalcd is 0 and if it’s not, issue error msg that we tried to recalc whilst a recalc was already underway and return.

If it is zero, for 0 to maxThreads – 1, call AddImageCalcThread for the relevant threadInfo element.

When any thread finishes, the backgroundWorkerCalcs\_RunWorkerCompleted function

* Copies the updated bitmap to the main bitmap and disposes of the section bmp and invalidate that section of the main bmp image so it is repainted.
* checks the static noRowsCalcd value to see if all rows have been issued for calculation.
* If not all issued for calc, calls AddImageCalcThread to reactivate this thread for the next section.
* If all issued for calc, if this is the last thread running (check activeThreadCount) remove wait mousepointer and reset noRowsCalcd to 0. If not the last active thread running then simply return so that the next finishing thread can do the finalisation work.

AddImageCalcThread(threadInfo element)

Find the specified threadInfo element in the ThreadInfo array, update its info, create a new section bmp and activate it’s background worker.

createMandlebrotImage()

Calcs the image values for the specified rows in the main bitmap.

* mainBmpSide = the width and height of the main bitmap in pixels
* startRow = the pixel row number (first row = 0) of the first row in the main image to calc
* endRow = the row number of last row to calc
* minX = double precision min X value of scaled bounds of the main Bmp image
* maxX = double precision max X value of scaled bounds of the main Bmp image
* minY = double precision min Y value of scaled bounds of the main Bmp image
* maxY = double precision max Y value of scaled bounds of the main Bmp image
* bmpSection = working bmp with width mainBmpSide and height (endRow – startRow + 1)

ThreadInfo array contains, for each thread:

* mainBmpSide = the width and height of the main bitmap in pixels
* startRow = the pixel row number (first row = 0) of the first row in the main image to calc
* endRow = the row number of last row to calc
* minX = double precision min X value of scaled bounds of the main Bmp image
* maxX = double precision max X value of scaled bounds of the main Bmp image
* minY = double precision min Y value of scaled bounds of the main Bmp image
* maxY = double precision max Y value of scaled bounds of the main Bmp image
* bmpSection = working bmp with width mainBmpSide and height (endRow – startRow + 1)
* bwThread = background worker thread

Consider creating an image builder class which implements:

isBuildInProgress(Image)

returns true if an image build is in progress.

buildImage(Image, ImageCalcData, buildImageSectionCompletedEventHandler, buildImageCompletedEventHandler)

Starts off a new image build.

buildImage(Image, ImageCalcData)