Fractal X plans

1/16/18

See Direct3D Win32 Game1 in code\directx code

Made simple vertex and indices and loaded

Next try to set matrices and try rotating -ok

Then try making a texture

1/20/18

It rotates but it seems like some sections are transparent – try to figure out

1. Add a menu with something to stop the timer
2. Add something to set a fixed phi and theta
3. Figure out if the vertexes are drawing correctly
4. Add lights and menu to set
5. Add texture

In looks like the current project is Win32 see if we can transfer the code to a MFC project. Call the new project FractalX

Made project FractalX in git\_working

Find out what property view and others are about

need to import direct x libraries

Seem to have libraries but still a conflict with *D3D11CreateDevice and precompiled headers*

After that:

1. Connect renderer to view and get it to display
2. Clean up exceptions, etc.

d3d11.lib

dxguid.lib

1/27/18

Got it to display

Commit -ok

Next clean up. Examine and see how it can be refactored. Maybe replace exceptions or throw specific one and catch.

2/3/18

Make exceptions with descriptions

Cleaned up.

Moved vertex creation to a factory and set before initialization.

Then

It rotates but it seems like some sections are transparent – try to figure out

1. Add a menu with something to stop the timer
2. Add something to set a fixed phi and theta
3. Figure out if the vertexes are drawing correctly
4. Add lights and menu to set
5. Add texture

2/10/18

Change namespace to DXF

Make new folder DFX and put all DXF files there –ok

Change implementation of Renderer to use new settings – problem is I can’t find MathHelper::ToRadians() may have to implement manually –ok

2/13/18

Implemented different rotation modes –ok

commit -ok

Fix dialog to do input validation -ok

Add z angle -ok

Add texture

2/17/18

Look at how to implement texture

we need to make the colors first in the form of

*std*::*vector*<*uint32\_t*> GradientColors;

*uint32\_t* ColorrefToXmcolor(*COLORREF* cr, *uint32\_t* alpha)

{

// XMCOLOR is structured 0xaarrggbb

// COLORREF is structured 0x00bbggrr

return (alpha << 24) | (*GetRValue*(cr) << 16) | (*GetGValue*(cr) << 8) | *GetBValue*(cr);

// An alternative method

// return XMCOLOR(GetRValue(cr) / 255.0f, GetGValue(cr) / 255.0f, GetBValue(cr) / 255.0f, alpha / 255.0f);

}

violet -> blue -> Cyan -> Green -> Yellow -> Red -> White

0 -> 0.167 -> 0.333 -> 0.5 -> 0.666 -> 0.833 -> 1

1. Make function that spreads colors taking RGB and index ratio (0 to 1) and spreads colors between those values to make palette
2. Check Bulb and see if we can use texture

2/20/18

1. Put Dx code in DLL
2. link to it
3. Create a unit test project

2/24/18

Need to used an MFC DLL because windows is included in DirectX (kind of)

So use project Dxf. Delete project DxHelper when done

Main problem remaining is how to use shared\_ptr in export.

Seems like it’s not solvable. Try making the vertex calculation internal in the DLL so it doesn’t need to be passed back and forth.

2/27/18

Updated FractalX3 from work code

Commit

Changed format to VertexPositionNormalTexture and added texture and DxfColorFactory

Next try making a static library and see if we can export a vector

3/3/18

Make new project DxSupport as static link library

Commit

Next make a test project

Commit

Added DxTests

Remove changes to old FractalX sandbox -ok

Add more tests

3/6/18

1. Move menu up for rotation dialog -ok
2. Have it start on rotation x? -ok
3. Make test for enums -ok
4. Move texture to external method – ok
5. Add buffer creation to external methods –ok
6. Commit
7. Make vertex buffer a cube –ok
8. commit
9. Allow selection of different textures – need to finish dialog
10. Research fractal code
11. Consider error log

3/17/18

Added texture dialog

Cleaned up menus

Next think about how to implement fractal vertexes

Look at solution Fracture, project Bulb

Algorithm Ray Tracing and Stretch Ray Tracing

Plain ray tracing is RayTracingBulbCalculator

Stretch ray tracing is – DistanceRayTraceBulbCalculator

Study plain first

Also BulbDistanceCalculator

RayTracingBulbCalculator

double distance = RayMarch(start, direction, rayMarchParams, distanceEstimateParams);

public double RayMarch(Vertex start, Vec3D direction, RayMarchParameters rayMarchParams, DistanceEstimatorParameters distanceEstimateParams)

{

double totalDistance = 0.0;

int steps;

double lastDistance = *Double*.*MaxValue*;

for (steps = 0; steps < distanceEstimateParams.MaxRaySteps; steps++)

{

\_p = VectorMethods.Add(totalDistance \* direction, start);

double distance = EstimateDistance(\_p);

totalDistance += distance / distanceEstimateParams.StepDivisor; // note change

if (distance < distanceEstimateParams.MinRayDistance || distance > lastDistance)

break;

lastDistance = distance;

}

return 1.0 - ((double)steps) / distanceEstimateParams.MaxRaySteps;

}

public double EstimateDistance(Vec3D pos)

{

Vec3D z = new Vec3D(pos);

double dr = 1.0;

double r = 0.0;

for (int i = 0; i < Iterations; ++i)

{

r = z.Length();

if (r > Bailout)

break;

if (*Double*.*IsNaN*(r))

break;

BulbCalculator.CalculateNextCycle(ref z, ref r, ref dr);

z += pos;

}

return 0.5 \* *Math*.*Log*(r) \* r / dr;

}

public void CalculateNextCycle(ref Vec3D z, ref double r, ref double dr)

{

// convert to polar coordinates

double theta = *Math*.*Acos*(z.Z / r);

double phi = *Math*.*Atan2*(z.Y, z.X);

dr = *Math*.*Pow*(r, \_power - 1.0) \* \_power \* dr + \_C;

// scale and rotate the point

double zr = *Math*.*Pow*(r, \_power);

theta = theta \* \_power;

phi = phi \* \_power;

// convert back to Cartesian coordinates

z = zr \* new Vec3D(*Math*.*Sin*(theta) \* *Math*.*Cos*(phi), *Math*.*Sin*(phi) \* *Math*.*Sin*(theta), *Math*.*Cos*(theta));

}

// calculate normal

// vec3 n = normalize(vec3(DE(pos + xDir) - DE(pos - xDir),

// DE(pos + yDir) - DE(pos - yDir),

// DE(pos + zDir) - DE(pos - zDir)));

Vec3D CalculateNormal(Vec3D pos, double normalDelta)

{

double plusX = EstimateDistance(VectorMethods.AddX(pos, normalDelta));

double minusX = EstimateDistance(VectorMethods.AddX(pos, -1.0 \* normalDelta));

double plusY = EstimateDistance(VectorMethods.AddY(pos, normalDelta));

double minusY = EstimateDistance(VectorMethods.AddY(pos, -1.0 \* normalDelta));

double plusZ = EstimateDistance(VectorMethods.AddZ(pos, normalDelta));

double minusZ = EstimateDistance(VectorMethods.AddZ(pos, -1.0 \* normalDelta));

var norm = new Vec3D(plusX - minusX, plusY - minusY, plusZ - minusZ);

return norm.Normalize();

}

Problem – how to scan a sphere and convert to vertices and indices

Vertices to approximate a sphere

Start with 6 vertices

0 (0,1,0),

1 (1,0,0),

2 (0,0,-1),

3 (-1,0,0),

4 (0,0,1),

5 (0,-1,0)

Make counter clockwise

Make 8 triangles with indices: 0,1,2, 0,2,3, 0,3,4, 0,4,1, 5,2,1, 5,3,2, 5,4,3, 5,1,4

Then for n cycles divide each triangle into 4. Divide each line in half and add a new vertex between the two points. Create a new triangle for each old vertex with 2 new points and a new triangle with all 3 new vertices. Need to search if a new vertex was recently created.

Total number of triangles T = 2 x 4^(n+1)

Total number of vertices V = T/2 + 2 or 2 + 4^(n+1)

3/27/18

Next

1. Make unit tests for TriangleData GenerateCrudeTriangles(int depth);-ok
2. Normalize vertices -ok
3. Make conversion function for models -ok
4. Test models

3/31/18

n 0 1 2 3 4 5

V 6 18 66 258 1026 4098

T 8 32 128 512 2048 8192

It works!

Commit

4/3/18

Committed changes including SphereApproximator

Next plan fractal calculation

Started on Ray Tracing method with distance calculation.

4/14/18

3 problems

1. Have to scale world. Currently using 0.1 for each dimension- temp fix
2. The object is still a sphere – the return vector is always the same distance away?
3. The color is always white

EstimateDistance is returning a greater value after the initial and it always seems to be the same? Fracture returns smaller values.

VS not working. Try repair. Then see

<https://social.msdn.microsoft.com/Forums/en-US/aaaa3e11-7835-47b8-9c4c-15171dce05b1/visual-studio-not-starting-error-no-inprocserver32-registered-for-package-visual-studio-logging?forum=vsx>

4/21/18

Have the program building again use FractalX2 sandbox

Don’t know what the problem was, seems like solution was broke.

Not sure why we’re not getting a fractal. The distance does not decrease but it does in fracture. Math looks the same but something is not.

Compare distance calculation carefully to fracture. Maybe a function behaves differently.

4/24/18

Got it working in FractalX2 sandbox. The problem was the direction was the same as the starting point.

Committed

Next add a dialog to set the parameters, such as view distance, both perspective and distance from the camera to start the calculation. Then other tracing parameters. Also the number of points or n depth.

4/28/18

Remove Properties, Output and File view –ok

Consider adding a ribbon bar

<https://docs.microsoft.com/en-us/cpp/mfc/walkthrough-updating-the-mfc-scribble-application-part-1>

c:\users\steve\source\repos

Working on ribbon bar.

m\_wndRibbonBar commented out in mainframe because resources not loading right. Problem with IDR\_RIBBON copied from c:\users\steve\source\repos\TempRibbonBar

I think the problem is with the IDB\_WRITESMALL IDB\_WRITELARGE which weren’t copied correctly. Give up for now.

5/1/18

Tried to set perspective but it doesn’t work

5/5/18

Fixed perspective and added View dialog

Works good n= 7 is slow > 10 min

Next save to hard disk. OK

Next add tracing params and consider not making whole sphere

5/8/18

Created CModelSheet and ModelData

Need to set a DXF Render SetVertexData to take a ModelData and also need it to return one. -ok

Then add handler in view to call sheet to fill in data and a method to call renderer to recalculate

Need to work out how to determine if we can call the method to recalculate. Some kind of progress dialog

5/12/18

Hook up property sheet in view

Add method to recalculate model -ok

Progress

Display number of vertexes in property page

Try calculating single octant of sphere

5/15/18

Added progress. The last stage of GenerateCrudeTriangles()

is a little slow. When n is big the last cycle is long. Leave for now and hook up octants.

Next display in Param dialog the number of vertexes for n

Total number of triangles T = 2 x 4^(n+1)

Total number of vertices V = T/2 + 2 or 2 + 4^(n+1)

This can’t be right, check

1. Display number of vertexes and triangles
2. Add cancel button
3. Add partial sphere calculation

IDC\_VERTEX\_NUMBER\_EDIT

IDC\_TRIANGLE\_NUMBER\_EDIT

5/19/18

Got the display of the vertices and triangles.

Cancel is more complicated. Will need to do the calculations in a worker thread and check an event or modern equivalent set from the progress dialog.

Need to research. Don’t do same thing as before use more modern approach.

Old method had the progress dialog send a message to the parent that set an event. The worker thread method checked the event.

5/22/18

Examples for threading

bool failed = false;

CDialogTaskMonitor ProgressDialog(*AfxGetMainWnd*());

std::*thread* exportThread( [&](){ExportToCsvThreadMethod( SaveFileDlg.*GetPathName*(), pMatrix, &ProgressDialog, failed, isGCxGC, isSaturn);} );

// Show the progress dialog (until the task has completed)

ProgressDialog.SetTitle( *\_T*("Exporting Mass Calibration Matrix") );

ProgressDialog.*DoModal*();

exportThread.*join*();

std::mutex m\_mutex;

std::lock\_guard<std::mutex> guard(m\_mutex);

#include <condition\_variable>

m\_condVar.notify\_one();

    std::unique\_lock<std::mutex> mlock(m\_mutex);

    // Start waiting for the Condition Variable to get signaled

    // Wait() will internally release the lock and make the thread to block

    // As soon as condition variable get signaled, resume the thread and

    // again acquire the lock. Then check if condition is met or not

    // If condition is met then continue else again go in wait.

    m\_condVar.wait(mlock, std::bind(&Application::isDataLoaded, this));

[std::condition\_variable](http://en.cppreference.com/w/cpp/thread/condition_variable) cv;

[std::mutex](http://en.cppreference.com/w/cpp/thread/mutex) cv\_m;

int i;

void waits(int idx)

{

[std::unique\_lock](http://en.cppreference.com/w/cpp/thread/unique_lock)<[std::mutex](http://en.cppreference.com/w/cpp/thread/mutex)> lk(cv\_m);

    if(cv.wait\_for(lk, idx\*100ms, []{return i == 1;}))

[std::cerr](http://en.cppreference.com/w/cpp/io/cerr) << "Thread " << idx << " finished waiting. i == " << i << '**\n**';

    else

[std::cerr](http://en.cppreference.com/w/cpp/io/cerr) << "Thread " << idx << " timed out. i == " << i << '**\n**';

Plans for progress

1. Create progress dialog as member of view -ok
2. Make messages for progress and closing progress -ok
3. Add handlers to view for those -ok
4. Call methods on progress dialog or close from the handlers -ok
5. Close handler should update the renderer
6. Add method to view that calculates the model in worker thread
7. Thread method should take a method for updating progress – it should call postmessage and a condition variable to wait on for cancel
8. Thread method should also take a method for done
9. How to pass model back from worker? Used to return data as a pointer in message but there must be a better way.
10. Use a shared variable that is passed to the thread and mutex
11. promise <https://cpppatterns.com/patterns/pass-values-between-threads.html>
12. task <https://cpppatterns.com/patterns/execute-task-asynchronously.html>

I think you can put a shared or unique ptr in a future. Store the future in the view while the thread is calculating then when the complete message is done retrieve from the future

5/25/18

Have it set up to do calculation with future but OnInitialUpdate() tries to initialize renderer without any vertices which fails.

Have it start without vertexes and then add cancel.

5/27/18

Got it working with registered messages and std::thread

Commit -ok

Pimple and commit -ok

Add cancel – couldn’t get it to work with condition\_variable or atomic??????

6/1/18

Got it to work with an atomic bool.

Commit

Start on Seed portions – work out math.

6/4/18

Vertices and Triangle counts for number of starting triangle

Starting triangles == 8

T = 2 x 4^(n+1)

V = T/2 + 2 or 2 + 4^(n+1)

Starting triangle = 1

T= 4^n

levels =

l = 2;

for x = 1 to n: l =+ l-1

V = Sum(0 to levels)

n l V

0 2 3

1 3 6

2 5 15

3 9 45

4 17 153

Starting triangle = 2

T = 2\* T= 4^n

V = 2\*V(1) – l

Starting triangle = 4

T= 4^n

V = 4\*V(1) – 3l

6/9/18

1. Add seed triangles combo -ok
2. Calculate triangles and vertices -ok
3. Implement in calculation –ok
4. Commit –ok
5. Update back up drive-ok

Implemented SeedTriangles so partial sphere of vertices could be calculated for model.

Put hard drive in ceiling

6/23/18

1. Allow setting of TraceParams in SetModel
2. Compare BulbDistanceEstimator and FractionalDistanceEstimator in Fracture and figure out what we’re missing
3. Add C
4. Allow setting colors for texture. Consider how to enter and store textures

struct *TraceParams*

{

float *Distance* = 10.0f;

*DirectX*::*SimpleMath*::*Vector3* *Origin* = *DirectX*::*SimpleMath*::*Vector3*(0.0f, 0.0f, 0.0f);

// Distance estimate params

int *MaxRaySteps* = 1000;

double *MinRayDistance* = 0.0001;

double *StepDivisor* = 10.0;

double *Bailout* = 2.0;

int *Iterations* = 256;

float *NormalDelta* = 0.01f;

};

For 1:

Add a TraceParams to the view -ok

Pass to CreateBulb() in VertexFactory.cpp -ok

Add property page to CModelSheet -ok

Make new page IDD\_TRACE\_PARAMS\_PAGE CTraceParamsPage -ok

Add fields

Just added max ray steps so far – commit

Added CTraceParamsPage and TraceParams were added to view.

6/26/18

Commit

Added the rest of the trace params to the trace params page.

Next see 2 and 3 above

Then experiment

Consider saving flat image to bmp

Consider saving image to file with vertex data

Consider speeding up. Maybe storing seed triangles

Consider adding progress for calculating seed triangles

6/30/18

There are two calculations we can add 1)fractional and 2) Stretch.

1. Add a bool for fractional to params. This adds distances/minDistance to the number of steps when calculating color index. See fracture, FractionalDistanceEstimator::RayMarch()
   1. Add bool to params
   2. Implement calculation
   3. Connect
2. Implement stretch method as an option. In fracture this uses RayMarchDistance() it adds an extra step when the distances is stretched before calculating the color index
   1. Study
   2. Implement method to use min max
   3. Add params
   4. Connect
   5. Figure out how to calculate min/max
   6. Implement
3. Add constant C to input params and connect
4. Study how to implement colors/palettes
5. Save image
6. Save to file
7. Save triangles to file to speed up
8. Add progress for calculating seed triangles
9. Maybe make a common method for creating progress methods or think about a better approach

7/3/18

Did 1 and committed

7/5/18

Hooked up Stretch Parameters

Next implement

Two big differences

1. distance is calculated directly and not from the number of steps
2. There can be an earlier estimation calculation which changes progress too

Now we have to return the traceParams with the vertex data because the min/max could change. This is possible because the data is returned in FractalXView::OnCalcFinished()

So when returning to this:

1. put StretchDistanceParams in DxVertexData, set it in the RayTrace() and RayTraceStretch(). Fill in all intermediate steps as needed. -ok
2. Add a method to RayTraceStretch() to calculate min max as needed-ok
3. Commit -ok
4. Update progress in RayTraceStretch().-ok
5. Commit –ok

7/10/18

1. Added constant C param -ok
2. Added Power param -ok
3. Make new page for Fractal params –ok
4. Commit – Added FractalParamsPage -ok
5. Update view when calculation finished in case rotation is fixed

7/14/18

Made it redraw when rotation is fixed and a new model is calculated. Commit

Looking at how colors are saved in other programs

Fractals stores all colors and the pins

Chaos – can’t really tell. Looks like it serializes pins. The pins can include 3 pairs of colors. It include an enum for the spread type and double for the curve

Fracture has alpha. Has pins and each pin has a color and an index (double between 0 and 1)

Bulb has alpha

Plans for colors

1. Make a ColorPin with 3 colors for up to 3 stripes, an enum for strips, and an index. Color should be compatible with texture and include alpha. -ok
2. Make a palette class that stores a vector of pins. Add a serialization method to this. Think about how to serialize in and external to doc. -ok
3. Make method to convert to texture format -ok
4. Make a modeless dialog to display the palette. First just colors then pins. For now just add child dialog to set pin position and color. Later allow sliding pin with mouse.
5. Add method to save and load pin files
6. Connect so texture of renderer can be set
7. Work on serialization of whole doc.

Texture uses uint32\_t and just seems to spread whatever colors are given it. So we will store the palette as a bunch of rgbs (4 bytes one for each argb) and a double for the index or position between 0 and 1. The doc will have a resolution for the total number of colors in the palette, maybe 10,000. When converting from palette to texture, the app will generate the resolution number of colors (10,000). The colors will be calculated at each position by stretching.

Don’t worry about different types of stretching or the other 2 colors stored for each pin. Just serialize something for those values for now.

There is a gdiplus class called color which is close to what we want but I don’t want to have to use a strange header that might not always be supported. The documentation isn’t clear (<https://docs.microsoft.com/en-us/windows/desktop/api/gdipluscolor/nl-gdipluscolor-color>).

Make our own struct

RawColor

{

unsigned char red,

unsigned char green,

unsigned char blue,

unsigned char alpha

}

And make namespace with conversion methods

enum ColorStretchType {Linear = 0};

ColorPin

{

RawColor rawColor1;

RawColor rawColor2;

RawColor rawColor3;

double index;

ColorStretchType colorStretchType;

double curve;

}

7/17/18

For exporting colors see Chaos void cImageDoc::OnFileExportcolors()

Implemented serialization

1. Think about how to use with texture -ok
2. Make hard coded pins collections -ok
3. Make method to convert to texture
   1. Include method to make sure pin indices are from 0 to 1 and in order –put in DxColor -ok
4. Add one to view class and hook up to texture
5. Make a modeless dialog to display, load and save to file.
6. Hook up dialog
7. Test
8. Work on serializing doc with image.
9. Think about serializing raw triangles to speed up calculations

texture colors are in *std*::*vector*<*uint32\_t*>

we use this to make the color

static *uint32\_t* ColorrefToXmcolor(*COLORREF* cr, *uint32\_t* alpha)

{

// XMCOLOR is structured 0xaarrggbb

// COLORREF is structured 0x00bbggrr

return (alpha << 24) | (*GetRValue*(cr) << 16) | (*GetGValue*(cr) << 8) | *GetBValue*(cr);

}

We need a fixed number of colors like 10,000 and spread the colors using pins. This means making a color for each one.

The renderer doesn’t have to know about the palette, only the view. The view can call SetTextureColors with the color array

Before continuing we need to include linkage to DxSupport and DxColor.

7/24/18

Corrected ConvertPalette, etc

Correct library linking for both debug and release, may want to change later

commit

Added initial palette implementation with serialization and conversion. Fixed linking to DxColors.lib

Up to 4

Got exception trying to create texture

D3D11 ERROR: ID3D11Device::CreateTexture2D: The Dimensions are invalid. For feature level D3D\_FEATURE\_LEVEL\_10\_0, the Width (value = 9981) must be between 1 and 8192, inclusively. The Height (value = 1) must be between 1 and 8192, inclusively. And, the ArraySize (value = 1) must be between 1 and 512, inclusively. [ STATE\_CREATION ERROR #101: CREATETEXTURE2D\_INVALIDDIMENSIONS]

D3D11: \*\*BREAK\*\* enabled for the previous message, which was: [ ERROR STATE\_CREATION #101: CREATETEXTURE2D\_INVALIDDIMENSIONS ]

7/28/18

Commit

Reduced number of texture colors to 5000 because of Dx limits

We need a 2D texture because the vertex coordinates specify it. Maybe if we didn’t use an Effect we could code a 1D texture but it’s not worth it.

So revert and discard changes.

inclusively. And, the ArraySize (value = 1) must be between 1 and 512, inclusively. [ STATE\_CREATION ERROR #101: CREATETEXTURE2D\_INVALIDDIMENSIONS]

7/31/18

Commit

Got it to build with *D3D\_FEATURE\_LEVEL\_11\_0 and 10000 colors.*

Next

Have the initial colors created using this method. –ok

Commit

Work on dialog to load, save and display palettes

After pin palette dialog, move fields to doc

Plans for colors

Try a modeless dialog to display the colors then add a dialog that displays all the palettes in a directory. Look at chaos.

See ColorView.cpp

For palette dialog see ColorTreeDlg.h. But this serializes in a bitmap from the file. This draws the bitmaps in a treeview. I don’t know what we can draw out image in.

Maybe a picture control or just draw on the surface. For the palette selection dialog we can use the treeview approach.

8/4/18

Make a modeless dialog

Created a modeless dialog CPaletteViewDlg

Next add call back method to set palette back -ok

Then make double buffered bitmap and draw in dialog

[https://docs.microsoft.com/en-us/cpp/mfc/reference/cdc-class#stretchblt](https://docs.microsoft.com/en-us/cpp/mfc/reference/cdc-class" \l "stretchblt)

[https://docs.microsoft.com/en-us/cpp/mfc/reference/cdc-class#bitblt](https://docs.microsoft.com/en-us/cpp/mfc/reference/cdc-class" \l "bitblt)

Plan for double buffering

See void cColorView::OnDraw(*CDC*\* pDC) for example in Fractals

Make background image of fixed size with width = number of colors and height ~10

Fill background (solid now, stripes later for transparency)

Draw vertical lines, one pixel wide for each color

StretchBlt onto main bitmap/dc

draw pins (later)

Draw to screen

Need 2 CBitmap and CDC s

8/11/18

Removed dxgi.lib from DxSupport but still need in FractalX

Commits

Implemented DoubleBuffer and part of PaletteViewDlg.

Drew bmp on dialog

Next: fix position relative to controls. The draw colors.

8/14/18

1. Fixed bitmap position on dialog – commit
2. Populate palette name -commit
3. Draw colors in bitmap

Two problems

1. The number of colors does not always come out to the requested number – maybe fixed
2. The colors are not smooth, there are some lines – first bad color is about 100

Make sure StretchPaletteIndices(PinPalette& palette, double oldMin, double oldMax) in ColorPin.cpp is correct

8/17/18

Added a pin.

Working on CPinTracker

Made it display all the pins and committed

Next:

1. Add mouse handlers to move the pins
2. Make it update the palette and colors
3. Add methods to save and load the palette to a file
4. Serialize everything

8/21/18

Tried to implement pin movement but failed. Need to figure out

commit bad pins

8/25/18

Fixed at work.

Added NextUp.txt file to directory

This is at the start of it:

1) Fix so left most pin can go to end. Currently it's block about 1/4 th inch to right. -ok

2) Make pins stay in the plane. -ok

3) When pin is let go and it's exactly on top of another pin index, make it move 1 position rt or left -ok

4) Check pin movement

5) Add grid below

First: Palette is sometimes coming out with more or less than 1000 colors? It might be moving the pins to the ends

Commit

Fixed number of colors generated for palette so it should be the requested number.

Fixed problem with space on left of palette dialog

Made pins stay on the x plane

Made sure pins have different indices

8/28/18

Updated at work to import / export palette

Next allow editing of name -ok

Fixed palette color calculation –ok

Add context menu with empty handler to delete pin

IDR\_PALETTE\_VIEW\_CONTEXT\_MENU

ID\_PALETTE\_DELETE\_PIN

9/1/18

Commit

Converted PinEditDlg (not hooked up or tested)

I D\_PALETTE\_EDIT\_PIN

hook up dialog -ok

get working

make max index 999

Commit – Corrected pin index

remove junk from dialog and resources

Remove connect checks IDC\_CONNECT\_CHECK1 remove split checks

IDC\_SPLIT\_CHECK1 remove 3rd color bands

IDC\_BANDC\_EDIT2

Commit - Removed unused resources

Move controls to shrink (maybe add more pins later)

test remaining controls

9/4/18

Move controls

Commit

Hook up to parent so palette is passed back - Transferred pins back from pin edit dlg if OnOk

Test each function

Next - ok

Previous -ok

Delete 1 -ok

Delete 2 -ok

Delete 3 -ok

Insert 1 – inserts correct index but before first pin

Insert 2 – inserts correct index before first pin

Commit - Fixed insert function

Color Index 1 -ok

Color Index 2 -ok

Color Index 3 – bad mixes up other indices

Commit - Fixed pin index 3 bug

Color top 1 -ok

Color top 2 -ok

Color top 3 -ok

Color bottom 1-ok

Color bottom 2 -ok

Color bottom 3 -ok

Color 1 band 1 –isn’t enabled until you reopen the dialog

Color 1 band 2

Color 2 band 1

Color 2 band 2

Curve 1

Curve 2

9/8/18

Commit – Fixed all controls on pin edit dlg

Implement curve palette – it works – no changes needed

Implement banded palette

Commit - Implemented double banded pins

Implement Update button

Commit Made pin edit dlg modeless and implemented Update button

9/11/18

Working on serialization of doc

make extension frc

Chaos\nChaos\nChaos\nChaos Files (\*.cao)\n.cao\nChaos.Document\nChaos.Document

\nFractalX\nFractalX\n\n\nFractalX.Document\nFractalX.Document

FractalX\\nFractalX\nFractalX\ nFractalX Files (\*.frc)\n.frc\nFractalX.Document\nFractalX.Document

9/15/18

Try to remove shared\_ptr’s in doc members

Fix serialization

Commit Fixed serialization

Need SetModified in handlers

Commit: Added Save As button and SetModified() to setters

Need to add mutex to set/get in doc because it can be accessed from another thread. Maybe just vertex data

Commit: Added locks to doc data called from worker thread

Rethink color conversion from palette so we don't need to pass non-const palette all the time

When you change the palette it doesn’t update if it’s not rotating. Thought I fixed this before.

9/18/18

Add function to save file first 6 iterations 8 segments

save as triangles6\_8.triangles

31 sec for triangles only

55 sec for whole thing

24 for the whole thing with the file

0x00000083bdafe9e0 L"C:\\code\\github\\FractalXgit\\FractalX\\x64\\Debug\\triangles6\_8.triangles was not found."

C:\code\github\FractalXgit\FractalX\x64\Debug\Triangles

triangles6\_8.triangles

Next

triangles7\_8.triangles

Commit

Implemented loading triangle files

9/22/18

Commit

Added Triangle folder and ReadMe.txt describing them. – undid because files are too big

Note: Triangles10\_8 and Triangles10\_4 were too big to commit

Working on ConvertPalette

This should be 2 functions

ValidatePalette() – which exists and CalculatePaletteColors

When called:

View:: InitializeColors() – no need for validate

View::OnColors() – no need but should call it

View::SetNewPalette() – should call validate\*

PaletteViewDlg – OnInitDialog() no need to validate

PaletteViewDlg – PaletteChanged() should call or method that sets the palette should validate\*

// this can throw std::exception if there are less than 2 colors

DxColor::ValidatePalette(palette);

Next:

1. Rethink color conversion from palette so we don't need to pass non-const palette all the time-ok
2. Add ReadMe file to explain Triangle files
3. When you change the palette it doesn’t update if it’s not rotating. Thought I fixed this before.
4. Contrast and other color modes HVS?
5. Transparency

Commit

Spit ConvertPalette into ValidatePalette and CalculatePaletteColors –ok

Commit

Made it render if palette, colors, view or perspective changed.

Commit

Added Readme.txt explaining Triangle files

To speed up rendering files containing coordinates of triangles used to build the model can be placed in a folder named "Triangles" within the same path as the executable. The files have the form of triangles10\_2.triangles where the first digit is the number of iterations and the second the number of Seed Triangles (see the Model dialog vertex page). These can be built using the Export Triangles menu item under the Help menu. But the source has to be changed for each different number of iterations and seed triangles. See CFractalViewImpl:: void OnExportTriangles().

The seed triangles are used to calculate the model and if they need to be calculated they take almost the same amount of time as the model which can be over an hour. Using triangle files speeds up the calculation almost 2x.

9/29/18

Contrast params consist of two byte[3] for min and max

HSL params consist of 3 pairs of double for min/max hue, saturation and lightness

Also need something to determine which mode to use none, contrast or HSL

Apply when calculating colors from palette. Can just scan palette pins since these include min max colors

Dialog should set the values (min/max) and calculating the colors should apply

1. Scan code for locations of where to put dialog, where colors are generated, and where to store params
2. Make param class, ColorContrast-ok
3. Make dialog
4. Make calculation for contrast
5. Hook up
6. Test
7. Make calculation for HSL

1 – First put dialog button on palette dlg then on main menu/toolbar

ColorUtilities:: CalculatePaletteColors() and internally at AddColor()

Store Contrast Params in Doc

2 – put params and calculations in DxColors

Commit

Added ColorContrast Params

Made IDD\_CONTRAST\_DLG with IDC\_MODE\_COMBO

CContrastDlg

Made skeleton of dlg. Need to add to palette Dlg and connect params

10/2/18

Commit

Partial implantation of contrast dialog

10/6/18

1. Implement OnHScroll -ok
2. Add buttons for Palette dialog and handler -ok
3. Display and test -ok
4. Hook up edit boxes
   1. Implement DDX for HSL edit -ok
   2. Fix so it doesn’t say enter a number and the initial values are correct -ok
   3. OnLoseFocus for Edits-ok
   4. Make sure min < max-ok
5. Add implementation to use contrast
6. Add implementation to use HSL
7. Maybe make modeless
8. Add to main window

Commit- Implemented more contrast dlg

Commit – Fixed OnModeChanged()

Commit – Connected Edits in contrast dlg

10/9/18

Commit

Fixed contrast dlg validation

Commit Added contrast to color method

Next

HSL – look at Fracture RawImage.cs – not sure how to handle because it determine HSL min max from colors in image. Apply to palette colors instead of image colors

Modeless contrast update

Add to main window

10/13/18

Implemented converting colors with contrast but

1. Palette looks choppy -ok
2. The contrasted colors aren’t used in the model –ok

Commit - Implemented contrast

10/19/18

see <https://stackoverflow.com/questions/39118528/rgb-to-hsl-conversion>

<https://stackoverflow.com/questions/3018313/algorithm-to-convert-rgb-to-hsv-and-hsv-to-rgb-in-range-0-255-for-both>

Commit HSL implementation with bugs

Got bugs in RGB to HSL Hue calculation –

1) for magenta hue is -60 and should be 300. This might just be adding 360 when the hue is negative but I don’t see that in Fracture or other examples

2) I also think there is bug converting back in rounding since only primary colors come back

Commit Fixed to hsl

Try the calculation from

<https://gist.github.com/fairlight1337/4935ae72bcbcc1ba5c72>

First increase H to 360

Made ToHSL() function

need to replace ToRgb()

10/27/18

Commit: Fixed hsl/rgb conversion

11/3/18

Hook up CCOlorSelectorDlg to CPinEditDlg -ok

Add button to bring up windows dlg –ok

Add color square with DC -ok

Change to gdiplus

draw strips behind square

convert the rest of the colors in pin edit dlg

Add color square – draw stripes in background and color on top with transparency – will need gdiplus

Next work on transparency

1. Display a with RGB in pin dialog when you select a pin
2. When you click on a color allow entering an argb and add another button to use color dialog
3. Turn on gdiplus for app
4. Display stripes behind color square and draw color square with gdiplus.

Commit: Display ARGB in PinEdit dlg

Next Features

1. Transparency at model level and pin level
2. Add buttons to the main window/toolbar
3. Background image
4. Mouse handlers or buttons to move model or rotate

12/15/18

Commit

Got transparency working

Determine how hard it would be to display tranparency in palette dialog

Add update button to palette dialog

Add buttons to toolbar for most tasks

12/18/18

Commit

Implemented Update button on palette dialog and enable/disabling.

Commit

Implemented transparancyin PinEditDlg

12/22/18

Commit

Implemented transparency in PaletteViewDlg

Next

1) Toolbar buttons

2) Mouse handlers or buttons to move model or rotate

3) Export bitmap, jpg, png

4) Zoom region by selecting

5) Check what is saved with image. Seems like rotation isn’t saved not sure what is.

Current menu New menu Buttons

Rotation Calculate Model -make shapes circle square

Set Colors – remove Set Perspective -mave two rects

Set Perspective Set View - little camera

Set View Rotation - curved arrows

Set Model Edit Palette -like other apps

Edit Palette

ID\_EDIT\_SETPERSPECTIVE

ID\_EDIT\_SETVIEW

ID\_EDIT\_ROTATION

ID\_EDIT\_PALETTE

Commit

Removed Set Colors and rearranged edit menu

IDR\_FractalXTYPE IDR\_FRACTALX\_TYPE

Commit

Changed IDR\_FractalXTYPE to IDR\_FRACTALX\_TYPE

Problem: Can’t get IDR\_FRACTALX\_TYPE toolbar to show up

12/25/18

Can’t add separate toolbar for IDR\_FRACTALX\_TYPE. Just add to IDR\_MAINFRAME\_256

Commit

Added toolbar buttons

Next consider dialog and mouse moves for rotating and moving forward and back

There are rotations around x, y and z

There are target positions x, y and z

12/28/18

ID\_EDIT\_ANGLEPOSITION

Make structure to hold current angles and positions (if doesn’t exist)

*RotationParams RotationDlg.h*

*tuple<float, float, float>*

Make modeless dialog -ok

Make coordinate bitmap and display in dialog

Make button images

Make buttons

Connect each

Commit

Added shell of CpositionAngleDlg

12/31/18

Making a custom picture control

Making picturectrl

image is in: C:\docs\programming FractalX\images\coords.bmp

Got it to display a bmp.

Next

1) Make a parent Cwnd in the ctrl and clean up -ok

2) Have it catch mouse move and post message to parent with coord. Will use message later for clicks in rect. Have parent display x,y so we can figure out rectangles. Make it an option to send these messages.-ok

3) Make IDs for commands +x, -x, cwX, ccwX, +y, -y, ccY, ccwY, +z, -z, ccZ, ccwZ -ok

4) Create a list of objects that include a Crect and an int. Make members for these in control and setters. Make actual list object in dlg. Add on L but down handler that checks rectangles.-ok

5) Send message to parent when rect is hit. -ok

6) Figure out how to implement actual conversion from ctrl message to rendering -ok

7) Implement and test

7.1) Need to make position update the view (rotation OK)

7.2) Remove fractx view handlers and call imp directly

7.3) Fix direction of position change (not sure about rotation)

7.5) Update the position angle dlg when rotation changes

7.7) Update OnKillAnglePosition so it doesn’t do anything if rotating

8) Figure out if GDI+ would be better so we could use png and jpg

9) Use a local resource for the image or at least a safer path

Commit

Implemented CclickMappedPictureCtrl and used in CpositionAngleDlg

1/5/19

1) Make fixed the default rotations -ok

2) Make setting view update image -ok

3) Fix direction -ok

4) Commit – Made changing position update rendering and reversed directions of position and angles. -ok

5) Experiment with mapped image control (when you click make color change on symbol)

It didn’t draw the inverted colors?

1/8/19

SRCCOPY works but the small bitmap is always black, can’t set the colors. Consider using cdib. Added Cdib from Fractals.

Eventually need to figure out why moving z doesn’t work as expected.

Eventually clean up Cdib to remove unnecessary MFC stuff.

1/12/19

Forget HBITMAP use gdiplus

<https://docs.microsoft.com/en-us/windows/desktop/api/gdiplusheaders/nl-gdiplusheaders-bitmap>

<https://stackoverflow.com/questions/3340017/gdiplusbitmap-to-byte-array>

https://mfranc.com/programming/operacje-na-bitmapkach-net-1/

LockBits not working right???

1/15/18

Commits

Got basic drawing working of bitmap in ClickMappedPictureCtrl.cpp

Added function for shading

Next need to add separate rects for area to shade in addition to hot click rect.

1/19/19

Study existing code and decide if it should be moved out, cleaned up etc

Add second rect to structure to use for showing selection

Commit

Finished CclickMappedPictureCtrl.

Commit

Remove cdib

Next

1) Compare old dialogs with new ones and if values are updated in old dialogs. - seems OK

2) Figure out why z movement doesn’t seem right. -ok

3) Mouse move image directly. Maybe just position and not angles

4) Save bitmap. Eventually will want to save different sizes and resolutions.

Moving the target z has no effect but moving the camera does?

So far it seems like the target vector is put into the view matrix but we are transforming and it should go into the world matrix.

Commit

Fixed translation especially for z

1/22/19

Move x,y with right click

Zoom z with mouse wheel

Hook up x and y but it doesn’t move. Maybe it should be divided by less than 1000?

<https://docs.microsoft.com/en-us/windows/desktop/inputdev/wm-mousewheel>

1/26/19

Commit

Made it move model along x and y using control and right mouse button down.

Commit

Made it move model along the z axis using control button and mouse wheel

Commit

Make it redraw when the window is resized.

Commit

Make all rendering include translation and rotation around fixed angles. Translation matrix is multiple after rotation

Next figure out how to save screen to bitmap

This is a start

<https://social.msdn.microsoft.com/Forums/vstudio/en-US/bcdbb1b7-29ba-4478-b69c-815c7353f89a/how-can-i-do-fullscreen-capture-using-directx10-or-directx11?forum=vcgeneral>

1/29/19

Saving a bitmap

1) Find handler and extend -ok

2) Have renderer return a Cbitmap

It actually might make more sense to return a Cimage Gdiplus object

2/6/19

// look at

<https://docs.microsoft.com/en-us/windows/desktop/gdiplus/-gdiplus-retrieving-the-class-identifier-for-an-encoder-use>

Can’t get hdc to swap chain- see

<https://docs.microsoft.com/en-us/windows/desktop/api/dxgi/nf-dxgi-idxgisurface1-getdc>

I think the problem is

You must specify that the texture requires read and write access by the GPU. For example, set the **Usage** member of the  [D3D11\_TEXTURE2D\_DESC](https://msdn.microsoft.com/90c0f877-daf5-4b3d-9846-5bb414c55461) structure to D3D11\_USAGE\_DEFAULT.

<https://docs.microsoft.com/en-us/windows/desktop/api/d3d11/ne-d3d11-d3d11_resource_misc_flag>

So see if we can change the immutable buffers to this instead of *D3D11\_USAGE\_IMMUTABLE*

*Tried changing IMMUTable to D3D11\_USAGE\_DEFAULT but that still produces an invalid creation error??*

*See DxFactoryMethods.cpp line 78*

2/9/19

Got it copying the CDC from the swap chain but when you open the bmp file there is nothing in there. It looks like the bitmap info is written to the structures. Not sure if the Gdiplus Bitmap is working right.

Maybe try copying to something else before creating the bitmap?

Some things to consider

https://github.com/Microsoft/DirectXTK/wiki/ScreenGrab

<https://social.msdn.microsoft.com/Forums/en-US/2c702b43-7920-401c-8252-5e9f4dbe6541/how-can-i-capture-the-screen-shot-rendered-by-directx?forum=wingameswithdirectx>

*CDC* dcBitmap;

*CBitmap* Bitmap;

dcBitmap.*CreateCompatibleDC*(&dcScreen);

Bitmap.*CreateCompatibleBitmap*(&dcScreen, rectView.*right*, rectView.*bottom*);

dcBitmap.*SelectObject*(&Bitmap);

bool fReturn = Render(&dcBitmap, rectView);

<https://blogs.msmvps.com/peterritchie/2006/09/25/save-cbitmap-to-file/>

        CBitmap bitmap;

        bitmap.CreateBitmap(width, height, 1, 32, rgbData);

        CImage image;

        image.Attach(bitmap);

        image.Save(\_T("C:\\test.bmp"), Gdiplus::ImageFormatBMP);

Here is render code

*HDC*      hdc3D;

// new stuff - might be wrong :(

// see https://msdn.microsoft.com/en-us/library/windows/desktop/ff471345(v=vs.85).aspx

*ComPtr*<*IDXGISurface1*> pSurface1;

*HRESULT* hr = m\_swapChain->GetBuffer(0, \_\_uuidof(*IDXGISurface1*), (void\*\*)pSurface1.*GetAddressOf*());

if(FAILED(hr))

         return false;

pSurface1->*GetDC*(FALSE, &hdc3D);

auto hbm3D = static\_cast<*HBITMAP*>(*GetCurrentObject*(hdc3D, *OBJ\_BITMAP*));

*BITMAPINFO*       BitmapInfo;

BitmapInfo.*bmiHeader*.*biSize* = sizeof(BitmapInfo.*bmiHeader*);

BitmapInfo.*bmiHeader*.*biBitCount* = 0;

auto layout = m\_pCore->GetLayout();

VERIFY(*GetDIBits*(hdc3D, hbm3D, 0, layout.rectFull.*Height*(), nullptr, &BitmapInfo, *DIB\_RGB\_COLORS*));

// Force away from BI\_BITFIELDS so it doesn't stomp on other stack variables

BitmapInfo.*bmiHeader*.*biCompression* = *BI\_RGB*;

BitmapInfo.*bmiHeader*.*biWidth* = layout.rectFull.*Width*();

BitmapInfo.*bmiHeader*.*biHeight* = layout.rectFull.*Height*();

std::*unique\_ptr*<BYTE[]> pDIBits(new BYTE[(BitmapInfo.*bmiHeader*.*biWidth* + 1) \* BitmapInfo.*bmiHeader*.*biHeight* \* sizeof(DWORD)]);

VERIFY(*GetDIBits*(hdc3D, hbm3D, 0, layout.rectFull.*Height*(), pDIBits.get(), &BitmapInfo, *DIB\_RGB\_COLORS*));

VERIFY(*StretchDIBits*(\*pDC, rectRender.*left*, rectRender.top, rectRender.Width(), rectRender.Height(),

         0, 0, BitmapInfo.*bmiHeader*.*biWidth*, BitmapInfo.*bmiHeader*.*biHeight*, pDIBits.get(), &BitmapInfo, *DIB\_RGB\_COLORS*, *SRCCOPY*) != *GDI\_ERROR*);

pSurface1->*ReleaseDC*(nullptr);

2/12/19

Commit

Got it to save a bitmap

Next include other image types.

2/16/19

1) Add png and jpeg exporting -ok Commit Implemented saving .jpg and .png formats

2) Check that everything is serialized like zoom and positions – only palette and image not pos and angles-ok

3) Change background color -ok

4) Change light position

4.5) consider setting diffuse, emissive and spectacular color

5) Allow saving specific size images

6) Make more example palettes

7) Make images and see if we’re missing anything

8) Impl. Mandelbulb 2 and others?

9) Palette selection dialog

Data members that need to be serialized

*ModelData* ModelDesc; -in doc

*std*::*shared\_ptr*<*DxVertexData*> VertexData; -in doc

*TraceParams* CalculationParams; -in doc

DxColor::PinPalette Palette; -in doc

DxColor::ColorContrast Contrast; -in doc

*RotationParams* rp

tuple<float,float,float> camera

tuple<float,float,float> target (position)

Image size? -in doc

I don’t see how these fields are being moved from the doc to the view

They are being called from OnInitialUpdate()→Initialize()→InializeColors()

First figure out how to set the view size on inialialUpdate() then store the size and try to recall from doc.

Add member for size to view and doc. Hook up to serialization and OnSize the add other missing members.

You don't size a view, but you can size the view's parent frame window:

GetParentFrame()->MoveWindow(x, y, width, height);

This seems to work.

2/19/19

Added Csiize to doc. The problem is the window doesn’t appear at the top but down a little. Need to figure out how to reposition.

2/23/19

It may be too tricky to resize the frame. See if we can just size the image. It’s even more difficult to separate the DC from renderer.

Commit

Added serialized window size member to doc. Also allowed resizing of main window.

Next serialize

*RotationParams* rp

tuple<float,float,float> camera

tuple<float,float,float> target (position)

Commit

Serialized RotationParams

2/26/19

Make something to serialize a tuple.

Commit

Stored camera and target in doc, synchronized with view and serialized

Also need to store nearView and farView

Commit

Stored perspective in doc, synchronized with view and serialized

Commit

Reset doc version in serialization to 1

3/2/19

We need to convert a DxColor::colorArgb to a *XMVECTORF32*

*Got it to save color in doc and transfer to renderer on start up but need to add a handler to allow user to set color. Reuse ColorSelectorDlg.h. Need to re-render after setting background color in renderer.*

Commit

Added background color to doc and render

3/5/19

Commit

Allowed changing the background color

Avoided setting modified from resize and initialization

Next – allow Change light position

see

<https://github.com/Microsoft/DirectXTK/wiki/IEffectLights>

<https://github.com/Microsoft/DirectXTK/wiki/BasicEffect>

3/9/19

void PrepareLightForRender()

{

         auto light = m\_pCore->GetLight();

         if(!light.fChanged)

                 return;

         if(!m\_textureVertexEffect)

                 return;

*XMFLOAT3* direction = {light.eDirectionX, light.eDirectionY, -light.eDirectionZ};

         if(light.fViewportRelative)

         {

                 SimpleMath::Matrix inverse = *XMMatrixInverse*(nullptr, m\_viewMatrix);

                 SimpleMath::Vector3 source(0, 0, 0);

                 const SimpleMath::Vector3 target(light.eDirectionX, light.eDirectionY, -light.eDirectionZ);

                 auto sourceTransformed = *XMVector3Transform*(source, inverse);

                 auto targetTransformed = *XMVector3Transform*(target, inverse);

                 direction.x = *XMVectorGetX*(targetTransformed) - *XMVectorGetX*(sourceTransformed);

                 direction.y = *XMVectorGetY*(targetTransformed) - *XMVectorGetY*(sourceTransformed);

                 direction.z = *XMVectorGetZ*(targetTransformed) - *XMVectorGetZ*(sourceTransformed);

         }

         m\_textureVertexEffect->SetLightDirection(0, *XMLoadFloat3*(&direction));

         m\_textureVertexEffect->SetAmbientLightColor(*XMLoadColor*(&m\_lightColors.AmbientColor));

         m\_textureVertexEffect->SetLightDiffuseColor(0, *XMLoadColor*(&m\_lightColors.DiffuseColor));

         m\_textureVertexEffect->SetLightSpecularColor(0, *XMLoadColor*(&m\_lightColors.SpectacularColor));

         m\_pCore->SetLightChanged(false);

}

struct LightColors

{

         LightColors()

         {

                 AmbientColor.*a* = ALPHA\_OPAQUE;

                 AmbientColor.*r* = 64;

                 AmbientColor.*g* = 64;

                 AmbientColor.*b* = 64;

                 DiffuseColor.*a* = ALPHA\_OPAQUE;

                 DiffuseColor.*r* = 192;

                 DiffuseColor.*g* = 192;

                 DiffuseColor.*b* = 192;

                 SpectacularColor.*a* = ALPHA\_OPAQUE;

                 SpectacularColor.*r* = 192;

                 SpectacularColor.*g* = 192;

                 SpectacularColor.*b* = 192;

         }

*XMCOLOR* AmbientColor;

*XMCOLOR* DiffuseColor;

*XMCOLOR* SpectacularColor;

};

constexpr float SpecularPower = 16.0;

void SetSpecularColors(BasicEffect& effect)

{

         effect.SetAmbientLightColor(*XMLoadColor*(&m\_lightColors.AmbientColor));

         effect.SetDiffuseColor(*XMLoadColor*(&m\_lightColors.DiffuseColor));

         effect.SetEmissiveColor(*Colors*::*Black*);

         effect.SetSpecularColor(*XMLoadColor*(&m\_lightColors.SpectacularColor));

         effect.SetSpecularPower(SpecularPower);

}

// For 3D model

void SetSpecularLightFields(BasicEffect& effect)

{

         auto light = m\_pCore->GetLight();

         effect.SetLightingEnabled(true);

         effect.SetLightEnabled(0, true);

         effect.SetLightEnabled(1, false);

         effect.SetLightEnabled(2, false);

         SetSpecularColors(effect);

*XMFLOAT3* direction = {light.eDirectionX, light.eDirectionY, -light.eDirectionZ};

         effect.SetLightDirection(0, *XMLoadFloat3*(&direction));

}

Does each light have a Diffuse and Specular color?

Add members for general ambient, diffuse, spectacular, emissive and specularPower

Add to renderer -ok

Add to UI

Make dlg – in progress CEffectColorDlg

Hook up to view

Test

Serialize

Add to doc

Add members

Connect to view

Serialize

Add 3 lights to renderer

Add to UI

Serialize

3/12/19

Almost done with EffectColorDg

Next

1) Add OnButtonUp to dlg -ok

2) Connect to renderer -ok

3) Implement in renderer -ok

4) Add member to doc and serialized

5) Connect doc to view

6) Make light class objects

7) Dialog to set lights...

3/16/19

Commit

Added EffectColors and EffectColorDlg

Set Effect Colors in Renderer

Added effect colors to doc and serialized

Next add button for effect colors

Then figure out why ambient and specular have no effect

Note only the Diffuse color has an effect???

Figure out why after serialization

3/19/19

Commit

Added Effect Colors toolbar button

Need to add emissive color to effect colors

This is in Dx11RenderingContext

m\_effect->*SetTextureEnabled*(true);

m\_effect->*SetPerPixelLighting*(true);

m\_effect->*SetLightingEnabled*(true);

m\_effect->*SetLightEnabled*(0, true);

m\_effect->*SetVertexColorEnabled*(false);

m\_effect->*SetAmbientLightColor*(*Colors*::*White*);

m\_effect->*SetDiffuseColor*(*Colors*::*White*);

m\_effect->*SetLightDiffuseColor*(0, *Colors*::*White*);

m\_effect->*SetLightDirection*(0, -*Vector3*::*UnitZ*); // NEED TO UPDATE SO WE CAN CHANGE LIGHTING

m\_effect->*DisableSpecular*();

m\_effect→SetAlpha(1.0f);

basicEffect->EnableDefaultLighting();

<https://github.com/Microsoft/DirectXTK/wiki/BasicEffect>

effect->SetLightingEnabled(true);

effect->SetLightEnabled( 0, true );

static const XMVECTORF32 light { 0.f, -1.f, 0.f, 0.f };

effect->SetLightDirection( 0, light );

<https://github.com/Microsoft/DirectXTK/wiki/IEffectLights>

The default lighting set by **EnableDefaultLighting** consist of three lights:

* Ambient: (0.05333332, 0.09882354, 0.1819608)
* Light 0: Direction (-0.5265408, -0.5735765, -0.6275069), Diffuse (1, 0.9607844, 0.8078432), Specular (1, 0.9607844, 0.8078432)
* Light 1: Direction (0.7198464, 0.3420201, 0.6040227), Diffuse (0.9647059, 0.7607844, 0.4078432), Specular (0, 0, 0)
* Light 2: Direction (0.4545195, -0.7660444, 0.4545195), Diffuse (0.3231373, 0.3607844, 0.3937255), Specular (0.3231373, 0.3607844, 0.3937255).

We don’t want vertex coloring

Next try

*SetLightingEnabled*() -ok

basicEffect→EnableDefaultLighting(); -ok

m\_effect->*SetPerPixelLighting*(true); - make an option

Then add emmisive

Then add light structure

Commit

Added SetLightEnabled and EnableDefaultLighting()

Added emissive color but need to test more. Also fix rect positions in dlg. Not committed

3/23/19

Fixed color rects and tested serialization.

Commit

Added Emissive color to effect.

Lights

3 lights each has a diffuse and spectacular color, a direction and an enable

Set alpha float

SetPerPixelLighting

SetVertexColorEnabled - no

No pixel or vertex color



pixel coloring on, vertex off



You can’t turn on vertex coloring

Start on lights

1) Make Light and Lights structures -ok

2) Make DxLight and DxLights -ok

3) Make serialization methods -ok

4) Make Light Dlg

a) Sell with setters -ok

b) resource ctrls -ok

c) Draw rects and buttons-ok

d) pop up color dialogs-ok

e) DDX -ok

f) menu item and handlers -ok

g) Kill focus for edit boxes to enable validation -ok

h) handler for default and enable should disable edit boxes and colors ok

5) Test dlg -ok

6) Add member to view -ok

7) Add member to renderer -ok

8) Connect view to renderer -ok

9) Add to doc

10) Serialize and test

11) Add OnUpdate for OnEditLights() and OnEditEffectColors()

12) Put common dlg methods in util file

3/30/19

LightDlg should be ready.

4/2/19

Commit

Implemented lightDlg

Added lights to view and renderer

Figure out why ambient doesn’t work

Figure out why lights are not aligned with x,y,z. Might be related to normals

Remember when done with lights figure out why coloring/lights change on rotation, probably normals.

4/6/19

Next

1) Serialize Lights -ok

2) Add OnUpdate for OnEditLights() and OnEditEffectColors() -ok

3) Put common dlg methods in util file -ok

4) Button for lights -ok

5) Study normals and transformation of normals – see book 7.2.1

6) Fix normals -ok

7) Fix normal transformation – changed lights instead

7.5) Fix view x coordinate. If you change it in dialog it doesn’t update/change. Target x. -ok

8) Figure out how lights should work -ok

9) Fix lights if needed -ok

10) Allow saving specific size images -ok

11) Make more example palettes

12) Make images and see if we’re missing anything

13) Impl. Mandelbulb 2 and others?

14) Palette selection dialog

Commit

Serialized lights

Added OnUpdate for Editing effect colors and lights.

Added Edit Lights button

Refactored common dialog methods

4/9/19

Commit

Fixed vertex normals

light doesn’t look right when 180 < y < 270 and 180 < x < 270

Check if there is a way to transform normals with effect – can’t find anything

Check normal calculation

Not sure if this is best but set the default light directions.

Update dialog so tab order is correct.

Commit

Set light to dxtk default directions

Fix tab order on LightsDlg

4/16/19

Commit

Fixed view dialog target x

Light 1 and 3 resets y from z

Light 2 resets y from x

alpha doesn’t do anything obvious on light dialog

ambient material works with diffuse (not spectacular nor emissive) and not alone

Next

Fixed light dialog

Commit Fixed light dialog direction

Test

Remove alpha from effect colors on material dialog

Commit Removed alphas from effect dialog

Next

1) Light directions should be reversed -ok

2) Model image should switch z +,- both image and handler -ok

3) The display position on dialog and allow editing -ok

4) Disable alpha in material and lights dlg color chooser -ok

Commit Reversed light directions

4/20/19

Commit

Change z direction in PositionAngleDlg

*Change file used for PositionAngleDlg to Coords2.bmp this should be in the exe folder.*

Changed direction of z axis in PositionAngleDlg and resource file Coord.bmp to coords2.bmp

IDC\_POS\_X\_EDIT

IDC\_X\_ANGLE\_EDIT

Commit

Added position and angle Edit boxes to PositionAngleDlg

4/23/19

Commit

Disabled alpha for material and light colors

Next - Allow saving specific size images

First add dialog to set size in View::SaveImage()

See ImageSizeDlg

4/27/19

dlg is hooked up but image isn’t stretched.

Commit

Fixed stretching when saving resized images

4/30/19

Next

Make palettes and test images -ok

Add tics to bottom of palette image – look at Chaos in Fractals, maybe score tick to 1000? See ColorView.cpp DrawOnScreenBM where m\_bShowMarks. PaletteViewDlg

I think we will have to make a separate rect below the main rect for the ticks because the colors and pins are in the main double buffer

Add ‘Add Pin’ handler to palette dialog

Create palette selection dialog

5/4/19

Commit - Added ticks to palette dlg

Next

Add Pin handler

Spread evenly handler

Palette Selection dlg

5/6/19

Check AddPinBetween

5/11/19

Commit - Added Add Pin handler

Something is wrong when multiple pins are added or pins deleted after adding. Maybe index is not updated

Commit – Fixed index when adding first pin

Commit- fixed new pin index

When there is only a pin in the middle and you add one between 0 and the middle pin it shows up closer to the middle pin

Commit- Fix adding pin on the left

It would be nice to be able to set the color when adding a pin rather than setting it automatically

It would be nice to be able to edit a single pin.

Next implement spreading pins then adding color index, editing a specific pin

5/14/19

work on spreading pins

Commit

Implemented evenly spreading pins

Next

1) Allow deleting pins by lifting up

2) Set Pin color and index when adding a pin

3) Allow editing a single pin

5/18/19

Commit

Allow deleting pin by raising pin

For 2 & 3 we need a new dialog that shows a pin index and 2 colors and allows those values to be changed

Don’t forget when done to update the ProgramFiles\FractalX

5/21/19

Allow editing single pin

SinglePinEditDlg

1) Have it return pins -ok

2) Clean resources -ok

3) When color changes have it update the pin with the given index -ok

4) Commit -Added SinglePinEditDlg -ok

5) Allow selecting colors 1 or 2 -ok

6) Commit -ok

7) Display both colors even though 1 is selected. Maybe label color 1 and 2 -ok

8) commit – Display both colors on SinglePinEditDlg -ok

9) Display index in dialog -ok

10) When it is edited have it determine if it’s allowed -ok

11) When index is edited have it update the pin order -ok

12) Commit – Allow setting the pin position index -ok

5/25/19

Did 5-12 above

Next use this dlg when adding a pin

5/18/19

Commit - Implemented editing new pin

Update release version on computer and backup on external drives

Next Palette selection dialog

6/1/19

Add palette list dlg

use CfolderPickerDialog

<https://stackoverflow.com/questions/1304784/cfiledialog-browse-folders>

IDD\_PALELETTE\_SELECTION\_DLG

CpaletteSelectionDlg

1) Create resource -ok

2) Create class -ok

3) Add handlers to open dlg -ok

4) Implement browse button to open folder -ok

5) Make structure to store palette file path, name, palette object (pins etc?)-ok

6) Load palettes-ok

7) Add rect to palette struct to store location and drawing area

8) Draw palettes

9) Allow selection of palettes, maybe draw black rect around selected item.

10) Implement OK

For 5, 6, etc see PaletteViewDlg OnImport();

6/4/19

Commit

Added loading palettes in CpaletteSelectionDlg

For making color images see Chaos project in Fractals. IDD\_COLOR\_TREE\_DLG and CcolorTreeDlg

It has a TreeView control.

Need Hbitmap for each palette

see

*HBITMAP* CColorTreeDlg::GetBmp(const *CString* &filename)

This uses Cdib but we can try doing it manually

try something like this:<https://www.codeproject.com/questions/348546/creating-hbitmap-from-gdi>

1) Add tree view control IDC\_TREE1 -ok

2) Add tree methods and DDX -ok but not complete

3) Add method to create HBITMAPs – in progress -ok

6/8/19

Got palettes drawn

Next

1) Have it allow selecting a palette – make sure the selected one is highlighted in some way

2) Commit

3) Show path in dialog -commit

4) Store default path in registry & commit

5) Clean up – such as if a bmp is bad & commit

After this make sure everything is backed up

6/11/19

Commit Made it select palette and return from selection dialog.

Next

1) Update palette name in PaletteViewDlg

a) When exporting have it suggest a file name based on the palette name. -does this

b) The replace the serialized name with the file name -ok

c) When reading replace the serialize name with the file name-ok

2) Allow changing path in selection dlg edit box - ok

3) Store palette path in registry

The displayed name is the file name. When serialized it stores what was in the dialog

6/15/19

Commit

Force saved palette name to be the same as the file and override imported file name to filename.

Allow changing path in selection dlg edit box

Store and retrieve path in palette dlg

Next

Test

Make palettes

6/18/19

Make more palettes

remove blue 2

Possible improvements

0) Palette selection directly from view -ok

1) Allow updates from palette selection dlg -ok commit

2) Allow negative angles in rotation dialog -ok

3) Add a rotation rate

4) When you switch to fixed rotation save the position

5) Add copy pin handler

6) When you change from rotating to fix keep the last angles when it was stopped

6/22/19

Updated MFC packages

Working on 0:

1) Add menu item ID\_EDIT\_SELECTPALETTE -ok

2) Add Handler -ok

3) Implement handlers -ok

4) Test -ok

5) Add toolbar button -ok

6) Test -ok

7) Make a color menu

8) commit - Added palette selection dialog directly to view with toolbar button -also update to VS 2019

Commit – Allow negative angles in rotation dlg

Next

Build release version and check link -ok

make more palettes -ok

install VA -ok

install tab studio -ok

2) Try to remove warnings in release build -ok most are gone. Fixed the rest

3) Add a rotation rate -ok

4) When you switch to fixed rotation save the position

5) Add copy pin handler

6) When you change from rotating to fix keep the last angles when it was stopped

6/25/19

Commit Replaced some Cstring formats with wstringstream to avoid warnings.

Commit – added rotation rate (not serialized)

6/29/19

Commit Fixed rotation angles in renderer.

Add stop button – don’t keep rotation angles if setting fixed from the rotation dlg

7/3/19

Commit

Added Stop rotation handler.

7/6/19

Add copy and paste pin to palette view dlg

ID\_PALETTE\_COPYPIN ID\_PALETTE\_PASTEPIN

Commit

Added copy and paste of pins

Next time try to get the context menu working see -

<https://social.msdn.microsoft.com/Forums/vstudio/en-US/a664802b-3d00-4ca5-9891-e416478e4244/how-can-i-disable-a-context-menu-item-when-i-use-mfc-feature-pack>

use this to disable menu when the menu is shown. Need to change the handlers.

7/9/19

ID\_PALETTE\_EDIT\_PINS

ID\_PALETTE\_ADD\_PIN

ID\_PALETTE\_SPREAD\_PINS

ID\_PALETTE\_EDIT\_PIN

ID\_PALETTE\_COPYPIN

ID\_PALETTE\_PASTEPIN

Commit

Disabled context menu items in PaletteViewDlg when appropriate

Disabled some buttons when PinEdit dialog is visable

Next try to get rid of some warnings in release.

7/13/19

Only saw one minor warning and fixed and committed

Now update working exe in [C:\Program](../../../Program) Files

7/18/19

Add an enum for fractal type and work into serialization

Added serialization method.

Serialized FractalType

7/26/19

First add changes to formula such as the power to use instead of sqrt for r.

See [https://www.skytopia.com/project/fractal/2mandelbulb.html#julia](https://www.skytopia.com/project/fractal/2mandelbulb.html" \l "julia)

Add swapping the trig functions there too

Then try the juliabulb. See JuliaBulbEq.odt and <http://www.fractalforums.com/mandelbulb-implementation/better-de-estimate-using-orbit-traps/>

8/10/19

FractallType is in FractalParams which is embedded in TraceParams

TraceParam is passed to BasicRayTracer

Commit

Refactored BasicRayTracer to take a cartesian conversion function.

Add CartesianConverterAlt1() where x is sin(theta)\*tan(phi) instead of sin(theta)\*cos(phI) web site said it was for Y but I think it was a typo and it was really meant for X.

IDC\_MODEL\_TYPE\_COMBO

Commit

Implemented *CartesianConvertAltX1.*

Next implement other alternative conversions

8/13/19

normal = sin(theta)cos(phi)

Alt X1 = sin(theta)tan(phi)

Alt X2 = cos(theta) cos(phi)

Alt Y1 = sin(phi) tan(theta)

Alt Z1 = sin(theta)cos(theta)

These all look similar and spiky. Somewhat interesting if blowup is < 1.5

Commit

Added alt Cartesian conversion equations

Next

Change parameters

Make FractalParams represent IrayTracer to include Standard Bulb and Bulb Plus

3 enums

FractalType= IrayTracer

CartesianConversionType – conversion method

BulbNormalizeType – this will be either standard (rad 2) or AltRoots. When it’s Altroots the UI will allow the entry of a double between 0.001 and 1. This will only be selectable when FractalType is Bulb Plus

BulbPlus will support CartesianConversionType and NormalizeType

Step 1

Break FractalType out of TraceParams file. Put other enums in other files too. -ok

Add NormalizationRoot.

Add to FractalParams -ok

Work out serialization -ok

Add to UI

combos -ok

add members -ok

implement combos -ok

Add normalizationRoot -ok

Hook up new fields to page -ok

Hook up new types to rayTracer creation;

Disable cartesian and normal when appropriate

Implement new IrayTracer

Hook up

8/20/19

Commit

Implemented different cartesian coordinate conversions. Added CartesianConversionType and BulbNormalizationType.

Normalization method not working

8/24/19

Figure out why normalization is not working!

It might be because the calculations are done in float. Make a ray tracer that does the calculations in doubles. Need a Vector3 in double.

Examine all occurrences of Vector3 and XMFLOAT3

Make a class called Vector3Double

8/27/19

Made struct Vector3Double.

Make DoubleRayTracer – ok

Revert BasicRayTracer

update factory method -ok

8/31/19

DoubleRayTracer produces the same results but it has warnings in the code.

See if we can make the warnings go away. C26444 -ok

Next-

Clean up BasicRayTracer – include RayTracerCommon

Commit

Update backup repository on external drive

See if we can get better results.

9/3/19

Cleaned up

Commit

Implement DoubleRayTracer

9/6/19

Studying altx2.

They come out better with Step Divisor of 2

Iterations of 2 makes less fine. Higher iterations add more detail

Reducing Max Ray steps to 50 makes the colors spread better to the left

AltX1 and AltY1 yuck

AltZ1 is OK

Look at this next

<https://softologyblog.wordpress.com/2011/07/21/new-mandelbulb-variations/>

Next Juliabulb

Then Quilez’s julia 3D

we want a function that can be k\*trig(theta)\*trig(phi) for each x, y, z

where k is 1 by default, trig is sin(angle), cos(angle), tan(angle) or 1

put code in DxSupport

Code

1) enum TrigOption -ok

2) struct CartesianConverterGroup-ok

3) CartesianConverterFactory -ok

4) Add to params -ok

5) Serialization -ok

6) Add to RayTracers – just double -ok

7) UI

Check for memory leaks before proceeding

C:\Program Files (x86)\Visual Leak Detector\bin

Path %SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\;%SYSTEMROOT%\System32\OpenSSH\;C:\Program Files\dotnet\;C:\Program Files\Microsoft SQL Server\130\Tools\Binn\;C:\Program Files\TortoiseSVN\bin;C:\Program Files\Git\cmd;C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\170\Tools\Binn\;C:\Program Files (x86)\Visual Leak Detector\bin\Win32;C:\Program Files (x86)\Visual Leak Detector\bin\Win64 <SYSTEM>

9/10/19

Fixed memory leak. Memory leak was because ray tracers did not have destructors

Do not commit Visual leak detector changes now

9/14/19

Commit

Implemented CartesianConversion factory etc. Need to add to UI.

Create a Cartesian Conversion Dialog.

Pop up with button Fractal Params page.

Enable with Double Bulb model. Disable drop list with double bulb model

Check what else has to be disabled.

IDD\_CARTESIAN\_CONVERSION\_DLG

IDC\_MULTIPLIER\_X

IDC\_THETA\_X\_COMBO

IDC\_PHI\_X\_COMBO

IDC\_CUSTOM\_CONVERSION\_BUT

9/17/19

Need to pass the DXF::*CartesianConverterGroup*

to the FractalParamPage -ok

Set in CartesianConvertionDlg -ok

Fix errors -ok

test -ok

Implement edit boxes-ok

Implement combos -ok

DDX combos -ok

test -ok

Commit – Implemented Cartesian Conversion Options

Enable with Double Bulb model. Disable drop list with double bulb model

Check what else has to be disabled.

when done see if we can use more shred\_ptrs for structs

9/21/19

Bulb Normalization drop list repopulates when it’s set active – fixed

Cartesian conversion also repopulates - fixed

Disable Custom button when not double bulb-fixed

Disable conversion list when double bulb -fixed

Add a default button to custom cartesian conversion Dlg

Check that property page and sheet classes are correctly -They’re fine

pass structure as shared\_ptr

TraceParams

Commit

Fixed combo box re-initialization

Disable/enabled Cartesian conversion controls

Added default button to Custom Cartesian conversion dlg

Used shared\_ptr for trace params

9/24/19

Check for leaks -ok

Put release version in folder

Commit

Fixed parameter ranges on TraceParamPage

Next

Fix coloring

Specular only lights up bottom rt quadrant

See images in Trials folder

It doesn’t seem to be lighting because if you rotate the model, the lights stay relative to the model.

9/28/19

The problem is the normalization. The old method looks a little better than the one used in CT.

Specular works in other quadrants but not as intense. Need to reduce specular factor.

No clue:

Try testing CT

Google

Write a shader

If step divisor is greater than 2 then specular shows on 7/8

If step divisor is 1 the it shows on 1/8th? - so look at step divisor

10/1/19

It seems like the 2nd 1/8 is coming out backwards. The triangles are clockwise

When you make 1-4 sections it looks correct now but still a problem with 8?

Vertices and triangles appear to be correct so not sure now?

Found problem.

1) Need to recalculate old triangle files

2) Commit

10/5/19

Commits

Fixed bug in source triangles

Added alternative normal method but not using it, CalculateNormals2()

Made new triangle files

10/8/19

Added Triangle folder to FractalX folder and commit

10/12/19

Working on Julia sets

The current C value is really just the first derivative and should be fixed at 1

Need to add a x,y,z value for C to use for Julian sets

Commit

Rename ConstantC to Derivative

Added constantC tuple to fractal params and page

Implemented constantC for Julia sets

10/15/19

Bumped version to 1.2.0.0

Features to consider

1) Contrast

2) Calculating sections so we can zoom in at high resolutions

3) Ingles’ fractals

10/26/19

Start on zoom in

The hard part will be trying to determine the coordinates to zoom in

Currently:

ctrl key down and middle mouse zoom in z directions

ctrl key down and right mouse move in x and y

Use ctrl left mouse to select an area

3 Stages to implement

1) Mouse handlers and display of coordinates in UI -ok

2) Determining the vertex coordinates

3) Methods that calculate triangles and models for zoomed in portion

Mouse handlers: OnLButtonDown, Up and Mouse movement

Capture on ctrl key down

Display on pixels on status bar?

10/29/19

Commit

Got coordinates displayed on the status bar

11/3/19

Start on mapping 2D to3D

Commit

Added mapping to 3D and display of vertices

Next show rectangles – look at Fractals2 for example

Then recalculate specified vertex range

11/5/19

Implemented drawing rectangle when left mouse button and ctrl key are down.

11/9/19

Start calculating new triangles.

1) Identify code used to start process now - OnEditSetModel()

2) Find triangle generation code and use as example (GenerateCrudeTriangles())

3) Determine new parameters needed- enum for vertex model, Two tuples of x,y,z

Steps

0) Make new vertex type – probably float – see what value is usually used or make a template -ok

1) Add members to ModelData: enum for vertex model, Two tuples of x,y,z -ok

2) Implement serialize -ok

3) Add members to doc? -NA

4) Add menu item to set selected rectangle. This will normalize coordinates before setting in the doc then set the vertices in ModelData -ok

5) Add controls and data to CvertexPage -ok

6) In TriangleLoader GetTriangles() have it skip LoadTriangles if custom vertex mode

7) Make a new method GenerateTrianglesFromCrudeVertices() modeled on GenerateCrudeTriangles(). Call that when custom vertices.

8) Fix unit test project so it builds

3D tuples

Member name struct type library

ConstantC FractalParams double DxSupport

Direction Light float DxColors

m\_camera FractalDoc float FractalX

m\_target FractalDoc float FractalX

CustomVertex ModelData float DxSupport

Commit

Added vertex struct and used for constC

Used vertex for target and camera

Note there is a problem building DxTests

11/12/19

Implemented changes to ModelData and Serialize (not committed)

Add menu item to save vertices

Disable if there are no vertices

Normalize

Save to Doc

11/16/19

Commits

Implemented changes to ModelData and Serialize

Changed view vertices to vertex instead of tuple and added a handler for OnSaveSelectedVertices() to view.

Moved view vertex members to pImp

We need to save four vertices not 2 to make sure we have valid corners

Commit

Save 4 vertices in ModelData

Store vertices in ModelData in view.

Next Add to UI

1) Vertex Source

2) 4 Vertices

IDC\_SPHERICAL\_RAD

IDC\_CUSTOM\_VERTEX\_RAD

IDC\_TL\_X\_EDIT

IDC\_TL\_Y\_EDIT

IDC\_TL\_Z\_EDIT

IDC\_TR\_X\_EDIT

IDC\_TR\_Y\_EDIT

IDC\_TR\_Z\_EDIT

IDC\_BL\_X\_EDIT

IDC\_BL\_Y\_EDIT

IDC\_BL\_Z\_EDIT

IDC\_BR\_X\_EDIT

IDC\_BR\_Y\_EDIT

IDC\_BR\_Z\_EDIT

Hooked up radio buttons but not new edit controls

11/19/19

Commit

Displayed vertex source and vertices on VertexPage

Next need to implement custom vertex calculation

Rough custom vertex calculations

1) Needs to zoom x,y

2) Maybe needs to adjust z dimension

3) May not always get coords right

4) Sometimes it says point is outside the shape but it shouldn’t be

Find out why 4 happens

Commit

Fixed vertex coordinates

Change window size from 800, 600 to 800, 800

Added WorldScale to renderer and CworldScaleDlg (not serialized)

Next

1) Serialize WorldScale, figure out where to store it -ok

2) Do we need more mutexes in the doc? No but do anyway to be safe for now-ok

3) Automatically figure new scale when zooming

4) Test when model is rotated

Commit

Added WorldScale to doc and serialized

Added Mutex to all fields

11/26/19

First test if scaling works when model is rotated

Then decide on shifting position and scale automatically

Zooming works even when model is rotated but it will be tricky to zoom and position

Can figure out zooming size with relative window size Original is 800x800 and if selected rect is 400x400 the change from 0.1 to 0.2? - this applies when camera z is 0.3

For positioning about 0.25 change in x move whole model off screen

Camera z zoom x,y

default 0.3 0.1,0.1

1 0.33, 0.33

2 0.67, 0.67

11/30/19

Add auto zoom to vertex page -ok

Connect to view -ok

Have it calculate zoom and set -ok

Reset m\_topleft and m\_bottomRight after rendering, and anyplace else.

Commit – Implemented auto zooming scale when using custom vertexes (spherical)

12/3/19

When vertices change set auto zoom to false and disabled -ok but need to update enable when reset to same (similar values). Right now it will remain disabled if values are set back to equal

Change so Custom vertexes radio button and auto zoom are disabled when vertices are no unique. Get rid of enable auto zoom.

Commit Update VertexPage to disable custom vertexes when vertexes are not unique

Next Calculate position

12/7/19

First estimate for x, use center of image where c1 = first image center, c2 = new center, and w = width

pos += 0.1\*(c1-c2)/(w/2)

OK

It works pretty good on the first zoom but on the second zoom it doesn’t translate far enough. I think because it needs to know the center position relative to the original screen or it needs to use the previous zoom in the calculation instead of 0.1. The later doesn’t make sense because that would make the translation even smaller.

To test zoom in on a square to the left of about ¼ the size. This looks OK. Then zoom again on a center square of less than ½ the size. The new image is way off on the left because it wasn’t shifted enough.

Consider automatically saving the zoom coordinates

12/10/19

Auto zoom works, it’s just translation

Still can’t get it. The first round is close but the second varies a lot.

12/13/19

Got x ok

1) Do y

2) Clean code

Commit – Implemented auto zoom and auto translate

3) Make sure the variable is cleared – not sure where or when, maybe when translation is set and when calculation is restarted without the auto zoom

4) Automatically set coord so handler isn’t needed

5) Add auto translate

12/17/19

Commit

Reset translation, scale and target when changed back to spherical vertices.

Added Auto Translate as separate option

Prompt user to set new coordinates

Fixed flicker when resizing

Next

1) Find out why drawing rect causes flickering after first round -ok

2) Reset points after calculation -ok

3) Contrast

4) Ingles’ fractals

12/20/19

Change version from 1.2.0.0 to 1.2.1.0

Commit

Bumped version to 1.2.1.0

Bumped version in about box to 1.2.1.0

Do release build and store in Program Files\

Next

Noticed zoom does not work when model is rotated!!! Fix

Next research and plan for Ingles’ fractals

- <https://www.iquilezles.org/www/articles/juliasets3d/juliasets3d.htm>

Consider constrast

Fix test project

12/23/19

Selection of coordinates works and so does zoom but the translation is off.

When rotated z may need to be scaled also.

Got an approximate fix.

Commit – Reset Z scale and camera z when zooming so translation comes out closer.

12/28/19

Commit: Bump version to 1.2.1.3

Update local exe

Work on Ingles’ fractals

For new 3D fractals

1) Make new IRayTracer – see DoubleRayTracer.cpp InglesRayTracer

2) Main difference will be in CalculateNextCycle – this will use cartesian coords rather than spherical -ok

3) Make new enum value – *InglesFractal-ok*

4) Update param page for new type -ok

5) Test

Remember old version will not work with new enum value

When calculating change Baliout to 10

Max ray steps to 10

Step divisor to 2

They are coming out interesting but I’m not sure they are correct. Need to check math. Maybe some coordinates are not being processed correctly because there is usually a ring in the middle.

Fix angles in position dlg so they don’t go past 360.

Need to commit change to SphereApproximator.cpp this is a possible bug fix in debug.

1/4/20

Commits

Fixed possible bug in SphereApproximator.cpp

Limit rotation to 0-360 degrees in PositionAnagleDlg

Add another type Ingles2

Commit: Added Ingles and Ingles2 ray tracers (currently they are the same)

Commit: Updated Ingles2 ray tracer

Next look at the return value from estimateDistance()

Added a different distance calculation from https://www.shadertoy.com/view/MsfGRr

It might be better, maybe make it an option

1/11/20

Experiment with 2nd version of Ingles

What fields work best

**Bailout** 2-100 works **10** about best Slight more color (less red when >= 10)

**Max Ray Steps**: 10000-10, best ~**100**, 10 black, 10000 white

**Min Ray Distance**: 0.0001-0.1, best around **0.001**, 0.1 smoother & less detail

**Step Divisor**: 1-100, 1 whiter, 100 darker and smooth, **10** best

**Iterations**: 2-10000, **256** OK, 2 less detail, 10,000 about the same as 256

**Fractional Steps**: when true it comes out spikes, unless Max ray steps <1000 then black ball

How to get more color? Max Ray Steps of 300-350

Best for Igles2 without fractional Steps

**Bail Out**: 20

**Max Ray Steps**: 350

**Min Ray Distance**: 0.001

**Step Divisor**: 15

**Iterations:** 256

Consider more changes? - no it’s about right

Commit

Ingles2 Algorithm implemented close to expected

Next:

How to improve color with fractional steps – stretch but then it comes out like non-fractional? But it doesn’t look like fractional shape?

Try Max Ray Steps **300**, Min Ray Distance of **0.1** to 0.2, step divisor **15**

Test stretching -it works about right except for fractional

\*See if we can implement stretching for fractional

How should we implement other formulas – make separate tracers (algorithms) or options under Ingles2?

Look at disabling some fields

1/14/20

Test stretching with fractional and normal bulb-fixed

RayMarch

return 1.0 - ((double)steps) / m\_traceParams.*Bulb*.*MaxRaySteps*;

RayMarchFractional

double delta = steps;

if (steps < m\_traceParams.*Bulb*.*MaxRaySteps* - 1 && distance < m\_traceParams.*Bulb*.*MinRayDistance*)

delta = steps + distance / m\_traceParams.*Bulb*.*MinRayDistance*;

delta = *std*::*min*(delta, static\_cast<double>(m\_traceParams.*Bulb*.*MaxRaySteps*));

return 1.0 - delta / m\_traceParams.*Bulb*.*MaxRaySteps*;

RayMarchDistance

return lastDistance;

Split RayTrace to 2 functions, one will run when on stretch (a new parameters)  
 Without the stretch field it will run old RayTrace. With Stretch it will estimate the distance from the steps first. Then calculate all and stretch.

1) Add parameters (TraceParams.cpp StretchDistanceParams) – Stretch -ok

2) Add to UI (Model Page) IDD\_POSITION\_RANGE\_PAGE -ok

3) Serialize - OK

4) Implement in basic ray tracer and test

RayTrace() →

RayTraceNormal() - same as before

RayTraceStepStretch() - stretch the step output

1/18/20

Commit

Implemented Step Stretch method for basic bulb

Implemented Step Stretch method for double bulb

1/21/20

Commit

Implemented Step Stretch method for Ingles

Implemented Step Stretch method for Ingles2

Next

First compare differences for Ingles and Ingles2

Fractional doesn’t work good for Ingles & Ingles2 try to fix

Try to move some double functions into common place. It won’t work for basic ray tracer.

Disable some parameters that aren’t necessary.

1/25/20

Ingle and Ingles2 work in general and so does stretch the only problem may be Ingles in Fractional mode produces a lot of spikes that go to infinity. See if it can be fixed.

Commit

Put some common Double functions in RayTracerDoubleCommon

The only difference between the 3 double implementations is EstimateDistance()

Tried to Moved most implementation to *InternalDoubleRayTracer*

but Ingles and Ingles2 are coming out funny

There seems to be a difference is the implementation – Don’t commit until it’s worked out. The change is interesting thought.

*1/28/20*

Found issues. MakeStartingPoint was using constantC for all double ray tracers but it originally only used it for the original and not the Ingles.

Removed constant C even for plain double ray tracer and it seems to work better even for double ray tracer.

Fractional only works for low constant C values and Double Ray tracer. May need to adjust other parameters.

For Ingles and Fractional use a low number of iterations ~5

For all use low iterations and possibly smaller C values

Commit

Moved most implementation to *InternalDoubleRayTracer*

Next

Adjust min max stretch range. Limit the calculated values and allow a wider range in the UI

Disable some parameters that aren’t necessary.

*2/1/20*

Commit

Adjust min max stretch range. Set max stretch to1e7 and limit in calculations to this value.

Power field is not used for Ingles or Ingles2 IDC\_POWER\_EDIT

Commit: Disable power when Ingles or Ingles2 is selected

Not sure if bulb normalization is working with alt normalization

m\_normalizationType *AltRoots*

*m\_normalizationRoot NormalizationRoot*

*IDC\_ROOT\_EDIT*

*Normalization root should be disabled if normalizationType is Std*

*Derivative should be disabled for Igles and Igles2*

*Maybe default derivative should be zero*

Commit: Disabled Normalization root if not AltRoots

Commit: Disable Derivative for Ingles and Ingles2

The derivative should probably be zero but leave it so we are backward compatible at least for now.

NormalDelta is not used any place? Why?

All Params on the IDD\_TRACE\_PARAMS\_PAGE are used and enabled correctly except NormalDelta

Next look at Origin and Range page and then figure what to do about normal delta- probably a DX property

2/4/20

All other params work for all settings. Just figure out how to implement normDelta

Normal Delta – we don’t need it, this is from the Bulb project

Commit: Removed normal delta and update version to 1.2.1.4

Bump version to 1.2.1.4

Update release version.

Next fix test project

Then implement other functions for ingles1 & 2

2/22/20

C1189

Building MFC application with /MD[d] (CRT dll version) requires MFC shared dll version. Please #define \_AFXDLL or do not use /MD[d]

C:\Program Files (x86)\Microsoft Visual Studio\2019\Professional\VC\Tools\MSVC\14.24.28314\atlmfc\include\afx.h

Problem was Carchive in Vertex.h – moved to another file SerializeVertex.h and made test project use MFC in shared DLL.

Commit – Fixed building of DxTest project

3/24/20

Make an ingles 3 tracer and update the derivative.

*Commit*

Implemented Ingles3 tracer

looks similar, maybe better – derivative is different -

Next – the distance estimator is for a quarterion – try to update for this derivative

3/28/20

Distance should be

(|zn|\*log|zn|)/dZn

It looks similar to 2 but less color. Need to reduce the max ray steps or something else

Commit

Updated distance formula on Ingles3

4/4/20

Thinking about how to implement other formulas in ingles3 such as z^3

1) Make CalculateNextCycle() a function ptr. So we can assign to different functions on intialization. - ok

2) Add a new enum for the formula just for ingles3

3) Place on Fractal parameter page just below model type and hide when not ingles3

4/18/20

1) Add enum Ingle3EquationType -ok

2) Add to fractal parameters -ok

3) Serialized -ok

4) Implement CalculateNextCycle methods for cubed and sine X -ok

5) Add to dialog

a) Add field -ok

b) Add combo -ok

c) implement combo-ok

d) connect to params -ok

e) enable/disable

Commit

Implemented some new Ingles3 equations but they aren’t hooked up yet

I think the derivative equations might be wrong, all the components for x should be multiplied by dx, not dx, dy, dz, and similarly for y and z partials. Test later

4/21/20

IDC\_INGLES\_EQUATION\_TYPE\_COMBO

It works. Not spectacular but different. Think about other formulas.

Commit

Hooked up options for various Ingles3 equations

Bump version to 1.2.2.0

Updated release version in Program files

4/25/20

Implement Ingles Alt 1 and Alt 2.

Alt 1 has i\*j = 1

Alt 2 has j\*j = i and i\*j= 0

1) Implement Next Cycle -ok

2) Ingles3EquationType enum -ok

3) Hook up tracer3 -ok

4) UI

Look good – different than original, but not necessarily more detail but interesting

Commit

Implemented Ingles 3 Alt1 and 2

5/2/20

Implement Ingle3 Alt3 – 5

All OK nothing spectacular – will have to test with different params one at a time.

Next time save angles and zoom when recalculating a fractal or allow user to select whether to or not

5/9/20

The angles and positions are saved when serialized

Only the angles are saved when recalculating

Target and World scale not stored when recalculating

Added option to set default target and world scale.

Bump version to 1.2.3.0

Update Program Files

5/12/20

Allow to add a second VB so we can add shapes and planes.

Start with SetModel()

m\_vertices and m\_indices

CreateBuffers()

m\_vertexBuffer

m\_indexBuffer

1) have possible indices and triangles for plane at y = -2

2) Need to figure out how to make vertex buffer and indices from triangles

3) Connect

4) Figure out how to draw image on it

5/16/20

Need to fill in a *DxVertexData*

struct *DxVertexData*

{

*std*::*vector*<DirectX::*VertexPositionNormalTexture*> *Vertices*;

*std*::*vector*<unsigned int> *Indices*;

*StretchDistanceParams* *StretchParams*;

};

Create TriangleData then DxVertexData

Got the plane drawn in position. Problem is it rotates with the primary model. Do we want separate world for it?

Probably want it on an angle between the Y and Z planes

1) See if we can fix it a specific location

2) Figure out how to draw an image onto it. Do we need many vertices and another texture or can we paste an image onto the surface?

5/19/20

Don’t worry about positioning yet

1) Add code to open an image file -ok

1.5) Need to check that file exists before closing (Kill Active) if name is not empty-ok

2) Pass to renderer-ok

3) Only create second model if there is a filename?

4) Create texture using WIC

5) Hook this texture up to the second model

5/22/20

Did up to 2

Commit

Added texture filename to params and UI and passed texture filename into renderer

Have m\_texture2 and m\_textureView2.

Temp function void SetTexture2(*std*::*vector*<*uint32\_t*> colors)

Try making a function to create a texture using WIC in DxFactoryMethods

working on DXF::WicTextureFactory::*CreateWicTexture2D()*

Should be ready. Have to change some params that are passed in.

5/23/20

Fixed wic texture function and it seems to work but when the model is drawn it still uses the other texture. Also not sure about setting the texture coordinate. The primary texture is 1D but the secondary can be 2D?

5/30/20

1) Try to get it to draw image – it works

2) Increase vertices for secondary texture – not needed, just need higher resolution image

3) Fix positioning – see below

Bump version from to 2.0.0.0 – do not update Program files

Commit – Added support for secondary model and bumped version to 2.0.0.0

Add option to show secondary model

Commit – Added option to show background model and bumped version to 2.0.1.0

Add coordinates for plane to UI

Add separate position and rotation?

Clean up code especially renderer

6/2/20

Added *std*::*vector*<*Vertex*<float>> *BackgroundModel* = *GetDefaultBackgroundModel*();

to background params.

Added edit boxes to background page

Looks like the dialog is ready – next connect to construction

Add default button to reset

6/6/20

Commit

Implemented vertex model vertices on background page with default button.

Next plan and implement separate world coordinates for the background model

6/9/20

m\_worldScale and m\_rotationParams are used to create the world matrix

*Vertex*<float> m\_worldScale = *Vertex*<float>(0.1f, 0.1f, 0.1f);

*RotationParams* m\_rotationParams;

*RotationAction* *Action*;

float *AngleXDegrees*;

float *AngleYDegrees*;

float *AngleZDegrees*;

*RotationSpeed* *Speed*;

Look at IDD\_POSITION\_ANGLE\_DLG

*Vertex*<float> m\_target;

*Vertex*<float> CFractalXDoc::GetTarget() const

IDD\_ROTATION\_DLG

Make separate RotationParams and Vertex<float> parameters and a flag to lock rotation params and target on first ones. Or copy from the first, then a lock to rotate with 1st params.

Plan

0) Decide whether to add target fields to RotationGroup – keep separate

1) Make an enum RotationSelection MainModel, Background, LockBackgroundOnModel -ok

2) Make a RotationGroup structure with two RotationParams and a RotationSelection -ok

3) Add these to doc and serialize -ok

4) Update PositionAngleDlg – to allow setting either params or fixing background = Model -ok

4.5) Update setters and getters to dialog, skip messages for now -ok

5) Update RotationDlg to show check box for fixing background = Model – only allow background to be fixed or rotate with main. -ok

5.5) Update setters and getters. -ok

6) Add new params to renderer -ok

7) Add another world matrix to renderer -ok

8) Connect world 2 to rotation params. -ok

8.2) Check messages from position angle dlg -ok

8.5) test / commit

9) Add second target param to doc and serialize

10) Connect to position rotation dlg

11) Connect to renderer

12) Make sure target 2 gets copied when LockBackgroundOnModel

6/13/20

Did 3 above.

IDC\_MAIN\_MODEL\_RAD

IDC\_BACKGROUND\_MODEL\_RAD

IDC\_COPY\_MAIN\_TO\_BKGND\_RAD

IDC\_COPY\_MAIN\_TO\_BKGND\_CHECK

Up to 6.

**6/16/20**

Two problems

1) It doesn’t show the background with or without the image

2) DDX and model options don’t work with the PositionAngleDlg

6/20/20

Fixed problem with background not showing.

Now the main, background RotationSelectionType is not stored

Also there is not option to set it in the setup sheet

1) Find out why it isn’t saved in the rotation angle dlg and fix

Because it isn’t copying the reference -ok

2) Fix when sending messages – NA

Commit Implemented rotation of background model

Commit – Fixed initialization of rotation selection type

3) Add to sheet

4) check rotation dlg

6/23/20

Commit – Fix rotation dlg -ok

Next find out when you save an image with a background the background isn’t saved?

1) new struct and serialize – ok

2) Add to doc -ok

3) Call doc to set -ok

4) Get from doc and set renderer -ok

Commit – Fixed serialization with background model

Do not add option to lock background with main to model sheet.

Next

1) Add target params

a) Add to structure -ok

b) serialize -ok

c) connect to UI – PositionAngleDlg -ok

d) Connect to renderer -ok

e) Check for other UI - no

2) Find out why exception when loading image for background -ok

3) Refactor – catch DX exceptions and display an error and return to working state -ok

4) Refactor serialization functions to be just called Serialize()

5) Refactor serialization functions to throw a serialization exception and catch it. -ok

6) Bump version

7) Save to program files

OnInitialUpdate()

Initialize()

m\_renderer->*SetModel*(\*pDoc->GetVertexData());

RunRotationParams()

ForeModelRender()

6/27/20

Add another target paramsAdded separate params for Background target.

Commit Added separate params for Background target.

Note that when you switch from copy Background from main, and switch back to main you see funny things because the old background params are used.

Commit Fixed exception when creating texture from file

7/1/20

Commit – Fixed catching exceptions in renderer

Needs more testing

Then Refactor serialization functions to be just called Serialize()

7/4/20

1) Make a new project. Call DxCore and use that as a namespace.

Start by putting SerializationException in it.

Commit – Created DxCore static library with CSerializationException.

Commit - Added throws and catch for CSerializationException.

Next try catches for other serialization exceptions

Added catch for CarchiveException

Next use std::lock\_guard<std::mutex>() for all locks and test exception catching again

Added

7/6/20

Commit Used lock\_guard in serialization code

Next

Refactor serialization functions to be just called Serialize()

Commit – Renamed serialization functions

Look for more things to serialize

Look at view

Commit Move fields to impl -ok

Change string functions to remove warnings -ok

Move code to another file or group like methods in a file -ok

Look at renderer

7/11/20

Commit Refactored view

Next refactor renderer.

Bump version and save to program files

7/14/20

Commit Refactored Renderer

discard below

winmerge key: 000014-1VGKFM-8FFUGE-RB67ZU-8X9K2A-DCUKEQ-6MJ1GT-UAKP8N-C71Y5H-QAEFP8

license name: prtgtrial

Visual assist

Serial Number(s):

HDFK-GRA8WM-LSNUKL-32ND

Tab studio

User Name: DONNA BATAGIANIS

Key: VGFic1N0dWRpbwABYacDGdvHW2Xpb9aOZYQyUvrgh5UBvrwtlKTIrOaqSEwyKukGYXbqV6YbEW3rRzn3dkmmDV0R84KfdXTUycmbMy/Q4d8fd9ue