

## Blender Unit Tests

### Output Generated By unittest

Read new prefs: C:\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\config\userpref.blend  
found bundled python: C:\Program Files\Blender Foundation\Blender\2.69\python  
test\_mist (USGS.test\_blender.TestBlenderDTMViewerRenderContext) ... ok  
test\_stars (USGS.test\_blender.TestBlenderDTMViewerRenderContext) ... ok  
test\_dtm\_loading (USGS.test\_blender.TestBlenderLoad) ... ok  
test\_dtm\_loading\_with\_null\_image (USGS.test\_blender.TestBlenderLoad) ... ok

-----  
Ran 4 tests in 7.568s

OK

### Execution Output:

Creating mistic mist...  
Mist created  
Adding stars to background...  
Stars applied successfully  
Bin Mode: BIN12-FAST  
Scale: 0.010000  
Color Mapping: None  
Flyover Mode: No flyover  
Processing image, saving at: C:\Users\Andrew\DTEED\_020492\_1830\_021481\_1830\_A01.blend  
Processing image in Blender, please be patient...  
Saved image at: C:\Users\Andrew\DTEED\_020492\_1830\_021481\_1830\_A01.blend  
DTM\_IMG: C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
DTM\_TEXTURE: None  
Bin Mode: BIN12-FAST  
Scale: 0.010000  
Color Mapping: None  
Flyover Mode: No flyover  
Processing image, saving at: C:\Users\Andrew\blend  
Processing image in Blender, please be patient...  
Unable to load the DEM.

**Test Code:**

```
__author__ = 'Andrew'
from . import blender_module
import bpy
import unittest
import bpy.props

#The DEM you want to use for blender unit tests
dem = 'C:\\Users\\Andrew\\Desktop\\DEMs\\DTEED_020492_1830_021481_1830_A01.IMG'

#This class of tests focuses on the DTMViewrRenderContext
#It tests the newly coded stars and mist functions
class TestBlenderDTMViewerRenderContext(unittest.TestCase):
    def test_stars(self):
        world = bpy.context.scene.world
        render = blender_module.DTMViewerRenderContext(dem,'1080p',True, False)

        [r, g, b] = world.zenith_color
        self.assertNotEqual([r, g, b], [0.040, 0.040, 0.040])
        self.assertNotEqual(world.use_sky_paper, True)
        self.assertNotEqual(world.use_sky_blend, True)
        self.assertNotEqual(world.use_sky_real, True)

        render.createStars()
        [r, g, b] = world.horizon_color
        self.assertAlmostEqual(r, 0.0)
        self.assertAlmostEqual(g, 0.0)
        self.assertAlmostEqual(b, 0.0)

        [r, g, b] = world.zenith_color
        self.assertAlmostEqual(r, 0.040)
        self.assertAlmostEqual(g, 0.040)
        self.assertAlmostEqual(b, 0.040)

        self.assertEqual(world.use_sky_paper, True)
        self.assertEqual(world.use_sky_blend, True)
        self.assertEqual(world.use_sky_real, True)
        self.assertEqual(world.star_settings.use_stars, True)

    TestBlenderDTMViewerRenderContext.cleanup(self)

    def test_mist(self):
        _DTMViewerRenderContext = blender_module.DTMViewerRenderContext(dem,'1080p', False,
True)
        world = bpy.context.scene.world
        mist = bpy.context.scene.world.mist_settings

        #Before mist is created
        #testing background colors
```

```
[r, g, b] = world.zenith_color
self.assertNotAlmostEqual(r, 0.040)
self.assertNotAlmostEqual(g, 0.040)
self.assertNotAlmostEqual(b, 0.040)
self.assertNotEqual(world.use_sky_paper, True)
self.assertNotEqual(world.use_sky_blend, True)
self.assertNotEqual(world.use_sky_real, True)
```

```
#testing mist properties
self.assertNotEqual(mist.use_mist, True)
self.assertNotEqual(mist.start, 1.0)
```

```
_DTMViewerRenderContext.createMist(_DTMViewerRenderContext)
#After mist is created
#test to insure all properties are as expected
[r, g, b] = world.horizon_color
self.assertAlmostEqual(r, 0.0)
self.assertAlmostEqual(g, 0.0)
self.assertAlmostEqual(b, 0.0)
```

```
[r, g, b] = world.zenith_color
self.assertAlmostEqual(r, 0.040)
self.assertAlmostEqual(g, 0.040)
self.assertAlmostEqual(b, 0.040)
self.assertEqual(world.use_sky_paper, True)
self.assertEqual(world.use_sky_blend, True)
self.assertEqual(world.use_sky_real, True)
```

```
#testing mist properties
self.assertEqual(mist.use_mist, True)
self.assertEqual(mist.start, 1.0)
self.assertAlmostEqual(mist.intensity, 0.15)
```

```
TestBlenderDTMViewerRenderContext.cleanup(self)
```

```
def cleanup(self):
    world = bpy.context.scene.world
    world.horizon_color = (0.051, 0.051, 0.051)
    world.zenith_color = (0.01, 0.01, 0.01)
    world.ambient_color = (0.0, 0.0, 0.0)
    world.use_sky_paper = False
    world.use_sky_blend = False
    world.use_sky_real = False

    world.mist_settings.use_mist = False
    world.mist_settings.start = 5.0
    world.mist_settings.depth = 25.0
    world.mist_settings.height = 0.0
    world.mist_settings.intensity = 0.0
```

```
world.star_settings.use_stars = False
world.star_settings.size = 2.0
world.star_settings.distance_min = 0.0
world.star_settings.color_random = 0.0
world.star_settings.average_separation = 15.0
```

#This whole class of tests is dedicated to insuring our newly coded loader function is working correctly.

#The tests that it covers are focused on saving and dtm loading.

```
class TestBlenderLoad(unittest.TestCase):
```

```
    #This is a very simple test to insure that a mesh is being created
```

```
    #We assume that the mesh logic is correct, so we are not testing to see if the DTM for correctness
```

```
    #Only that it exists
```

```
    def test_dtm_loading(self):
```

```
        blender_module.DTMViewerRenderContext.clearScene(self)
```

```
        blender_module.load(self, None,
```

```
            filepath=dem,
```

```
            scale=0.01,
```

```
            bin_mode='BIN12-FAST',
```

```
            color_pattern='None',
```

```
            flyover_pattern='No flyover',
```

```
            texture_location=None,
```

```
            cropVars=False,
```

```
            resolution='1080p',
```

```
            stars=False,
```

```
            mist=False)
```

```
        self.assertTrue(TestBlenderLoad.find_image(self))
```

```
        blender_module.DTMViewerRenderContext.clearScene(self)
```

# This test insures that if a bad file path is given that the process fails out and

# no image is loaded

```
    def test_dtm_loading_with_null_image(self):
```

```
        blender_module.DTMViewerRenderContext.clearScene(self)
```

```
        blender_module.load(self, None,
```

```
            filepath="",
```

```
            scale=0.01,
```

```
            bin_mode='BIN12-FAST',
```

```
            color_pattern='None',
```

```
            flyover_pattern='No flyover',
```

```
            texture_location=None,
```

```
            cropVars=False,
```

```
            resolution='1080p',
```

```
            stars=False,
```

```
            mist=False)
```

```
        self.assertFalse(TestBlenderLoad.find_image(self))
```

```
        blender_module.DTMViewerRenderContext.clearScene(self)
```

```
def find_image(self):  
    for item in bpy.data.objects:  
        if (item.type == 'MESH'):  
            return True  
    return False
```

## GDAL Unit Tests

### Output Generated By unittest

```
test_gdal_color_relief_blue_steel (USGS.test_gdal.TestGdalColorrelief) ... ok
test_gdal_color_relief_diverging_blue_red (USGS.test_gdal.TestGdalColorrelief) .
.. ok
test_gdal_color_relief_diverging_brown_blue (USGS.test_gdal.TestGdalColorrelief)
... ok
test_gdal_color_relief_diverging_green_red (USGS.test_gdal.TestGdalColorrelief)
... ok
test_gdal_color_relief_diverging_red_blue (USGS.test_gdal.TestGdalColorrelief) .
.. ok
test_gdal_color_relief_diverging_red_brown (USGS.test_gdal.TestGdalColorrelief)
... ok
test_gdal_color_relief_diverging_red_gray (USGS.test_gdal.TestGdalColorrelief) .
.. ok
test_gdal_color_relief_earth (USGS.test_gdal.TestGdalColorrelief) ... ok
test_gdal_color_relief_fail (USGS.test_gdal.TestGdalColorrelief) ... ok
test_gdal_color_relief_rainbow_light (USGS.test_gdal.TestGdalColorrelief) ... ok

test_gdal_color_relief_rainbow_medium (USGS.test_gdal.TestGdalColorrelief) ... o
k
test_gdal_color_relief_rainbow_saturated (USGS.test_gdal.TestGdalColorrelief) ..
. ok
test_gdal_color_relief_sequential_blue (USGS.test_gdal.TestGdalColorrelief) ...
ok
test_gdal_color_relief_sequential_bluegreen (USGS.test_gdal.TestGdalColorrelief)
... ok
test_gdal_color_relief_sequential_green (USGS.test_gdal.TestGdalColorrelief) ...
ok
test_gdal_color_relief_sequential_red (USGS.test_gdal.TestGdalColorrelief) ... o
k
test_gdal_color_relief_sequential_yellowbrown (USGS.test_gdal.TestGdalColorrelie
f) ... ok
test_gdal_hillshade (USGS.test_gdal.TestGdalHillshade) ... ok
test_gdal_hillshade_alt_file (USGS.test_gdal.TestGdalHillshade) ... ok
test_gdal_hillshade_bad_input (USGS.test_gdal.TestGdalHillshade) ... ok
test_hsv_merge_dem1_sequential_red (USGS.test_gdal.TestGdalHsvMerge) ... ok
test_hsv_merge_dem2_sequential_red_fail (USGS.test_gdal.TestGdalHsvMerge) ... ok
```

-----  
Ran 22 tests in 802.357s

DTM\_TEXTURE: None

Skipping flyover

No path or camera to attach to one another in attach\_camera\_to\_path.

No path or target to attach to one another in add\_target\_to\_path.

## Execution Output:

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Blue\_Steel.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Diverging\_BlueRed.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Diverging\_BrownBlue.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender

Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Diverging\_GreenRed.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Diverging\_RedBlue.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Diverging\_RedBrown.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Diverging\_RedGray.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.



Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Earth.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief "C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Sequential\_YellowBrown.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
FAIL

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Rainbow\_Light.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Rainbow\_Medium.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"

0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Rainbow\_Saturated.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Sequential\_Blue.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Sequential\_BlueGreen.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Sequential\_Green.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Sequential\_Red.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Sequential\_YellowBrown.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem hillshade  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\hillshade.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Hillshade Created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem hillshade  
C:\Users\Andrew\Desktop\DEMs\DTEEC\_017569\_1645\_016857\_1645\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\hillshade.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Hillshade Created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem hillshade "C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\hillshade.tiff"  
FAIL

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Sequential\_Red.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Running Command: OSGeo4W gdaldem hillshade  
C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\hillshade.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Hillshade Created.

Running Command: OSGeo4W python "C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\hsv\_merge.py"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"

"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\hillshade.tiff"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\texture.tiff"  
This process takes a while. Please be patient...  
0...10...20...30...40...50...60...70...80...90...100.  
Texture created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief "C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Sequential\_Green.txt"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
"FAIL

Running Command: OSGeo4W gdaldem hillshade "C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\hillshade.tiff"  
FAIL

Running Command: OSGeo4W python "C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\hsv\_merge.py"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\hillshade.tiff"  
"C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\texture.tiff"  
This process takes a while. Please be patient...  
FAIL

## Test Code:

```
__author__ = "Andrew"
from . import gdal_module
import unittest
import os

#####
###
#           The DEMs for the following tests           #
#####
###
dem1 = "C:\\Users\\Andrew\\Desktop\\DEMs\\DTEED_020492_1830_021481_1830_A01.IMG"
dem2 = "C:\\Users\\Andrew\\Desktop\\DEMs\\DTEEC_017569_1645_016857_1645_A01.IMG"

#####
###
#           Helpful variables used in all of the tests           #
#####
###
project_location = os.path.dirname(__file__)
color_destination = os.path.normpath(""+project_location+"/maps/colorrelief.tiff")
hillshade_destination = os.path.normpath(""+project_location+"/maps/hillshade.tiff")
hsv_merge_location = os.path.normpath(""+project_location+"/hsv_merge.py")
texture_location = os.path.normpath(""+project_location+"/maps/texture.tiff")

# Assumptions
#1.) The gdal and hsv_merge utilities are assumed to be pre-tested for correctness
#2.) We will be looking for a return code of 0 to see if an executed test successfully passes
#3.) We will be looking for a return code of 1 to see if an executed test fails

#All of these tests focus on the functions within the gdal driver
class TestGdalHillshade(unittest.TestCase):
    def test_gdal_hillshade(self):
        gdal = gdal_module.GDALDriver(dem1)

        gdal.gdal_clean_up(hillshade_destination, "")
        return_code = gdal.gdal_hillshade(hillshade_destination)
        self.assertEqual(return_code, 0)
        gdal.gdal_clean_up(hillshade_destination, "")

    def test_gdal_hillshade_alt_file(self):
        gdal = gdal_module.GDALDriver(dem2)

        gdal.gdal_clean_up(hillshade_destination, "")
        return_code = gdal.gdal_hillshade(hillshade_destination)
        self.assertEqual(return_code, 0)
        gdal.gdal_clean_up(hillshade_destination, "")

#No DEM
def test_gdal_hillshade_bad_input(self):
```

```

gdal = gdal_module.GDALDriver("")

gdal.gdal_clean_up(hillshade_destination, "")
return_code = gdal.gdal_hillshade(hillshade_destination)
self.assertEqual(return_code, 1)
gdal.gdal_clean_up(hillshade_destination, "")

#All of these tests do not test for correctness of GDAL
#These tests check for correctness of our created color files by processing them through gdal
#If they process, the color file was formed correctly... there's better ways to do this
# #But i'm lazy and this fulfills the concept logically
class TestGdalColorrelief(unittest.TestCase):
    def test_gdal_color_relief_blue_steel(self):
        gdal = gdal_module.GDALDriver(dem1)
        color_relief = os.path.normpath(">"+project_location+"/color_maps/Blue_Steel.txt")

        gdal.gdal_clean_up(color_destination, "")
        return_code = gdal.gdal_color_relief(color_relief, color_destination)
        self.assertEqual(return_code, 0)
        gdal.gdal_clean_up(color_destination, "")

    def test_gdal_color_relief_diverging_blue_red(self):
        gdal = gdal_module.GDALDriver(dem1)
        color_relief = os.path.normpath(">"+project_location+"/color_maps/Diverging_BlueRed.txt")

        gdal.gdal_clean_up(color_destination, "")
        return_code = gdal.gdal_color_relief(color_relief, color_destination)
        self.assertEqual(return_code, 0)
        gdal.gdal_clean_up(color_destination, "")

    def test_gdal_color_relief_diverging_brown_blue(self):
        gdal = gdal_module.GDALDriver(dem1)
        color_relief =
os.path.normpath(">"+project_location+"/color_maps/Diverging_BrownBlue.txt")

        gdal.gdal_clean_up(color_destination, "")
        return_code = gdal.gdal_color_relief(color_relief, color_destination)
        self.assertEqual(return_code, 0)
        gdal.gdal_clean_up(color_destination, "")

    def test_gdal_color_relief_diverging_green_red(self):
        gdal = gdal_module.GDALDriver(dem1)
        color_relief = os.path.normpath(">"+project_location+"/color_maps/Diverging_GreenRed.txt")

        gdal.gdal_clean_up(color_destination, "")
        return_code = gdal.gdal_color_relief(color_relief, color_destination)
        self.assertEqual(return_code, 0)
        gdal.gdal_clean_up(color_destination, "")

```

```

def test_gdal_color_relief_diverging_red_blue(self):
    gdal = gdal_module.GDALDriver(dem1)
    color_relief = os.path.normpath("\\\\"+project_location+"/color_maps/Diverging_RedBlue.txt\\")

    gdal.gdal_clean_up(color_destination, "")
    return_code = gdal.gdal_color_relief(color_relief, color_destination)
    self.assertEqual(return_code, 0)
    gdal.gdal_clean_up(color_destination, "")

def test_gdal_color_relief_diverging_red_brown(self):
    gdal = gdal_module.GDALDriver(dem1)
    color_relief = os.path.normpath("\\\\"+project_location+"/color_maps/Diverging_RedBrown.txt\\")

    gdal.gdal_clean_up(color_destination, "")
    return_code = gdal.gdal_color_relief(color_relief, color_destination)
    self.assertEqual(return_code, 0)
    gdal.gdal_clean_up(color_destination, "")

def test_gdal_color_relief_diverging_red_gray(self):
    gdal = gdal_module.GDALDriver(dem1)
    color_relief = os.path.normpath("\\\\"+project_location+"/color_maps/Diverging_RedGray.txt\\")

    gdal.gdal_clean_up(color_destination, "")
    return_code = gdal.gdal_color_relief(color_relief, color_destination)
    self.assertEqual(return_code, 0)
    gdal.gdal_clean_up(color_destination, "")

def test_gdal_color_relief_earth(self):
    gdal = gdal_module.GDALDriver(dem1)
    color_relief = os.path.normpath("\\\\"+project_location+"/color_maps/Earth.txt\\")

    gdal.gdal_clean_up(color_destination, "")
    return_code = gdal.gdal_color_relief(color_relief, color_destination)
    self.assertEqual(return_code, 0)
    gdal.gdal_clean_up(color_destination, "")

def test_gdal_color_relief_rainbow_light(self):
    gdal = gdal_module.GDALDriver(dem1)
    color_relief = os.path.normpath("\\\\"+project_location+"/color_maps/Rainbow_Light.txt\\")

    gdal.gdal_clean_up(color_destination, "")
    return_code = gdal.gdal_color_relief(color_relief, color_destination)
    self.assertEqual(return_code, 0)
    gdal.gdal_clean_up(color_destination, "")

def test_gdal_color_relief_rainbow_medium(self):
    gdal = gdal_module.GDALDriver(dem1)
    color_relief = os.path.normpath("\\\\"+project_location+"/color_maps/Rainbow_Medium.txt\\")

```



```

        gdal.gdal_clean_up(color_destination, "")
        return_code = gdal.gdal_color_relief(color_relief, color_destination)
        self.assertEqual(return_code, 0)
        gdal.gdal_clean_up(color_destination, "")

def test_gdal_color_relief_rainbow_saturated(self):
    gdal = gdal_module.GDALDriver(dem1)
    color_relief = os.path.normpath("{}".format(project_location + "/color_maps/Rainbow_Saturated.txt"))

    gdal.gdal_clean_up(color_destination, "")
    return_code = gdal.gdal_color_relief(color_relief, color_destination)
    self.assertEqual(return_code, 0)
    gdal.gdal_clean_up(color_destination, "")

def test_gdal_color_relief_sequential_blue(self):
    gdal = gdal_module.GDALDriver(dem1)
    color_relief = os.path.normpath("{}".format(project_location + "/color_maps/Sequential_Blue.txt"))

    gdal.gdal_clean_up(color_destination, "")
    return_code = gdal.gdal_color_relief(color_relief, color_destination)
    self.assertEqual(return_code, 0)
    gdal.gdal_clean_up(color_destination, "")

def test_gdal_color_relief_sequential_bluegreen(self):
    gdal = gdal_module.GDALDriver(dem1)
    color_relief = os.path.normpath("{}".format(project_location + "/color_maps/Sequential_BlueGreen.txt"))

    gdal.gdal_clean_up(color_destination, "")
    return_code = gdal.gdal_color_relief(color_relief, color_destination)
    self.assertEqual(return_code, 0)
    gdal.gdal_clean_up(color_destination, "")

def test_gdal_color_relief_sequential_green(self):
    gdal = gdal_module.GDALDriver(dem1)
    color_relief = os.path.normpath("{}".format(project_location + "/color_maps/Sequential_Green.txt"))

    gdal.gdal_clean_up(color_destination, "")
    return_code = gdal.gdal_color_relief(color_relief, color_destination)
    self.assertEqual(return_code, 0)
    gdal.gdal_clean_up(color_destination, "")

def test_gdal_color_relief_sequential_red(self):
    gdal = gdal_module.GDALDriver(dem1)
    color_relief = os.path.normpath("{}".format(project_location + "/color_maps/Sequential_Red.txt"))

    gdal.gdal_clean_up(color_destination, "")
    return_code = gdal.gdal_color_relief(color_relief, color_destination)
    self.assertEqual(return_code, 0)
    gdal.gdal_clean_up(color_destination, "")

```

```

def test_gdal_color_relief_sequential_yellowbrown(self):
    gdal = gdal_module.GDALDriver(dem1)
    color_relief =
os.path.normpath(""+"project_location+"/color_maps/Sequential_YellowBrown.txt\"])

    gdal.gdal_clean_up(color_destination, "")
    return_code = gdal.gdal_color_relief(color_relief, color_destination)
    self.assertEqual(return_code, 0)
    gdal.gdal_clean_up(color_destination, "")

def test_gdal_color_relief_fail(self):
    gdal = gdal_module.GDALDriver("")
    color_relief =
os.path.normpath(""+"project_location+"/color_maps/Sequential_YellowBrown.txt\"])

    gdal.gdal_clean_up(color_destination, "")
    return_code = gdal.gdal_color_relief(color_relief, color_destination)
    self.assertEqual(return_code, 1)
    gdal.gdal_clean_up(color_destination, "")

##This set of tests could have a lot more, the problem is it takes a very long time to process
##to make sure it is working correctly, it needs to generate color reliefs and hill shades
##and then merge them together.
class TestGdalHsvMerge(unittest.TestCase):
    def test_hsv_merge_dem1_sequential_red(self):
        gdal = gdal_module.GDALDriver(dem1)

        gdal.gdal_clean_up(color_destination, hillshade_destination)
        color_relief = os.path.normpath(""+"project_location+"/color_maps/Sequential_Red.txt\"])
        gdal.gdal_color_relief(color_relief, color_destination)
        gdal.gdal_hillshade(hillshade_destination)

        return_code = gdal.hsv_merge(hsv_merge_location, hillshade_destination, color_destination,
texture_location)
        self.assertEqual(return_code, 0)
        gdal.gdal_clean_up(color_destination, hillshade_destination)
        gdal.gdal_clean_up(texture_location, "")

    def test_hsv_merge_dem2_sequential_red_fail(self):
        gdal = gdal_module.GDALDriver("")

        gdal.gdal_clean_up(color_destination, hillshade_destination)
        color_relief = os.path.normpath(""+"project_location+"/color_maps/Sequential_Green.txt\"])
        gdal.gdal_color_relief(color_relief, color_destination)
        gdal.gdal_hillshade(hillshade_destination)

        return_code = gdal.hsv_merge(hsv_merge_location, hillshade_destination, color_destination,
texture_location)

```

```
self.assertEqual(return_code, 1)
gdal.gdal_clean_up(color_destination, hillshade_destination)
```

#At this point gdal cleanup has been run over 44 times no tests are needed for gdal cleanup to verify it's functionality

## Flyover Unit Tests

### Output Generated By unittest

```
test_circle_pattern (USGS.test_flyover.TestFlyoverPatterns) ... ok
test_diamond_pattern (USGS.test_flyover.TestFlyoverPatterns) ... ok
test_linear_pattern (USGS.test_flyover.TestFlyoverPatterns) ... ok
test_no_flyover (USGS.test_flyover.TestFlyoverPatterns) ... ok
test_distance_two_points_normal_usage (USGS.test_flyover.TestMiscellaneousFunctions) ... ok
test_distance_two_points_x_test (USGS.test_flyover.TestMiscellaneousFunctions) .
.. ok
test_distance_two_points_y_test (USGS.test_flyover.TestMiscellaneousFunctions) .
.. ok
test_distance_two_points_z_test (USGS.test_flyover.TestMiscellaneousFunctions) .
.. ok
test_get_center_general_usage (USGS.test_flyover.TestMiscellaneousFunctions) ...
ok
test_get_dem_boundaries_big_resolution (USGS.test_flyover.TestMiscellaneousFunctions) ... ok
test_get_dem_boundaries_normal_dem (USGS.test_flyover.TestMiscellaneousFunctions) ... ok
test_get_dem_boundaries_small_resolution (USGS.test_flyover.TestMiscellaneousFunctions) ... ok
test_midpoint_two_points_general_usage (USGS.test_flyover.TestMiscellaneousFunctions) ... ok
test_midpoint_two_points_x (USGS.test_flyover.TestMiscellaneousFunctions) ... ok

test_midpoint_two_points_y (USGS.test_flyover.TestMiscellaneousFunctions) ... ok

test_midpoint_two_points_z (USGS.test_flyover.TestMiscellaneousFunctions) ... ok
```

-----  
Ran 16 tests in 24.922s

OK

### Execution Output:

The flyover doesn't have any output to the console, except when things don't work, so there is nothing to put in this section.

**Test Code:**

```
__author__ = "Andrew"

from . import flyover_module
from . import blender_module
import unittest
import math
import bpy

#The DEM you want to use for blender unit tests
dem = 'C:\\Users\\Andrew\\Desktop\\DEMs\\DTEED_020492_1830_021481_1830_A01.IMG'

class TestFlyoverPatterns(unittest.TestCase):
    def test_no_flyover(self):
        #Verify normal pass through
        flyover = flyover_module.FlyoverDriver()
        bool = flyover.no_flyover()
        self.assertTrue(bool)
        TestFlyoverPatterns.cleanup_flyover(self)

    def test_linear_pattern(self):
        flyover = flyover_module.FlyoverDriver()
        bool = flyover.linear_pattern()
        self.assertTrue(bool)
        TestFlyoverPatterns.cleanup_flyover(self)

    def test_diamond_pattern(self):
        flyover = flyover_module.FlyoverDriver()
        bool = flyover.diamond_pattern()
        self.assertTrue(bool)
        TestFlyoverPatterns.cleanup_flyover(self)

    def test_circle_pattern(self):
        flyover = flyover_module.FlyoverDriver()
        bool = flyover.circle_pattern()
        self.assertTrue(bool)
        TestFlyoverPatterns.cleanup_flyover(self)

    #Wasn't needed in the normal flyover but i provided this here since there will be multiple flyovers
    #constructed
    def cleanup_flyover(self):
        #Delete the Camera, CameraTarget, and the Path
        for item in bpy.data.objects:
            if item.type == 'CAMERA':
                item.select = True
                bpy.ops.object.delete()
            if item.type == 'EMPTY':
                item.select = True
                bpy.ops.object.delete()
```

```
if item.type == 'CURVE':  
    item.select = True  
    bpy.ops.object.delete()
```

```
class TestMiscellaneousFunctions(unittest.TestCase):  
    def test_distance_two_points_x_test(self):  
        flyover = flyover_module.FlyoverDriver()  
  
        point_one = (1.0, 0.0, 0.0)  
        point_two = (0.0, 0.0, 0.0)  
  
        val = flyover.distance_two_points(point_one, point_two)  
        self.assertAlmostEqual(1.0, val)  
  
    def test_distance_two_points_y_test(self):  
        flyover = flyover_module.FlyoverDriver()  
  
        point_one = (0.0, 1.0, 0.0)  
        point_two = (0.0, 0.0, 0.0)  
  
        val = flyover.distance_two_points(point_one, point_two)  
        self.assertAlmostEqual(1.0, val)  
  
    def test_distance_two_points_z_test(self):  
        flyover = flyover_module.FlyoverDriver()  
  
        point_one = (0.0, 0.0, 1.0)  
        point_two = (0.0, 0.0, 0.0)  
  
        val = flyover.distance_two_points(point_one, point_two)  
        self.assertAlmostEqual(1.0, val)  
  
    def test_distance_two_points_normal_usage(self):  
        flyover = flyover_module.FlyoverDriver()  
  
        point_one = (1.0, 1.0, 2.0)  
        point_two = (0.0, -1.0, -1.0)  
  
        val = flyover.distance_two_points(point_one, point_two)  
        self.assertAlmostEqual( math.sqrt(14.0), val)  
  
    def test_midpoint_two_points_x(self):  
        flyover = flyover_module.FlyoverDriver()  
  
        point_one = (2.0, 0.0, 0.0)  
        point_two = (0.0, 0.0, 0.0)  
  
        val = flyover.midpoint_two_points(point_one, point_two)
```

```

self.assertAlmostEqual(val[0], 1.0)
self.assertAlmostEqual(val[1], 0.0)
self.assertAlmostEqual(val[2], 0.0)

def test_midpoint_two_points_y(self):
    flyover = flyover_module.FlyoverDriver()

    point_one = (0.0, 2.0, 0.0)
    point_two = (0.0, 0.0, 0.0)

    val = flyover.midpoint_two_points(point_one, point_two)
    self.assertAlmostEqual(val[0], 0.0)
    self.assertAlmostEqual(val[1], 1.0)
    self.assertAlmostEqual(val[2], 0.0)

def test_midpoint_two_points_z(self):
    flyover = flyover_module.FlyoverDriver()

    point_one = (0.0, 0.0, 2.0)
    point_two = (0.0, 0.0, 0.0)

    val = flyover.midpoint_two_points(point_one, point_two)
    self.assertAlmostEqual(val[0], 0.0)
    self.assertAlmostEqual(val[1], 0.0)
    self.assertAlmostEqual(val[2], 1.0)

def test_midpoint_two_points_general_usage(self):
    flyover = flyover_module.FlyoverDriver()

    point_one = (4.0, 2.0, 2.0)
    point_two = (2.0, -2.0, -1.0)

    val = flyover.midpoint_two_points(point_one, point_two)
    self.assertAlmostEqual(val[0], 3.0)
    self.assertAlmostEqual(val[1], 0.0)
    self.assertAlmostEqual(val[2], 0.5)

#This one seemed really odd to me, i got lost analyzing what data was being passed in...
#Not really sure why the midpoint of midpoints ends up being the actual midpoint...
#The image might be skewed...
def test_get_center_general_usage(self):
    flyover = flyover_module.FlyoverDriver()

    x_cross_mid = (100.0-0.0)/2
    y_cross_mid = (100.0-0.0)/2

    val = flyover.get_center(((100.0,0.0,0.0),(0.0,0.0,0.0),(0.0,100.0,0.0),(0.0,0.0,0.0)))
    self.assertAlmostEqual(x_cross_mid/2, val[0]) #x pos
    self.assertAlmostEqual(y_cross_mid/2, val[1]) #y pos

```

#Warning: these values had to be hard coded  
#If you don't have this DEM then you may spend a lot of time getting this test to work correctly.

```
def test_get_dem_boundaries_normal_dem(self):  
    #Load in a DTM (we need some type of mesh!)  
    blender_module.DTMViewerRenderContext.clearScene(self)  
    blender_module.load(self, None,  
        filepath=dem,  
        scale=0.01,  
        bin_mode='BIN12-FAST',  
        color_pattern='None',  
        flyover_pattern='No flyover',  
        texture_location=None,  
        cropVars=False,  
        resolution='1080p',  
        stars=False,  
        mist=False)
```

```
#Fetch the dem_boundaries  
flyover = flyover_module.FlyoverDriver()  
vals = flyover.get_dem_boundaries()  
x_max = vals[0]  
#check x y z positions  
self.assertAlmostEqual(x_max[0], 32.0400009)  
self.assertAlmostEqual(x_max[1], -51.9000015)  
self.assertAlmostEqual(x_max[2], 5.97231769)
```

```
x_min = vals[1]  
self.assertAlmostEqual(x_min[0], 0.0)  
self.assertAlmostEqual(x_min[1], -3.18000006)  
self.assertAlmostEqual(x_min[2], 1.376258969)
```

```
y_max = vals[2]  
self.assertAlmostEqual(y_max[0], 25.92000007)  
self.assertAlmostEqual(y_max[1], 0.0)  
self.assertAlmostEqual(y_max[2], 6.119607925)
```

```
y_min = vals[3]  
self.assertAlmostEqual(y_min[0], 6.42000007)  
self.assertAlmostEqual(y_min[1], -55.13999938)  
self.assertAlmostEqual(y_min[2], 6.86862039)
```

```
blender_module.DTMViewerRenderContext.clearScene(self)
```

#Small Scale

```
def test_get_dem_boundaries_small_resolution(self):  
    #Load in a DTM (we need some type of mesh!)  
    blender_module.DTMViewerRenderContext.clearScene(self)  
    blender_module.load(self, None,
```



```
    filepath=dem,  
    scale=0.000001,  
    bin_mode='BIN12-FAST',  
    color_pattern='None',  
    flyover_pattern='No flyover',  
    texture_location=None,  
    cropVars=False,  
    resolution='180p',  
    stars=False,  
    mist=False)
```

```
#Fetch the dem_boundaries  
flyover = flyover_module.FlyoverDriver()  
vals = flyover.get_dem_boundaries()  
x_max = vals[0]  
#check x y z positions  
self.assertAlmostEqual(x_max[0], 32.0400009)  
self.assertAlmostEqual(x_max[1], -51.9000015)  
self.assertAlmostEqual(x_max[2], 5.97231769)
```

```
x_min = vals[1]  
self.assertAlmostEqual(x_min[0], 0.0)  
self.assertAlmostEqual(x_min[1], -3.18000006)  
self.assertAlmostEqual(x_min[2], 1.376258969)
```

```
y_max = vals[2]  
self.assertAlmostEqual(y_max[0], 25.92000007)  
self.assertAlmostEqual(y_max[1], 0.0)  
self.assertAlmostEqual(y_max[2], 6.119607925)
```

```
y_min = vals[3]  
self.assertAlmostEqual(y_min[0], 6.42000007)  
self.assertAlmostEqual(y_min[1], -55.13999938)  
self.assertAlmostEqual(y_min[2], 6.86862039)
```

```
#Massive scale
```

```
def test_get_dem_boundaries_big_resolution(self):  
    #Load in a DTM (we need some type of mesh!)  
    blender_module.DTMViewerRenderContext.clearScene(self)  
    blender_module.load(self, None,  
        filepath=dem,  
        scale=1.0,  
        bin_mode='BIN12-FAST',  
        color_pattern='None',  
        flyover_pattern='No flyover',  
        texture_location=None,  
        cropVars=False,  
        resolution='1080p',  
        stars=False,
```

```
mist=False)

#Fetch the dem_boundaries
flyover = flyover_module.FlyoverDriver()
vals = flyover.get_dem_boundaries()
x_max = vals[0]
#check x y z positions
self.assertAlmostEqual(x_max[0], 32.0400009)
self.assertAlmostEqual(x_max[1], -51.9000015)
self.assertAlmostEqual(x_max[2], 5.97231769)

x_min = vals[1]
self.assertAlmostEqual(x_min[0], 0.0)
self.assertAlmostEqual(x_min[1], -3.18000006)
self.assertAlmostEqual(x_min[2], 1.376258969)

y_max = vals[2]
self.assertAlmostEqual(y_max[0], 25.92000007)
self.assertAlmostEqual(y_max[1], 0.0)
self.assertAlmostEqual(y_max[2], 6.119607925)

y_min = vals[3]
self.assertAlmostEqual(y_min[0], 6.42000007)
self.assertAlmostEqual(y_min[1], -55.13999938)
self.assertAlmostEqual(y_min[2], 6.86862039)
```

Creating mistic mist...  
Mist created  
Adding stars to background...  
Stars applied successfully  
Bin Mode: BIN12-FAST  
Scale: 0.010000  
Color Mapping: None  
Flyover Mode: No flyover  
Processing image, saving at: C:\Users\Andrew  
\DTEED\_020492\_1830\_021481\_1830\_A01.blend  
Processing image in Blender, please be patient...  
Saved image at: C:\Users\Andrew  
\DTEED\_020492\_1830\_021481\_1830\_A01.blend  
DTM\_IMG: C:\Users\Andrew\Desktop\DEMs  
\DTEED\_020492\_1830\_021481\_1830\_A01.IMG  
DTM\_TEXTURE: None  
Bin Mode: BIN12-FAST  
Scale: 0.010000  
Color Mapping: None  
Flyover Mode: No flyover  
Processing image, saving at: C:\Users\Andrew\blend  
Processing image in Blender, please be patient...  
Unable to load the DEM.

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief C:\Users\Andrew\Desktop  
\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG "C:\Users\Andrew\AppData  
\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS  
\color\_maps\Blue\_Steel.txt" "C:\Users\Andrew\AppData\Roaming\Blender  
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief C:\Users\Andrew\Desktop  
\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG "C:\Users\Andrew\AppData  
\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS  
\color\_maps\Diverging\_BlueRed.txt" "C:\Users\Andrew\AppData\Roaming  
\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps  
\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG "C:\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Diverging\_BrownBlue.txt" "C:\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG "C:\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Diverging\_GreenRed.txt" "C:\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG "C:\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Diverging\_RedBlue.txt" "C:\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG "C:\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Diverging\_RedBrown.txt" "C:\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"

```
\colorrelief.tiff"
0...10...20...30...40...50...60...70...80...90...100.
Color-Relief created.
```

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

```
Running Command: OSGeo4W gdaldem color-relief C:\Users\Andrew\Desktop
\DEMs\DTEED_020492_1830_021481_1830_A01.IMG "C:\Users\Andrew\AppData
\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS
\color_maps\Diverging_RedGray.txt" "C:\Users\Andrew\AppData\Roaming
\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps
\colorrelief.tiff"
0...10...20...30...40...50...60...70...80...90...100.
Color-Relief created.
```

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

```
Running Command: OSGeo4W gdaldem color-relief C:\Users\Andrew\Desktop
\DEMs\DTEED_020492_1830_021481_1830_A01.IMG "C:\Users\Andrew\AppData
\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS
\color_maps\Earth.txt" "C:\Users\Andrew\AppData\Roaming\Blender
Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"
0...10...20...30...40...50...60...70...80...90...100.
Color-Relief created.
```

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

```
Running Command: OSGeo4W gdaldem color-relief "C:\Users\Andrew
\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS
\color_maps\Sequential_YellowBrown.txt" "C:\Users\Andrew\AppData
\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps
\colorrelief.tiff"
FAIL
```

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

```
Running Command: OSGeo4W gdaldem color-relief C:\Users\Andrew\Desktop
\DEMs\DTEED_020492_1830_021481_1830_A01.IMG "C:\Users\Andrew\AppData
```

```
\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS
\color_maps\Rainbow_Light.txt" "C:\Users\Andrew\AppData\Roaming
\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps
\colorrelief.tiff"
0...10...20...30...40...50...60...70...80...90...100.
Color-Relief created.
```

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

```
Running Command: OSGeo4W gdaldem color-relief C:\Users\Andrew\Desktop
\DEMs\DTEED_020492_1830_021481_1830_A01.IMG "C:\Users\Andrew\AppData
\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS
\color_maps\Rainbow_Medium.txt" "C:\Users\Andrew\AppData\Roaming
\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps
\colorrelief.tiff"
0...10...20...30...40...50...60...70...80...90...100.
Color-Relief created.
```

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

```
Running Command: OSGeo4W gdaldem color-relief C:\Users\Andrew\Desktop
\DEMs\DTEED_020492_1830_021481_1830_A01.IMG "C:\Users\Andrew\AppData
\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS
\color_maps\Rainbow_Saturated.txt" "C:\Users\Andrew\AppData\Roaming
\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps
\colorrelief.tiff"
0...10...20...30...40...50...60...70...80...90...100.
Color-Relief created.
```

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

```
Running Command: OSGeo4W gdaldem color-relief C:\Users\Andrew\Desktop
\DEMs\DTEED_020492_1830_021481_1830_A01.IMG "C:\Users\Andrew\AppData
\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS
\color_maps\Sequential_Blue.txt" "C:\Users\Andrew\AppData\Roaming
\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps
\colorrelief.tiff"
0...10...20...30...40...50...60...70...80...90...100.
Color-Relief created.
```

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG "C:\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Sequential\_BlueGreen.txt" "C:\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG "C:\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Sequential\_Green.txt" "C:\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG "C:\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Sequential\_Red.txt" "C:\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps\colorrelief.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief C:\Users\Andrew\Desktop\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG "C:\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\color\_maps\Sequential\_YellowBrown.txt" "C:\Users\Andrew\AppData

```
\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps
\colorrelief.tiff"
0...10...20...30...40...50...60...70...80...90...100.
Color-Relief created.
```

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

```
Running Command: OSGeo4W gdaldem hillshade C:\Users\Andrew\Desktop
\DEMs\DTEED_020492_1830_021481_1830_A01.IMG "C:\Users\Andrew\AppData
\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps
\hillshade.tiff"
0...10...20...30...40...50...60...70...80...90...100.
Hillshade Created.
```

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

```
Running Command: OSGeo4W gdaldem hillshade C:\Users\Andrew\Desktop
\DEMs\DTEEC_017569_1645_016857_1645_A01.IMG "C:\Users\Andrew\AppData
\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps
\hillshade.tiff"
0...10...20...30...40...50...60...70...80...90...100.
Hillshade Created.
```

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

```
Running Command: OSGeo4W gdaldem hillshade "C:\Users\Andrew\AppData
\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps
\hillshade.tiff"
FAIL
```

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

```
Running Command: OSGeo4W gdaldem color-relief C:\Users\Andrew\Desktop
\DEMs\DTEED_020492_1830_021481_1830_A01.IMG "C:\Users\Andrew\AppData
\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS
\color_maps\Sequential_Red.txt" "C:\Users\Andrew\AppData\Roaming
\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps
\colorrelief.tiff"
```



0...10...20...30...40...50...60...70...80...90...100.  
Color-Relief created.

Running Command: OSGeo4W gdaldem hillshade C:\Users\Andrew\Desktop  
\DEMs\DTEED\_020492\_1830\_021481\_1830\_A01.IMG "C:\Users\Andrew\AppData  
\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps  
\hillshade.tiff"  
0...10...20...30...40...50...60...70...80...90...100.  
Hillshade Created.

Running Command: OSGeo4W python "C:\Users\Andrew\AppData\Roaming  
\Blender Foundation\Blender\2.69\scripts\addons\USGS\hsv\_merge.py" "C:  
\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\scripts  
\addons\USGS\maps\colorrelief.tiff" "C:\Users\Andrew\AppData\Roaming  
\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps  
\hillshade.tiff" "C:\Users\Andrew\AppData\Roaming\Blender Foundation  
\Blender\2.69\scripts\addons\USGS\maps\texture.tiff"  
This process takes a while. Please be patient...  
0...10...20...30...40...50...60...70...80...90...100.  
Texture created.

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Cleaning up Gdal temp images...

Running Command: OSGeo4W gdaldem color-relief "C:\Users\Andrew  
\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS  
\color\_maps\Sequential\_Green.txt" "C:\Users\Andrew\AppData\Roaming  
\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps  
\colorrelief.tiff"  
"FAIL"

Running Command: OSGeo4W gdaldem hillshade "C:\Users\Andrew\AppData  
\Roaming\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps  
\hillshade.tiff"  
FAIL

Running Command: OSGeo4W python "C:\Users\Andrew\AppData\Roaming  
\Blender Foundation\Blender\2.69\scripts\addons\USGS\hsv\_merge.py" "C:  
\Users\Andrew\AppData\Roaming\Blender Foundation\Blender\2.69\scripts  
\addons\USGS\maps\colorrelief.tiff" "C:\Users\Andrew\AppData\Roaming  
\Blender Foundation\Blender\2.69\scripts\addons\USGS\maps  
\hillshade.tiff" "C:\Users\Andrew\AppData\Roaming\Blender Foundation

\Blender\2.69\scripts\addons\USGS\maps\texture.tiff"  
This process takes a while. Please be patient...