

Assignment 05 – Isosurface

Interface design

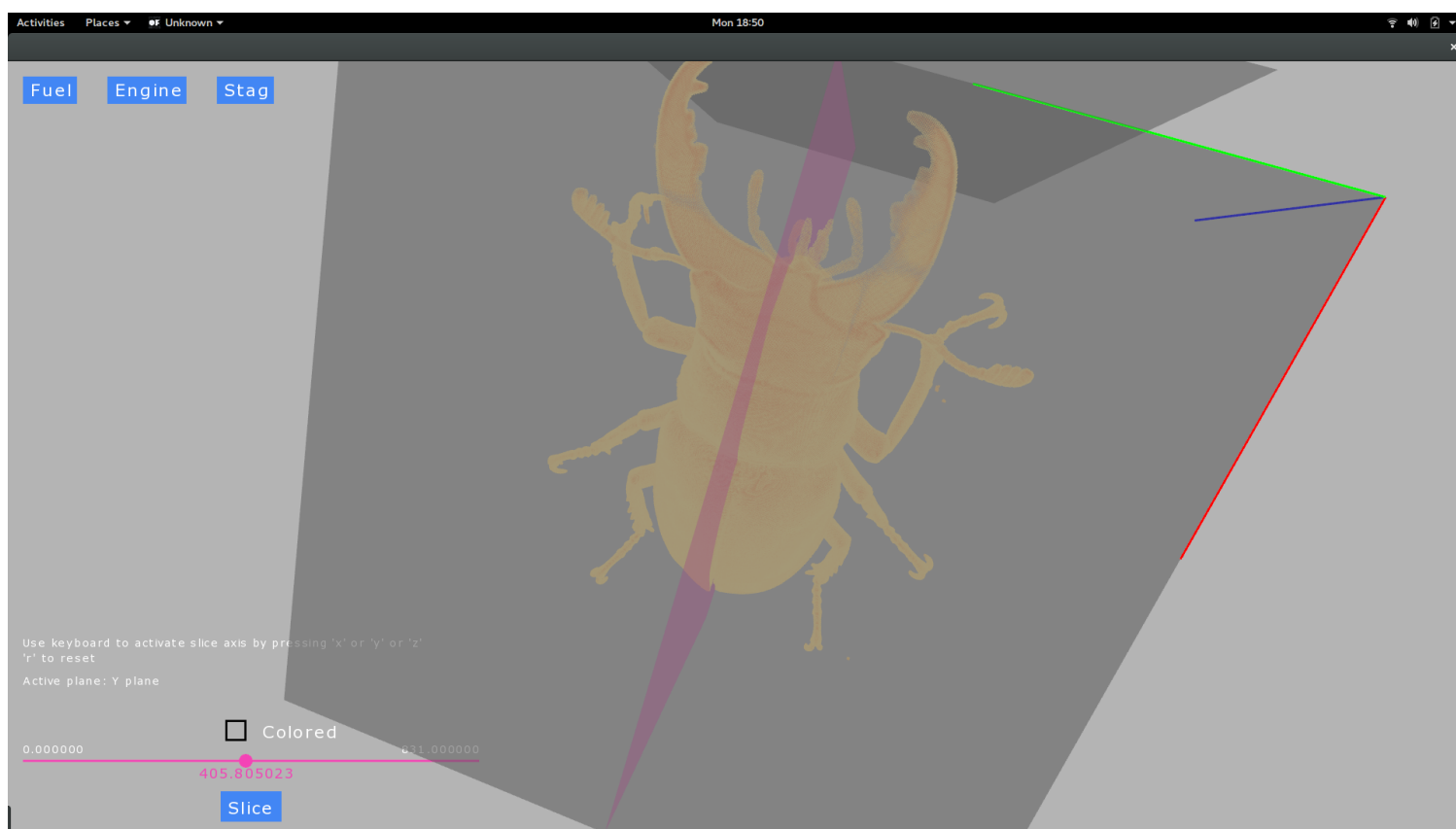
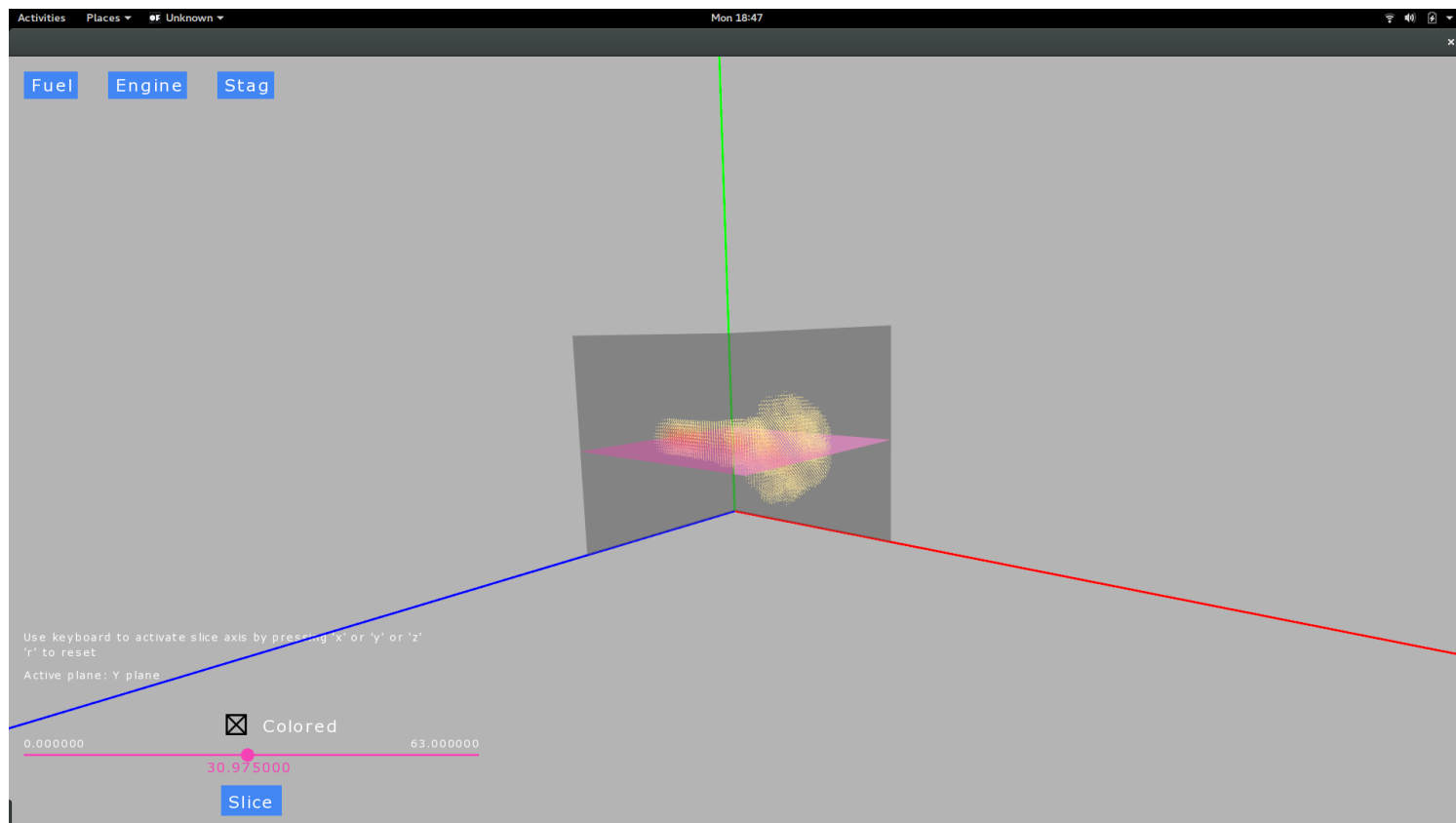
1. To depict the model, I'm plotting the points in a 3D volume. The points are color coded according to their intensity.
2. I also drew the reference X, Y, Z planes in gray along with 3 axes. This gives a notion of reference for the user when they rotate/zoom or do any kind of other interaction with the visualization.
3. Even before the user moves the plane, they can rotate and/or zoom in to see the model more closely.
4. To perform the slice operation -
 1. User should activate one of the planes – X or Y or Z plane by pressing 'X' / 'Y' / 'Z' on keyboard. Conversely, to reset the activation, user can hit 'R' key on keyboard.
 2. Once the plane is activated, it turns pink, thus standing out. So does the slider at bottom left corner of the screen.
 3. User can use the slider to move the plane forward/backward along its respective axis. Meanwhile, user can turn around / zoom the 3D model to get a better view.
 4. User can also now choose whether he wants a gray scale or color slice using the color check box.
 5. Once the user has move the plane to desired position & chosen color/gray scale configuration, he can click slice button below the slider.
 6. User should now see the sliced image. He can also find legend beside the image to decode the color to data value.
 7. The sliced image is scaled up or down to fit the image in the available space.

Interactions

1. User uses keys 'X'/'Y'/'Z'/'R' in combination with slider to perform slice.
2. User can use check box to indicate whether color/gray scale. If checked, color else gray scale.

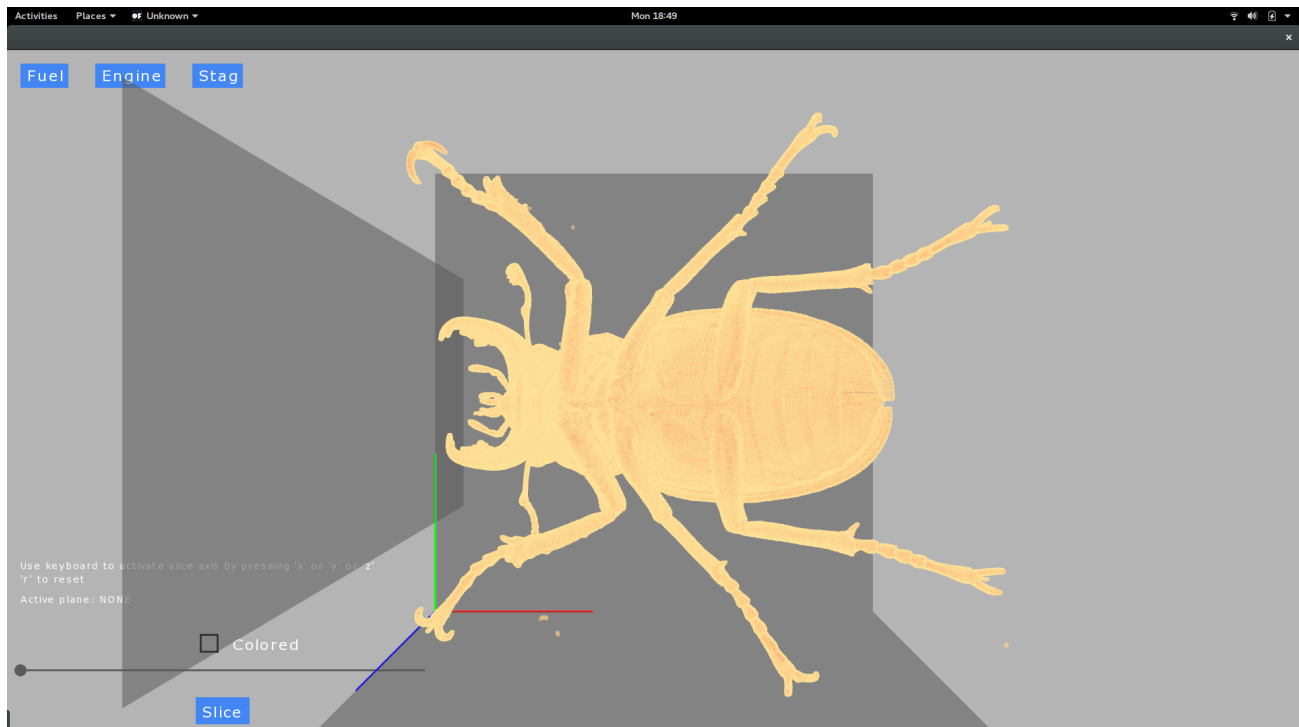
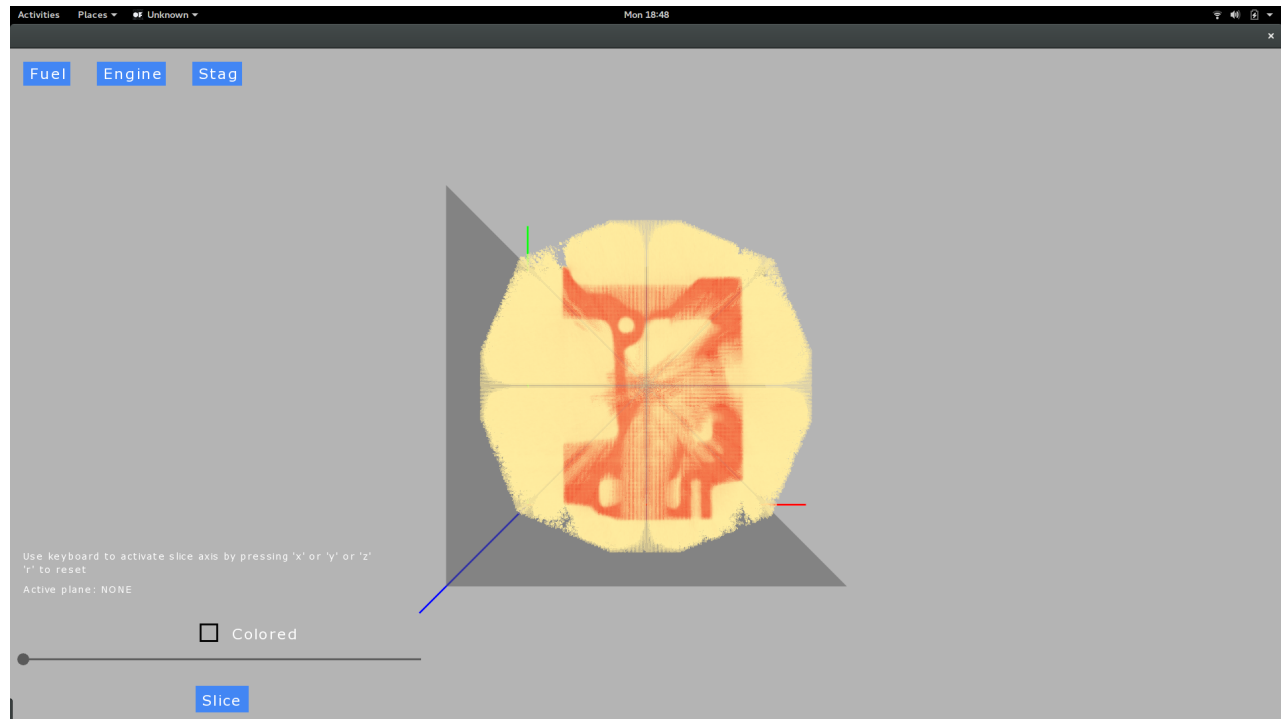
Other features

1. User can use the 3D rendering of the model to accurately position the plane in the desired plane. Depicted below in screenshots.



2. The points in the 3D model are color coded according to their data values, thus showing interesting internal patterns within the model even before slicing. Note: I didn't see these internal patterns when I had a single color for all the points. All I saw was the outline of the 3D model filled on the inside.

Depicted in below screenshots.



Color map

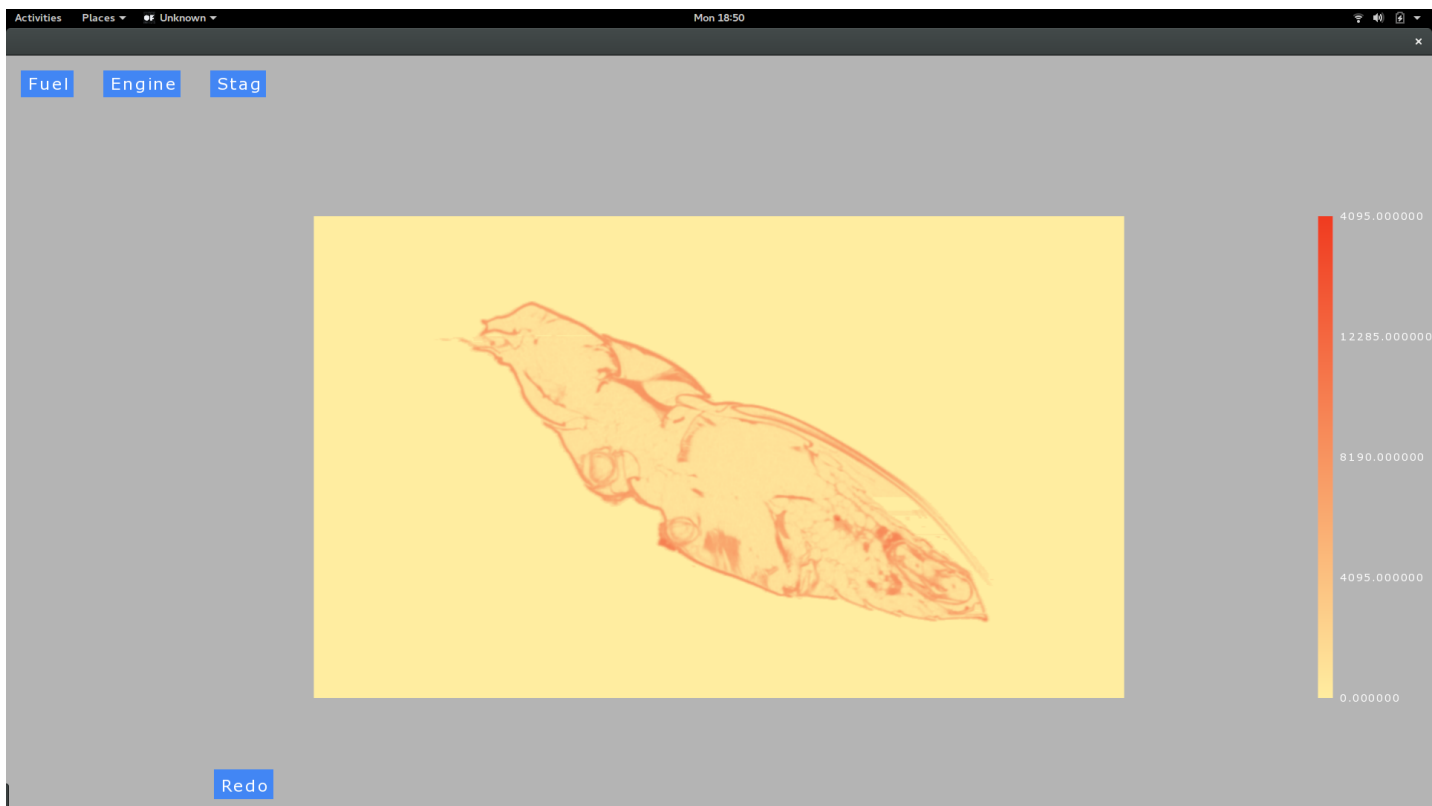
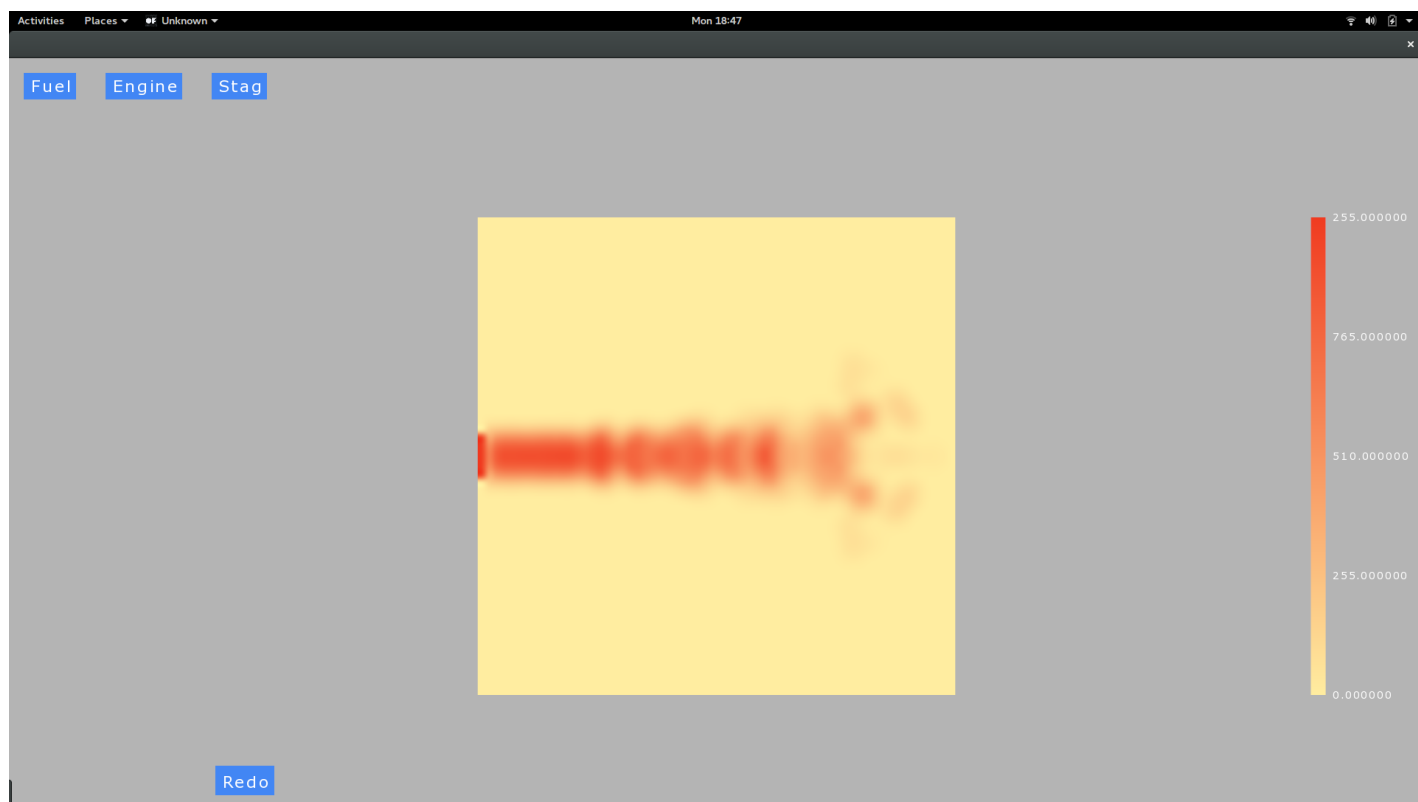
1. I used a linear color map with values from light yellow to dark orange going from min to max. I experimented with a few more colors like blue and green. Neither was able to depict the contrast between data values as good as yellow-to-orange color map.
2. I tried the multi hue color map in comparison. The the contrast was good, with many colors it was very difficult to say which color corresponds to above average value or below average value. This was easily solved with a continuous linear color map with light representing below average and dark representing above average.
3. I couldn't try rainbow color map but I'm guessing the result would have equally bad. Unfortunately I didn't take snapshots of multi hue color map to compare with. Didn't think I would need it in the report when I reverted the change.

File loading

Unlike last time, where I used `ofxSystemDialog` to let user choose dataset, I've buttons this time for Fuel, Engine and Stag dataset. This was because `ofxSystemDialog` was crashing in the Ubuntu 16.04 and I saw posts on `openframeworks` site on this issue. Seems like it is still open. However, I'm using `ofDataPath` and hence as long as these dataset are in the data path of the app, it should be able to read them.

Sliced images





Interesting observations

1. The slice of stag and engine was most interesting. I was able to see the internal organs of stag and subtle body structure. In case of engine I was able to see the internals of the engine which I failed to see in a 3D model.