Computer Graphics I

Submitted by Zubin Bhuyan

# Project Report
## Week 3

So far a 3D house structure, along with options to shear, rotate, and add lights to the scene have been implemented. An option to cast shadow on a plane below also has been added. Fig. 1 is the screenshots of the webpage.

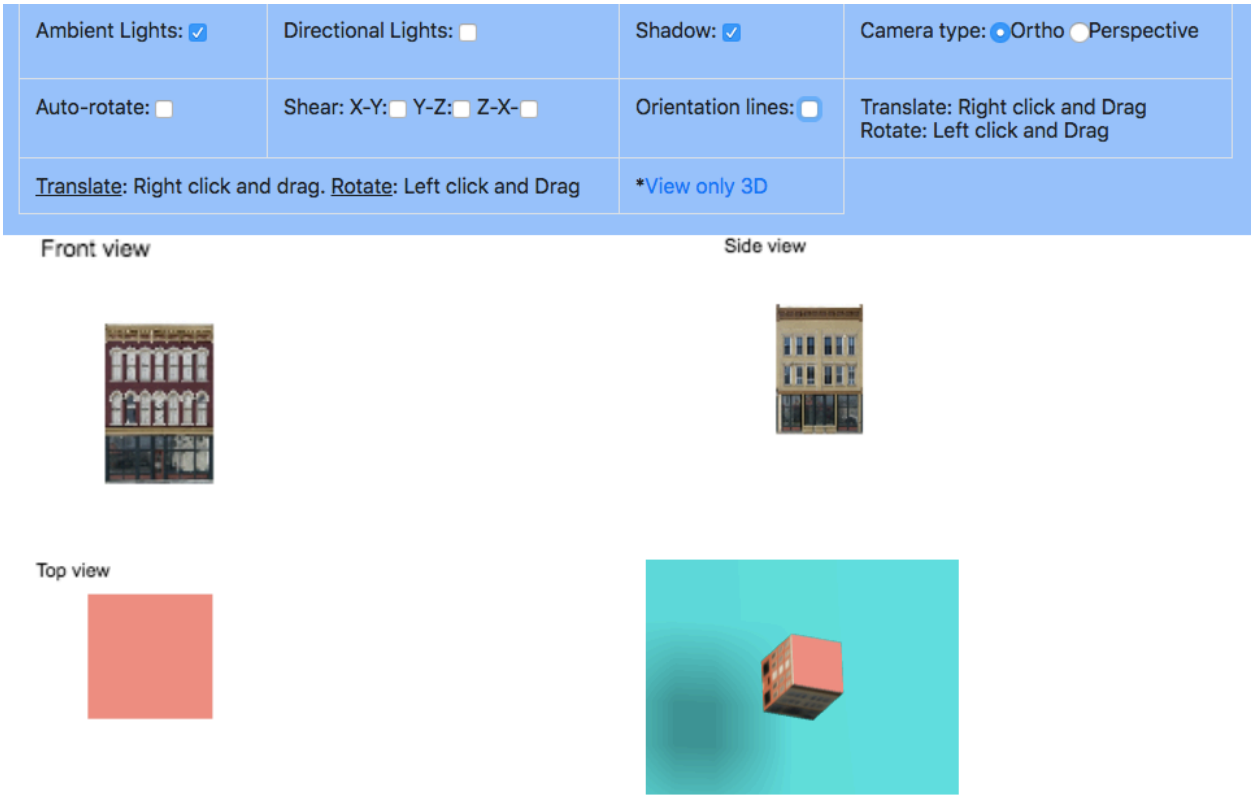


Fig. 1 Orthographic view of the structure from above with a shadow being cast below.

## Languages & libraries

The javascript library, THREE.js is used to define and render the objects and environment. This library is based on the WebGL open source library. THREE.js library is available under the MIT license.

## Viewing, transformations & projection

The object can be viewed from multiple directions. Simple mouse movements and keyboard inputs can be used to perform basic transformations. And the camera is set for perspective projection.

View direction can be set by clicking and dragging the left mouse button. Translation can either be performed by using the keyboard arrows, or bu right clicking and dragging. The zoom operation can be performed by scrolling in and out.

## Lighting, Texture and Material

Several light sources have been used to illustrate how lighting affects a scene. The webpage has option to turn on or off *ambient light* and two sources of directions light. Fig. 2 shows different types of lights which can be used.

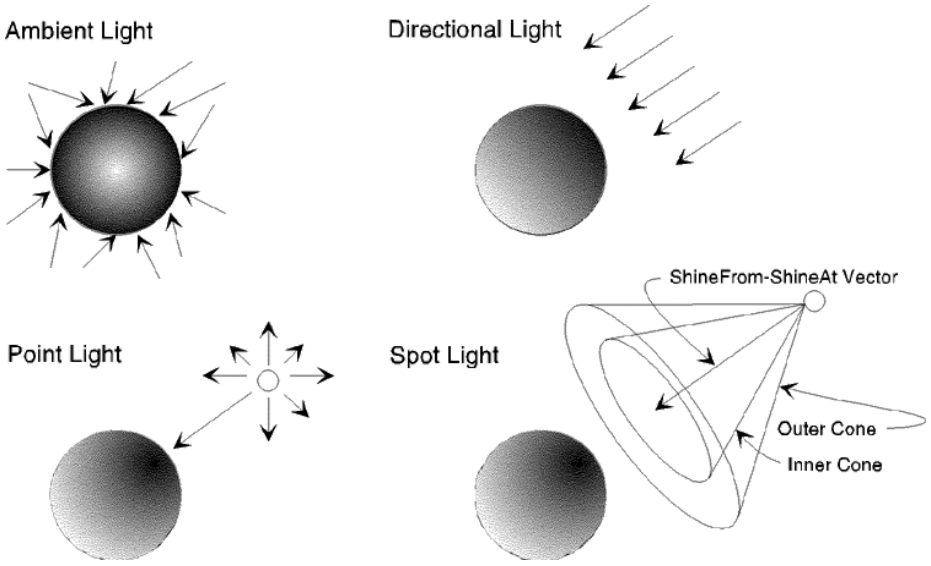


Fig. 2 Types of common light sources. The darker shades indicate higher intensity of light.

The type of material plays a crucial role in determining how the final objects are rendered. In this project, the two materials used are *basic material* and *Phong materials* on the surface of the building. Fig 3 lists common material types.

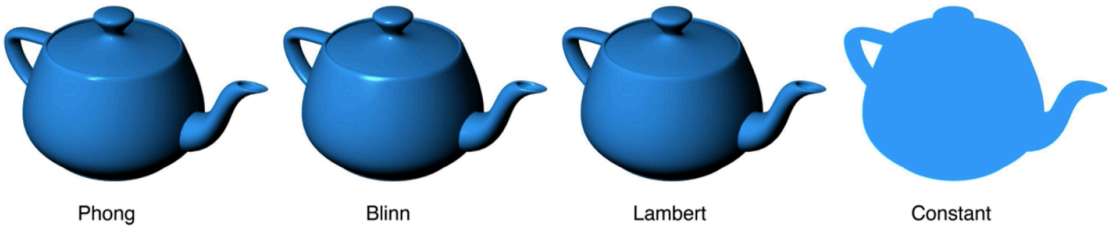


Fig 3 Common material types.

## WEEK 2 updates

1. Option to choose camera types has been added (**Orthographic and perspective**).The scene is rendered with a new camera every time the radio button selection is changed.
2. Skew option along 3 axes.
    1. Skewing can be done individually for each axis.
3. Front, side and top view canvases are added.
4. The webpage has been made responsive with ***bootstrap grid***.

### Shearing

Shearing is the transformation which slants the shape of an object. In 2-D it can be defined using the matrix $X_{sh}$:

$$X_{sh} = \begin{bmatrix} 1 & 0 & 0 \\ shx & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$X' = X + Sh_x \cdot Y$$

$$Y' = Y$$

```
var matrix = new THREE.Matrix4();

matrix.set(   1,   Syx,  Szx,  0,
            Sxy,     1,  Szy,  0,
            Sxz,   Syz,   1,   0,
              0,     0,   0,   1  );

// apply shear matrix to geometry
cube.geometry.applyMatrix( matrix );
scene.add( cube );
```

Fig. 3 shows the code snippet which is used to skew cube shapes in THREE.js.

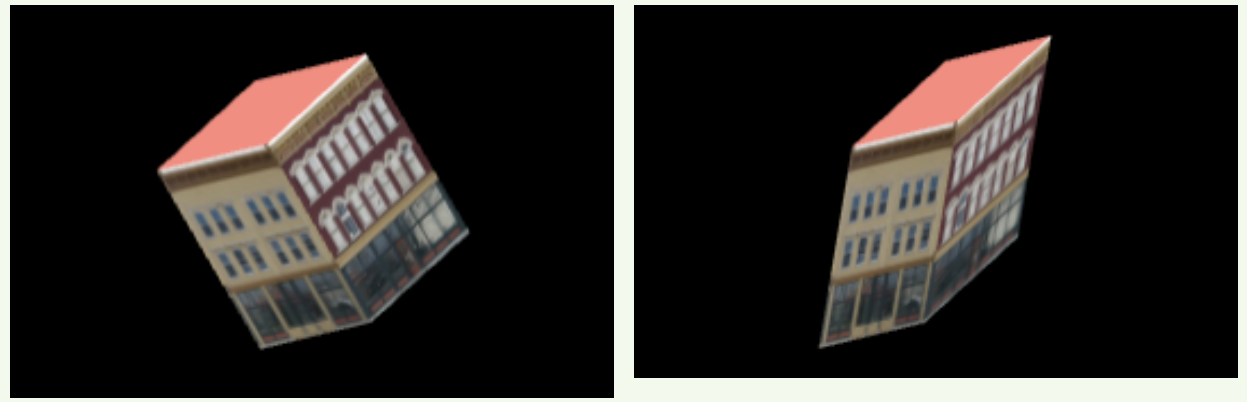

Fig 4. Normal image and a Skewed image (Orthographic)

## Orthographic vs Perspective

The following images show the difference between the perspective and orthographic cameras.

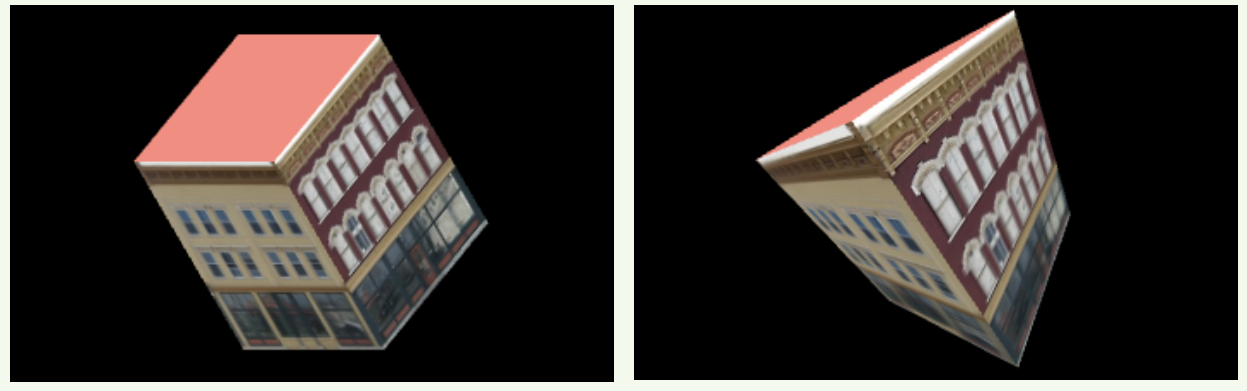

Fig 5 a, b. Orthographic and Perspective camera for the same object.

## Week 3 updates

1. Anti-aliasing for the renderer has been enabled.
2. Orientation lines: A check box to show or hide orientation.
3. Option to cast shadow on a plane below the structure.

## Shadows

Fig 6 illustrates how shadows are rendered. The near and far planes (with black crosses), and the top/left/bottom/right of the shadow camera define a box. This box has to bind objects needed for shadows.

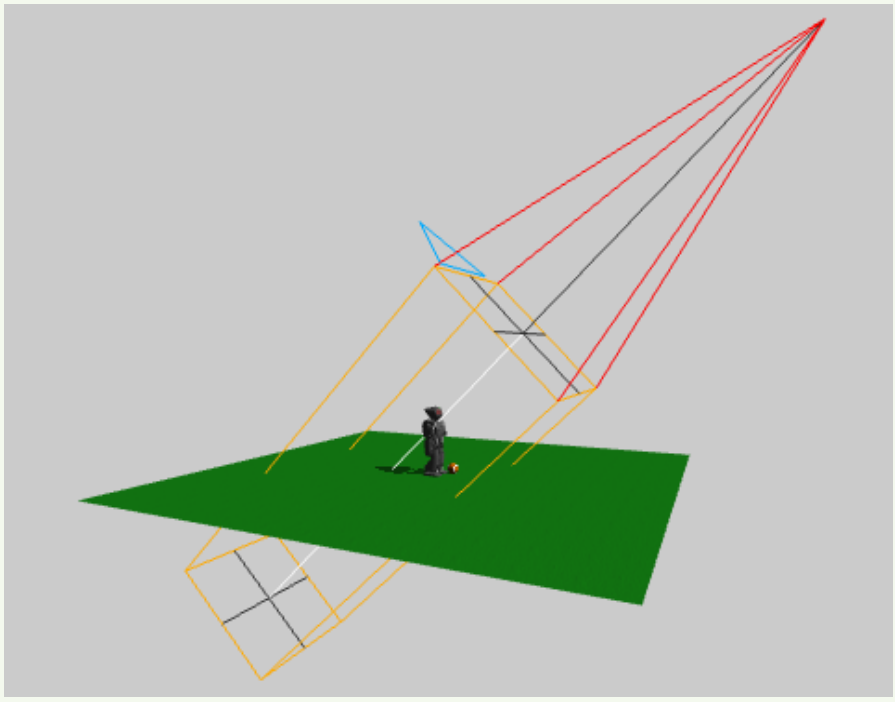

Fig 6. Shadow rendering.

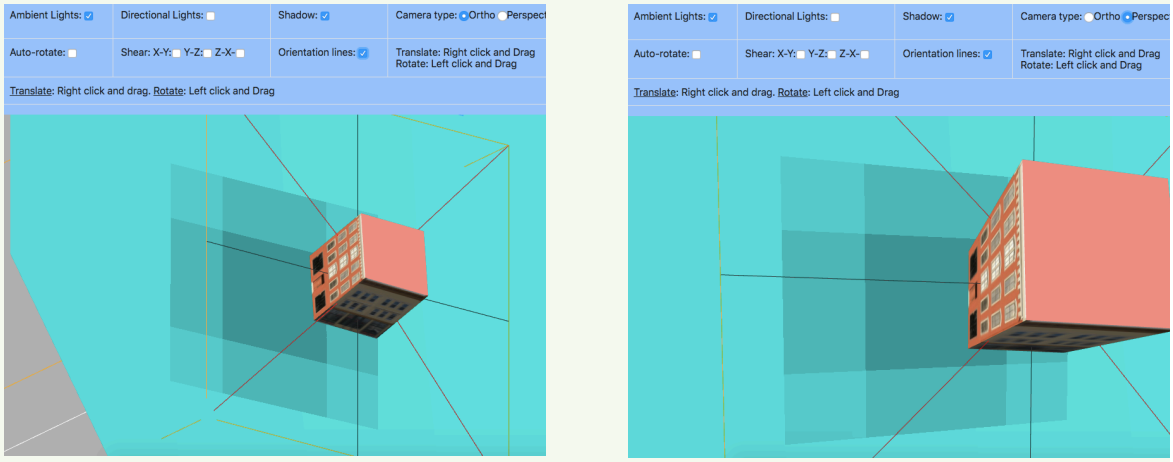


Fig 7 a, b Shadow rendering without Percentage closer Filtering (PCF) for both Orthographic and Perspective

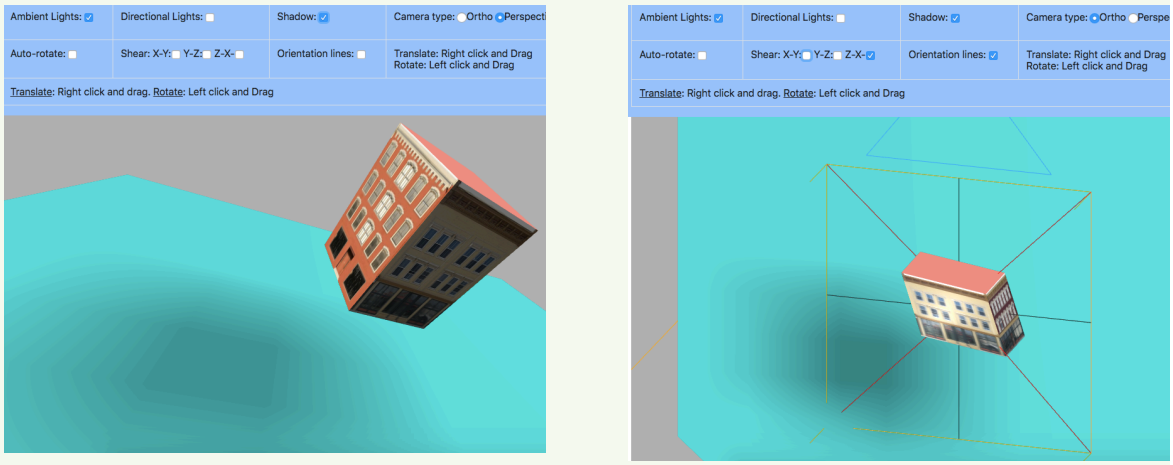


Fig 8 a, b PCF Shadow rendering. Fig 8b shows shadow casting with a sheared structure with helper orientation lines switched on.