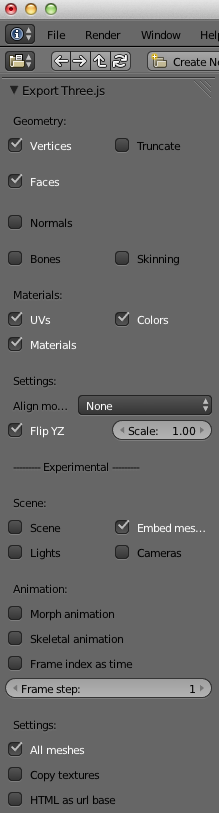
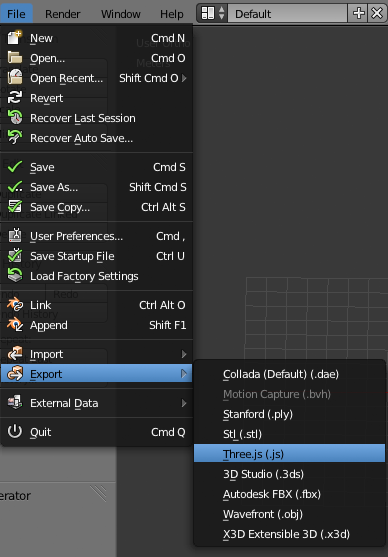
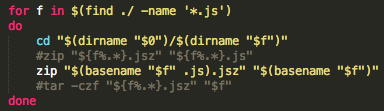
# Exporting Blender to Three.js

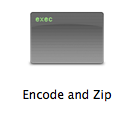
Uncheck Normals (makes smaller files, normal can be calculated during loading) and check UVs if any textures are present.



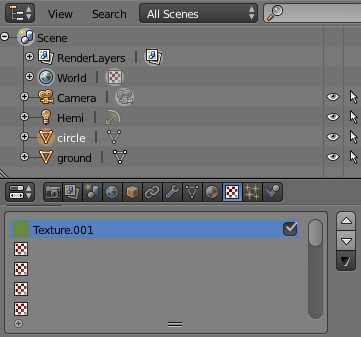
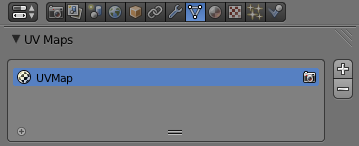
# Compressing js into jsz

Run *Encode and Zip* script



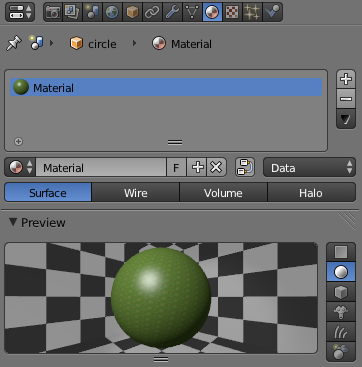


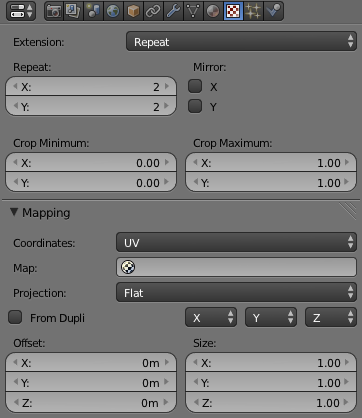
# Creating Textured Material

Make sure your create UVMap for each selected material

# Repeat Texture

For each selected material which requires a repeat texture

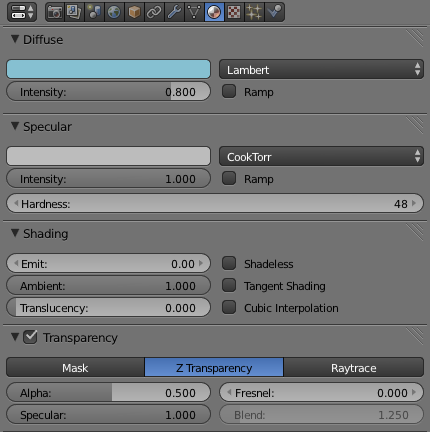




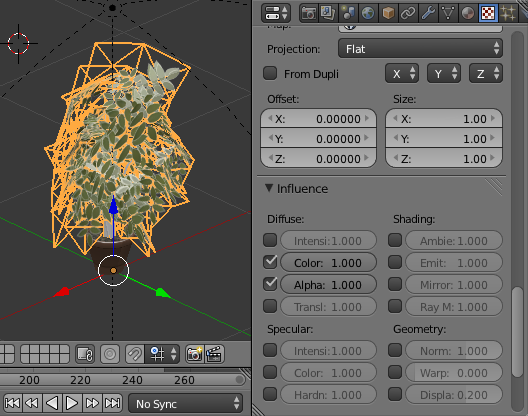
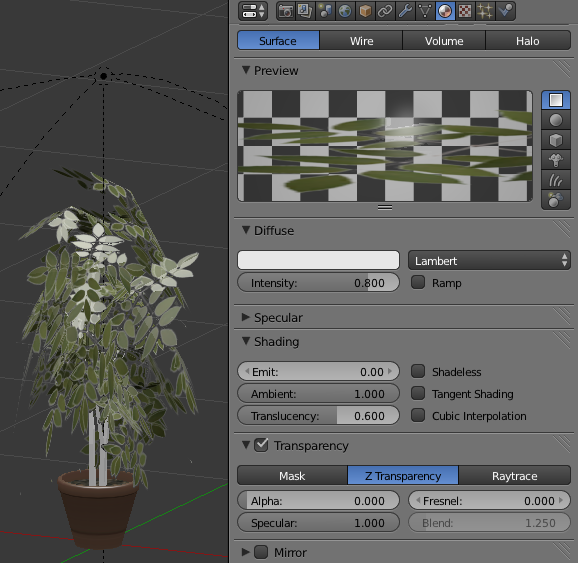
**Note:** Some shapes do not distribute repeat texture as square pattern; sometimes it is necessary to adjust mesh

# Creating Transparency Texture

Alpha less than 1 becomes transparent glass texture. Intensity can be kept at 1

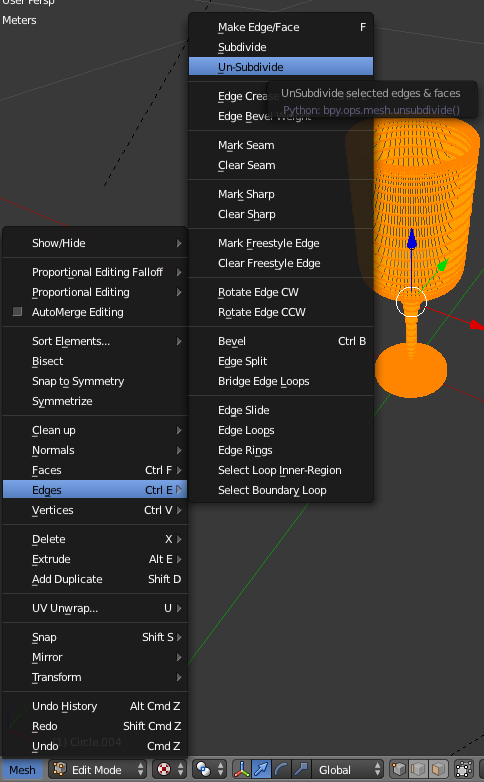
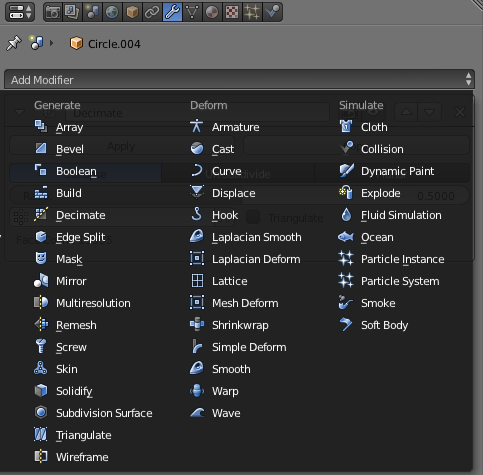
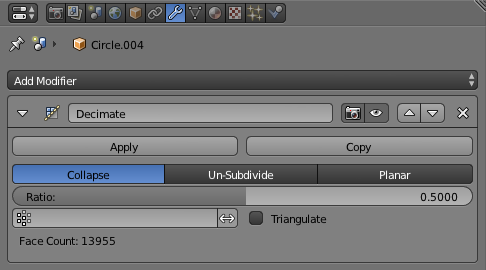


# Creating Transparency Texture for Plants

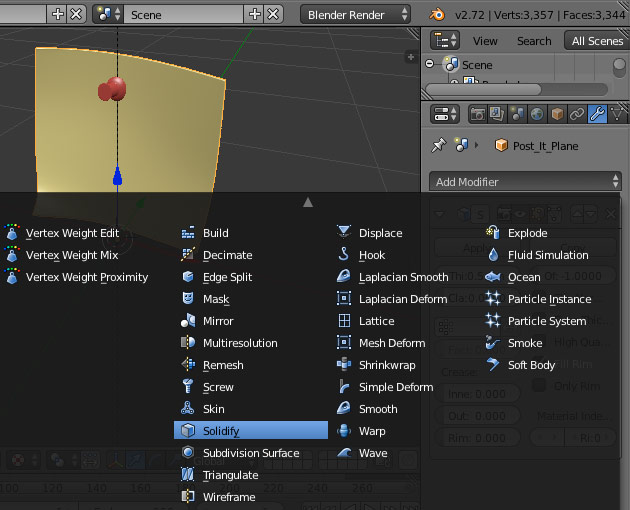
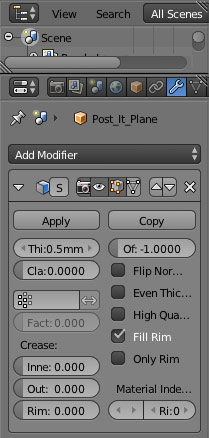
Make sure to use 24-bit png image as texture with transparency. Set Alpha 1 in Textures and 0 in Materials

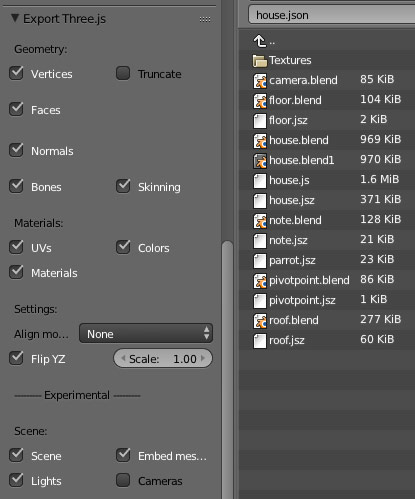
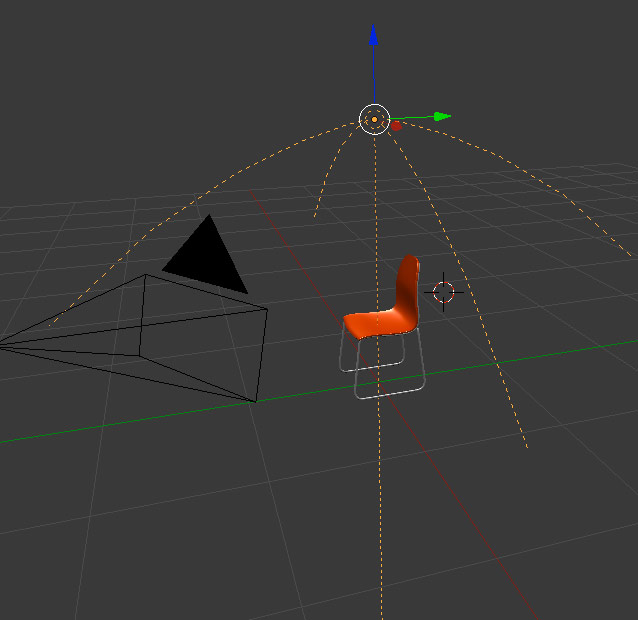
# Reducing the Polygons

There are two effective ways of reducing the polygon size: edit the mesh directly and use “Decimate” modifier



# DoubleSided Mesh

Sometime it is required to have surface color on both sides. Although Three.js can “flip” the material it comes at a render performance cost. For simplicity it is up to Blender to provide side-ready object.  
  
Embed Lights into Exported 3D Model  
  
This feature is useful when creating lamps and custom lighting. The trick is the export the model as “scene”. (**Scene and Lights must be checked**) Three.js treats these objects a little differently from THREE.JSONLoader() If embedding lights as part of the model you need to use THREE.SceneLoader() to load 3D model back into web-world.



# Notes

* Blender apparently has two types of rendering systems “Internal” and “Cycles”. Although cycles produces good rendered results - for this project cycles is not good, it takes too long to generate thumbnails and has problems with three.js conversion scripts (missing transparencies)
* Note that objects that rely on “Particle Engine” in blender will not be converted to three.js therefore you need to convert Particles Modifier into mesh in Blender before export.
* Textures JPG @ 64% quality 512px by 512px are considered high res. Anything over 512px will be loaded slow online and should be checked for final image size (typical max at around 150Kb/texture)