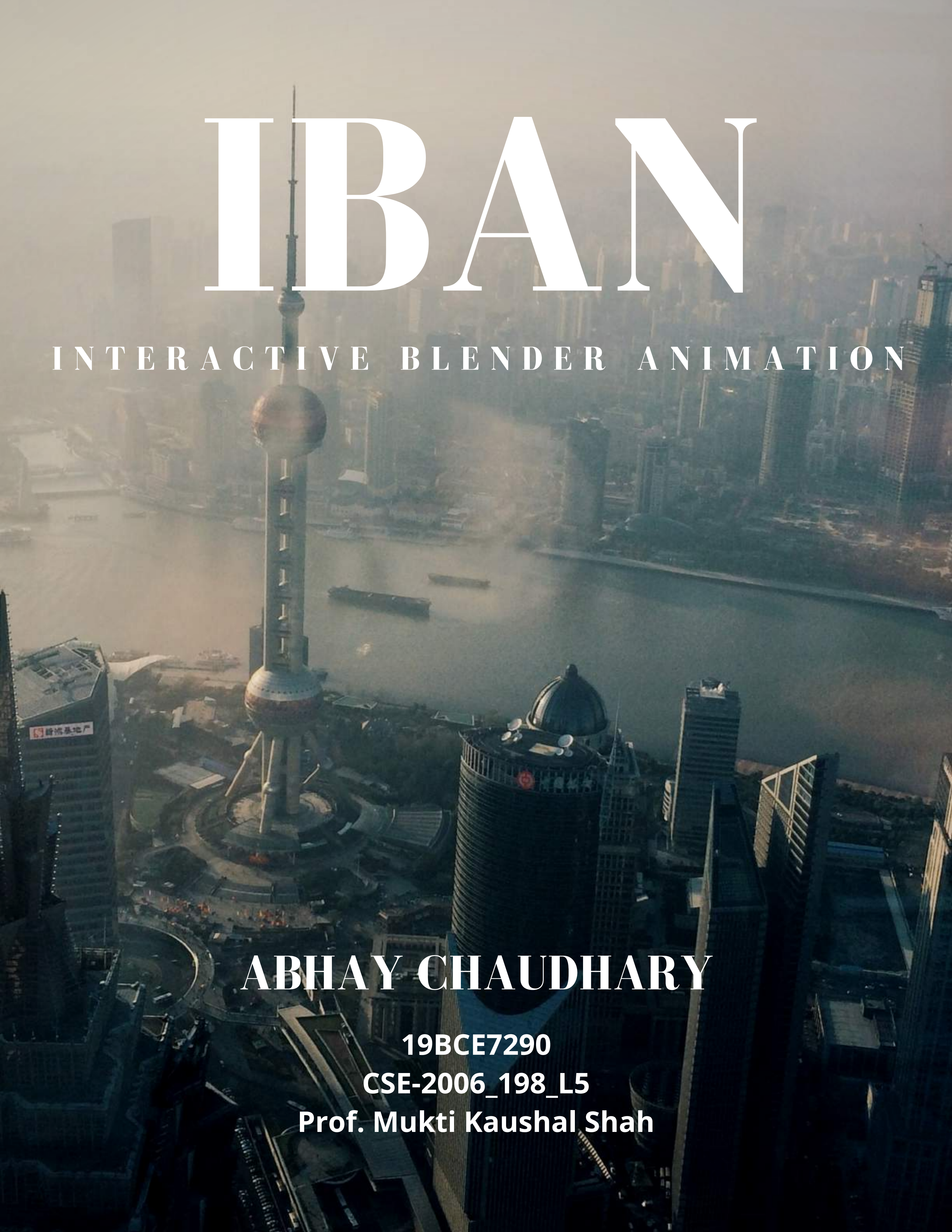


IBAN

INTERACTIVE BLENDER ANIMATION

ABHAY CHAUDHARY

19BCE7290
CSE-2006_198_L5
Prof. Mukti Kaushal Shah

Computer Graphics
(CSE-2006)

IBAN: Interactive Blender Animation

Abhay Chaudhary
(19BCE7290)
L5

Prof. Mukti Kaushal Shah
School of Computer Science and Engineering
(SCOPE)

Vellore Institute of Technology, Amravati
2020

All the source code, files, input files, output, videos everything is available on the GitHub for reference as this is a multiple files project.

The link to the GitHub repository is https://github.com/Abhayindia/IBAN-Interactive-Blender-Animation

Please have a look at the video for better understanding so

as to how the second phase of the project works.

INDEX

IBAN: Interactive Blender Animation

Abhay Chaudhary
Department of Computer Science and Engineering
Vellore Institute of Technology, Amaravati, AP, India

Abstract:

Interactive animation has its own uniqueness, and apart from games, with artistic expression of traditional animation at the same time, reinforces the audience's participation, and opera interact. Based on handheld mobile devices, making interactive animated short film by created with a lot of new features. First consideration in the animated feature for the structures of branch development, after that interactive participative forms that appropriately and cleverly designed, thereby strengthening the expression of works . When you are using Unity3D tool to achieve specific interactive tasks, such as animated elements resource limitations and interface design are also factors that in the creation of to be considered. So Verge3D is a good tool to make it work. The basic idea of the project id to help the user interact, visualise and bring the concept of human interaction with the animation which will be run.

Keywords: Blender Animation, Verge 3D, Animated works, Computer Graphics

INTRODUCTION

1.1. Blender

Blender is a free and open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, motion graphics, interactive 3D applications, virtual reality and computer games. Blender's features include 3D modeling, UV unwrapping, texturing, raster graphics editing, rigging and skinning, fluid and smoke simulation, particle simulation, soft body simulation, sculpting, animating, match moving, rendering, motion graphics, video editing, and compositing.

Features

Modeling

Primitives

Blender has support for a variety of geometric primitives, including polygon meshes, fast subdivision surface modeling, Bezier curves, NURBS surfaces, metaballs, icospheres, text, and an n-gon modeling system called B-mesh.

Modifiers

Modifiers apply non-destructive effects.

Sculpting

Blender has multi-res digital sculpting, which includes dynamic topology, maps baking, remeshing, re-symmetrize, and decimation.

Simulation

Blender can be used to simulate smoke, rain, dust, cloth, fluids, hair and rigid bodies.

Fluid simulation

The fluid simulator can be used for simulating liquids, like water hitting a cup. It uses the Lattice Boltzmann methods to simulate the fluids and allows for lots of adjusting of the amount of particles and the resolution.

The particle physics fluid simulation creates particles that follow the Smoothed-particle hydrodynamics method. Simulation tools for soft body dynamics including mesh collision detection, LBM fluid dynamics, smoke simulation, Bullet rigid body dynamics, and an ocean generator with waves. A particle system that includes support for particle-based hair. Real-time control during physics simulation and rendering.

In Blender 2.82 a new fluid sim system called mantaflow was added, replacing the old system.

Animation

Keyframed animation tools including inverse kinematics, armature (skeletal), hook, curve and lattice-based deformations, shape animations, non-linear animation, constraints, and vertex weighting.

Grease Pencil

Blender's Grease Pencil tools allow for 2D animation within a full 3D pipeline.

Rendering

Internal render engine with scanline rendering, indirect lighting, and ambient occlusion that can export in a wide variety of formats; A path tracer render engine called Cycles, which can take advantage of the GPU for rendering. Cycles supports the Open Shading Language since Blender 2.65.

EEVEE is a new physically based real-time renderer. It works both as a renderer for final frames, and as the engine driving Blender's realtime viewport for creating assets. Blender Internal was removed in 2.8.

Texture and shading

Blender allows procedural and node-based textures, as well as texture painting, projective painting, vertex painting, weight painting and dynamic painting.

Post-production

Blender also includes a non-linear video editor called the Video Sequence Editor (VSE), with support for effects like Gaussian blur, color grading, fade and wipe transitions, and other video transformations. However, there is no multi-core support for rendering video with the VSE.

Plugins/addons and scripts

Blender supports Python scripting for creation of custom tools, prototyping, game logic, importing/exporting from other formats and task automation. This allows for integration with a number of external render engines through plugins/addons.

Deprecated features

The Blender Game Engine was a built-in real-time graphics and logic engine with features such including collision detection, a dynamics engine, and programmable logic. It also allowed the creation of stand-alone, real-time applications ranging from architectural visualization to video games. In April 2018 it was removed from the upcoming Blender 2.8 release series, having long lagged behind other game engines such as the open-source Godot, and Unity. In the 2.8 announcement, the Blender team specifically mentioned the Godot engine as a suitable replacement for migrating Blender Game Engine users.

Blender Internal, a biased rasterization engine / scanline renderer used in the previous versions of Blender was also removed for the 2.80 release, in favor of the new "Eevee" renderer, a realtime PBR renderer.

1.2. Verge 3D

Verge3D uses WebGL for rendering. It incorporates components of the Three.js library and exposes its API to application developers.

Puzzles

> Application functionality can be added via JavaScript, either by writing code directly or by using Puzzles, Verge3D’s visual programming environment based on Google Blockly. Puzzles is aimed primarily at non-programmers allowing quick creation of interactive scenarios in a drag-and-drop fashion.

App Manager and web publishing

> App Manager is a lightweight web-based tool for creating, managing and publishing Verge3D projects, running on top of the local development server. Verge3D Network service integrated in the App Manager allows for publishing Verge3D applications via Amazon S3 and EC2 cloud services.

PBR

> For purposes of authoring materials, a glTF 2.0-compliant physically based rendering pipeline is offered alongside with the standard shader-based approach. PBR textures can be authored using external texturing software such as Substance Painter for which Verge3D offers the corresponding export preset. Besides the glTF 2.0 model, Verge3D supports physical materials of 3ds Max (with Autodesk Raytracer as reference), and Blender 2.80's real-time Eevee materials.

glTF and DCC software integration

> Verge3D integrates directly with Blender and Autodesk 3ds Max, enabling users to create 3D geometry, materials and animations inside the software, then export them in the JSON-based glTF format. The Sneak Preview feature allows for exporting and viewing scenes from the DCC tool environment.

Facebook 3D posts

> For Facebook publishing, Verge3D offers a specific GLB export option. The exported GLB files are displayed and can be opened in the App Manager.

Asset compression

> Exported files can optionally use LZMA compression, resulting in a reduction in file size of up to 6x.

UI and website layouts

> Interface layouts, created using external WYSIWYG editors, can be linked with Puzzles to trigger changes to a 3D scene being rendered in the browser and vice versa.

Animation

> Verge3D supports skeletal animation, including animation of bipeds and character rigs, and allows for animation of material parameters. Model parts can also be set up to be dragged by the user.

Physics

> The physics module can be linked separately to enable collision detection, dynamically moving objects, support for characters and vehicles, springs, ropes and cloth simulation. As of version 2.11, simple physics simulations can be created

and controlled without coding via Puzzles, the visual programming system used by Verge3D.

AR/VR

The 2.10 update added support for WebXR, an in-development open technology designed to enable virtual reality and augmented reality experiences to be displayed in web browsers. It works with both headsets with controllers, like the HTC Vive and Oculus Rift, and those without, like Google Cardboard. AR/VR experiences can enabled via Puzzles or JavaScript.

Code Implementation

Manifest.Json

```
{
    "name": "",
    "icons": [
        {
            "src": "/android-chrome-192x192.png",
            "sizes": "192x192",
            "type": "image/png"
        },
        {
            "src": "/android-chrome-512x512.png",
            "sizes": "512x512",
            "type": "image/png"
        }
    ],
    "theme_color": "#ffffff",
    "background_color": "#ffffff",
    "display": "standalone"
}
```

Safari-Pinned-Tab.svg

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN"
 "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg version="1.0" xmlns="http://www.w3.org/2000/svg"
 width="1000.000000pt" height="1000.000000pt" viewBox="0 0 1000.000000 1000.00
0000"
 preserveAspectRatio="xMidYMid meet">
<metadata>
Created by potrace 1.11, written by Peter Selinger 2001-2013
</metadata>
<g transform="translate(0.000000,1000.000000) scale(0.100000,-0.100000)"
fill="#000000" stroke="none">
<path d="M4830 9985 c-260 -45 -521 -151 -960 -390 -165 -90 -125 -67 -705
-400 -280 -161 -663 -379 -850 -485 -187 -106 -421 -240 -520 -298 -785 -456
-1046 -690 -1202 -1077 -23 -58 -63 -206 -77 -293 -14 -80 -18 -105 -30 -194
-3 -18 -8 -76 -11 -128 -3 -52 -8 -120 -11 -150 -10 -135 -14 -592 -17 -1780
-2 -1402 0 -1474 48 -1778 65 -402 227 -669 571 -938 45 -35 108 -82 140 -104
81 -56 196 -133 216 -145 397 -240 521 -314 708 -422 124 -72 239 -139 257
-149 17 -11 80 -47 140 -81 262 -152 830 -485 936 -549 49 -30 90 -54 93 -54
```

```
2 0 33 -18 69 -40 36 -22 67 -40 69 -40 2 0 35 -18 73 -39 151 -87 185 -105
355 -190 283 -142 511 -224 698 -250 68 -9 300 -7 360 4 284 51 582 182 1165
509 22 13 67 38 100 56 33 18 68 40 78 48 9 8 17 12 17 8 0 -4 12 1 28 12 15
11 135 81 267 157 132 76 254 145 270 155 95 55 607 347 645 368 25 13 74 42
110 62 36 20 88 50 115 65 107 59 417 242 562 331 526 326 759 568 881 914 33
95 67 236 77 320 4 30 8 57 10 60 2 3 6 39 10 80 3 41 8 91 10 110 20 199 25
552 27 1940 3 1356 1 1435 -38 1715 -52 378 -164 611 -408 849 -136 133 -276
237 -591 441 -70 46 -405 245 -411 245 -3 0 -17 8 -32 19 -20 14 -652 382
-947 551 -16 10 -169 100 -340 200 -614 362 -899 517 -1138 619 -103 44 -249
96 -300 106 -10 2 -26 6 -35 9 -9 3 -30 8 -47 11 -16 2 -41 7 -55 11 -54 12
-307 12 -380 -1z"/>
</g>
</svg>
```

Arrowleft.svg

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><!DOCTYPE svg PUBLIC "-
//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"><sv
g width="100%" height="100%" viewBox="0 0 10 10" version="1.1" xmlns="http://w
ww.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" xml:space="pres
erve" xmlns:serif="http://www.serif.com/" style="fill-rule:evenodd;clip-
rule:evenodd;stroke-linejoin:round;stroke-
miterlimit:1.41421;"><path d="M2.284,1.759l1.406,-1.403l4.466,4.455l-
4.466,4.456l-1.406,-1.403l3.06,-3.053l-3.06,-3.052Z"/></svg>
```

Interactive Architecture Scene.css

```
body {
    background-color: #FFF;
    margin: 0px;
    overflow: hidden;
    font-family: 'Raleway', sans-serif;
}

#container {
    position: absolute;
    top: 0px;
    left: 0px;
    width: 100%;
    height: 100%;
}

div#fullscreen_button {
    position: absolute;
    top: 10px;
    right: 10px;
    width: 50px;
    height: 50px;
```

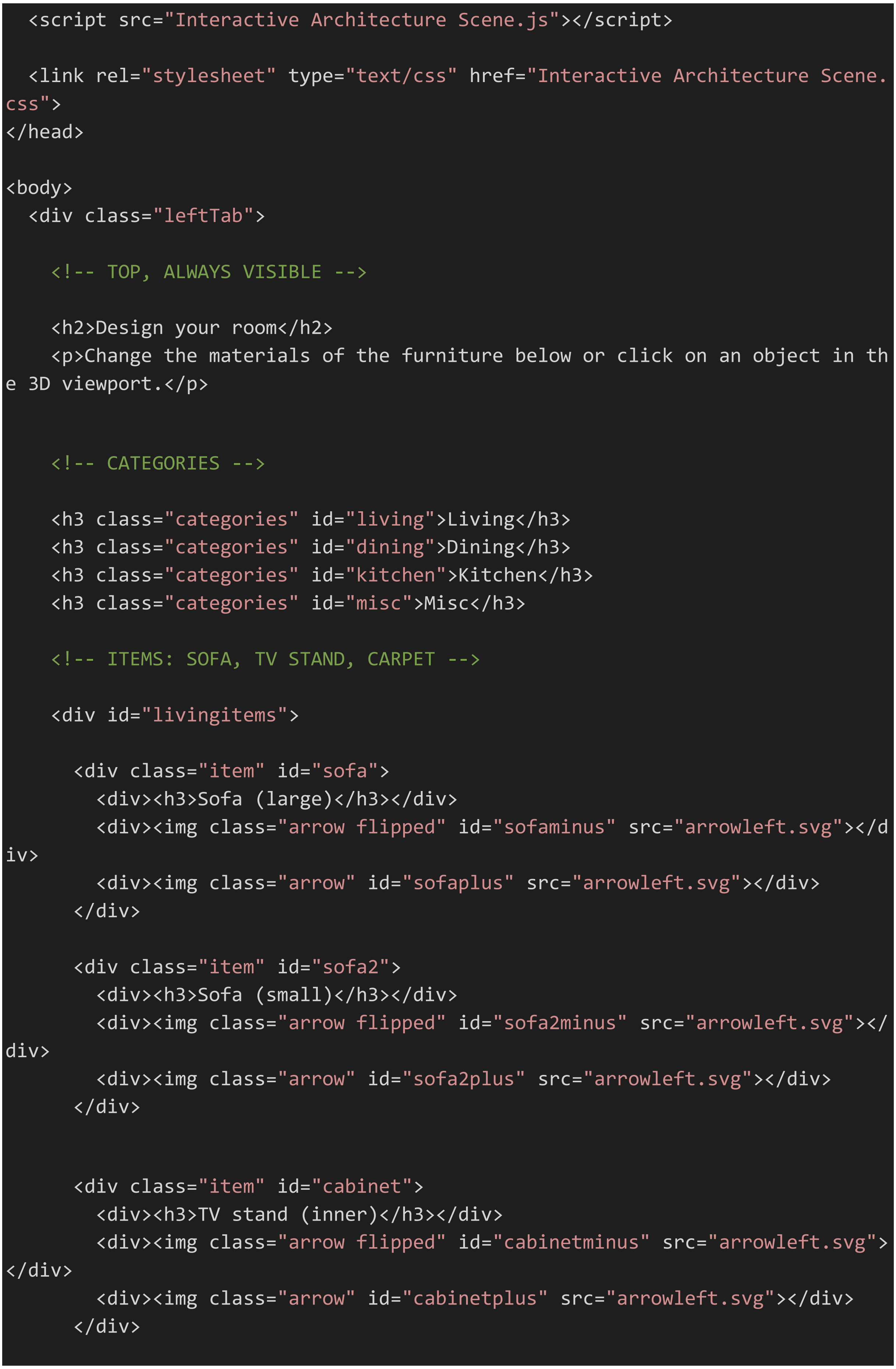
```css
    cursor: pointer;
    background-size: 100% 100%;
    display: none;
}

.fullscreen-open {
    background-image: url('media/fullscreen-open.png');
}

.fullscreen-close {
    background-image: url('media/fullscreen-close.png');
}

/* removes tap blinking on ios devices */
* { -webkit-tap-highlight-color:rgba(0,0,0,0); }


h2 {

    /* z-index: 2; */
    font-family: sans-serif;

    color: black;
   margin-top: 9px;
}

.leftTab {
    position: absolute;
    top: 0px;
    height: 100%;
    width: 25%;
    background-color: white;
    z-index: 1;
    padding: 8px;
    box-shadow: 2px 0 20px #00000069;
}

h3:hover{
    cursor:pointer;
}

.item {
    height: 75px;
    background-color: whitesmoke;
    border-radius: 3px;
    border: 2px solid #dadada;
    margin-top: 12px;
    grid-template-columns: 1fr 70px 70px;
    display: grid;
```

```css
}

.item div h3 {
    padding-top: 8px;
    padding-left: 12px;
}


.arrow {
opacity: 0.3;
    height: 50px;
    margin-top: 12px;
}

.arrow:hover {
    opacity: 0.9;
    cursor: pointer;
}

.flipped {
    -moz-transform: scaleX(-1);
        -o-transform: scaleX(-1);
        -webkit-transform: scaleX(-1);
        transform: scaleX(-1);
        filter: FlipH;
        -ms-filter: "FlipH";
}

#livingitems {
    display: none;
}

#diningitems {
    display: none;
}

#kitchenitems {
    display: none;
}

#miscitems {
    display: none;
}

.back {
    cursor: pointer;
    text-decoration: underline;
    color: gray;
```

```
}
```

Interactive Architecture Scene.html

```
<!DOCTYPE html>
<!--
__VERGE3D_PLAYER__ - delete this comment if you want to edit the player's code
-->
<html lang="en">
<head>
  <title>Verge3D Web Interactive</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">

  <!-- Search Engines -->
  <meta name="description" content="Interactive 3D Web application made with Verge3D. Immerse yourself in amazing graphics experience offered by state-of-the WebGL and HTML5 technologies.">
  <!-- Twitter -->
  <meta name="twitter:card" content="summary">
  <meta name="twitter:title" content="Verge3D Web Interactive">
  <meta name="twitter:description" content="Interactive 3D Web application made with Verge3D. Immerse yourself in amazing graphics experience offered by state-of-the WebGL and HTML5 technologies.">
  <meta name="twitter:image:src" content="https://cdn.soft8soft.com/images/player_socials.jpg">
  <!-- Open Graph -->
  <meta property="og:title" content="Verge3D Web Interactive">
  <meta property="og:description" content="Interactive 3D Web application made with Verge3D. Immerse yourself in amazing graphics experience offered by state-of-the WebGL and HTML5 technologies.">
  <meta property="og:image" content="https://cdn.soft8soft.com/images/player_socials.jpg">
  <meta property="og:type" content="website">

  <!-- favicons -->
  <link rel="apple-touch-icon" sizes="180x180" href="media/apple-touch-icon.png">
  <link rel="icon" type="image/png" sizes="32x32" href="media/favicon-32x32.png">
  <link rel="icon" type="image/png" sizes="16x16" href="media/favicon-16x16.png">
  <link rel="manifest" href="media/manifest.json">
  <link rel="mask-icon" href="media/safari-pinned-tab.svg" color="#5bbad5 ">
  <meta name="theme-color" content="#ffffff ">

  <script src="v3d.js"></script>
```

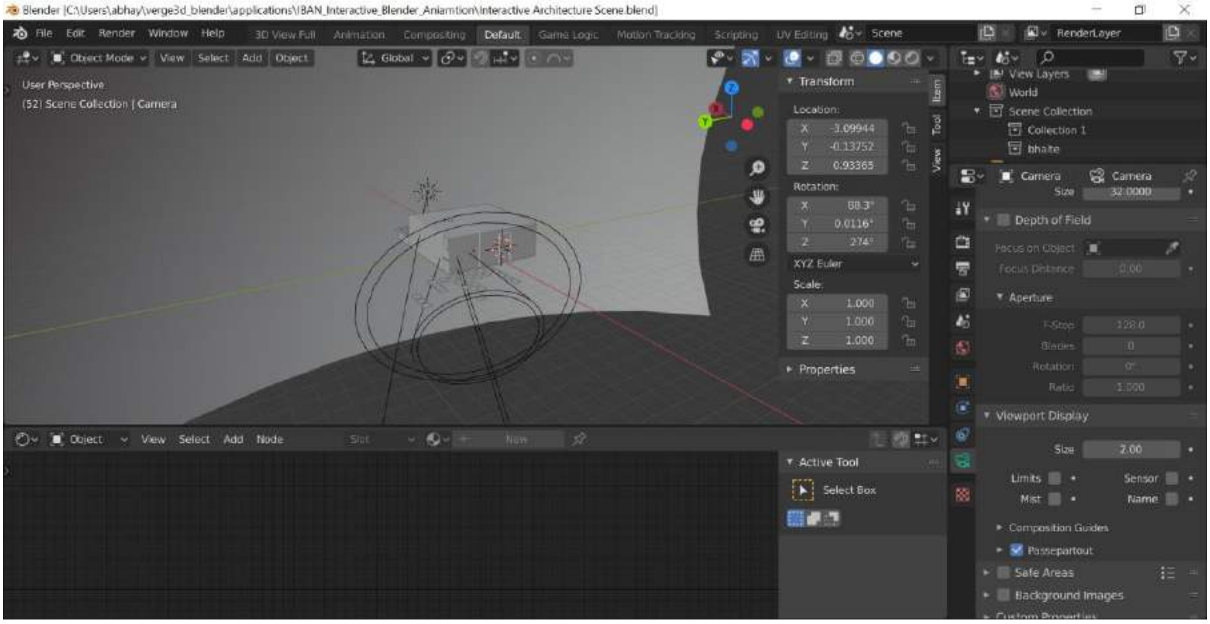


```
  <script src="Interactive Architecture Scene.js"></script>

  <link rel="stylesheet" type="text/css" href="Interactive Architecture Scene.
css">
</head>

<body>
  <div class="leftTab">

    <!-- TOP, ALWAYS VISIBLE -->

    <h2>Design your room</h2>
    <p>Change the materials of the furniture below or click on an object in th
e 3D viewport.</p>


    <!-- CATEGORIES -->

    <h3 class="categories" id="living">Living</h3>
    <h3 class="categories" id="dining">Dining</h3>
    <h3 class="categories" id="kitchen">Kitchen</h3>
    <h3 class="categories" id="misc">Misc</h3>

    <!-- ITEMS: SOFA, TV STAND, CARPET -->

    <div id="livingitems">

      <div class="item" id="sofa">
        <div><h3>Sofa (large)</h3></div>
        <div><img class="arrow flipped" id="sofaminus" src="arrowleft.svg"></d
iv>
        <div><img class="arrow" id="sofaplus" src="arrowleft.svg"></div>
      </div>

      <div class="item" id="sofa2">
        <div><h3>Sofa (small)</h3></div>
        <div><img class="arrow flipped" id="sofa2minus" src="arrowleft.svg"></
div>
        <div><img class="arrow" id="sofa2plus" src="arrowleft.svg"></div>
      </div>


      <div class="item" id="cabinet">
        <div><h3>TV stand (inner)</h3></div>
        <div><img class="arrow flipped" id="cabinetminus" src="arrowleft.svg">
</div>
        <div><img class="arrow" id="cabinetplus" src="arrowleft.svg"></div>
      </div>
      <div class="item" id="cabinet2">
        <div><h3>TV stand (outter)</h3></div>
        <div><img class="arrow flipped" id="cabinet2minus" src="arrowleft.svg"></div>
        <div><img class="arrow" id="cabinet2plus" src="arrowleft.svg"></div>
      </div>

      <div class="item" id="sofa">
        <div><h3>Carpet</h3></div>
        <div><img class="arrow flipped" id="carpetminus" src="arrowleft.svg"></div>
        <div><img class="arrow" id="carpetplus" src="arrowleft.svg"></div>
      </div>
    <p class="back">go back</p>
    </div>

    <!-- ITEMS: TABLE, LAMP, CHAIRS -->

    <div id="diningitems">

      <div class="item" id="table">
        <div><h3>Table</h3></div>
        <div><img class="arrow flipped" id="tableminus" src="arrowleft.svg"></div>
        <div><img class="arrow" id="tableplus" src="arrowleft.svg"></div>
      </div>

      <div class="item" id="chairsseat">
        <div><h3>Chairs (seat)</h3></div>
        <div><img class="arrow flipped" id="chairseatminus" src="arrowleft.svg"></div>
        <div><img class="arrow" id="chairseatplus" src="arrowleft.svg"></div>
      </div>


      <div class="item" id="chairframe">
        <div><h3>Chairs (frame)</h3></div>
        <div><img class="arrow flipped" id="chairframeminus" src="arrowleft.svg"></div>
        <div><img class="arrow" id="chairframeplus" src="arrowleft.svg"></div>
      </div>

      <div class="item" id="lamp">
        <div><h3>Lamp</h3></div>
        <div><img class="arrow flipped" id="lampminus" src="arrowleft.svg"></div>
        <div><img class="arrow" id="lampplus" src="arrowleft.svg"></div>
      </div>
      <p class="back">go back</p>
    </div>


    <!-- ITEMS: KITCHEN -->

     <div id="kitchenitems">

      <div class="item" id="kitchentop">
        <div><h3>Cabinet (top)</h3></div>
        <div><img class="arrow flipped" id="kitchentopminus" src="arrowleft.svg"></div>
        <div><img class="arrow" id="kitchentopplus" src="arrowleft.svg"></div>
      </div>

      <div class="item" id="countertop">
        <div><h3>Countertop</h3></div>
        <div><img class="arrow flipped" id="countertopminus" src="arrowleft.svg"></div>
        <div><img class="arrow" id="countertopplus" src="arrowleft.svg"></div>
      </div>


      <div class="item" id="kitchenbottom">
        <div><h3>Cabinet (bottom)</h3></div>
        <div><img class="arrow flipped" id="kitchenbottomminus" src="arrowleft.svg"></div>
        <div><img class="arrow" id="kitchenbottomplus" src="arrowleft.svg"></div>
      </div>

    <p class="back">go back</p>
    </div>

    <!-- ITEMS: MISC -->

     <div id="miscitems">

      <div class="item" id="kitchentop">
        <div><h3>Floor</h3></div>
        <div><img class="arrow flipped" id="floorminus" src="arrowleft.svg"></div>
        <div><img class="arrow" id="floorplus" src="arrowleft.svg"></div>
      </div>


    <p class="back">go back</p>
    </div>
```

```
  </div>
  <div id="container"></div>
  <div id="fullscreen_button" class="fullscreen-
open" title="Toggle fullscreen mode"></div>
</body>

</html>
```

Input Blender Files

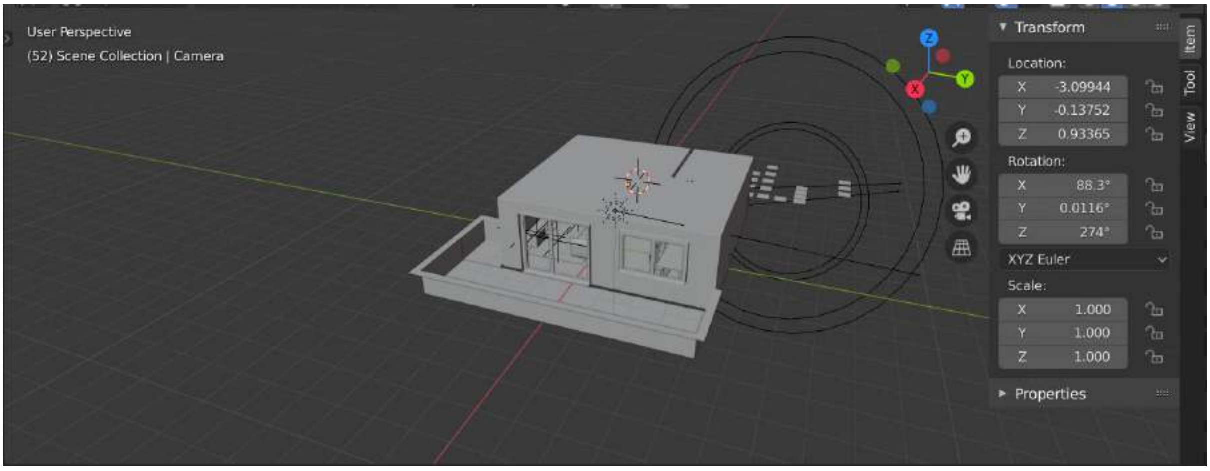
Blender File 1:



Blender [C:\Users\abhay\verge3d_blender\applications\IBAN_Interactive_Blender_Aniamtion\Interactive Architecture Scene.blend]

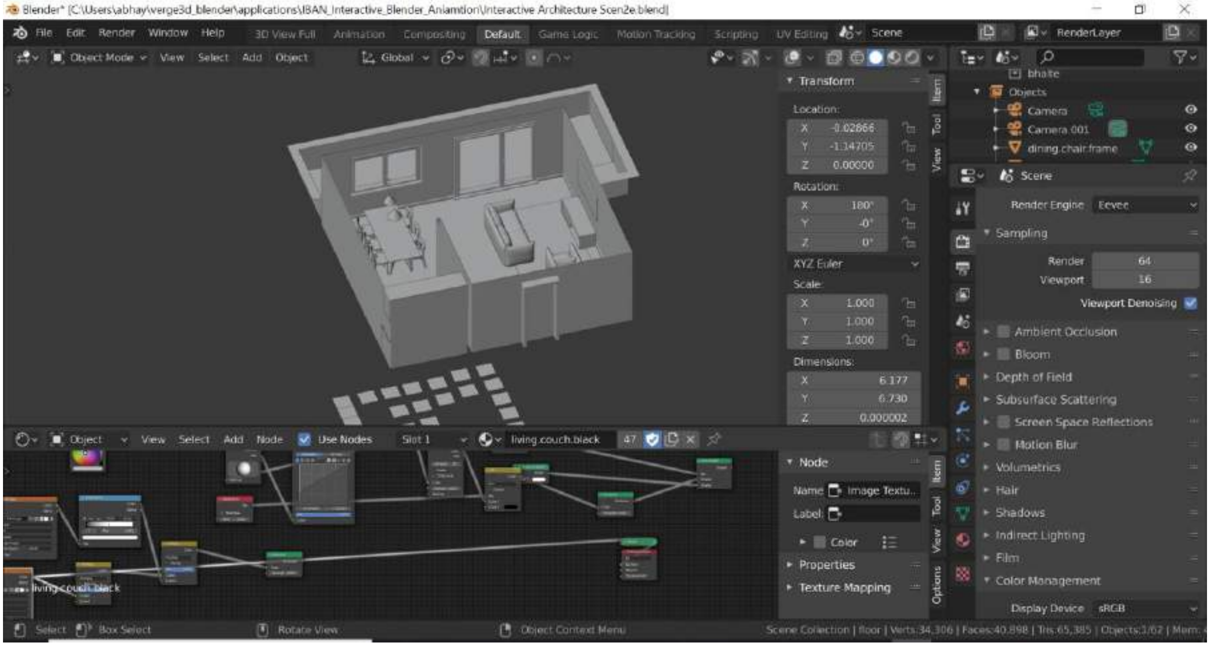
Location:
X -3.09944
Y -0.13752
Z 0.93365
Rotation:
X 88.3°
Y 0.0116°
Z 274°
XYZ Euler
Scale:
X 1.000
Y 1.000
Z 1.000

Camera Size 32.0000
Size 2.00



Location:
X -3.09944
Y -0.13752
Z 0.93365
Rotation:
X 88.3°
Y 0.0116°
Z 274°
XYZ Euler
Scale:
X 1.000
Y 1.000
Z 1.000

Blender File 2:



Blender* [C:\Users\abhay\verge3d_blender\applications\IBAN_Interactive_Blender_Aniamtion\Interactive Architecture Scen2e.blend]

Transform

Location:
X -0.02866
Y -1.14705
Z 0.00000

Rotation:
X 180°
Y -0°
Z 0°
XYZ Euler

Scale:
X 1.000
Y 1.000
Z 1.000

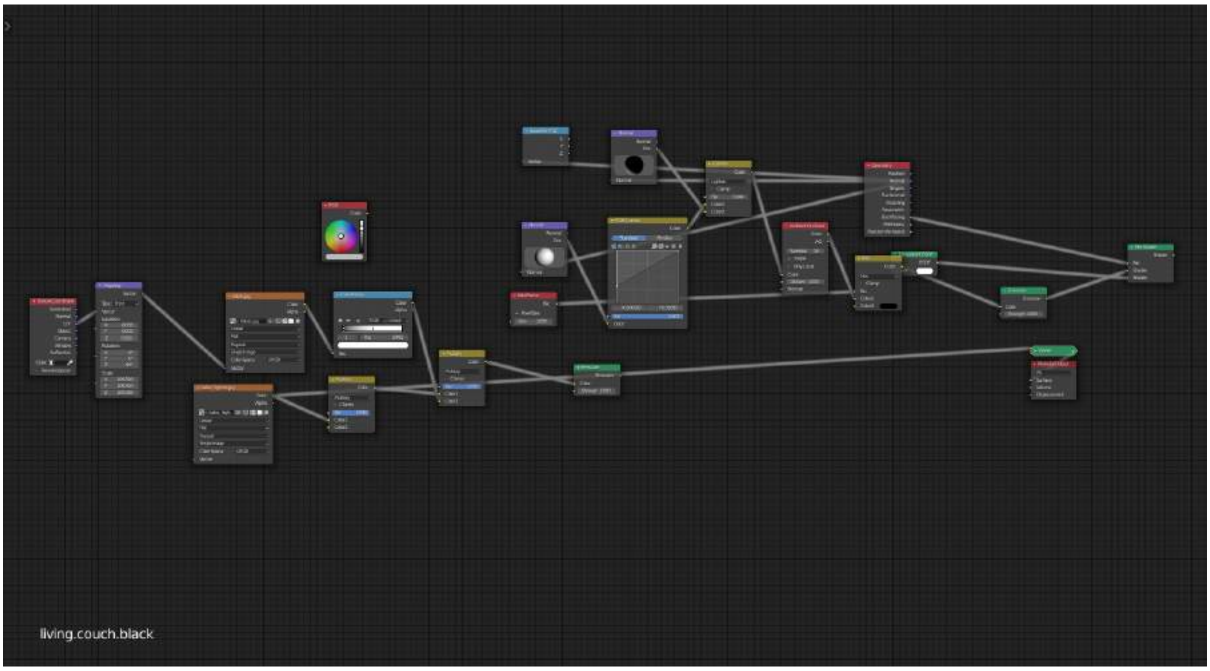
Dimensions:
X 6.177
Y 6.730
Z 0.000002

Node
Name: Image Textu...
Label:

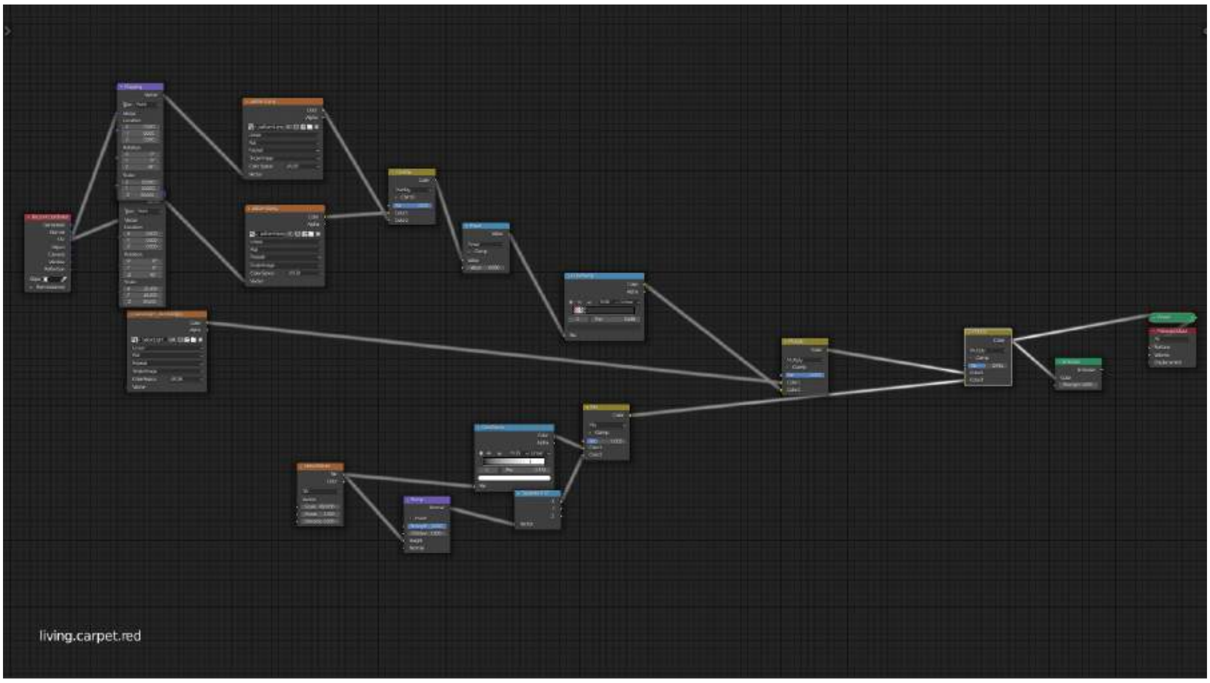
Scene
Render Engine: Eevee
Sampling
Render 64
Viewport 16
Viewport Denoising ☑

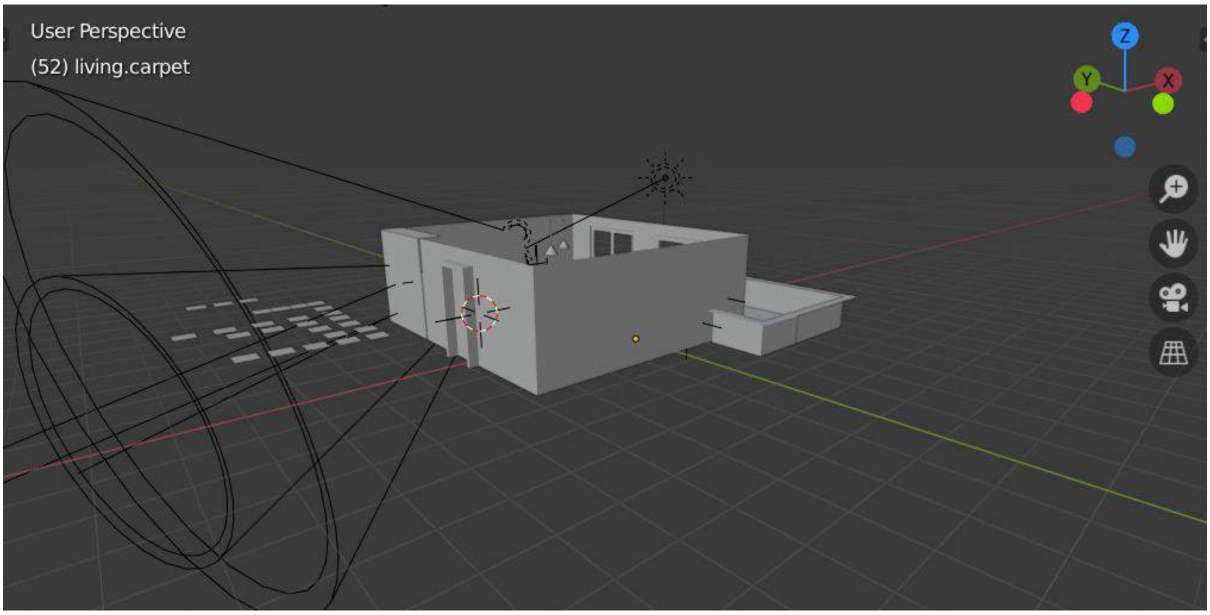
Display Device: sRGB

Scene Collection | floor | Verts:34,306 | Faces:40,898 | Tris:65,385 | Objects:1/62 | Mem:

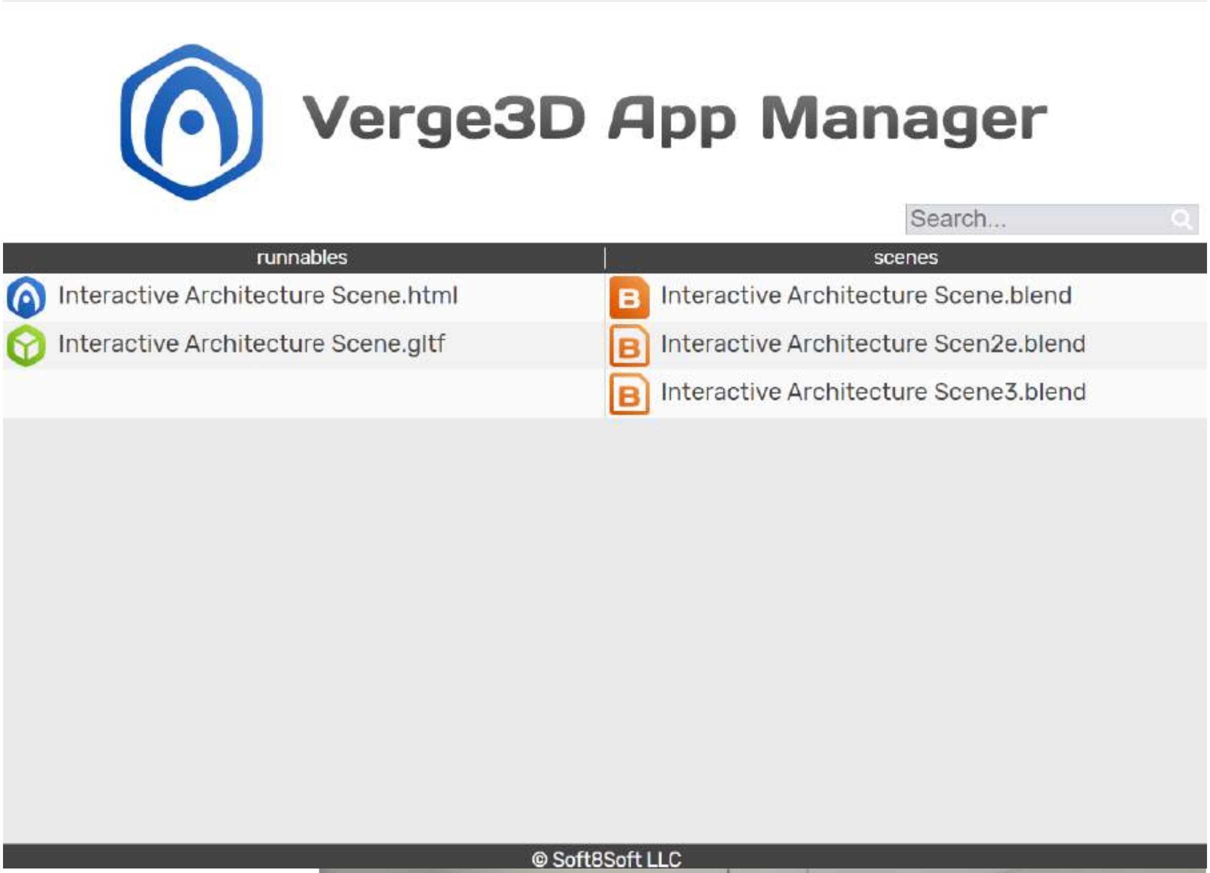


Blender File 3:





Implementation on Verge 3D App Manager

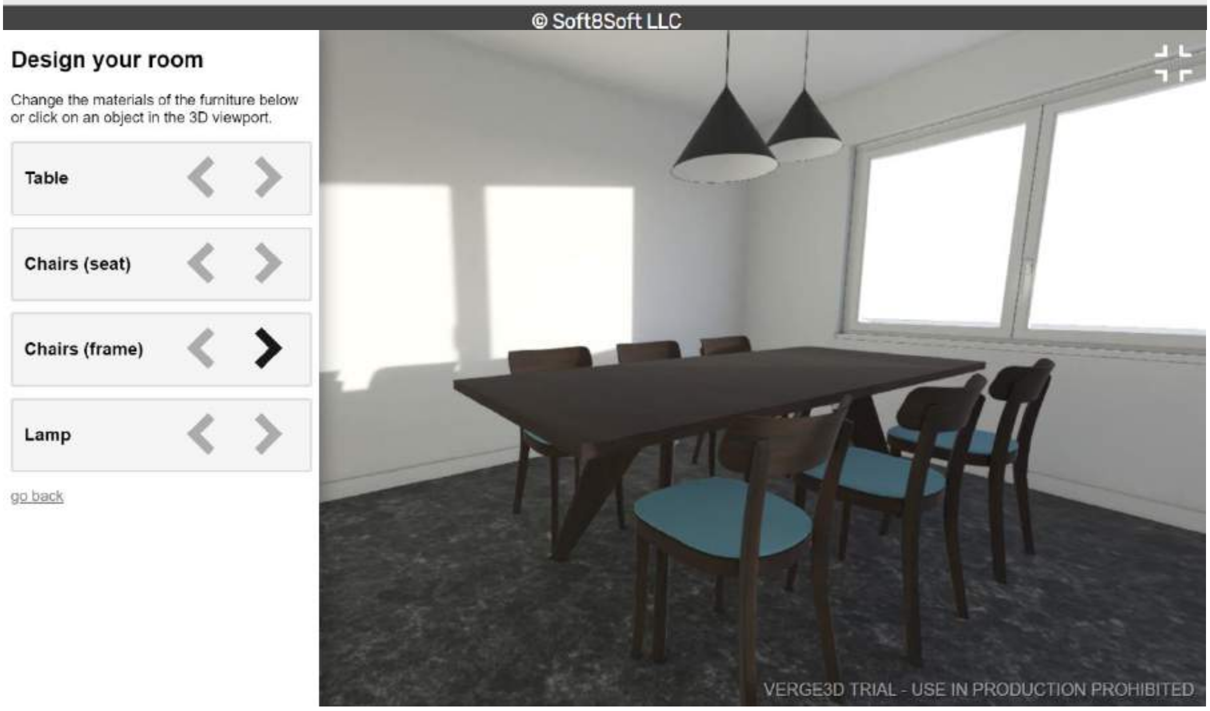


Verge3D App Manager

Search...

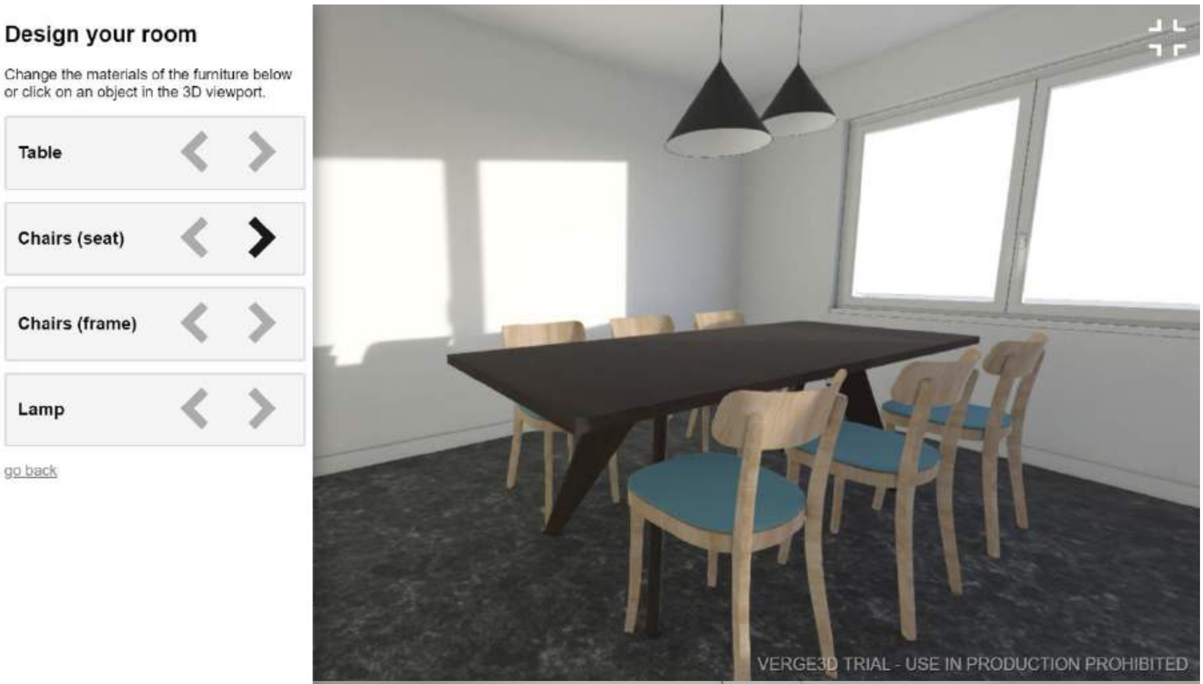
runnables	scenes
Interactive Architecture Scene.html	Interactive Architecture Scene.blend
Interactive Architecture Scene.gltf	Interactive Architecture Scen2e.blend
	Interactive Architecture Scene3.blend

© Soft8Soft LLC



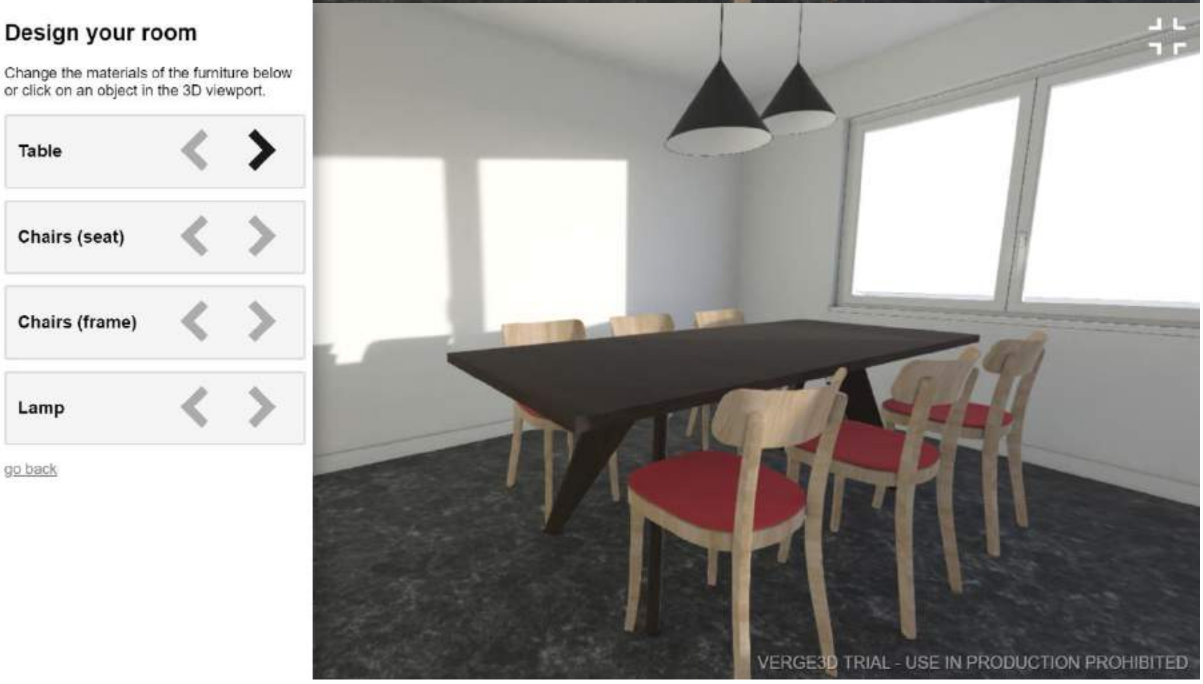
Design your room

Change the materials of the furniture below or click on an object in the 3D viewport.



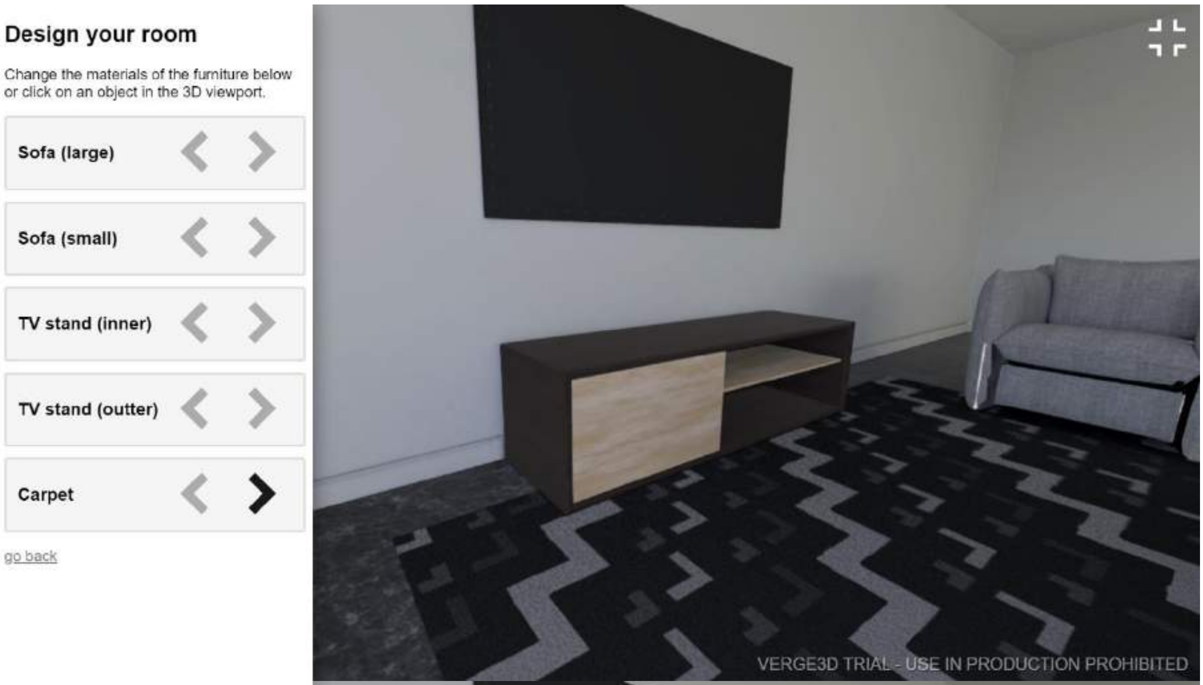
Design your room

Change the materials of the furniture below or click on an object in the 3D viewport.



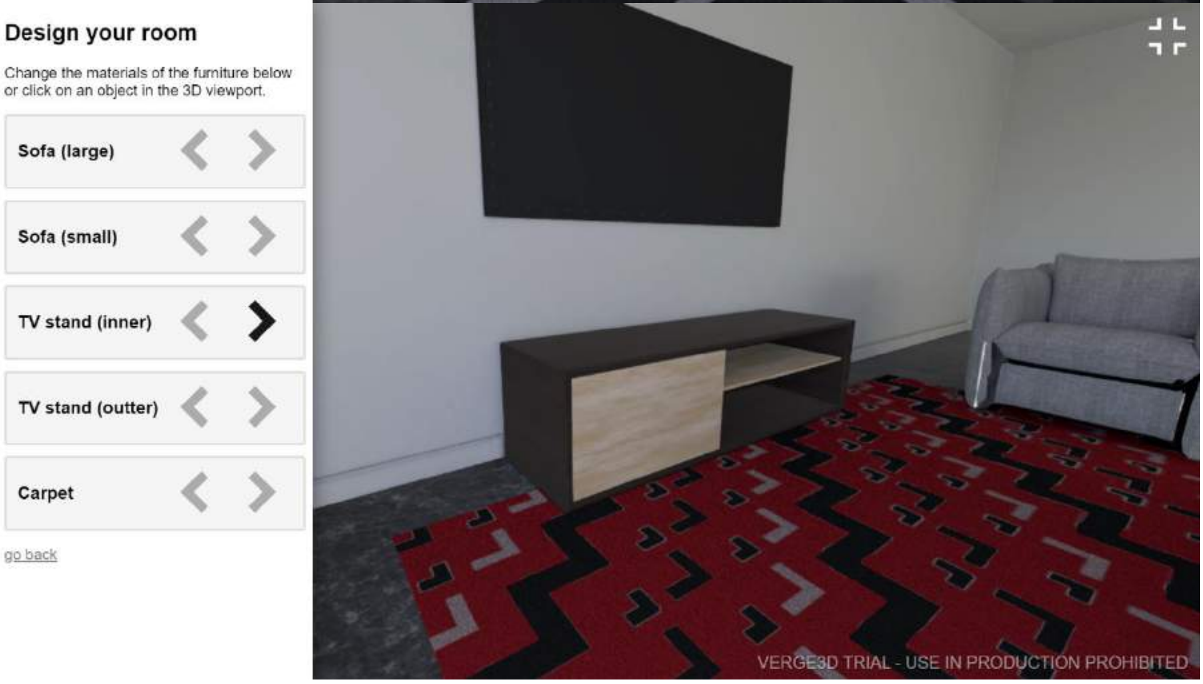
Design your room

Change the materials of the furniture below or click on an object in the 3D viewport.



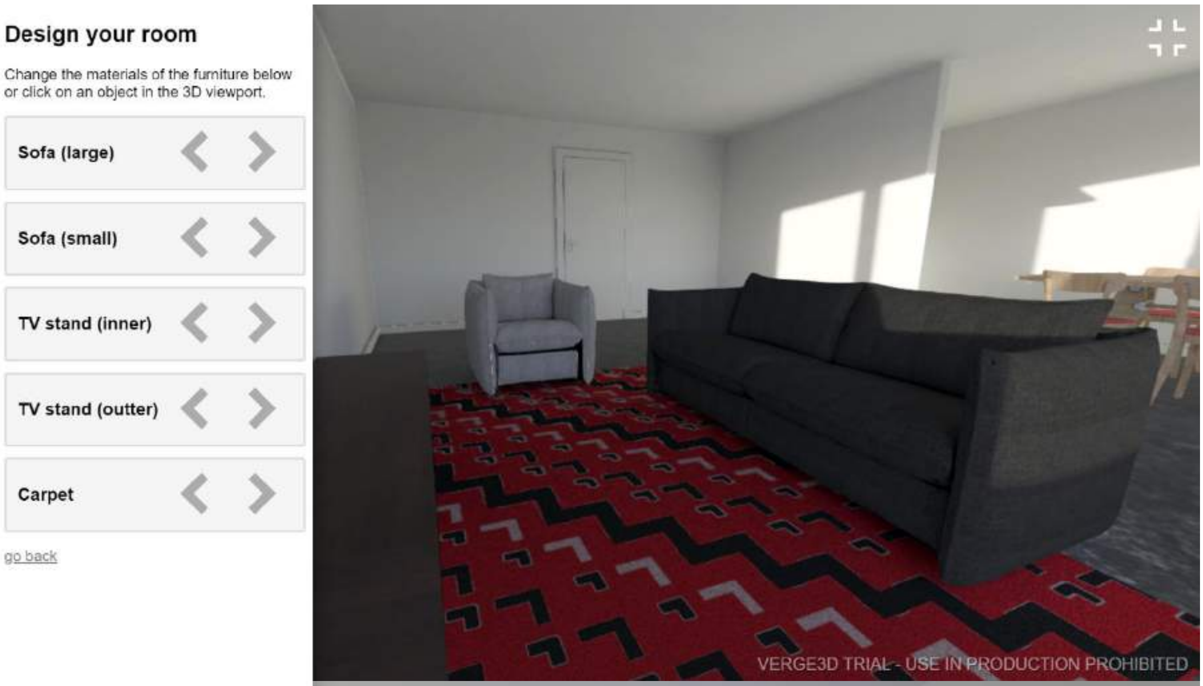
Design your room

Change the materials of the furniture below or click on an object in the 3D viewport.



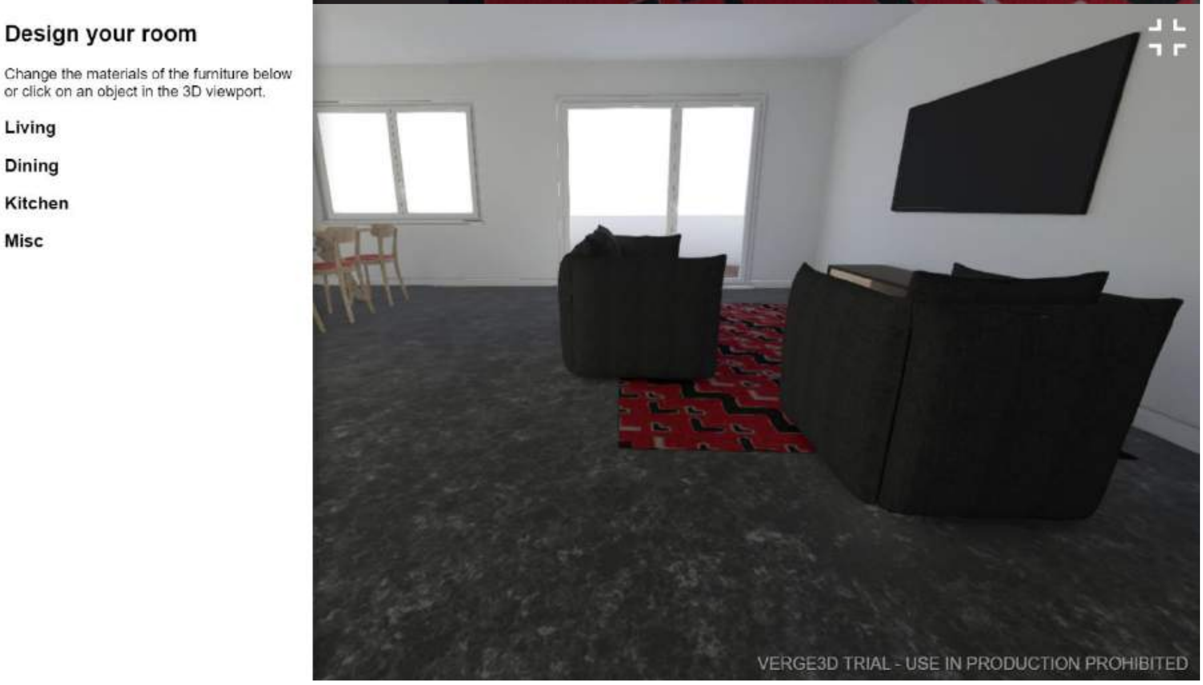
Design your room

Change the materials of the furniture below or click on an object in the 3D viewport.



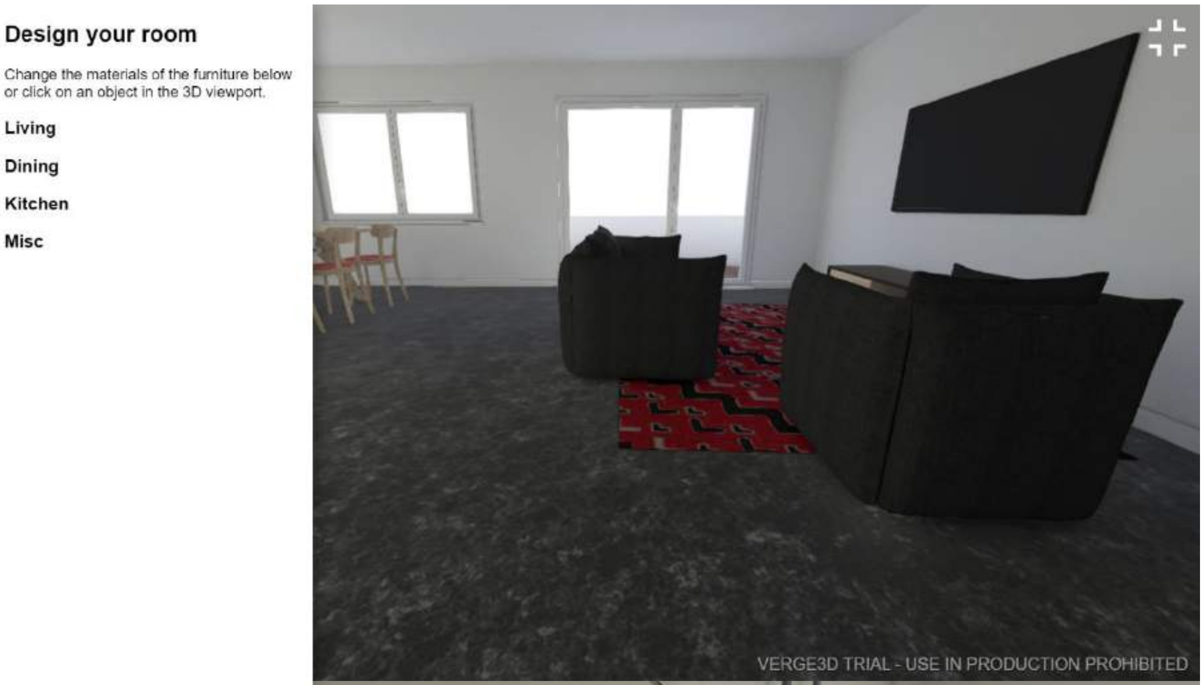
Design your room

Change the materials of the furniture below or click on an object in the 3D viewport.



Design your room

Change the materials of the furniture below or click on an object in the 3D viewport.



Design your room

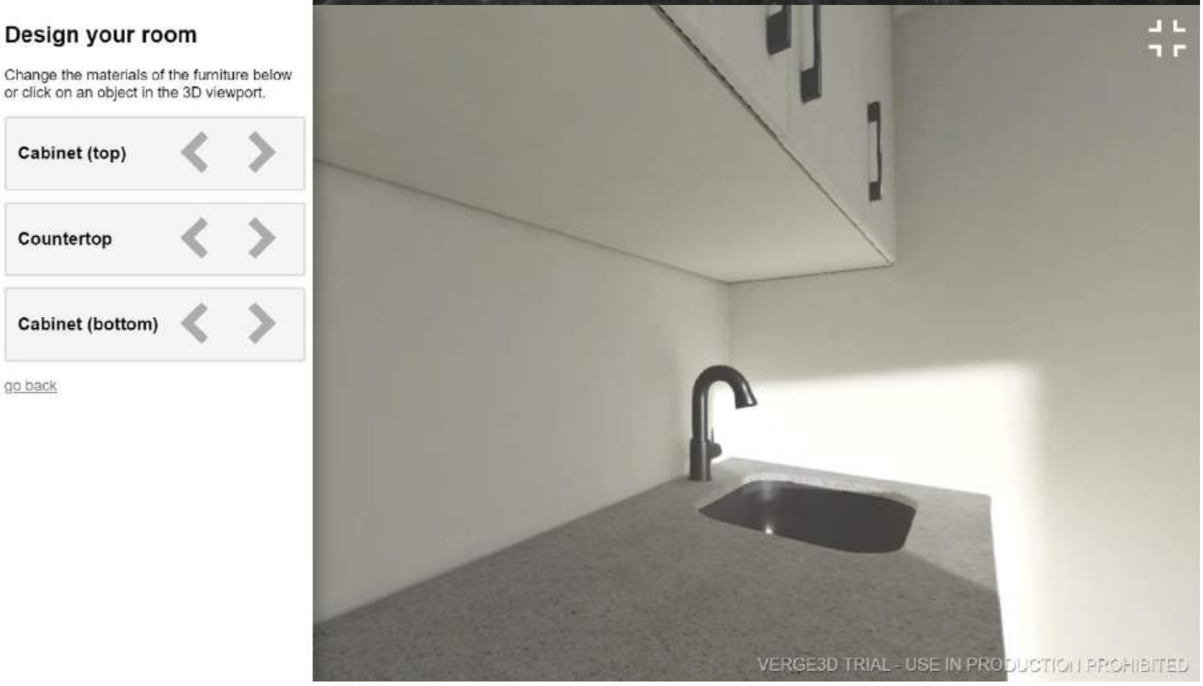
Change the materials of the furniture below or click on an object in the 3D viewport.

Living

Dining

Kitchen

Misc



Design your room

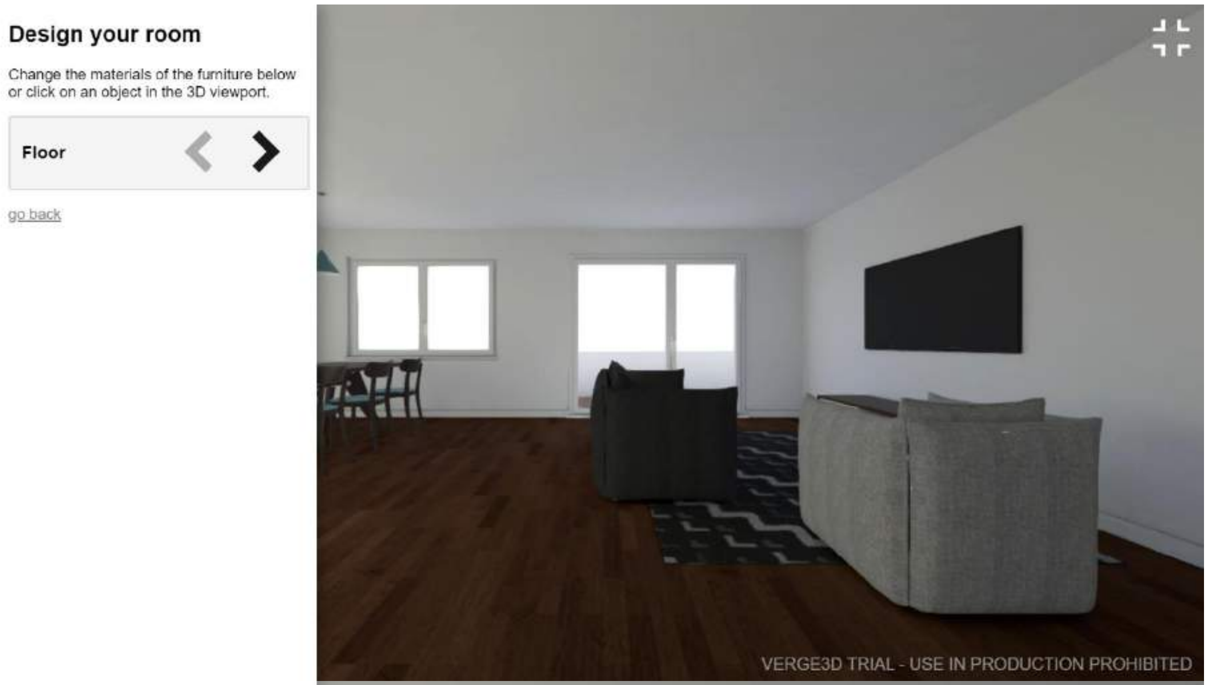
Change the materials of the furniture below or click on an object in the 3D viewport.

Cabinet (top) < >

Countertop < >

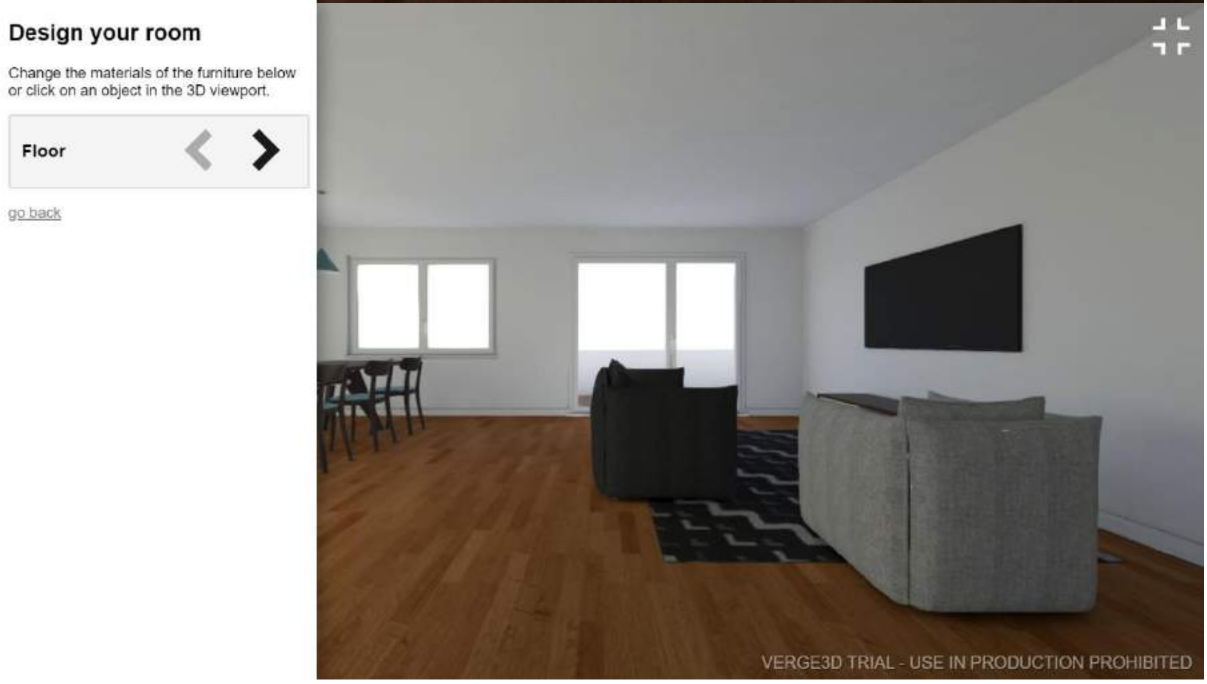
Cabinet (bottom) < >

go back



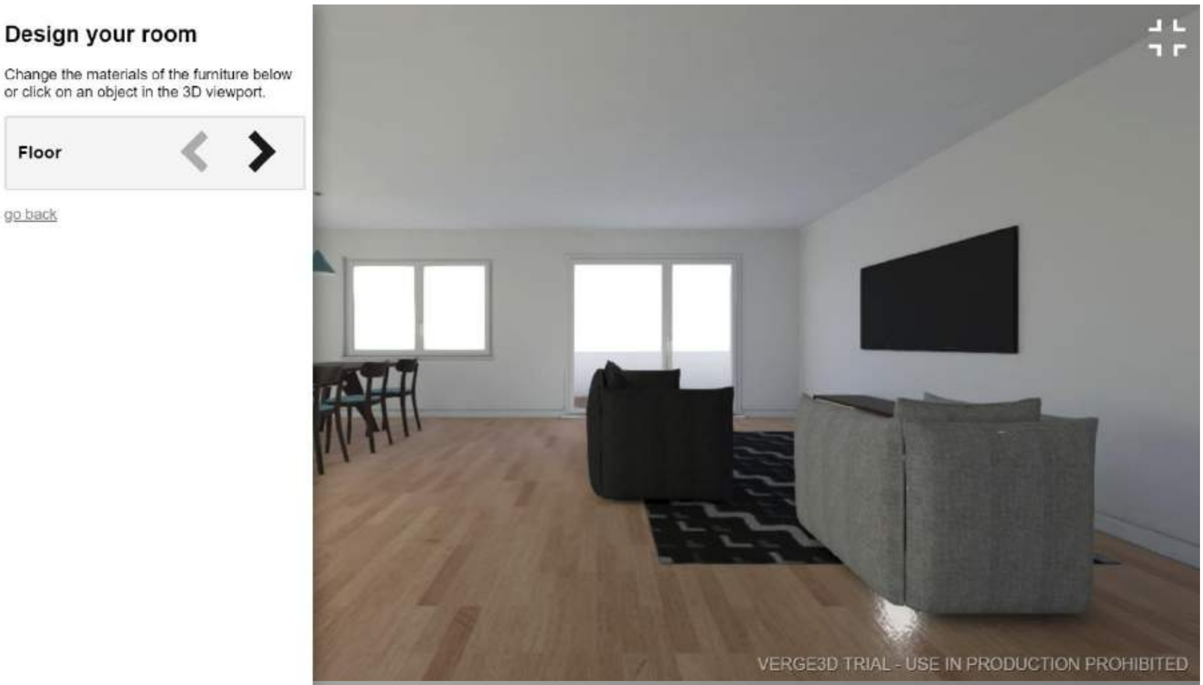
Design your room

Change the materials of the furniture below or click on an object in the 3D viewport.



Design your room

Change the materials of the furniture below or click on an object in the 3D viewport.

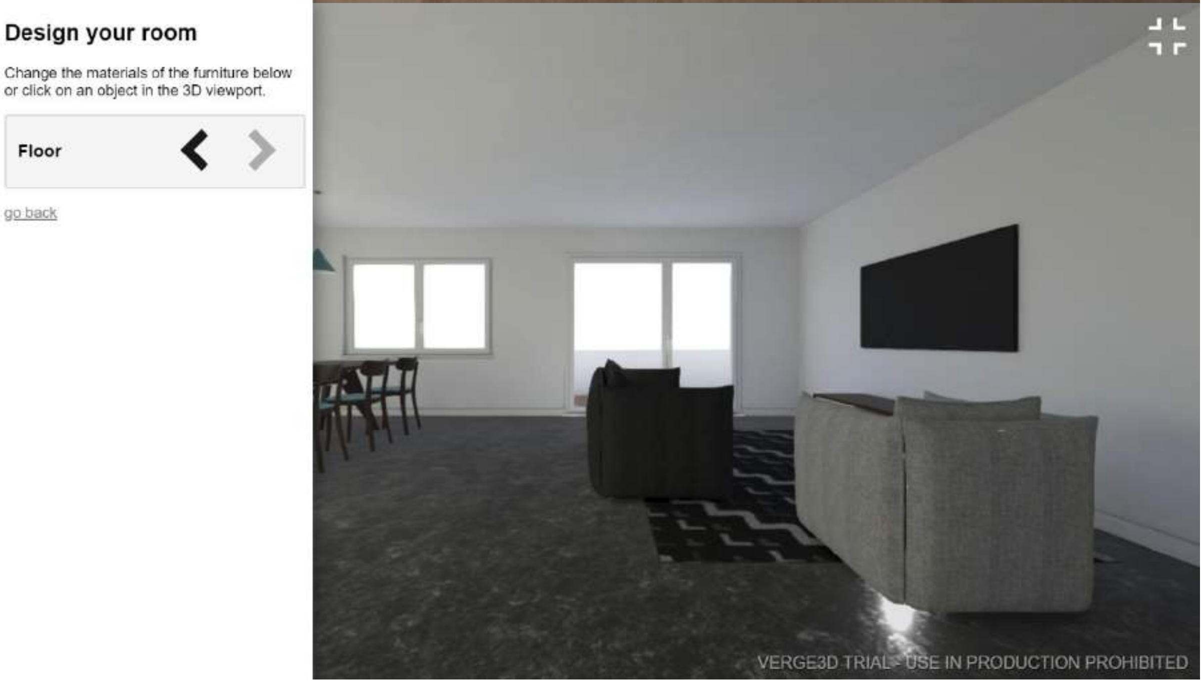


Design your room

Change the materials of the furniture below or click on an object in the 3D viewport.

Floor

go back

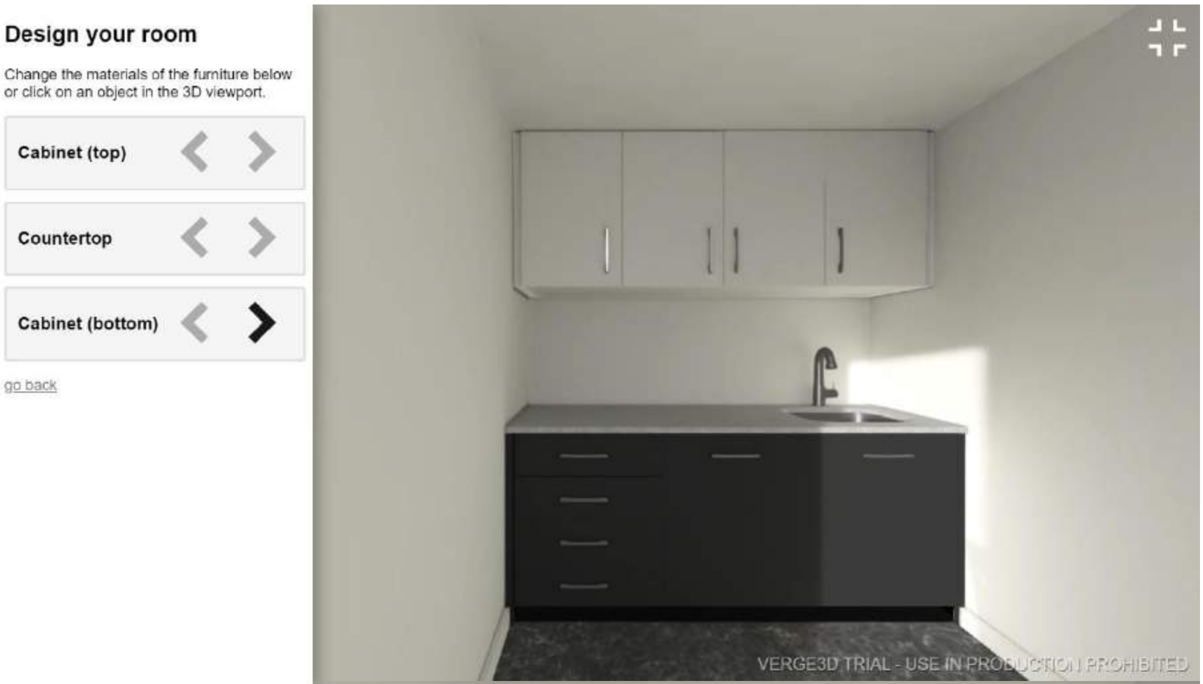


Design your room

Change the materials of the furniture below or click on an object in the 3D viewport.

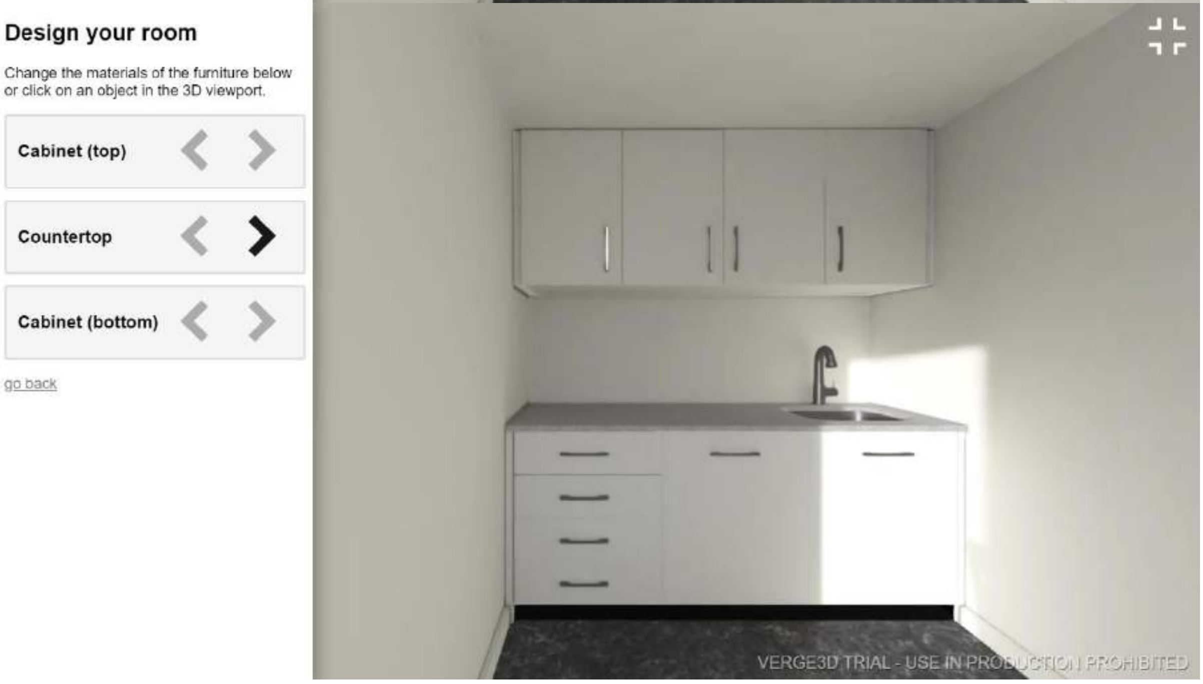
Floor

go back



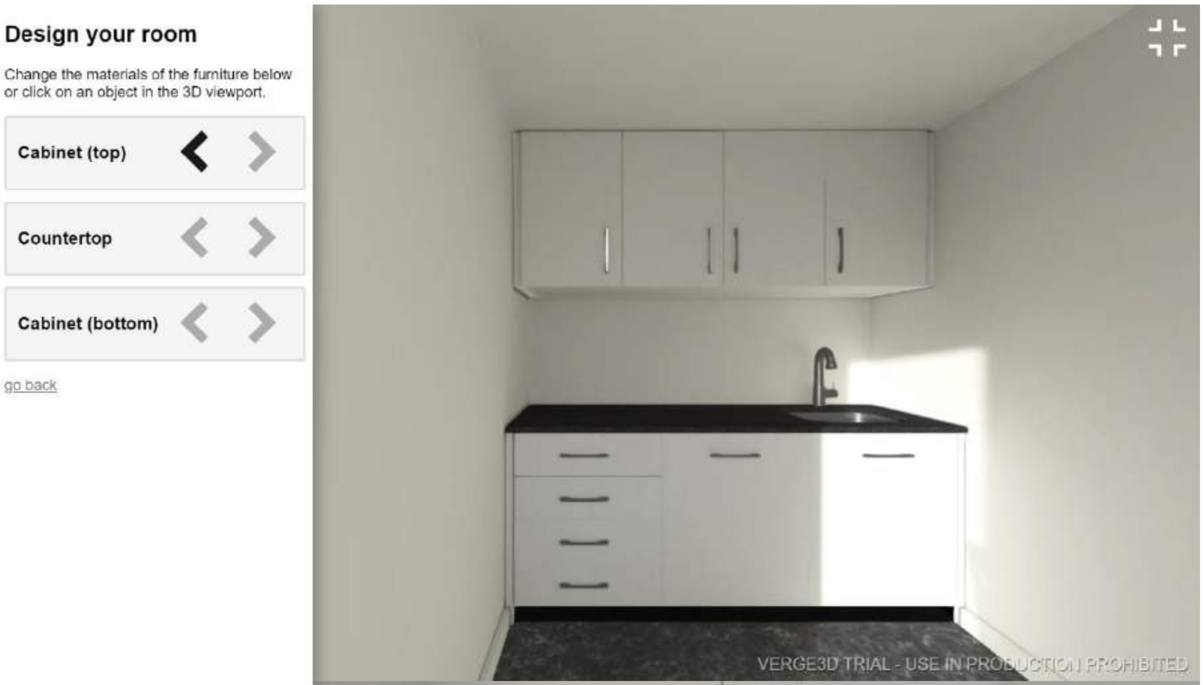
Design your room

Change the materials of the furniture below or click on an object in the 3D viewport.



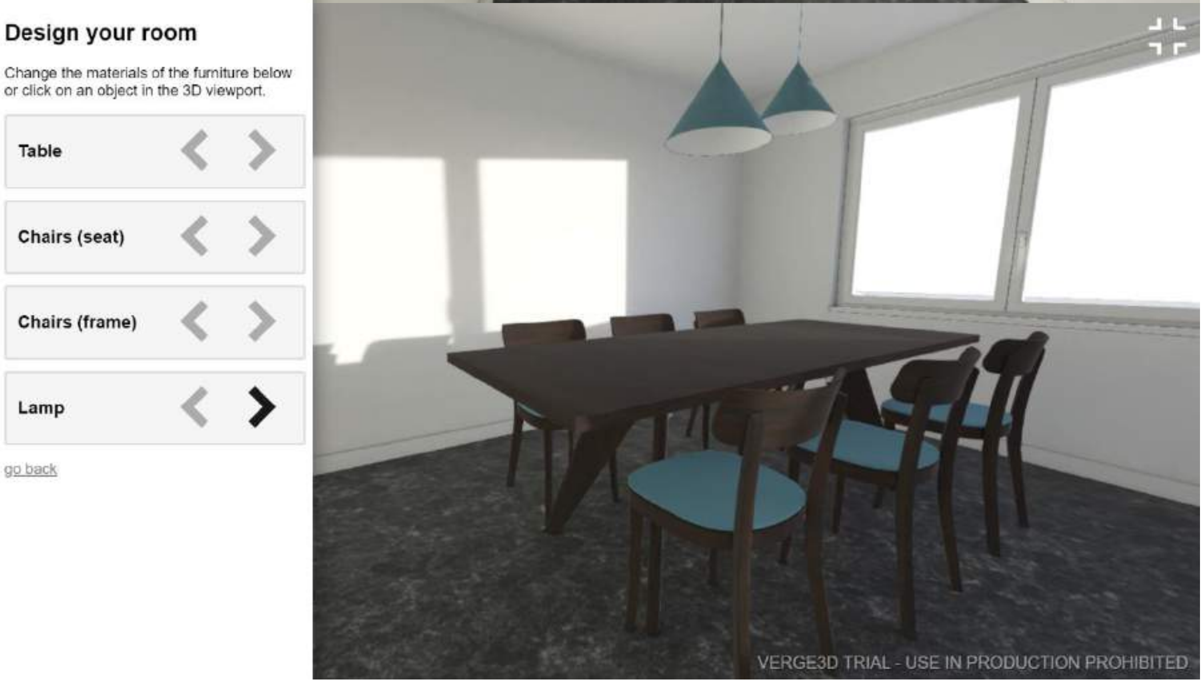
Design your room

Change the materials of the furniture below or click on an object in the 3D viewport.



Design your room

Change the materials of the furniture below or click on an object in the 3D viewport.



Design your room

Change the materials of the furniture below or click on an object in the 3D viewport.

References:

[1] E. ALLEN and K. MURDOCK, Body Language: Advance 3D Character Rigging, First Edit. Cybex. Wiley Publishing, INC, 2008.

[2] I. Baran and J. Popovic, “Automatic Rigging and Animation of 3D Characters,” ACM Trans. Graph., vol. 26, no. 3, p. 8, 2007.

[3] Z. Bhatti, “Procedural Model of Horse Simulation,” in 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry (VRCAI ’13)., 2013, pp. 139–146.

[4] Z. Bhatti and A. Shah, “Widget based automated rigging of bipedal character with custom manipulators,” Proc. 11th ACM SIGGRAPH Int. Conf. Virtual-Reality Contin. its Appl. Ind. -VRCAI ’12, p. 337, 2012.

[5] Z. Bhati, A. Shah, A. Waqas, H. Abid, and M. Malik, “Template based Procedural Rigging of Quadrupeds with Custom Manipulators,” in International Conference on Advanced Computer Science Applications and Technologies, 2013, pp. 259–264.

[6] P. Liu, F. Wu, W. Ma, R. Liang, and M. Ouhyoung, “Automatic Animation Skeleton Construction Using Repulsive Force Field,” in In Computer Graphics and Applications, 2003. Proceedings. 11th Pacific Conference on, 2003, pp. 409–413.

[7] S. Katz and A. Tal, “Hierarchical mesh decomposition using fuzzy clustering and cuts,” ACM Trans. Graph., vol. 22, no. 3, p. 954, Jul. 2003.

[8] L. Wade and R. E. Parent, “Automated generation of control skeletons for use in animation,” Vis. Comput., vol. 18, no. 2, pp. 97–110, Mar. 2002.

[9]N. Pantuwong and M. Sugimoto, “Skeleton growing_: an algorithm to extract a curve skeleton from a pseudonormal vector field,” Vis. Comput. Springer, vol. 1, 2012.

[10] N. Pantuwong and M. Sugimoto, “A fully automatic rigging algorithm for 3D character animation,” SIGGRAPH Asia 2011 Posters - SA ’11, p. 1, 2011.

[11] I. Baran and J. Popovic, “Automatic rigging and animation of 3D characters,” ACM Trans. Graph., vol. 26, no. 3, p. 72, Jul. 2007.

[12] S. Capell, M. Burkhart, B. Curless, T. Duchamp, and Z. Popovi, “Physically Based Rigging for Deformable Characters,” Eurographics/ ACM SIGGRAPH Symp. Comput. Animait., no. July, pp. 29–31, 2005.

[13] L. Moccozet, F. Dellas, and N. M. Thalmann, “Animatable Human Body Model Reconstruction from 3D Scan Data using Templates,” in In Proceedings of Workshop on Modelling and Motion Capture Techniques for Virtual Environments, CAPTECH ., 2004, pp. 73–79.

[14] D. Anderson, J. L. Frankel, J. Marks, A. Agarwala, P. Beardsley, D. Leigh, K. Ryall, E. Sullivan, and J. S. Yedidia, “Tangible Interaction + Graphical Interpretation_: A New Approach to 3D Modeling,” 2000.

[15] B. Liu and T. a. Davis, “A hybrid control scheme for facial rigging,” in Proceedings of The 18th International Conference on Computer Game (CGAMES’2013) USA, 2013, pp. 164–167.

[16] M. B. Nendya, E. M. Yuniarno, and S. Gandang, “Facial Rigging For 3D Character,” Int. J. Comput. Graph. Animat., vol. 4, no. 3, pp. 21–29, 2014.

[17] V. Orvalho, “Reusable facial rigging and animation: Create once, use many,” no. June, 2007.

[18] V. Orvalho, P. Bastos, F. Parke, B. Oliveira, and X. Alvarez, “A Facial Rigging Survey,” in In 33rd Annual Conference of the European Association for Computer Graphics-EUROGRAPHICS, 2012, vol. 51, pp. 13–18.

[19] T. Mclaughlin, L. Cutler, and D. Coleman, “(2011, August). Character rigging, deformations, and simulations in film and game production,” ACM SIGGRAPH 2011 Courses ACM., p. 5, 2011.

[20] H. Li, T. Weise, and M. Pauly, “Example-Based Facial Rigging,” ACM Trans. Graph. (TOG), 29(4), 32., vol. 29, no. 4, p. 32, 2010.