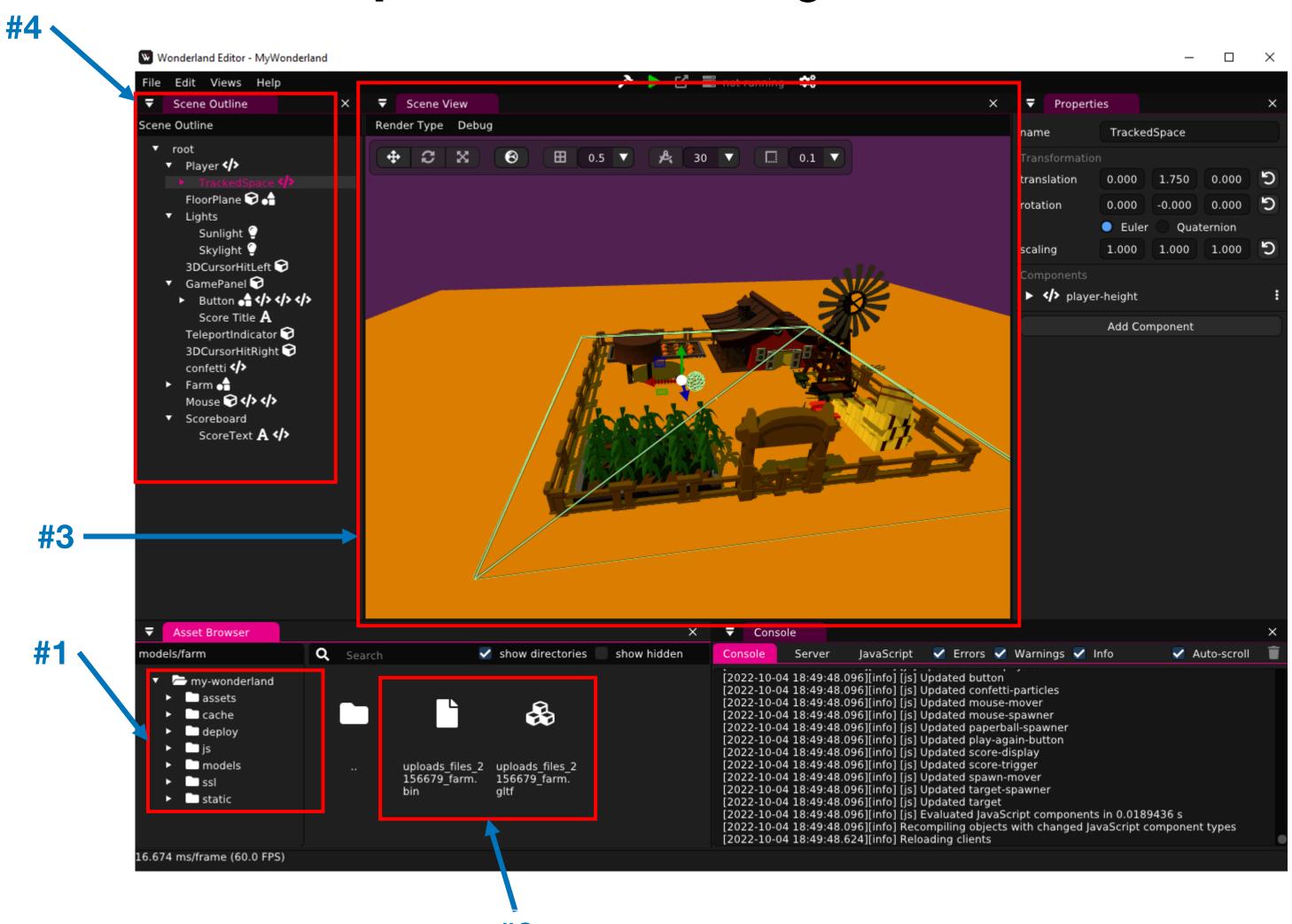
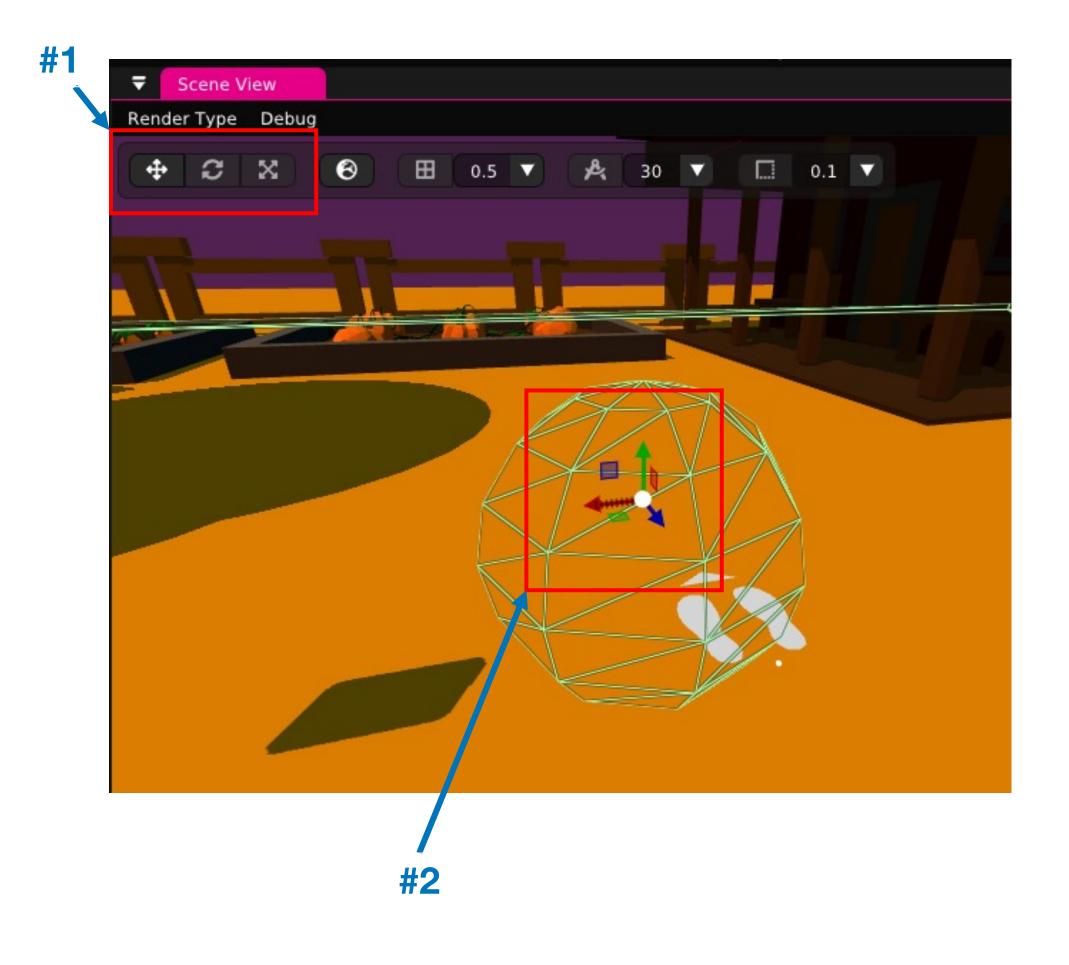
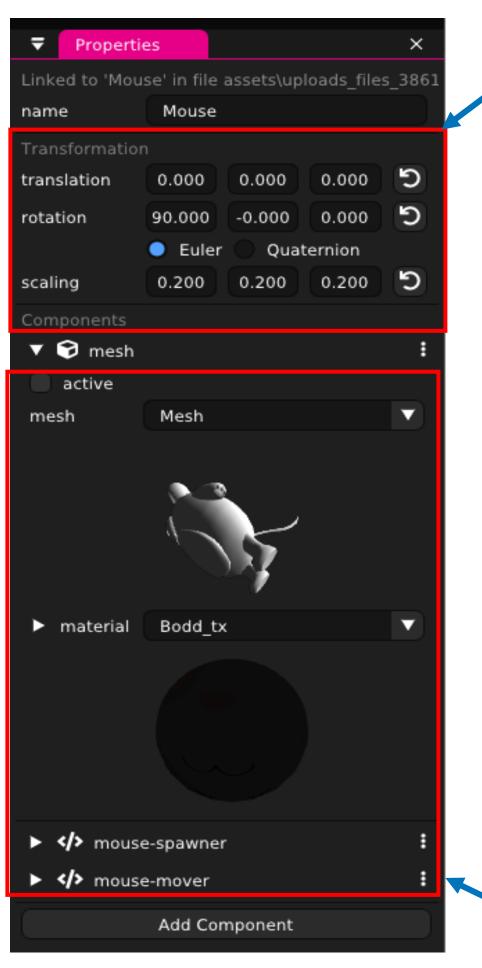
#### Import 3D objects into the scene



- 1. From the **Asset Browser**, open the models folder.
- 2. Drag and drop your 3D objects from your system into the **models folder**.
- 3. Drag and drop your 3D objects from the **models folder** into the Scene View.
- 4. Your object should show up in the **Scene View** as well as the **Scene Outline**.

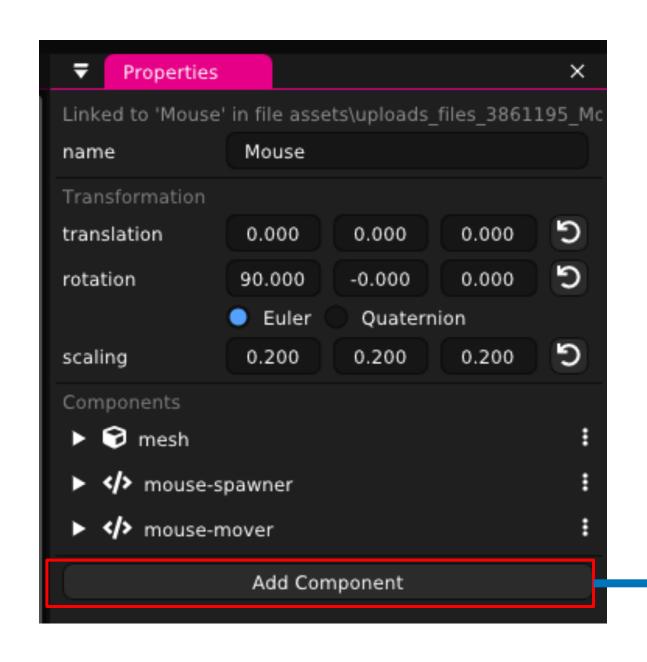
## Modify objects

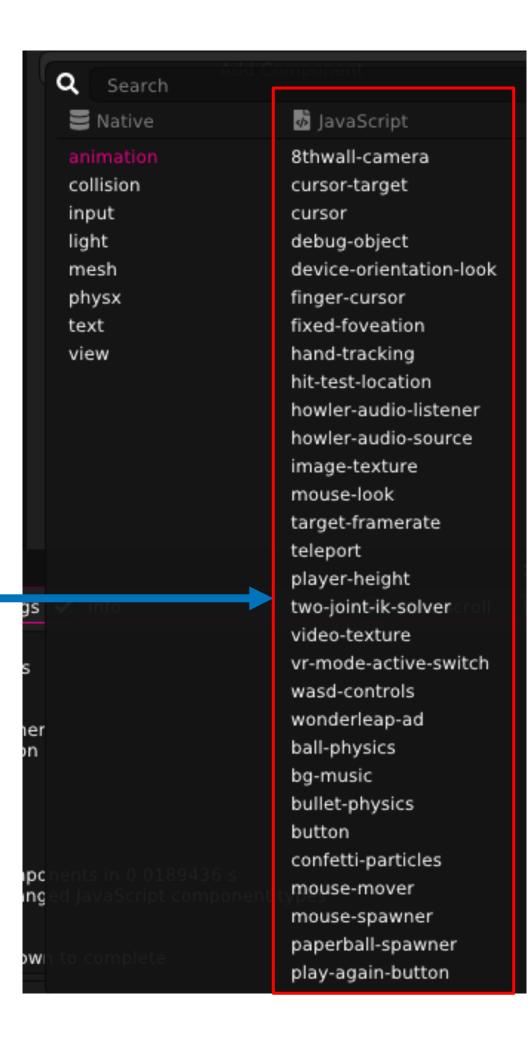




- 1. Objects can be modified directly in the Scene View Translation, rotation, and scaling by dragging the mouse/trackpad.
- 2. Changes can also be made from the **Properties view** by inputting values for greater precision.
- 3. Component values can also be modified in the Properties view.

### Adding JavaScript components to Objects





- JS components can be added to each object from the Properties view.
- Some of these components come with Wonderland (ex: 8thwall-camera, howler-audiosource) and the rest are user defined (ex: mouse-mover, mouse-spawner).

# Modify scene view

- 1. Zoom using the scroll wheel
- 2. Pan by right-clicking and dragging
- 3. Rotate by holding the middle-mouse button and dragging.

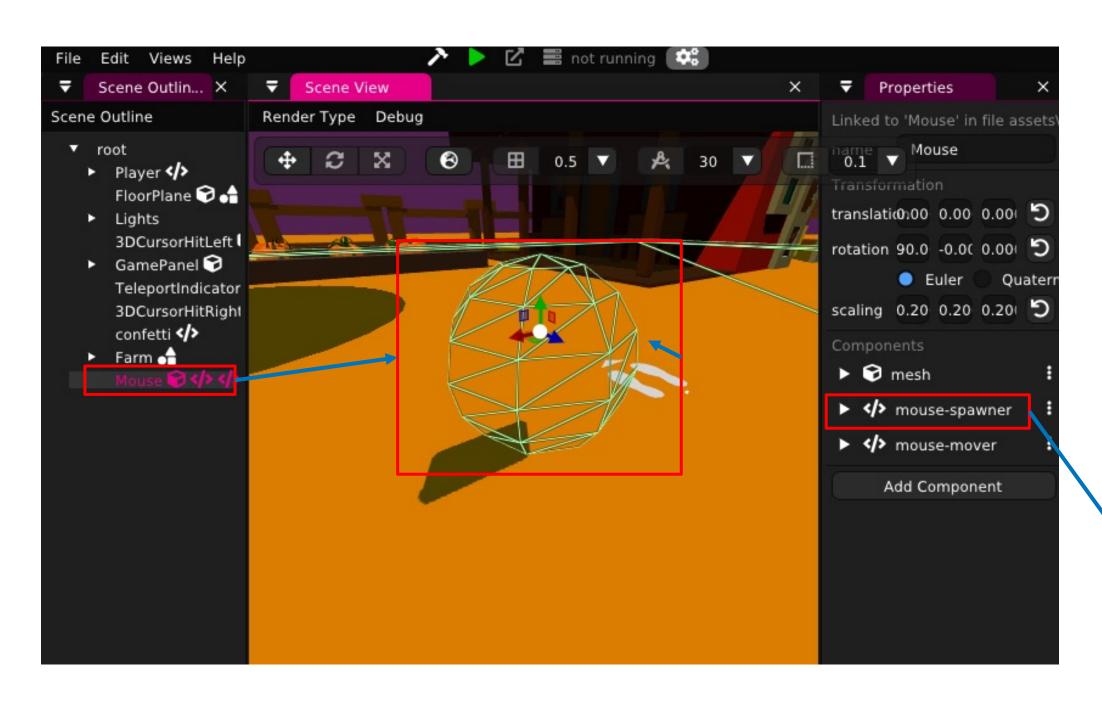






#1. Zoom #2. Pan #3. Rotate

### Apply Javascript logic to the scene



Use code editor like Visual Studio Code to add Javascript components and define the behavior of individual objects, such as movement, object spawning, adding/removing components, collision effects, etc.

```
EXPLORER
                                                         JS paperball-spawner is JS score-trigger is JS ball-physics is
OPEN EDITORS
                                    js > JS mouse-spawner.js > ...
                                      1 // var wastebinSpawner = null;
    JS play-again-button.js js
                                      var floorHeight = 0;
    JS button.js js
    JS paperball-spawner.js js
    JS score-trigger.js js
                                      5 @brief
    JS ball-physics.js js
    JS roomba.js js
                                          WL.registerComponent('mouse-spawner', {
                                               targetMesh: {type: WL.Type.Mesh},
    JS wastebin-spawner.js is
                                               targetMaterial: {type: WL.Type.Material},
 X JS mouse-spawner.js js
                                               spawnAnimation: {type: WL.Type.Animation},
   JS confetti-particles.js js
                                               maxTargets: {type: WL.Type.Int, default: 20},
                 ា ២១១
MY-WONDERLAND
                                               particles: {type: WL.Type.Object},
> assets
                                               init: function() {
                                                   this.time = 0;
                                                   this.spawnInterval = 3;
∨ js
                                                   this.critterSound = this.object.addComponent('howler-audio-sourd
 JS ball-physics.js
 JS bg-music.js
                                               start: function() {
 JS button.js
                                                   // WL.onXRSessionStart.push(this.xrSessionStart.bind(this));
 JS confetti-particles.js
                                                   this.targets = [];
 JS mouse-mover.js
                                                   // targetSpawner = this;
 JS mouse-spawner.js
                                                   this.spawnTarget();
JS paperball-spawner.js
 JS play-again-button.js
                                               update: function(dt) {
 JS score-display.js
                                                   this.time += dt;
 JS score-trigger.js
                                                   if(this.targets.length >= this.maxTargets) return;
 JS spawn-mover.js
                                                   if(this.time >= this.spawnInterval){
 JS target-spawner.js
                                                       this.time = 0;
 JS target.js
                                                        this.spawnTarget();
> models

✓ static

                                                   // updateScore("Place a target");
 > music

✓ sfx

                                               spawnTarget: function() {
                                                   // console.log("target-spawner >> spawnTarget");
 9mm-pistol-shoot-short-reverb-7...
                                                   // if(this.targets.length >= this.maxTargets) return;
  click.wav
                                                   /* Only spawn object if cursor is visible */
  cows-56001.mp3
  critter-40645.mp3
                                                   const obj = WL.scene.addObject();
  high-pitched-aha-103125.mp3
                                                   obj.transformLocal.set(this.object.transformWorld);
  pig_grunts_snorts_breathing_hack...
```

### View and debug in Huawei VR Glass

- 1. Enable USB debugging on Huawei VR compatible Huawei phone.
- 2. Enable ADB over Wi-Fi and connect to Huawei phone.
- 3. Generate SSL key, certificate, and DH parameter files.
- 4. Wonderland Engine navigate to Views > Preferences > Server, enable SSL server and enter the SSL files.
- 5. Repackage Wonderland project and launch the server.
- 6. Go to desktop Chrome browser, enter URL chrome://inspect#devices, click Port Forwarding button, add Port: 8081, IP address and port: localhost:8081. Check Enable port forwarding and click Done.
- 7. Put on Huawei VR Glass, launch either the Wolvic VR Browser and visit <a href="https://localhost:8081/index.html">https://localhost:8081/index.html</a>

