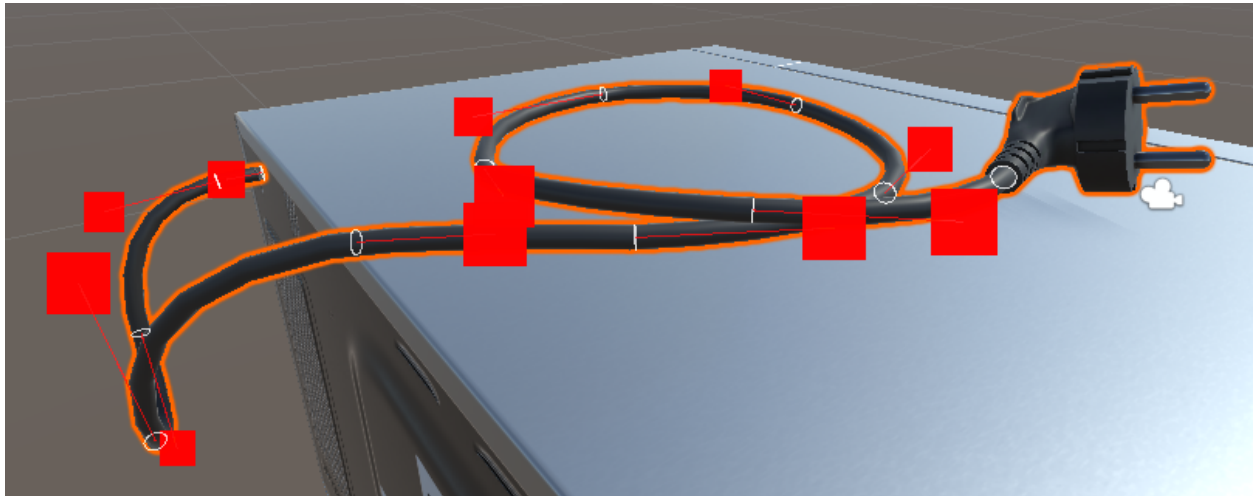


Thank you for purchasing the Ultra Microwave PBR asset!

Cable Editor	1
Inspector Properties	1
Troubleshooting	2
Microwave Display and Shaders	2
Render Texture	2
Shader Parameters	3
Realistic Microwave Controller	4
MicrowaveController	4

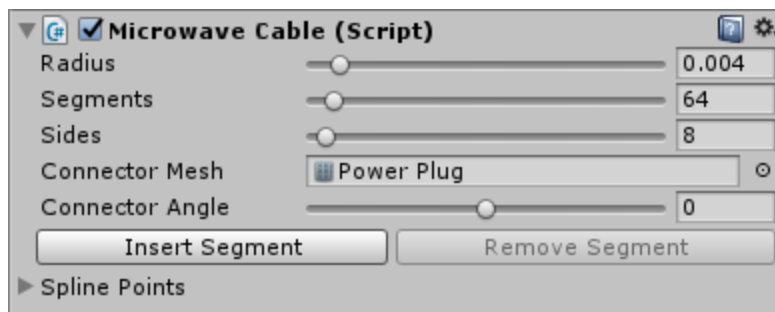
Cable Editor

The microwave includes a spline-based cable editor so you can wire it up realistically in your scene. This is controlled by the MicrowaveCable-Script.



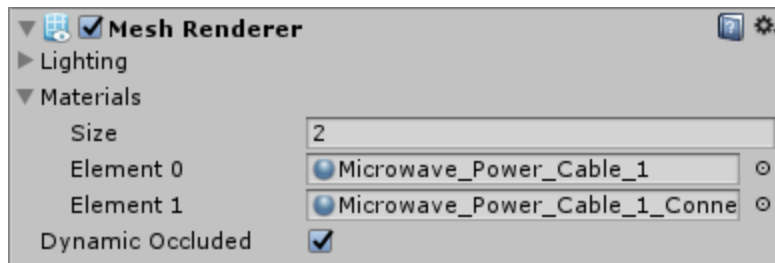
Inspector Properties

- **Radius** controls the diameter/thickness of the cable.
- **Segments** controls the number of cylinders generated to follow the spline. A higher number leads to better seamless results but more polygons are needed.
- **Sides** controls how many sides each segment has (the sides that make up each cylinder).
- The **Connector Mesh** is placed at the end of the cable and can be twisted with the **Connector Angle** if it does not match a socket/fixture in your scene.



Troubleshooting

If the connector mesh does not appear at the end of the hose or cable, please ensure that two material slots have been assigned in the mesh renderer.



Microwave Display and Shaders

The microwave has a simple user interface that displays the current time of the host computer and the remaining time of the cooking program. Font: <https://www.dafont.com/digital-7.font>



Render Texture

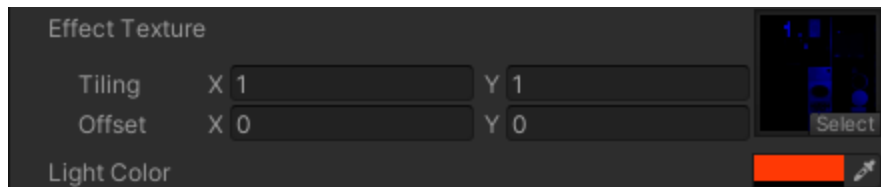
The MicrowaveController-Script creates a Render Texture and draws onto it using an instance of the MicrowaveGui material (as to not affect the original). This material uses the MicrowaveGui shader.

The shader has one texture slot and renders it with UVs, as you would expect. The trick to rendering many different textures, for example the digits for the microwave, is to swap the texture with another and render a quad. This is repeated for each digit.

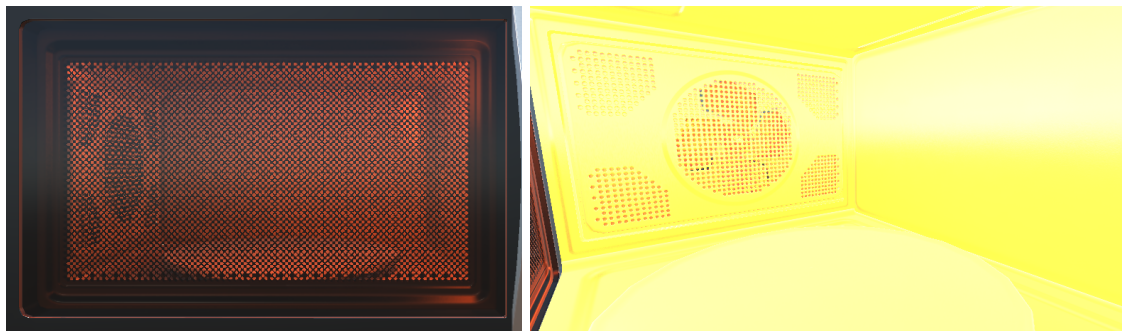
Shader Parameters

The microwave shader is based on the standard shader and uses the usual textures. The special Effect Texture contains a baked lightmap of the activated microwave in the blue channel, this is applied to the emission slot multiplied by the given light color.

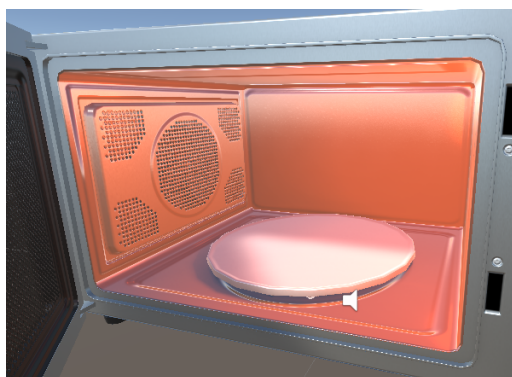
The alpha of the light color enables and disables the internal lighting.



From the outside, looking through the dark door, it appears like the microwave has been activated. To achieve this look, the light intensity has been significantly increased in the shader.



As such the microwave truly should not be used with the door open. If the user opens the door, the light intensity will be smoothly but quickly reduced, as a compromise.










Realistic Microwave Controller

The microwave comes with a realistic controller replicated exactly from a real microwave to faithfully simulate reality. It provides the configuration of the cooking time, controls the rotation of the oven fan and turntable, switches the interior lighting, and responds to the opening and closing of the door. It plays sound with start, loop and end segments. It crossfades audio samples and uses multiple audio sources to create a realistic experience.

MicrowaveController

This is the primary script that runs the microwave. It handles all animation and sound. It provides status information that other scripts in the scene can respond to.

 isCooking	Gets whether the microwave is currently cooking.
 isPaused	Gets whether the microwave has been paused by the user.
 isDoorOpen	Gets whether the door is currently open.
 doorAngle	Gets or sets the angle of the door in degrees.
 secondsRemaining	Gets the amount of seconds remaining while cooking.
 StartMicrowave	Starts the microwave or adds a minute to the cooking program (the time increment is customizable and can also be used to immediately set a long timer, like 3 minutes, if the microwave is not currently running).
 StopMicrowave	Pauses the microwave or stops it when called again.

The inspector provides a small user interface for *demonstration purposes only*.



The door can be rotated on Y using the inspector (or C# script using the doorAngle property).