Touchless.Design

Touchless.Design SDK: Version 3.0.0

Website: https://touchless.design
Leap Motion SDK: https://developer.leapmotion.com/sdk-leap-motion-controller

In light of the coronavirus crisis, touchless interaction has become the focus of many designers and developers who create kiosks and screen-based exhibits for public spaces. Using the Leap Motion Controller and V4 SDK, our Integrated Touchless System allows users to interact with our tables and exhibits with no physical touch. This system tracks the visitor's hand and watches for specific gestures, such as opening and closing, that give complete control over the mouse. The Integrated Touchless System is a mouse emulation system with multiple methods for onboarding and real-time feedback.

Alongside this system are three additional applications: a dynamic cursor overlay, a secondary instructional screen, and an LED system add-on. Download the latest release here. These applications are meant to provide both feedback and onboarding information to the visitor through color changes, gesture icons, and interaction information when hovering over buttons or drag areas. This repository also includes a Unity asset package that allows users to integrate support for the system into their own applications and a demo application for reference. The core system is built in .Net Framework 4.6.1 and the peripheral applications are built in the Unity Game Engine.

Important Notes:

- The Leap Motion V4 Orion SDK must be installed with a leap motion device connected for this application to function.
- This repository is meant to be built and run using the Touchless Add-on with a Leap Motion device and an Ideum Touchless Pedestal.
- The keyboard shortcut Ctrl+Alt+I will toggle mouse emulation on/off when the service is running.

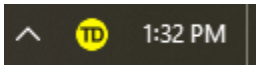
Preview Video: https://www.youtube.com/watch?v=apu0_l-zF6g

Repository Layout

All of the source code for the Integrated Touchless System is in the src directory, as well as the peripheral applications and dependencies with the exception of the Leap Motion V4 Orion SDK. The Leap Motion V4 Orion SDK should be downloaded and installed before running the Touchless Design applications.

Service

This is the core of the Integrated Touchless System. It operates as a Windows service and is responsible for integrating the Leap Motion SDK, controlling the mouse, managing the Overlay and Add-on applications, setting the color of the attached LEDs, and communicating with any client applications. Certain settings such as gesture toggles can be configured via a system tray icon as shown below. Other options can be configured by editing the configuration JSON files in the Integrated Touchless System's root directory. This is explained in greater depth here.



Overlay

The Overlay application is a replacement for the Windows cursor and provides feedback to the user as they interact with the Integrated Touchless System. When enabled, the System will automatically manage and communicate with the Overlay application.

When interacting with a client application that is connected to the Integrated Touchless System, the Overlay will have the following behaviors:

- When hovering over a button, it will show an animated select gesture.
- When hovering over a drag area, it will show a closed fist with arrows.
- When a selection is made, it will turn green and change size.
- If the screen is touched, it will turn red.

AddOn

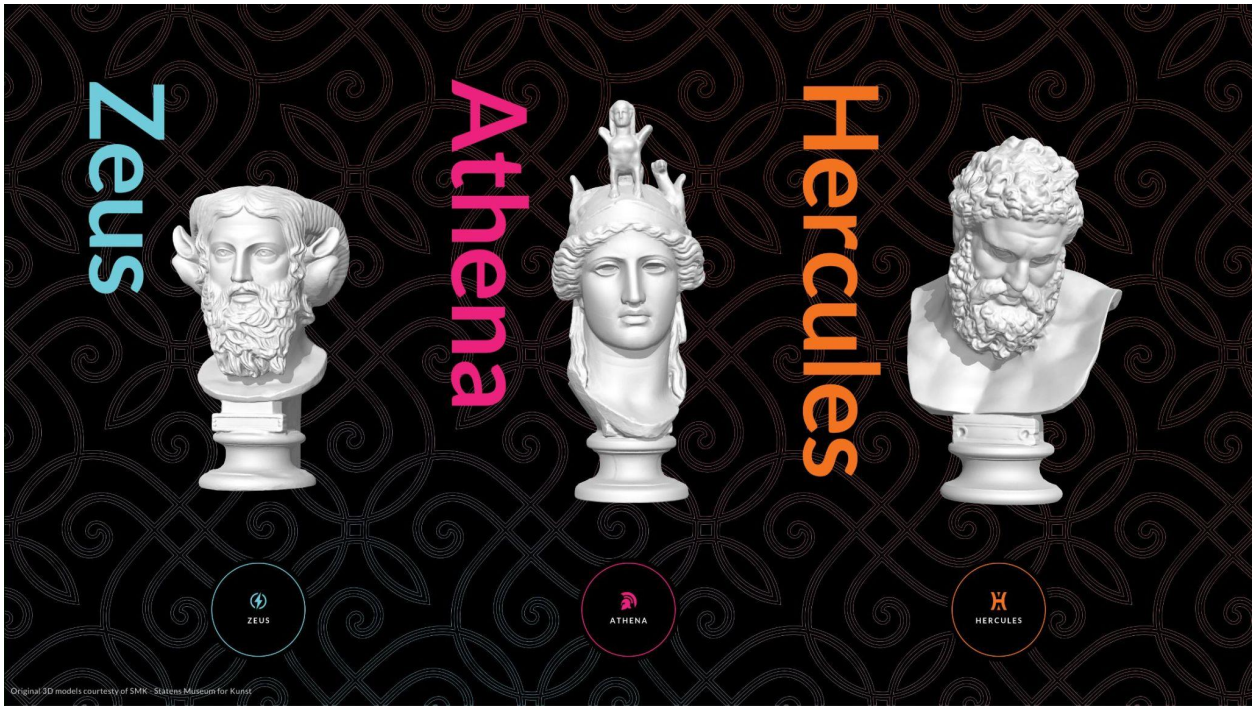
The Add-on application is designed to display information on the secondary monitor and control the LED lights available on the Touchless Pedestal. This application provides further feedback and onboarding information to supplement the Overlay cursor as users interact with the Integrated Touchless System. When enabled, the System will automatically manage and communicate with the Add-on application. In addition to the animated hand icon, the Add-on application also shows text feedback.

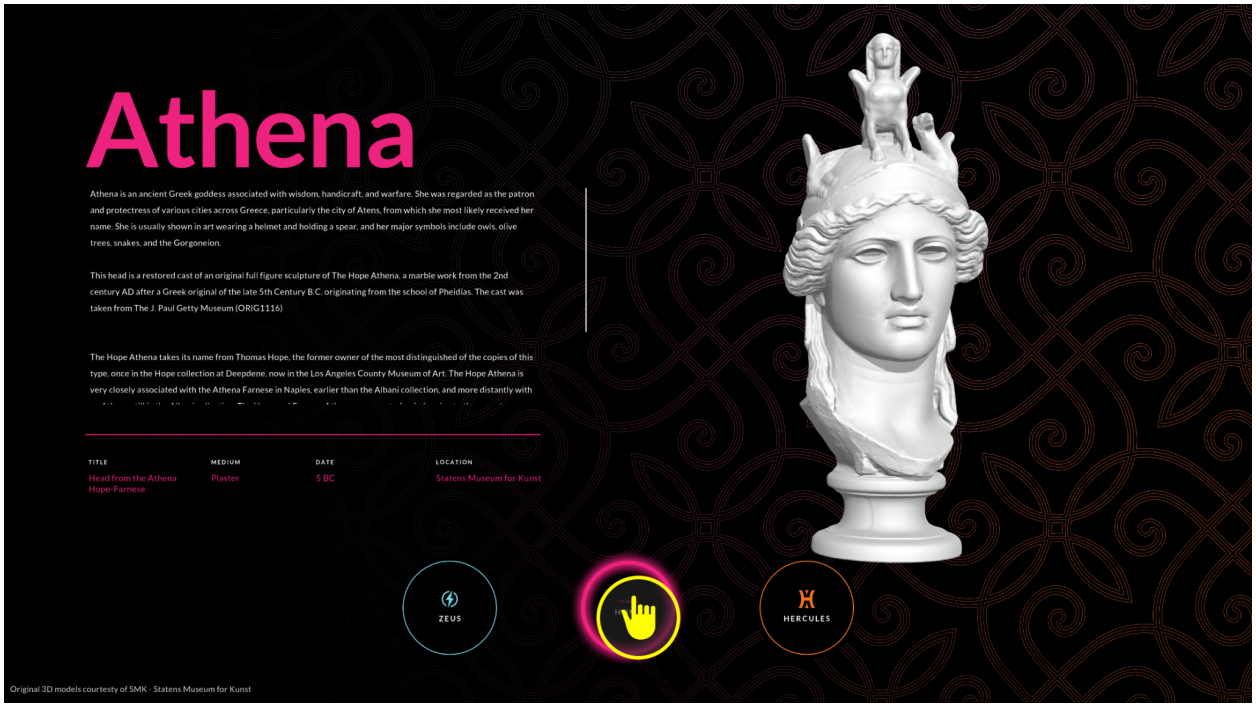
Asset Package

The Unity asset package, "TouchlessDesign.unitypackage", can be imported into a Unity Engine project and used to integrate that project application with the Integrated Touchless System. More detailed instructions for this can be found here.

Example Project

The Example application is a demo client application that uses the Unity asset package to integrate with the Integrated Touchless System.





Ideum.Logging

This is a logging abstraction we use in Unity and is included in all of the Unity projects, but we've included the source code here for posterity.

Deployment

Service

Building:

In order to build the core of the Integrated Touchless System, open the TouchlessDesign.sln file in Visual Studio, right-click the Solution and select "Build Solution." We have included instructions that will build the solution at the following path on your system:
`%appdata%/Ideum/TouchlessDesign`

Running:

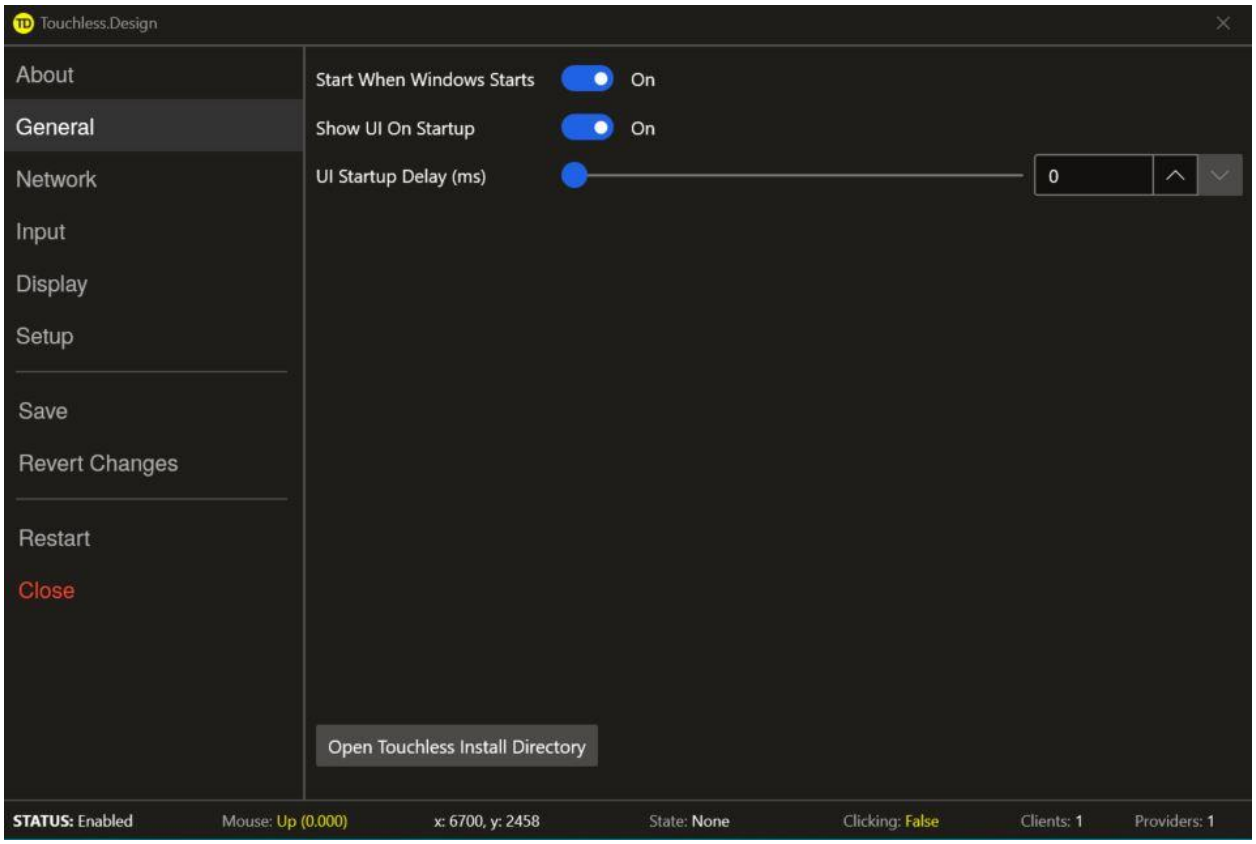
Navigate to the build directory (see above), open `TouchlessDesign/bin/Service/`, and run the `TouchlessDesign.exe`. If the AddOn and/or Overlay applications have been built and configured, they will also be launched and the mouse should immediately begin reacting to the Leap Motion controller. An icon will appear in the system tray and by left clicking on the icon, the service UI will be displayed.

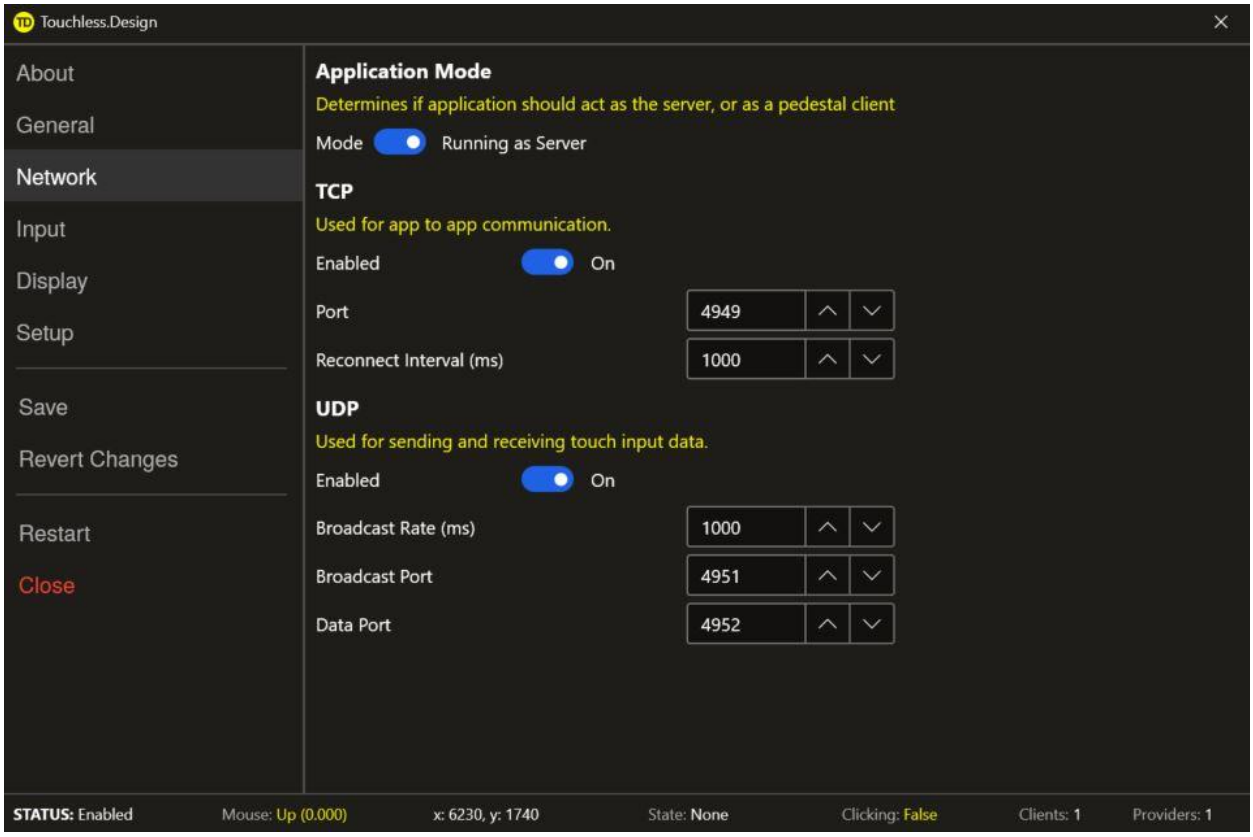
Configuration:

Configuration is primarily handled through the UI, but all configurable values are stored in JSON files that can be found in the touchless design directory.

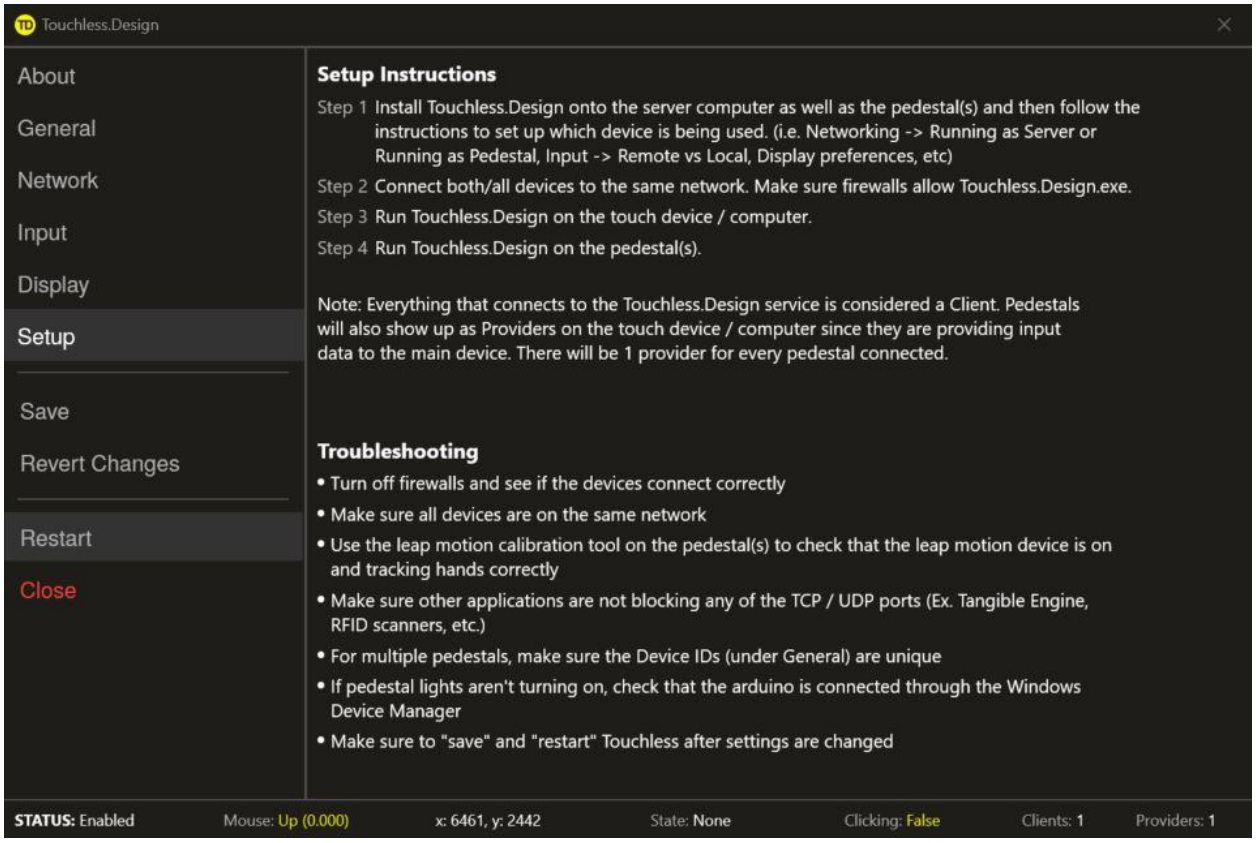
- **display.json**: Allows you to adjust functionality related to the overall user experience such as toggling the onboarding on and off or adjusting values relating to the arduino LED control.
- **general.json**: Allows you to decide if Touchless should start on startup.
- **input.json**: Allows you to adjust options related to mouse emulation and whether you're receiving touch input locally or remotely from a pedestal.
- **network.json**: Configures network settings for the Service and lets the program know whether it is the server or pedestal client.


Touchless.Design
About
General
Network
Input
Display
Setup
Save
Revert Changes
Restart
Close
Touchless.Design
v3.0.0
© 2023 Ideum, Inc. All rights reserved.
Releases | Issues | Code | Website | Ideum
STATUS: Enabled
Mouse: Up (0.000)
x: 6862, y: 2403
State: None
Clicking: False
Clients: 1
Providers: 1


Touchless.Design
About
General
Network
Input
Display
Setup
Save
Revert Changes
Restart
Close
Start When Windows Starts
On
Show UI On Startup
On
UI Startup Delay (ms)
0
Open Touchless Install Directory
STATUS: Enabled
Mouse: Up (0.000)
x: 6700, y: 2458
State: None
Clicking: False
Clients: 1
Providers: 1


Touchless.Design
About
General
Network
Input
Display
Setup
Save
Revert Changes
Restart
Close
Application Mode
Determines if application should act as the server, or as a pedestal client
Mode
Running as Server
TCP
Used for app to app communication.
Enabled
On
Port
4949
Reconnect Interval (ms)
1000
UDP
Used for sending and receiving touch input data.
Enabled
On
Broadcast Rate (ms)
1000
Broadcast Port
4951
Data Port
4952
STATUS: Enabled
Mouse: Up (0.000)
x: 6230, y: 1740
State: None
Clicking: False
Clients: 1
Providers: 1


Touchless.Design
About
General
Network
Input
Display
Setup
Save
Revert Changes
Restart
Close
General
Input Provider
Remote (UDP)
Server => Remote | Pedestal => Local | Can use local mode with server if a leap motion is plugged in.
Mouse Emulation
On
Keyboard Shortcut
Control+Alt+I
Grab Threshold
0.811
Hand Confidence
0.000
Click Emulation
On
Click Duration (ms)
500
Horizontal Axis (Left / Right)
Min (m)
-0.250
Max (m)
0.350
Vertical Axis (Forwards / Backwards)
Orientation
Z-Axis (Forwards / Backwards)
Invert
Off
Min (m)
-0.200
Max (m)
0.300
STATUS: Enabled
Mouse: Up (0.000)
x: 6895, y: 2579
State: None
Clicking: False
Clients: 1
Providers: 1



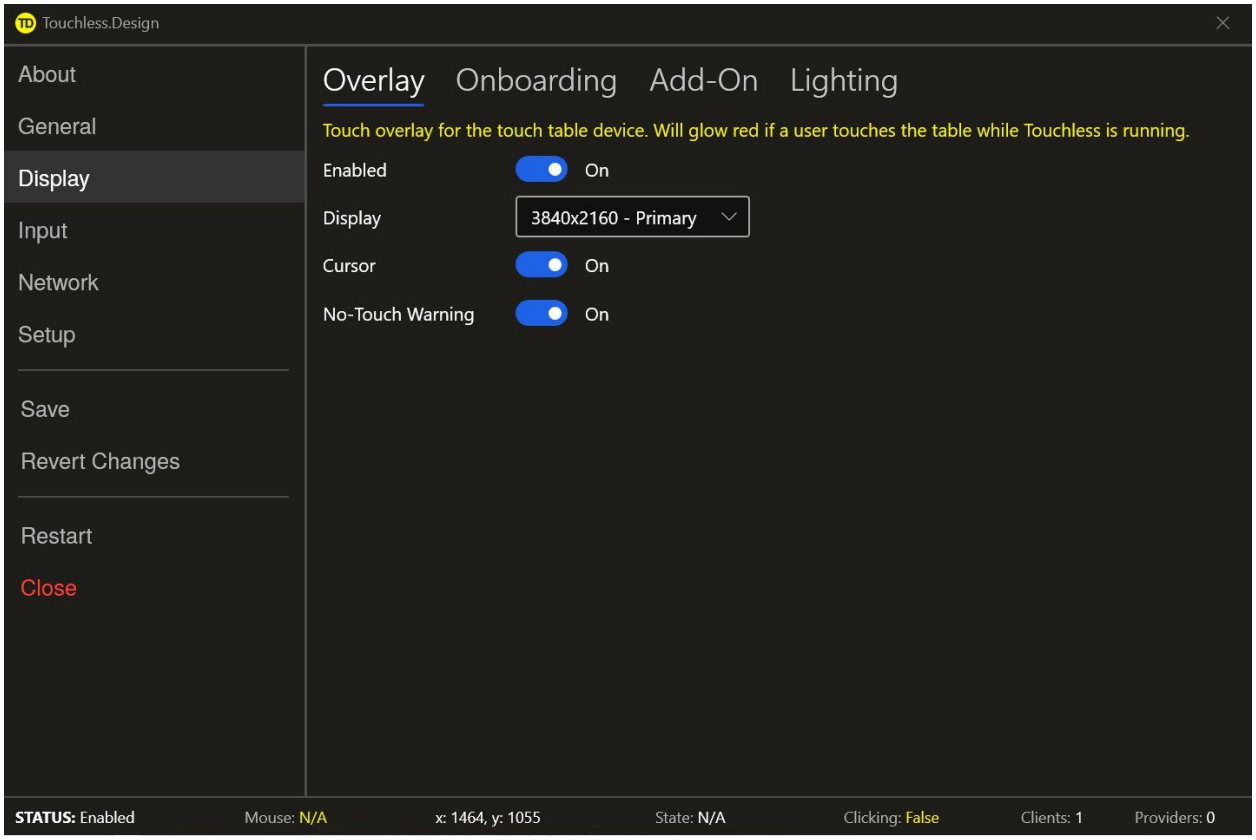
Main Device Settings

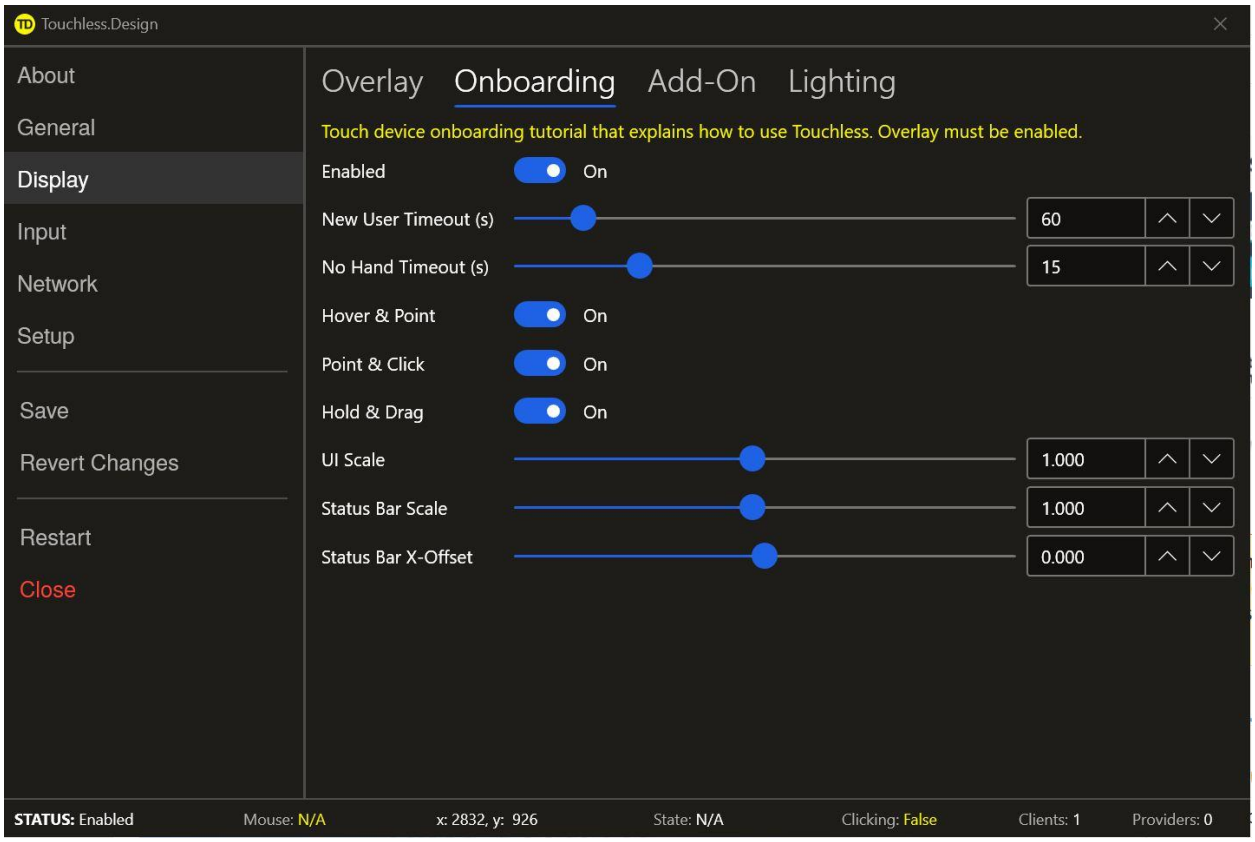
The main device used to host Touchless and any Touchless applications can be called the “Server” and will need to have the “Running as Server” toggle set in the network tab.

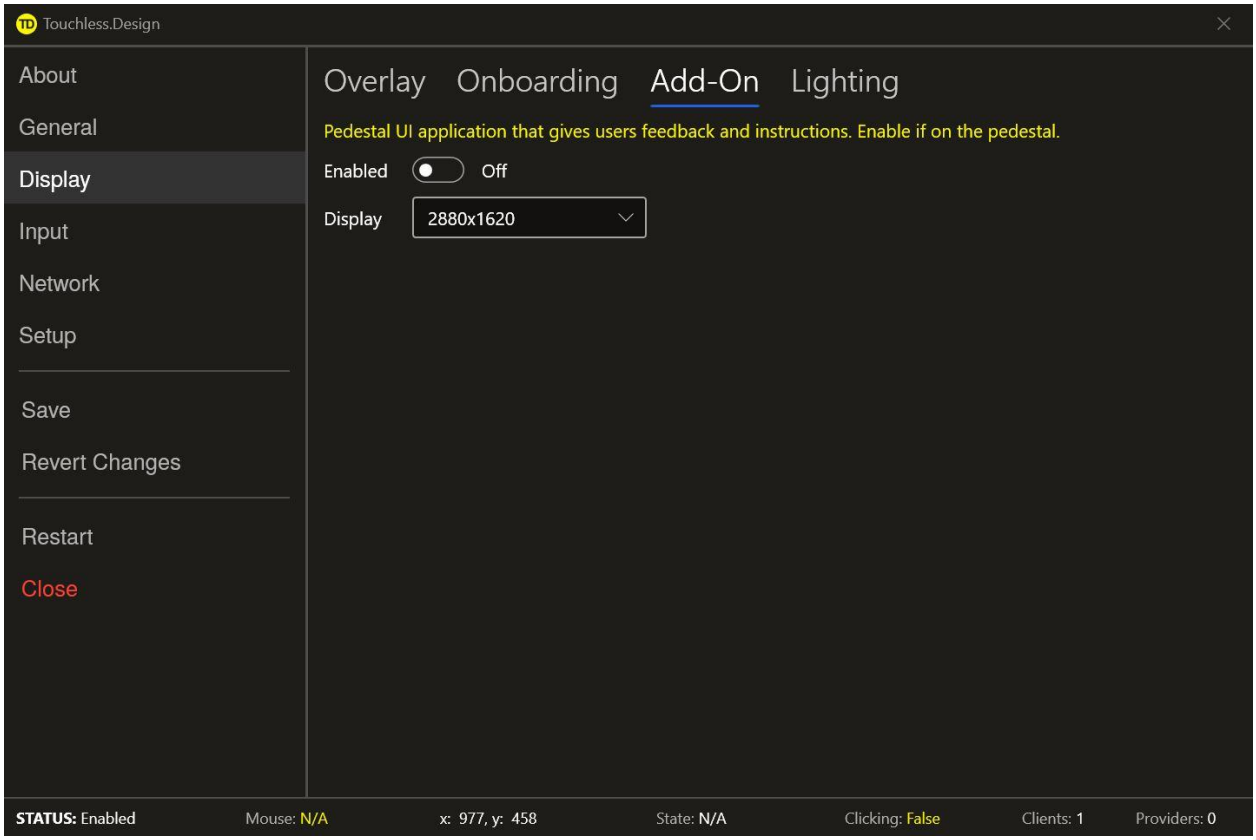
Next, go into the Input tab and set the “Input Provider” to “Remote UDP” so that the server knows to listen for touch input from a pedestal on the same network.

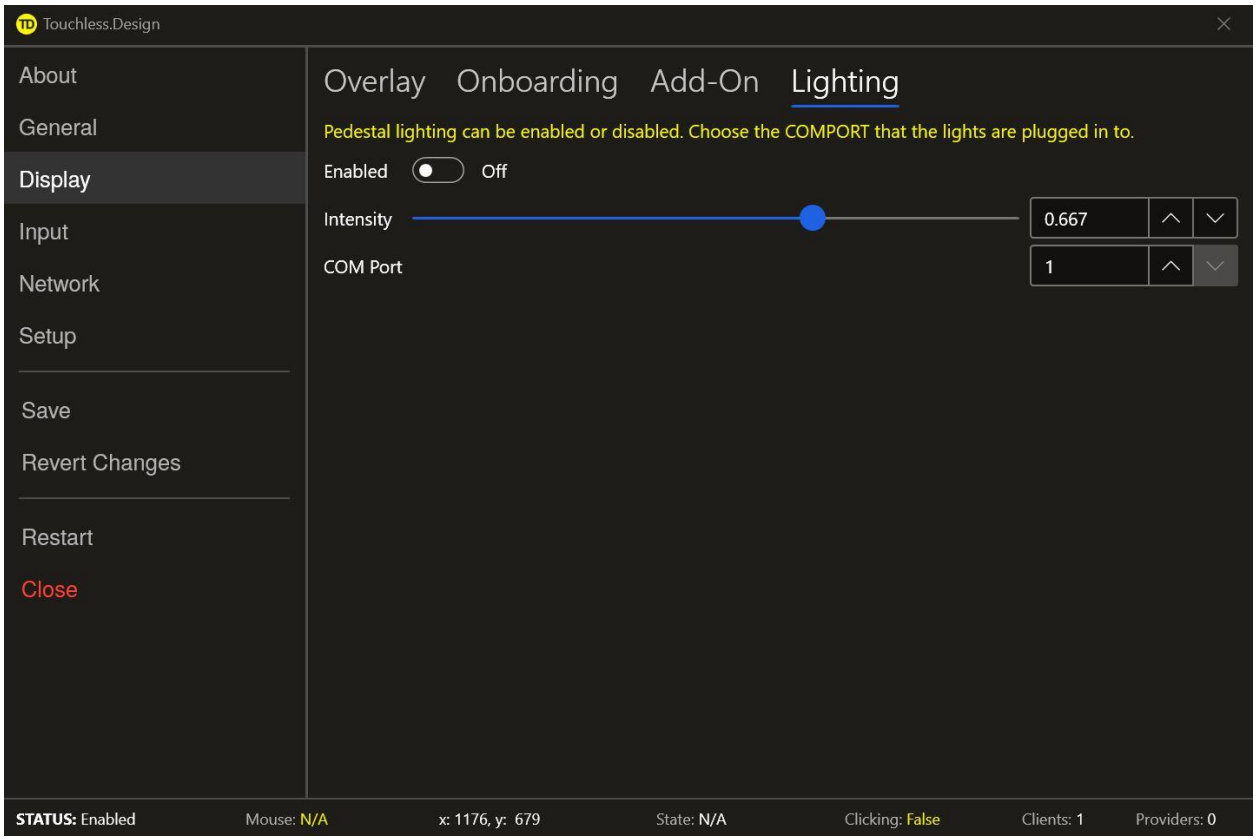
If you’re on a testing computer or sitting at your desk programming, you can connect a leap motion to your machine and run Touchless in server mode, with the “Input Provider” set to “Local” for a one-user test setup.

The following images show you which display options are meant to run on the server versus the pedestal.


Touchless.Design
About
General
Display
Input
Network
Setup
Save
Revert Changes
Restart
Close
Overlay
Onboarding
Add-On
Lighting
Touch overlay for the touch table device. Will glow red if a user touches the table while Touchless is running.
Enabled
On
Display
3840x2160 - Primary
Cursor
On
No-Touch Warning
On
STATUS: Enabled
Mouse: N/A
x: 1464, y: 1055
State: N/A
Clicking: False
Clients: 1
Providers: 0


Touchless.Design
About
General
Display
Input
Network
Setup
Save
Revert Changes
Restart
Close
Overlay
Onboarding
Add-On
Lighting
Touch device onboarding tutorial that explains how to use Touchless. Overlay must be enabled.
Enabled
On
New User Timeout (s)
60
No Hand Timeout (s)
15
Hover & Point
On
Point & Click
On
Hold & Drag
On
UI Scale
1.000
Status Bar Scale
1.000
Status Bar X-Offset
0.000
STATUS: Enabled
Mouse: N/A
x: 2832, y: 926
State: N/A
Clicking: False
Clients: 1
Providers: 0


Touchless.Design
About
General
Display
Input
Network
Setup
Save
Revert Changes
Restart
Close
Overlay
Onboarding
Add-On
Lighting
Pedestal UI application that gives users feedback and instructions. Enable if on the pedestal.
Enabled
Off
Display
2880x1620
STATUS: Enabled
Mouse: N/A
x: 977, y: 458
State: N/A
Clicking: False
Clients: 1
Providers: 0

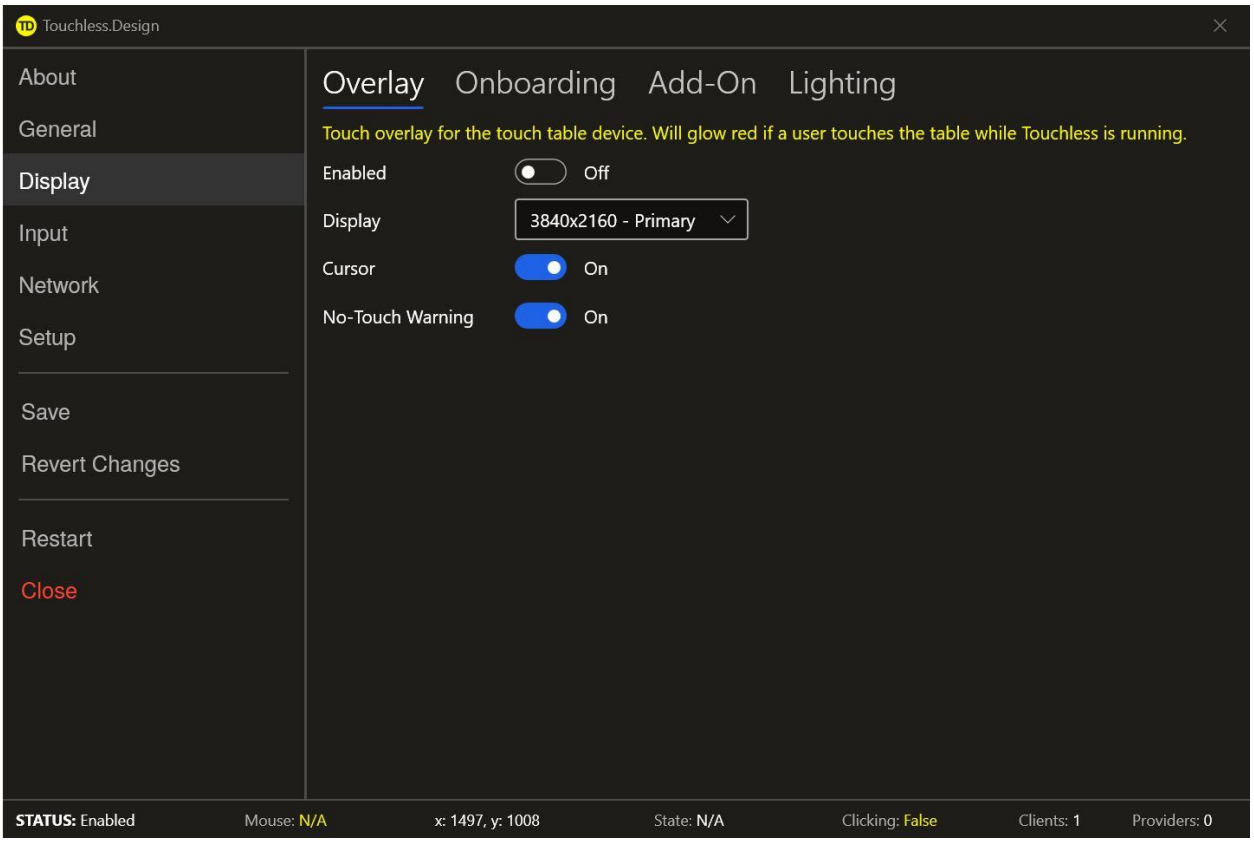

Touchless.Design
About
General
Display
Input
Network
Setup
Save
Revert Changes
Restart
Close
Overlay
Onboarding
Add-On
Lighting
Pedestal lighting can be enabled or disabled. Choose the COMPORT that the lights are plugged in to.
Enabled
Off
Intensity
0.667
COM Port
1
STATUS: Enabled
Mouse: N/A
x: 1176, y: 679
State: N/A
Clicking: False
Clients: 1
Providers: 0

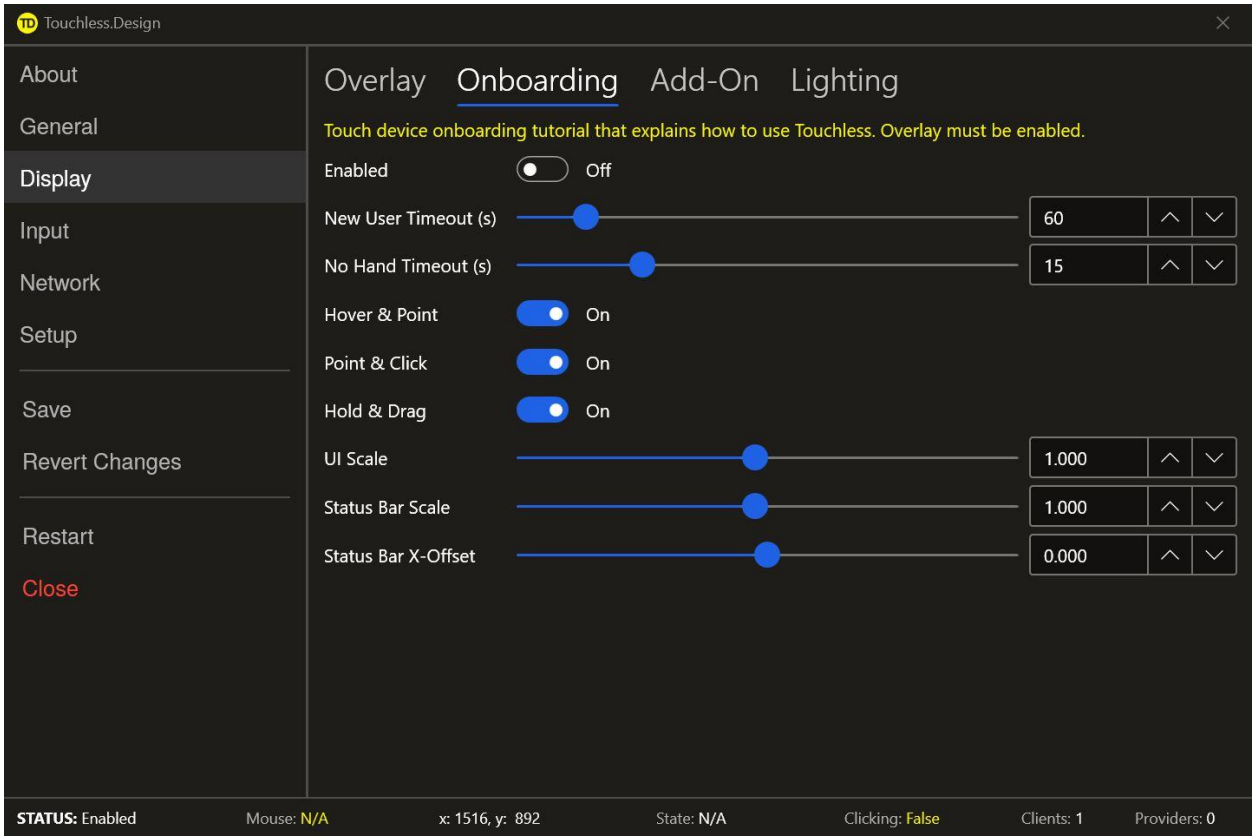
Pedestal Settings

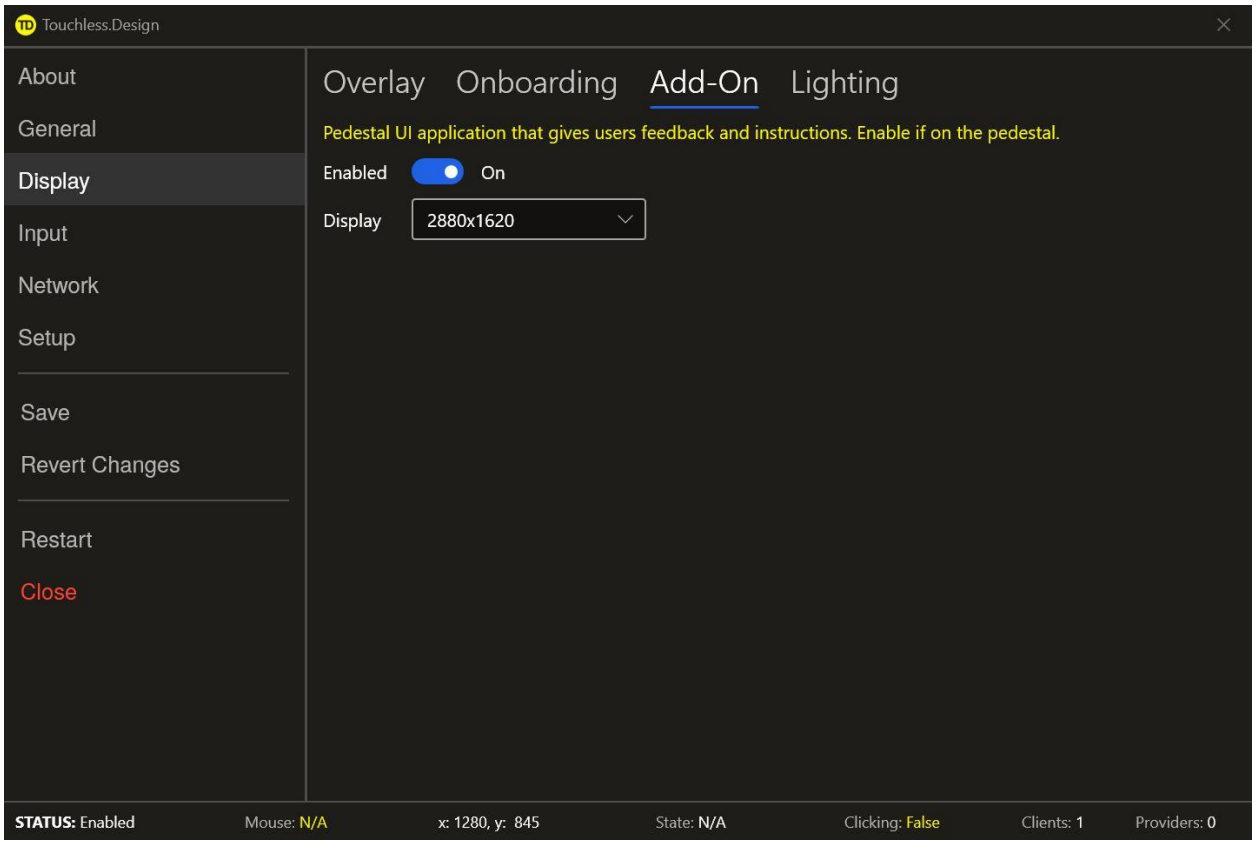
The pedestal will come installed with the same TouchlessDesign program, but will need to be configured slightly differently.

In the network tab, set the mode to “Running as Pedestal”, and in the input tab, set the “Input Provider” to “Local”. This will tell the pedestal to look for the leap motion data and then send it over to the server, over the network.

The following images show you which display options are meant to run on the pedestal versus the server..

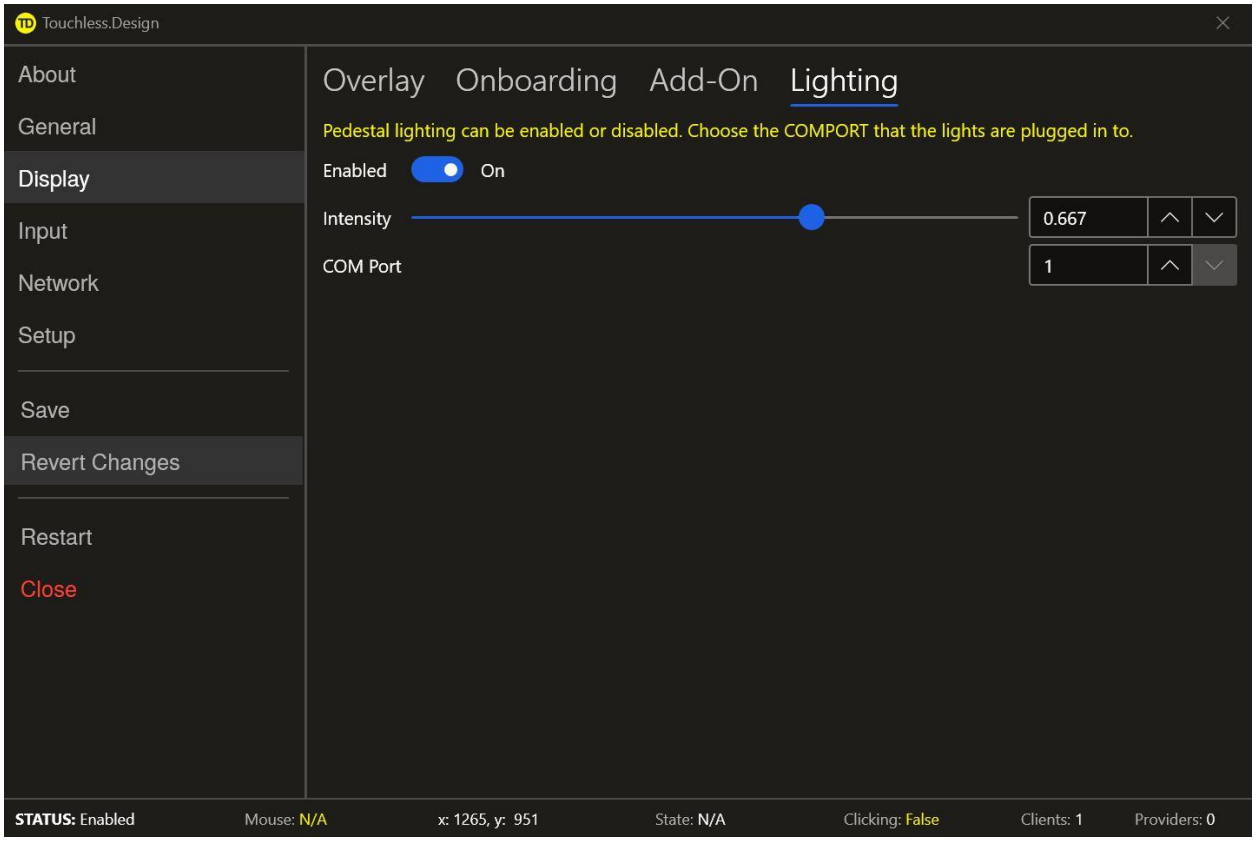



Touchless.Design
About
General
Display
Input
Network
Setup
Save
Revert Changes
Restart
Close
Overlay
Onboarding
Add-On
Lighting
Touch device onboarding tutorial that explains how to use Touchless. Overlay must be enabled.
Enabled
Off
New User Timeout (s)
60
No Hand Timeout (s)
15
Hover & Point
On
Point & Click
On
Hold & Drag
On
UI Scale
1.000
Status Bar Scale
1.000
Status Bar X-Offset
0.000
STATUS: Enabled
Mouse: N/A
x: 1516, y: 892
State: N/A
Clicking: False
Clients: 1
Providers: 0


Touchless.Design
About
General
Display
Input
Network
Setup
Save
Revert Changes
Restart
Close
Overlay
Onboarding
Add-On
Lighting
Pedestal UI application that gives users feedback and instructions. Enable if on the pedestal.
Enabled
On
Display
2880x1620
STATUS: Enabled
Mouse: N/A
x: 1280, y: 845
State: N/A
Clicking: False
Clients: 1
Providers: 0

The lighting has changed between version 2.0 and 3.0 from a FadeCandy server to an Arduino sketch. This requires the program to know which COM port the lights are plugged into in order to send state changes to the sketch.

To find the COM port value, look in Device Manager at the USB devices. Once you've set the value, save and restart the UI.



AddOn and Overlay

1. Download Unity version 2019.3.2f1
2. Open the application in Unity, and go to File -> Build Settings.
3. Make sure the _App/Scenes/App Scene is checked and press Build.
4. In order for the Integrated Touchless System to find and run either of the peripheral applications, they should be built at the following respective paths:
    a. `%appdata%/Ideum/TouchlessDesign/bin/AddOn/`
    b. `%appdata%/Ideum/TouchlessDesign/bin/Overlay/`

Now, when you run the Integrated Touchless System, it will automatically start up the applications specified in the display.json file. Likewise, when the System is exited, it will terminate those same applications.

***Note**: These applications will only provide gesture feedback when running alongside a client application that uses the Integrated Touchless System bindings, such as the Example application.*

Adding Support to a Custom Application

In order to integrate a project with the Integrated Touchless System, the project must first import the unity asset package located in the root src directory of this repository. Below are the minimum recommended steps to get the asset package loaded and integrated. The Example application can also be used as a demonstration.

1. Open the Unity project.
2. Open Assets-> Import Package -> Custom Package...
3. This will open a dialog. Select the TouchlessDesign.unitypackage file in the src directory of this repository. When it prompts you to select which parts of the package to import, select Import All.
4. In a MonoBehavior script, in the Start function, call the TouchlessDesign.Initialize function and subscribe to its `OnConnected` and `OnDisconnected` events. (Make sure to include using Ideum in the file). Also include the `Deinitialize` call when quitting the application.

```
void Start() {
  TouchlessDesign.Initialize("%appdata%/Ideum/TouchlessDesign");
  TouchlessDesign.Connected += OnConnected;
  TouchlessDesign.Disconnected += OnDisconnected;
}

void OnApplicationQuit() {
  TouchlessDesign.DeInitialize();
}
```

5. In order to query the state of the System, there are a number of static query calls. For each one, pass a delegate that will handle the response from the System. Below is an example on a script that queries both the click and hover state, and the no touch state on an interval, and passes two delegates to those calls. When the TouchlessDesign receives a response from the System for either call, it will execute the passed delegate.

```
private void Update() {
  if (_connected) {
    _timer += Time.deltaTime;
    if(_timer > _queryInterval) {
      TouchlessDesign.QueryClickAndHoverState(HandleQueryResponse);
```

```
      TouchlessDesign.QueryNoTouchState(HandleNoTouch);
      _timer = 0f;
    }
  }
}

private void HandleNoTouch(bool noTouch) {
  if (noTouch) {
    // Show no-touch warning information.
  }
}

private void HandleQueryResponse(bool clickState, HoverStates
hoverState) {
  // Do something with the hover and click state.
}
```

6. In order to set a hover state (for example, when the user hovers over a button), create a new script and place it on whatever component you want to react to the hover state. In that script, implement the IPointerEnterHandler and IPointerExitHandler interfaces.

```
public class TouchlessHoverCapture : MonoBehaviour,
IPointerEnterHandler, IPointerExitHandler
```

7. Implement the interface functions OnPointerEnter and OnPointerExit. In those functions call the TouchlessDesign `SetHoverState` function, and pass the desired hover state, or pass false to deactivate, as shown below:

```
public void OnPointerEnter(PointerEventData eventData) {
  TouchlessDesign.SetHoverState(HoverStates.Click);
}

public void OnPointerExit(PointerEventData eventData) {
  TouchlessDesign.SetHoverState(false);
}
```

New in 3.0.0

Multi-User Functionality

Version 3.0.0 now includes multi-user functionality to allow for multiple pedestals to interact with one main display.

The first client pedestal to connect will have control of the mouse, while latter pedestals will be accounted for using touch points.

At the bottom of the Touchless.Design popup window, text now provides information on the number of touch provider pedestals as well as overall clients (apps) that are connected and talking to one another.

An example Unity project (Pong) has been placed in the directory `examples/Touchless-Pong.unitypackage` as a way to see how this new functionality can be used.

Arduino Powered Lights

Originally, Touchless used Fade Candy lights to illuminate the pedestal and change colors to indicate certain gestures. Fade Candy is now deprecated, so a replacement set up has been implemented to achieve the same effect. Pedestals now use arduino powered lights (an Adafruit Trinket M0 wired to NeoPixel LED strips) and a COM Port to send commands back and forth. You can find the arduino board here.

First Time Hardware Setup

1. Install the touch device version of Touchless onto the touch device / computer, and the pedestal version of Touchless onto the pedestal(s).

2. Connect both/all devices to the same network. Make sure firewalls allow Touchless.Design.exe.

3. Run Touchless.Design on the touch device / computer.

4. Run Touchless.Design on the pedestal(s).

Note: Everything that connects to the Touchless.Design service is considered a Client. Pedestals will also show up as Providers on the touch device / computer since they are providing input data to the main device. There will be 1 provider for every pedestal connected.

Troubleshooting

- Turn off firewalls and see if the devices connect correctly
- Make sure all devices are on the same network
- Use the leap motion calibration tool on the pedestal(s) to check that the leap motion device is on and tracking hands correctly
- Make sure other applications are not blocking any of the TCP / UDP ports (Ex. Tangible Engine, RFID scanners, etc.)
- For multiple pedestals, make sure the Device IDs (under General) are unique
- If pedestal lights aren't turning on, check that the arduino is connected through the Windows Device Manager
- Make sure to "save" and "restart" Touchless after settings are changed