## Lesson Objectives: MLTK-Brain

<table>
<thead>
<tr>
<th>Indicators of Learner Success</th>
<th>Engagement Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Able to build the Brain App on their Magic Leap device</td>
<td>Content feels immediately useful in a professional capacity</td>
</tr>
<tr>
<td>Positive Engagement on Discord</td>
<td>Shared sense of accomplishment from learner-learner sharing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lower Order Objectives</th>
<th>Out of Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broad grasp of the current state of Magic Leap tooling</td>
<td>Zero Iteration workflows</td>
</tr>
<tr>
<td>Comfortable navigating MLTK feature set and example scenes</td>
<td>Multi-user features of MLTK</td>
</tr>
<tr>
<td>Able to drag-and-drop MLTK prefabs into their own projects</td>
<td>Unity Canvas, Tracked Pose Driver, Magic Leap Camera</td>
</tr>
<tr>
<td>Able to access and follow Discord chat</td>
<td>MLSpatialMapper, MLSceneOptimizerBehavior</td>
</tr>
<tr>
<td>Comfortable articulating:</td>
<td></td>
</tr>
<tr>
<td>○ Value of MLTK and its role in the tooling ecosystem</td>
<td></td>
</tr>
<tr>
<td>○ Value of this workshop to colleagues and managers</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Higher Order Objectives</th>
<th>Post-Workshop Paths</th>
</tr>
</thead>
<tbody>
<tr>
<td>Able to leverage MLTK code in new C# MonoBehaviours</td>
<td>Remaining MLTK features</td>
</tr>
<tr>
<td>Apply Best Known Practices for MR Interaction using MLTK</td>
<td>Zero Iteration workflows</td>
</tr>
<tr>
<td>Habitually give and receive help via Discord</td>
<td>Explore other Discord channels</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prerequisite Knowledge</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Intermediate Unity (Prefabs, MonoBehaviors, and Components)</td>
<td></td>
</tr>
<tr>
<td>Basic C# with learner’s preferred IDE</td>
<td></td>
</tr>
<tr>
<td>Foundational Spatial Computing Experience</td>
<td></td>
</tr>
<tr>
<td>○ Completed HelloCube on a Magic Leap headset</td>
<td></td>
</tr>
<tr>
<td>○ Prior experience with Unity XR/AR</td>
<td></td>
</tr>
</tbody>
</table>
Magic Leap
Online Workshop
Rapidly prototype a 3D visualization app

April 2nd

Join us on Discord: discord.gg/VSUWSMM
Have you joined us on Discord yet?
discord.gg/VSUWSMM
Have you joined us on Discord yet?
discord.gg/VSUWSMM
Join us on Discord: discord.gg/VSUWSMM

#workshop-general
Drop a meme and say hello

#workshop-feedback
Fill out the Workshop Survey and let us know we can improve for next time in the chat.

#workshop-help
Having trouble? This is your first line of support.

Workshop help 1, 2, 3
These are audio-only channels. Our Mentors may direct you here to talk through an issue as a group.
Join us on Discord: discord.gg/VSUWSMM

Pinned Links
Important resources are pinned to each channel. Click the Pin icon for easy reference.

[ML] Mentors
Users with the [ML] prefix are from the Magic Leap team, here to answer questions and provide support!
ENVIRONMENT SETUP
Workshop Resources

**Unity 2019.3 Project Template**
Open via Unity Hub

**.unitypackage file**
Import into the Unity Project Template

**MLTK**

**Workshop Assets**

- **.unitypackage file**
  Import into the Unity Project Template
  *MLTK should be imported first otherwise you may see broken references*

Join us on Discord: discord.gg/VSUWSMM
Common Environment Setup Gotchas

**Lumin Build Target**
- *File > Build Settings*
- Select *Lumin* and then click *Switch Platform*

**Lumin SDK v0.24.1**
- *Preferences > External Tools*
- Set *Lumin SDK* to:
  
  `<username>/MagicLeap/mlsdk/v0.24.1`

**Certificate & Private Key**
- *Project Settings > Player*
- Select the Lumin tab
- Expand *Publishing Settings*
- Select your ML Certificate

Join us on Discord: discord.gg/VSUWSMM
(re)Generating a Certificate

**Urgent:** The recent Lumin OS 0.98.10 update requires that you generate new certificates. Certificates generated before 3/15/2020 will not work.

### Certificates

<table>
<thead>
<tr>
<th>Name</th>
<th>Issued</th>
<th>Expiration</th>
<th>Status</th>
<th>Revoke</th>
<th>Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>kedar_msistalth</td>
<td>5/3/2019</td>
<td>5/2/2020</td>
<td>Active</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kedar_mlia4667</td>
<td>7/28/2019</td>
<td>7/27/2020</td>
<td>Active</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

developer.magicleap.com > Publish > Certificates
Magic Leap Workshop Troubleshooter

MLTroubleshooter.unitypackage is pinned in #workshop-help

Once imported, launch from the menu:
Magic Leap > Troubleshoot Project

Join us on Discord: discord.gg/VSUWSMM
Have you joined us on Discord yet?
discord.gg/VSUWSMM
Have you joined us on Discord yet?

discord.gg/VSUWSM
Today’s Workshop:
MLTK Brain App

Join us on Discord: discord.gg/VSUWSMM
Meet the Team

Kedar Shashidhar

Nate Aschenbach

Today’s Host
Developer Evangelist
naschenbach@magicleap.com
@inventonater

Today’s Primary Instructor
Developer Evangelist
kshashidhar@magicleap.com
@kedarshashi

[ML] in #workshop-help
- [ML] Slukas
- [ML] Tricia
- [ML] Filip
- [ML] Josh N
- [ML] Shane Engelman
- [ML] Daniel
- [ML] Alicia
- [ML] Chris R

Join us on Discord: discord.gg/VSUWSMM
Agenda

Environment Set Up
Let’s make sure everyone is ready to rock.

What is Magic Leap Tool Kit?
A high level introduction and map of the current tools.

MLTK-Brain Example
MLTK applied to a medical 3D visualization application.

Wrap Up
Recap what we learned and get ready for next time.

Let us know if we are going too fast!

Join us on Discord: discord.gg/VSUWSMM
WHAT IS MLTK?
What is the Magic Leap Tool Kit?

Magic Leap Toolkit is a collection of Components and Prefabs delivered as a .unitypackage file. These Prefabs provide developers with useful reusable tools that solve specific real-world problems or to extend functionality for developing Magic Leap apps.

Join us on Discord: discord.gg/VSUWSMM
Today...
We will learn about these features

- **Control Input**
  Unity event binding and interaction for all Control input events.

- **Control Pointer**
  A spatial targeting, selecting, and manipulation system that displays weight and other physical characteristics

- **Keep In Front**
  Keeps digital content in users view

- **Place on Floor**
  Provides a starting position for an app's main content without user input or complex setups

Join us on Discord: discord.gg/VSUWSMM
Today...
We will learn about these features

- **Control Input**
  Unity event binding and interaction for all Control input events.

- **Control Pointer**
  A spatial targeting, selecting, and manipulation system that displays weight and other physical characteristics

- **Keep In Front**
  Keeps digital content in users view

- **Place on Floor**
  Provides a starting position for an app’s main content without user input or complex setups

Not today...
Please explore these other features on your own!

- **Interactive Objects New!**
  A set of user inputs and objects that are part of the HandInput system

- **Hand Input New!**
  A plug-and-play tool that provides stable, smooth hand tracking for keypoints in any hand pose

- **Ramp Light**
  A shading technique which maximizes the visual quality of Magic Leap’s additive display.

- **Transmission**
  A cross-platform, multiplayer solution for connecting devices over LAN

- **Spatial Alignment**
  Visual alignment for peers connected with Transmission

- **Runtime Console**
  Viewing log messages while running an app

- **SimpleHandPointer New!**
  An alternative to the Control Pointer to enable hand input

- **Surface Details**
  A solution to identify different surface types

Join us on Discord: discord.gg/VSUWSMM
Keep In Front is a component that keeps content in the user view while respecting other objects in the scene and attempts to stay in front of them.

Join us on Discord: discord.gg/VSUWSMM
Place On Floor helps locate an area on the floor of the user’s space where an app can place digital content.
Control Input component provides you with a comprehensive set of Unity events for interactivity with the Magic Leap Control.
Control Pointer provides a pointer for the manipulation and movement of digital objects with the Control. Control Pointer includes components that let users target, select/deselect, and drag/drop objects.
Workshop Time
Models
- Brain.fbx - model

Prefabs
- Completed prefabs for each lesson module
- User Interfaces

Scenes
- Starter Scenes
- Completed Scenes for each lesson modules

Scripts
- Completed Logic Scripts for this project
## Completed Scenes & Associated Prefabs

<table>
<thead>
<tr>
<th>Scene Name</th>
<th>Associated Prefab</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLW.Brain.001.PlaceOnFloor -- Starter.unity</td>
<td>No associated prefab</td>
</tr>
<tr>
<td>MLW.Brain.001.PlaceOnFloor -- Complete.unity</td>
<td>BrainPrefab.001.PlaceOnFloor</td>
</tr>
<tr>
<td>MLW.Brain.003.FinesseControl -- Complete.unity</td>
<td>BrainPrefab.003.FinesseControl</td>
</tr>
<tr>
<td>MLW.Brain.004.UIControl -- Complete.unity</td>
<td>BrainPrefab.004.UIControl</td>
</tr>
</tbody>
</table>
Place On Floor
Add Brain Model to Our Scene

1. Open the MLW.001.PlaceOnFloor -- Starter Scene
2. Create a new gameobject and rename it to "[CONTENT]" and place it at the origin
3. Add the BrainPefab.001.PlaceOnFloor Prefab from the Assets/Prefabs folder to your Scene under the [CONTENT] Game Object in your Hierarchy
   a. Set position transform to (0,1,0)
   b. Set scale transform to (0.25,0.25,0.25)
4. Rebuild the Lighting. Open the lighting menu under Windows > Rendering > Lighting Settings. Scroll to the bottom of the window and hit the generate lighting button.
Set Up A Starting Place for our Application

1. Under “[UTILITIES]” create a new gameobject called “PlaceOnFloor” and place it at the origin
2. Add the component PlaceOnFloor to the gameobject
3. Create a new gameobject and rename it to “[GUI]” and place it at the origin
4. Add the prefab PlacementInstructions from Assets/Prefabs/ as a child of “[GUI]”
5. Add the component KeepInFront to the prefab
6. In the PlaceOnFloor gameobject:
   a. Add a reference to PlacementInstructions in the Instructions field
   b. Add a reference to [CONTENT] in the content field
MLW.001.PlaceOnFloor -- Starter
Build to Device / Run in Zero Iteration / Questions

Review of MLW.001.PlaceOnFloor

1. Removed default content in starter scene
2. Added the Brain Prefab to the scene
3. Created a reference to global [CONTENT] gameobject in PlaceOnFloor component
Control Pointer
Add a Control Pointer to the Scene

1. Create an empty game object in the Hierarchy and rename it as [INPUT]
   a. Set its transform to be at the origin.

2. Drag the ControlPointer prefab from /Assets/MagicLeap-Tools/Prefabs/Input into the Hierarchy as a child of the [INPUT] gameobject.
   a. Set the following parameters on your Pointer Script

| Parameter                | Value  
|--------------------------|--------
| Max Distance             | 2.5    
| Min Distance             | 0.5    
| Bendy Weight Multiplier  | 0.25   
| Bend Point Percentage    | 0.25   
| Bend Prediction Multiplier| 10     |
Adding Pointer Intractability to the Brain Model

Select all the children objects of the Brain Prefab. Add the following components and properties

1. Add a Mesh Collider component
   a. Check “Convex” in inspector
   b. Add “PointerInteractablePhysicMaterial” to the colliders Material Field

2. Add a Pointer Receiver component
   a. This will automatically add a rigidbody
   b. Check “Is Kinematic”

3. Add Input Feedback component
   a. Add audio clip references for each sound sample.
Add Rotatability and Scalability to All Interactable Objects.

1. Select all children components of the Brain Prefab and add a “Rotation Manipulator” and “Scale Manipulator” Script. This allows the use of:
   - The radial dial to rotate objects when selected
   - Left and Tight tap on the Touchpad to scale up and down
   - Force Press on the Touchpad to Reset the scale
Setting Up Double Tap Home to Exit App

1. In your scripts folder create a script called “ExitOnHome”
   a. Add a single function called ExitApp with the following code.

2. Add the ExitOnHome script as a component on the ControlPointer gameobject
   a. Select ExitOnHome.ExitApp as the function to call.

```csharp
public void ExitApp()
{
#if UNITY_EDITOR
    Debug.Log("Exit to Home: Unity Editor Quit");
    UnityEditor.EditorApplication.isPlaying = false;
#else
    Application.Quit();
#endif
}
```
Build to Device / Run in Zero Iteration / Questions

Review of MLW.002.ControlPointer

1. Added a controller pointer prefab to your scene and control parameters
2. Created a script ExitOnHome.cs that exits your application
3. Added an event callback in ControllInput to the ExitOnHome.cs Script on Double Tap Home
4. Enabled basic pointer interaction to each brain game object by adding colliders & MLTK scripts
5. Enabled rotation and scale manipulation to each brain game object by adding MLTK scripts
03

Finesse Control
Adding a way to globally control the brain model

1. Right Click on the Brain Prefab in your scene hierarchy and select “Open Prefab Asset”
2. Right click on the root gameobject and navigate to 3D Object > Sphere to add a sphere gameobject to your prefab
   a. Set position transform to (0,-0.5,0)
   b. Set scale transform to (0.2,0.2,0.2)
   c. Drag all brain game objects to be a child of the sphere
3. Add the following components to your Sphere.
   a. Pointer Receiver
   b. Rotation Manipulator
   c. Scale Manipulator
   d. Input Feedback
   e. MSA Source
4. Set “Is Kinematic” in Rigidbody to True
5. Add “PointerInteractablePhysicMaterial” to Sphere Collider
6. Add audio clip references in Input Feedback
Resetting the Brain using the Home Button

1. Create a script called ResetChildren.cs

2. Add the following private member variables

   ```csharp
   //Setting the following private member variables
   private Vector3[] _originalPositions;
   private Quaternion[] _originalRotations;
   private Vector3[] _originalScales;
   private bool _initialized = false;
   ```

3. Add the following functionality in Start()

   ```csharp
   void Start()
   {
   if (transform.childCount > 0)
   {
   _originalPositions = new Vector3[transform.childCount];
   _originalRotations = new Quaternion[transform.childCount];
   _originalScales = new Vector3[transform.childCount];

   for(int i = 0; i < transform.childCount; i++)
   {
   _originalPositions[i] = gameObject.transform.GetChild(i).localPosition;
   _originalRotations[i] = gameObject.transform.GetChild(i).localRotation;
   _originalScales[i] = gameObject.transform.GetChild(i).localScale;
   }
   _initialized = true;
   }
   ```
Resetting the Brain using the Home Button

1. In `ResetChildren.cs` add a function `ResetChildTransforms()`

   ```csharp
   public void ResetChildTransforms()
   {
       if(_initialized)
       {
           for (int i = 0; i < transform.childCount; i++)
           {
               Transform child = gameObject.transform.GetChild(i);
               child.localPosition = _originalPositions[i];
               child.localRotation = _originalRotations[i];
               child.localScale = _originalScales[i];
           }
       }
   }
   ```

2. Add `ResetChildren.cs` as a component on the `Sphere` gameobject

3. In your `ControlPointer` gameobject in the `ControlInput` component, add a callback to On Home Button Tap()
Adding Audio Spatialization to Interaction Sounds

1. On the **Main Camera** gameobject under rendering. Add an **MSA Listener** Component

2. Select all interactable brain components:
   a. Add an **MSA Source** Component
   b. On the **Audio Source**:
      i. Check the **Spatialize** box
      ii. Set **Spatial Blend** parameter to 3D
Build to Device / Run in Zero Iteration / Questions

Review of MLW.003.FinessePointer

1. Modified the Brain Prefab in the scene with a global sphere object that parents brain game objects
2. Added full pointer interaction to the Sphere gameobject
3. Created a ResetChildren.cs script that resets the transforms of all children gameobjects
4. Added an event callback in ControllInput to ResetChildren on Single Tap Home
5. Added audio spatialization to input feedback sounds
UI Control
Adding UIs to each Brain Component

1. In Assets/Prefabs/BrainUIs drag each brain component UI to the hierarchy as a child of each brain component model.
   a. Disable all UI components
Displaying and Hiding the UIs with the Control Bumper

1. Create a script called **ShowUI.cs**
2. Add MagicLeapTools as a namespace.
3. Add the following public and private member variables

   ```csharp
   using MagicLeapTools;
   private bool _enabled = false;
   public PointerReceiver pointer;
   ```

4. Create Two New Functions, EnableUI() and HideUI()

   ```csharp
   public void EnableUI(){
     if(pointer != null){
       if (pointer.Dragging){
         _enabled = !_enabled;
         gameObject.SetActive(_enabled);
       }
     }
     else{
       Debug.Log("Null Pointer in Parent");
     }
   }
   
   public void HideUI(){
     _enabled = false;
     gameObject.SetActive(_enabled);
   }
   ```
Displaying and Hiding the UIs with the Control Bumper

1. Inside `ResetChildren.cs` make the following modification to the `ResetChildTransforms()` function

```csharp
public void ResetChildTransforms()
{
    if (_initialized)
    {
        for (int i = 0; i < transform.childCount; i++)
        {
            Transform child = gameObject.transform.GetChild(i);
            child.localPosition = _originalPositions[i];
            child.localRotation = _originalRotations[i];
            child.localScale = _originalScales[i];

            ShowUI_Complete showUI = child.GetComponentInChildren<ShowUI_Complete>();
            if (showUI != null)
            {
                showUI.HideUI();
            }
        }
    }
}
```
Displaying and Hiding the UIs with the Control Bumper

1. Select all 6 BrainUI Objects. Add ShowUI as a script to all of them.
   a. For each ShowUI component add a reference to the "PointerReceiver" component in its parent game object.
Displaying and Hiding the UIs with the Control Bumper

1. In the ControlPointer game object on the ControlInput component, add callbacks for each of the ShowUI objects and their respective EnableUI() functions.
Adding Tooltip Instruction Displays on the Control

1. In Assets/Prefabs/ControllerTooltips drag all three tooltip prefabs to be children of the ControlPointer gameobject.
Build to Device / Run in Zero Iteration / Questions

Review of MLW.004.UIControl

1. Added Brain UI prefabs to each brain gameobject
2. Created a ShowUI script that enables, disables, and hides the UI components
3. Modified the ResetChildren script to disable all UIs on home button press
4. Added the ShowUI script as a component on each brain UI
5. Added callback references to ShowUI script on bumper press in Control Input
6. Added Controller Tooltip prefabs as children of the ControlPointer gameobject
WRAP UP
**Today...**
We learned about these features

- **Control Input**
  Unity event binding and interaction for all Control input events.

- **Keep In Front**
  Keeps digital content in users view

- **Place on Floor**
  Provides a starting position for an app’s main content without user input or complex setups

- **Control Pointer**
  A spatial targeting, selecting, and manipulation system that displays weight and other physical characteristics

**Not today...**
Please explore these other features on your own!

- **Interactive Objects New!**
  A set of user inputs and objects that are part of the HandInput system

- **Hand Input New!**
  A plug-and-play tool that provides stable, smooth hand tracking for keypoints in any hand pose

- **Playspace New!**
  Users define an area in their space for use in an app

- **SimpleHandPointer New!**
  An alternative to the Control Pointer to enable hand input

- **Ramp Light**
  A shading technique which maximizes the visual quality of Magic Leap’s additive display.

- **Transmission**
  A cross-platform, multiplayer solution for connecting devices over LAN

- **Spatial Alignment**
  Visual alignment for peers connected with Transmission

- **Runtime Console**
  Viewing log messages while running an app

- **Surface Details**
  A solution to identify different surface types
Help us Improve!
Link to the survey is pinned in #workshop-feedback

forms.gle/jz7Mt9W7bGvQUpq97
SPRING 2020 #MLDEVJAM

magi.ca/djsp20
APPLY FOR ACCESS HARDWARE

magi.ca/hardware
JOIN THE NIST CHARIoT CHALLENGE

Submit your idea for the Build Augmented Reality Interfaces for First Responders Contest by:
May 6, 2020

Go to chariotchallenge.com to learn more and register for the informational webinar on April 14th, 2020 at 11am MT

Take Action
Learn more at chariotchallenge.com

magi.ca/hardware
STAY CONNECTED!

Developer Forums:  
forum.magicleap.com

Twitter:  
@magicleapdevs

Discord:  
discord.gg/VSUWSMM
SEE YOU
NEXT TIME!