

Github link: <https://github.com/weibaozi/AR-VR-Guidance-of-Continuum-Manipulator>

The project contains two main components: The Manipulator and The guiding.

- The manipulator part contains the manipulator itself and the dashboard including radar and level bars objects.
- The guiding part contains the patient model, plane, and guiding points.

Next, I will explain the details of how I implement each component.

- Continuum Manipulator:
 - The model of the snake part is given by Prof. Alejandro. The drill part is from a free 3d asset website. We assembled the manipulator so that each join is a child object from the previous object.:



- This helps to bend the manipulator easily by rotating each component to a certain degree simultaneously.
 - We used the OpenXR toolkit to get input from Quest3. We used the joystick to control the bend using the preset OpenXR inputsystem.
 - Initially, we bind the manipulator to the right hand but we found out this won't allow the manipulator to collide with other components. Therefore we changed the strategy. We used the OpenXR grab scripts to grab and control the manipulator instead and track the controller's velocity instead of location so that the manipulator can collide with other components.
- Radar:
 - The radar part simply found the local coordinates of the next guiding point in the manipulator tool tip's coordinates. And use the x and y coordinates to move the dot in the radar. Also, there is a constraint for the dot to stay on the radar.
- Horizontal Bar:
 - Similar to radar but this is just the local coordinates of the tooltip instead. Since we bend the tooltip we need a relative coordinate for the tooltip to let the user know when and how to bend the manipulator.
- Vertical Bar:
 - This uses the world coordinates to find the displacement of the y-axis between the manipulator and the next guiding point.
- Guiding points:
 - Predefined spheres, when the tooltip collides with the spheres, it deactivates.
- Insert point:

- A red cross indicates where to drill. The red cross is at a plane with a mesh collider. But there is a hole at the inert point so that the manipulator can be constrained at the insert point.
- Guiding manipulator:
 - A copy of the manipulator model to help the user align with it at the beginning. I used the “Quaternion.LookRotation” built-in function to find the orientation of the manipulator to the first guiding point.
- Scene management:
 - I set a menu scene to show a tutorial and choose which level the user wants to play using Unity SceneManager.