All Implemented Functions

return_Axis_Amgle(R)

- Purpose: Calculates the principal axis and angle from a given rotation matrix R.
- Inputs:
  - R (numpy array): A 3x3 rotation matrix.
- Outputs:
  - Axis (numpy array): The principal axis of rotation.
  - Angle (float): The rotation angle in radians.

Euler_Angles_to_Rot(yaw, pitch, roll)

- Purpose: Converts Euler angles (yaw, pitch, roll) into a rotation matrix.
- Inputs:
  - yaw (float): Rotation around the Z-axis (radians).
  - pitch (float): Rotation around the Y-axis (radians).
  - roll (float): Rotation around the X-axis (radians).
- Outputs:
  - Rotation matrix (numpy array): A 3x3 matrix.

rotMa_To_Yaw_Pitch_Roll(matrix)

- Purpose: Extracts Euler angles from a given rotation matrix.
- Inputs:
  - matrix (numpy array): A 3x3 rotation matrix.
- Outputs:
  - Tuple of yaw, pitch, and roll (floats).

vector_to_RotMa(vector)

- Purpose: Converts a rotation vector to a rotation matrix.
- Inputs:
  - vector (numpy array): A 3D rotation vector.
- Outputs:
  - Rotation matrix (numpy array): A 3x3 matrix.

Create_RotMa(theta, axis_vector)

- Purpose: Generates a rotation matrix for a given axis and angle.
- Inputs:
  - theta (float): Angle of rotation (radians).
  - axis_vector (numpy array): Axis of rotation.
- Outputs:
  - Rotation matrix (numpy array): A 3x3 matrix.

Quat_RotMa(Q)

- Purpose: Converts a quaternion into a rotation matrix.
- Inputs:
  - Q (list): A quaternion [w, x, y, z].
- Outputs:
  - Rotation matrix (numpy array): A 3x3 matrix.

Two_Vectors_To_One(m0, m1)

- Purpose: Computes a quaternion for the rotation from vector m0 to m1.
- Inputs:
    - m0 and m1 (numpy arrays): Initial and final 3D vectors.
- Outputs:
    - Quaternion (numpy array): [w, x, y, z].

Magnitude(vector)

- Purpose: Calculates the magnitude of a given vector.
- Inputs:
    - vector (numpy array): Input vector.
- Outputs:
    - Magnitude (float).

Create_A_Vector(x, y)

- Purpose: Maps screen coordinates to a 3D vector based on a projection model.
- Inputs:
    - x and y (floats): Screen coordinates.
- Outputs:
    - 3D vector (numpy array).

Rotation_Actualitation(R)

- Purpose: Updates the GUI elements based on the current rotation matrix R.
- Inputs:
    - R (numpy array): A 3x3 rotation matrix.
- Outputs: None (modifies the GUI directly).
- Description:
  This function takes the rotation matrix R and updates all the interface fields to display the corresponding:
    - Axis and angle of rotation.
    - Rotation vector.
    - Euler angles (yaw, pitch, roll).
    - Quaternion values.

apply_AA()

- Purpose: Applies a rotation to the object using the axis-angle parameters entered by the user in the GUI.
- Inputs: None (reads values from the GUI).
- Outputs: None (applies the rotation and updates the GUI).
- Description:
    - Retrieves the axis and angle from the GUI fields.
    - Uses these values to calculate a rotation matrix.
    - Updates the rotation matrix of the object and refreshes the interface.

apply_rotV()

- Purpose: Applies a rotation to the object using a rotation vector entered by the user in the GUI.
- Inputs: None (reads values from the GUI).
- Outputs: None (applies the rotation and updates the GUI).
- Description:
    - Reads the rotation vector values from the GUI.
    - Converts the rotation vector to a rotation matrix.
    - Updates the object's rotation matrix and synchronizes the interface.

apply_EA()

- Purpose: Applies a rotation to the object using Euler angles (yaw, pitch, roll) entered by the user in the GUI.
- Inputs: None (reads values from the GUI).
- Outputs: None (applies the rotation and updates the GUI).
- Description:
    - Retrieves Euler angles from the GUI and converts them to radians.
    - Converts the angles into a rotation matrix.
    - Updates the object's rotation and refreshes the GUI.

apply_quat()

- Purpose: Applies a rotation to the object using quaternion values entered by the user in the GUI.
- Inputs: None (reads values from the GUI).
- Outputs: None (applies the rotation and updates the GUI).
- Description:
    - Reads the quaternion values from the GUI fields.
    - Converts the quaternion into a rotation matrix.
    - Updates the object's transformation matrix and refreshes the GUI.