Helical Gear System - Report/ User Manual

Table of Contents

Abstract

This Maya Python scripting project aims to develop a tool for creating and animating helical gears in Autodesk Maya. The project highlights customizable parameters and a user interface for controlling animations. The script enables real-time gear generation and animation, showcasing procedural modeling and animation techniques.

Background

Procedural modeling and animation are essential in computer animation, allowing for automated creation and motion of complex models. This project draws inspiration from the need for efficient gear creation in mechanical animations, demonstrating the educational and creative benefits of Python scripting in Maya.

Design and Architecture

It’s structured into three main modules: Gear Creation, Animation, and User Interface (UI). The Gear Creation module handles procedural generation, the Animation module manages gear rotation using Maya expressions, and the UI module provides a user-friendly interface for parameter adjustments. This modular design ensures scalability and ease of maintenance.

Implementation

- ### Program Structure

  The script includes initialization, a main loop for user input processing, and cleanup phases.

- ### Implementation of Algorithms and Techniques

  Key techniques include procedural modeling for gear creation, Maya expressions for animation, and a user interface built with Maya's GUI elements for real-time control.

- ### Flow of Control

  The user launches the script, sets parameters via the UI, and initiates the animation. The script generates gears, positions them, and applies color and animation based on user input. Real-time updates allow for immediate feedback.

- ### Important Algorithms and Techniques

  - **createHelicalGear()**: Implements the algorithm for generating helical gears with user-defined parameters, utilizing procedural modeling techniques to create the gear geometry and extrusion operations for forming gear teeth.

  - **animateGears()**: Utilizes Maya expressions to animate the rotation of gears, applying mathematical algorithms to control the rotation speed and direction based on user input.

  - **createUI()**: Develops the user interface using Maya commands, employing GUI design techniques to create an intuitive interface for controlling gear animation parameters.

  - **animateGearsCmd()**: Coordinates user input and function calls, employing event-driven programming techniques to respond to user actions and trigger gear creation and animation algorithms.

How to Use

1. Load and Run: Open Maya, load the provided script in the script editor, and run it.
2. UI Interaction: Set the **Rotation Speed** and **Animation Duration**, choose the **Gear Color**, and click **Animat**e to create and animate the gears.
3. **Adjust Parameters**: Modify the parameters in the UI and click **Animate** again to see updates in real-time.

Results

- ## Images and Screenshots

Figure 1: User interface for controlling gear animation.

```
def createUI():
    '''Simple UI for controlling gear animation.'''
    if cmds.window("gearAnimationUI", exists=True):
        cmds.deleteUI("gearAnimationUI")

    cmds.window("gearAnimationUI", title="Gear Animation UI")
    cmds.columnLayout(adjustableColumn=True)

    cmds.text(label="Animation Settings", align="center")

    cmds.separator(style="none", height=10)

    cmds.intFieldGrp("rotationSpeedField", label="Rotation Speed", value1=10)
    cmds.intFieldGrp("durationField", label="Animation Duration", value1=10)
    cmds.colorSliderGrp("gearColorSlider", label="Gear Color")

    cmds.separator(style="none", height=10)

    cmds.button(label="Animate", command=animateGearsCmd)
    cmds.showWindow("gearAnimationUI")
```

Figure 2: Helical gears generated with the script.

```
def createHelicalGear(numTeeth, gearHeight, gearRadius, gearThickness, teethHeight, rotationSpeed, scale=1, initialRotation=0):
    '''Create a helical gear model with specified parameters.'''
    # Creates the gear model
    gear = cmds.polyPipe(sa=numTeeth * 2, h=gearHeight, r=gearRadius, t=gearThickness, name="gear")
    intStartFace = numTeeth * 2 * 2
    intEndFace = numTeeth * 2 * 3 - 1

    # Selects faces for extrusion
    cmds.select(clear=True)
    for i in range(intStartFace, intEndFace, 2):
        cmds.select(gear[0] + ".f[%d]" % i, add=True)

    # Extrudes the selected faces
    cmds.polyExtrudeFacet(ltz=teethHeight * 0.2, lsx=1)
    cmds.polyExtrudeFacet(ltz=teethHeight * 0.8, lsx=0.5)

    # Sets attributes and clean up history
    cmds.select(gear[0])
    cmds.addAttr(longName='gearRadius', shortName='gr', attributeType='float')
    cmds.addAttr(longName='teeth', shortName='tee', attributeType='short')
    cmds.setAttr(gear[0] + '.gr', gearRadius)
    cmds.setAttr(gear[0] + '.tee', numTeeth)

    # Scales the gear
    cmds.scale(scale, scale, scale, gear[0])

    # Sets initial rotation
    cmds.setAttr(gear[0] + ".rotateY", initialRotation)

    return gear
```

Figure 3: Gears in motion, animated using the script.

```python
def animateGears(gear1, gear2, gear3, gear4, rotationSpeed):
    '''Animate the rotation of gears.'''
    cmds.expression(name="gear1RotationExp", alwaysEvaluate=True, s="{0}.rotateY = time * {1}".format(gear1, rotationSpeed))
    cmds.expression(name="gear2RotationExp", alwaysEvaluate=True, s="{0}.rotateY = -time * {1}".format(gear2, rotationSpeed))
    cmds.expression(name="gear3RotationExp", alwaysEvaluate=True, s="{0}.rotateY = time * {1}".format(gear3, rotationSpeed))
    cmds.expression(name="gear4RotationExp", alwaysEvaluate=True, s="{0}.rotateY = -time * {1}".format(gear4, rotationSpeed))
```

Conclusion

The project achieved its goals, developing a versatile tool for procedural modeling and animation in Maya.

References

- Autodesk Maya Documentation: https://knowledge.autodesk.com/support/maya

- Python Documentation: https://docs.python.org/3/

- Maya Python Commands: https://download.autodesk.com/us/maya/2011help/CommandsPython/