A purple and blue text

AI-generated content may be incorrect.

Progress Report 2

Date – 16.02.2025 to 24.02.2025

Intern Name – Poorna Bhati

Task Accomplished –

1. Successfully opened and analyzed the code files in Visual Studio Code.
2. Conducted an in-depth analysis of the key code files, including:

* Code responsible for initiating the D3 Dual Arch tool.
* The main code file related to the articulator functionality of the tool.
* Additional supporting code files associated with the articulator module.

1. Explored the D3 Splint tool and performed a comparative analysis with the D3 Dual Arch tool to understand their functionalities and differences.

Home Supervisor: Host Supervisor:

Dr. Rajlakshmi Nayak Dr. Abu Syed Kabir

Associate Professor, Associate Professor,

Institute of Engineering & Technology, Department of Mechanical &

JK Lakshmipat University, Jaipur Aerospace Engineering,

Carleton University, Canada

**1: Open all the code files of D3 tool in VS code.**

Open folder “D3DualArch\_v0.0.7\_prerelease” 🡪 open folder “2.79” 🡪 open folder “python” 🡪 open folder “lib”.

1. “lib” contains many folders which include the source code of the tool and code files.

**2: Analysing the source code files.**

* Starting the D3 dual arch tool.

Open the folder “2.79” go to “scripts” then open the “ addons” in addons open the folder with the name “**d3 guard**”.

In the “d3 guard” folder we have a code file with the name “**\_init\_.py**” this file is responsible for starting the D3 Dual Arch tool in Blender.

The code responsible for opening or starting the D3 Dual Arch tool in Blender is primarily located in the register function. This function registers various classes, operators, and handlers with Blender, making the add-on's functionality available within the Blender interface.

When the add-on is enabled in Blender, this function is called, which sets up the add-on and makes its tools available in the Blender interface.

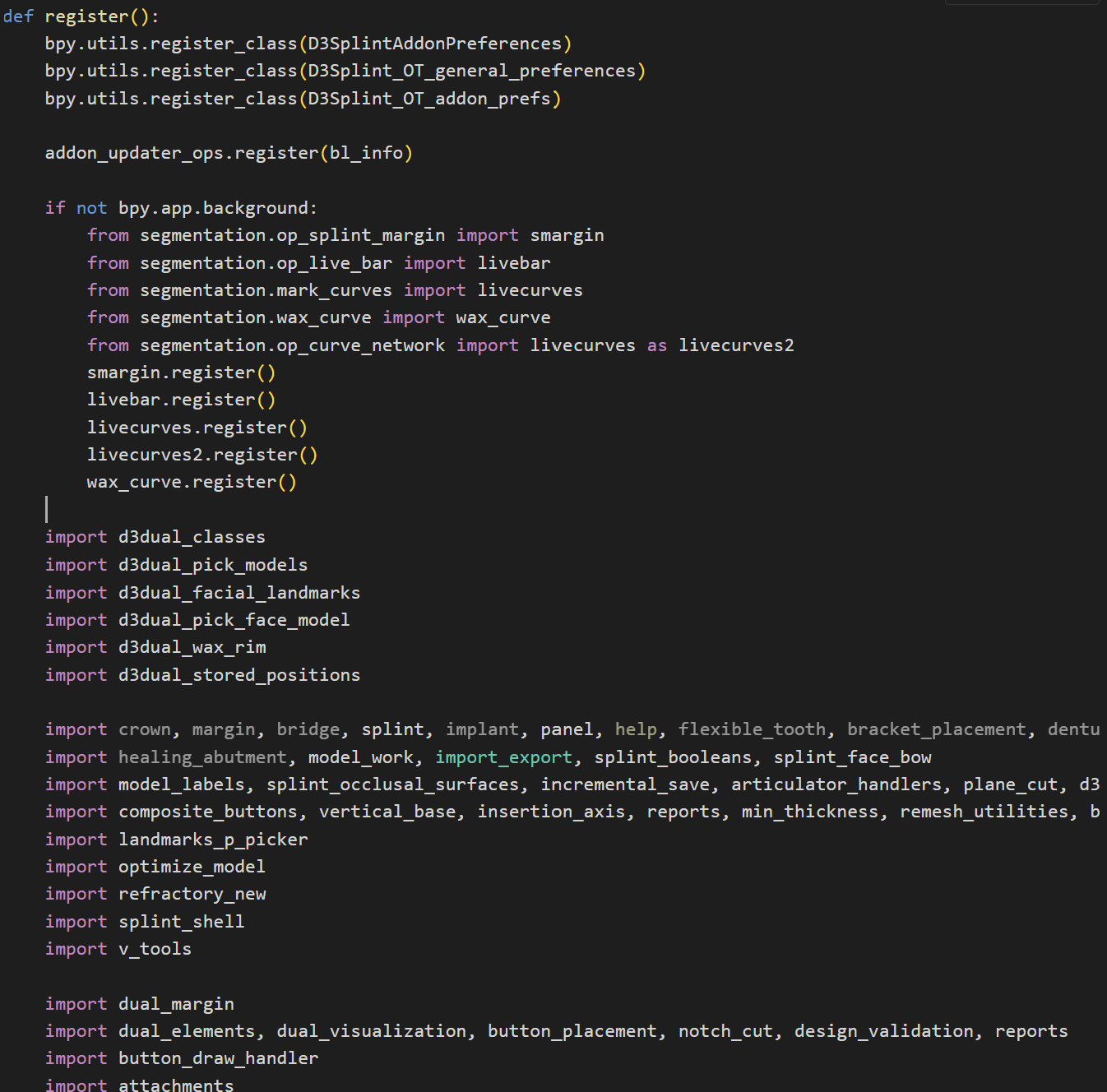


Figure 1: Snippet1 of code responsible for starting D3 dual arch tool.



Figure 2: Snippet2 of code responsible for starting D3 dual arch tool.

* Loading the articulator.

The file “articulator.py” manages the creation, manipulation, and animation of a virtual dental articulator within Blender. An articulator is a device used in dentistry to simulate the movement of the jaw, which helps in designing dental prosthetics and assessing occlusion.

The main tasks performed by this file:

1. Creating the Articulator:

The D3DUAL\_OT\_generate\_articulator operator creates or modifies an articulator based on user-defined parameters such as condyle width, condyle angle, and guidance angles.

1. Adjusting Articulator Components:

Functions like adjust\_bow\_size and link\_articulator\_decoration adjust the size and position of various components of the articulator based on the specified parameters.

1. Animating the Articulator:

The D3DUAL\_OT\_generate\_articulator\_keyframes operator generates keyframes for the articulator's animation, simulating different jaw movements such as protrusive, right excursion, left excursion, and full envelope motion.

1. Updating Parameters in Real-Time:

The D3DUAL\_OT\_live\_articulator\_parameters operator allows users to change the parameters of the articulator in real-time and see the effects immediately.

1. Managing Functional Surfaces:

Operators like D3DUAL\_OT\_splint\_create\_functional\_surface and D3SPLINT\_OT\_splint\_reset\_functional\_surface manage the creation and resetting of functional surfaces using the envelope of motion on the articulator.

1. Utility Functions:

Various utility functions such as create\_guide\_table, full\_envelope\_math, and protrusive\_math perform calculations and create necessary components for the articulator.

1. Registering and Unregistering Operators:

The register and unregister functions register and unregister all the defined classes and operators with Blender, making the functionality available within the Blender interface.

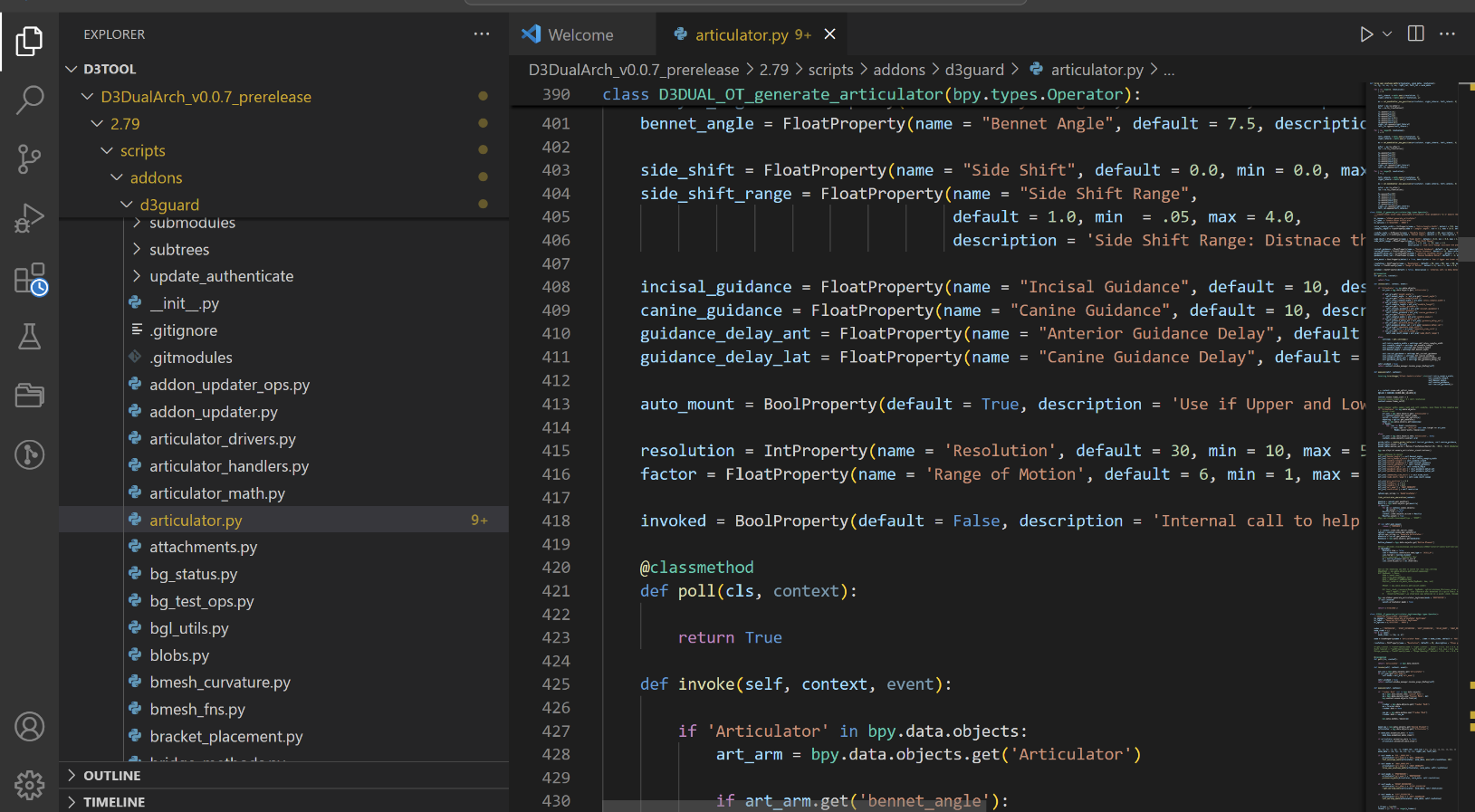


Figure 3: Snippet of articulator source code file.

* Other files related to articulator.

1. “Articulator\_drivers.py”

The articulator\_drivers.py file defines several functions that are used as drivers in Blender to control the animation of the virtual dental articulator. Drivers in Blender are a way to control the properties of objects using Python expressions or functions. This file also registers these functions in Blender's driver namespace so they can be used in the animation system.

**Functions**:

* saw\_tooth(frame): Generates a sawtooth wave pattern with a period of 30 frames. This function can be used to create repetitive linear motion.
* thirty\_steps(frame): Generates a stepped pattern with 30 steps. This function can be used to create discrete steps in the animation.
* full\_envelope\_with\_relax(frame, condy\_length, resolution, use\_relax, relax\_length, right\_left): Calculates the position of the articulator based on the frame number, condyle length, resolution, relaxation length, and direction (right or left). This function simulates the full envelope of motion with relaxation.
* three\_way\_envelope\_l(frame, factor, resolution): Calculates the left-side position of the articulator based on the frame number, factor, and resolution. This function simulates the three-way envelope of motion for the left side.
* three\_way\_envelope\_r(frame, factor, resolution): Calculates the right-side position of the articulator based on the frame number, factor, and resolution. This function simulates the three-way envelope of motion for the right side.

**Driver Namespace Registration**:

* load\_driver\_namespace(): Registers the defined functions in Blender's driver namespace. This makes the functions available for use as drivers in Blender's animation system.

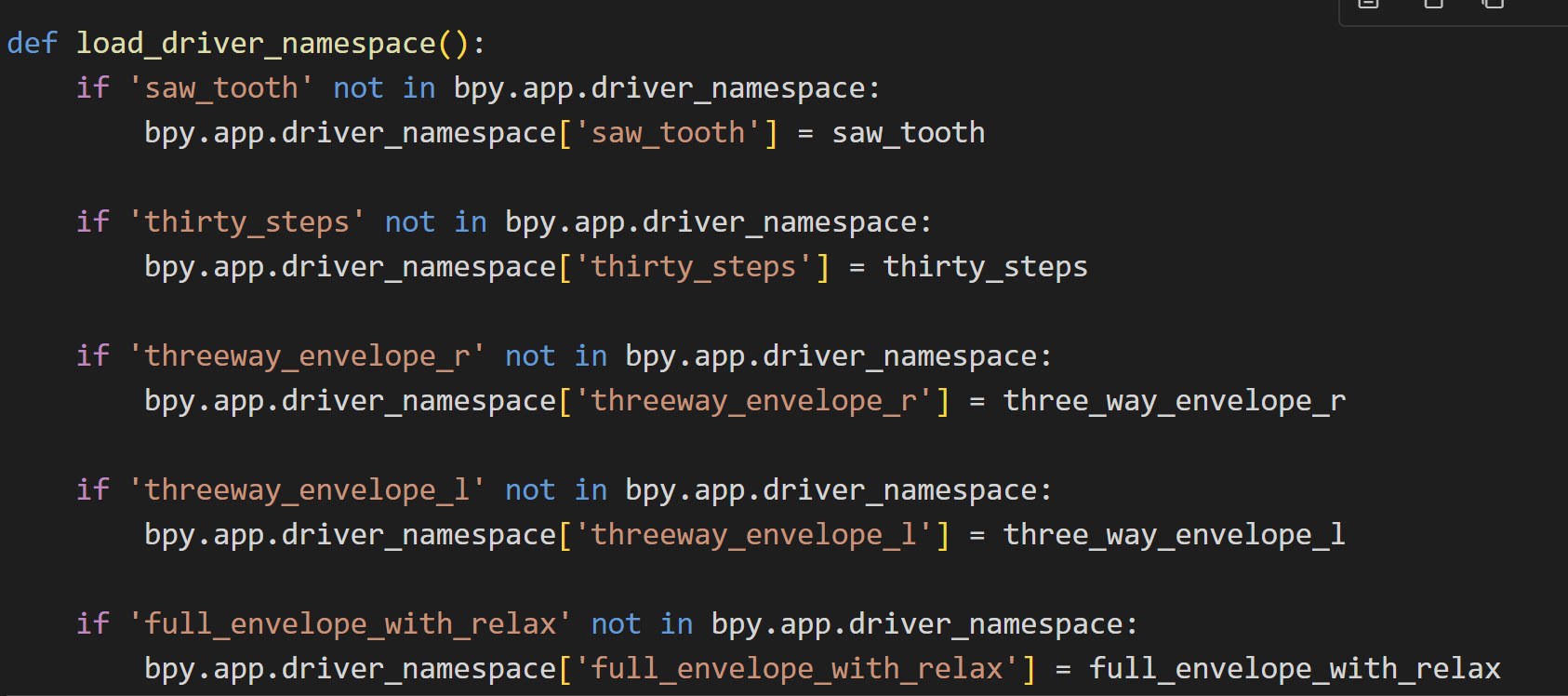


Figure 4: Relevant part of the code that registers the function.

1. “articulator\_handlers.py”

The [articulator\_handlers.py](vscode-file://vscode-app/c:/Users/ASUS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) file is responsible for managing the visualization and metrics of the virtual dental articulator within Blender. It provides functionality to draw visual aids and calculate metrics related to the articulator's position and movement.

The main role of this code file is:

* Drawing Visual Aids: The articulator\_draw function draws lines and text in the 3D view to represent the positions and metrics of the articulator's components.
* Calculating Metrics: The articulator\_metrics\_callback function calculates various metrics related to the articulator's position and movement, such as the lengths of the arms, the Balkwill angle, and the total translation.
* Managing Visibility: The articulator\_visibility function toggles the visibility of the articulator's components based on the visibility of the main articulator object.
* Enabling and Disabling Visualizations: The D3Splint\_OT\_enable\_articulator\_visualizations and D3Splint\_OT\_stop\_articulator\_visualizations operators enable and disable the visualizations by adding and removing the necessary callbacks.
* Handler Management: The clear\_articulator\_handlers function clears the handlers by removing the draw and metrics callbacks.

1. “articulator\_math.py”

The articulator\_math.py file contains mathematical functions and calculations related to the movement and positioning of a virtual dental articulator within Blender. This file is responsible for simulating the dynamics of the articulator, including immediate side shift and progressive side shift, and calculating the positions and rotations of the articulator's components.

The main components of this file are:

**Imports**: Imports necessary modules and functions from Blender (bpy), BMesh (bmesh), Mathutils (Matrix, Vector), and BVHTree (BVHTree).

**Global Variables**:

* incisal\_bvh: A global variable to store the BVH tree of the incisal table mesh for efficient ray casting.

**Functions**:

* intersect\_x\_axis\_sphere(a, b, c, R): Calculates the intersection points of a sphere with the X-axis. This function is used to find the positions of the condyles.
* set\_mandibular\_bow\_position(art\_obj, right\_lateral, left\_lateral, hinge\_opening): Calculates and sets the position of the mandibular bow based on the given parameters. This function simulates the movement of the mandibular bow, including lateral and hinge movements.
* calculate\_matrix(left\_lateral, right\_lateral, hinge\_opening, condyle\_width, condyle\_angle, bennet\_angle, condyle\_track\_length, immediate\_side\_shift, side\_shift\_range): Calculates the transformation matrix for the articulator based on the given parameters. This function simulates the movement of the condyles and calculates the resulting transformation matrix.

**3: Exploring D3 Splint**

D3 Splint is a powerful and versatile tool for dental professionals, offering a comprehensive set of features for designing and analyzing dental splints. Its integration with Blender, combined with its dental-specific tools and advanced modeling capabilities, makes it a unique and valuable addition to any dental practice or lab. Whether used for creating occlusal splints, training dental students, or improving the efficiency of dental labs, D3 Splint provides the tools needed to achieve precise and effective results.

D3 Splint and D3 Dual Arch Tool are both specialized add-ons for Blender, they focus on different areas of dental appliance creation and have distinct features.

**D3 Splint**-

Used for the creation and adjustment of occlusal splints, which are used to protect teeth and manage conditions like bruxism (teeth grinding).

* Occlusal Splint Design:

Tools specifically designed for creating occlusal splints.

Features for adjusting the fit and occlusion of the splint.

* Virtual Articulator:

Simulates jaw movements to assess occlusion and jaw dynamics.

Customizable settings for condyle angles, Bennet angle, and immediate side shift.

* Functional Surface Creation:

Creates functional surfaces using the envelope of motion.

Ensures the splint does not interfere with natural jaw movements.

* Mesh Manipulation:

Advanced Boolean operations for cutting and shaping splints.

Tools for refining the mesh to ensure a precise fit.

* Real-Time Feedback:

Provides real-time feedback on the fit and function of the splint.

Advanced visualization tools to see the splint in the context of the patient's mouth.

**D3 Dual Arch Tool**

Basically, used for the design and analysis of dual arch dental appliances, which involve both the upper and lower arches of the mouth.

* Dual Arch Design:

Tools for designing dental appliances that involve both the upper and lower arches.

Features for aligning and fitting both arches simultaneously.

* Articulator Simulation:

Simulates the movement of both the upper and lower jaws.

Allows for accurate assessment of occlusion and jaw movement for dual arch appliances.

* Tooth Library:

Includes a library of tooth models for use in dual arch designs.

Allows for the customization and placement of individual teeth.

* Batch Processing:

Capabilities for batch processing multiple designs.

Improves efficiency for dental labs working on multiple cases.

* Visualization and Analysis:

Real-time visualization of the dual arch appliance within the patient's mouth.

Tools for analyzing the fit and function of the appliance, including checking for interferences and ensuring proper occlusion.