18CSE304J

AR VR

School of Computing

By –

Shravya Sharan

(RA1911026010055)

K1 Section (Batch 1)

Arpita Muleva

(RA1911026010046)

K1 Section (Batch 1)

E1: AI-BASED BLENDER ANIMATION

1. **Aim –** To integrate an AI-based model over a predefined character animation using the Blender application.

2. **Requisites –**
    a. *Blender*
        i. The Blender application version 2.8 is installed on the device.
        ii. Permission is granted to execute scripting codes.

    b. *Character Model –*
        i. A predefined character model, which supports Blender version 2.8, is downloaded from Blender Studio.
        ii. The character model is adjusted preset viewpoint. The character's background and features are modified as per personal preference.

    c. *AI-based Model –*
        i. The AI model chosen for this project involves Facial Expression tracking using OpenCV.
        ii. This model enables Blender to capture the User's facial expressions and movements using their webcam and implement them onto the character model's facial features.
        iii. The python scripts necessary to execute the model are downloaded.

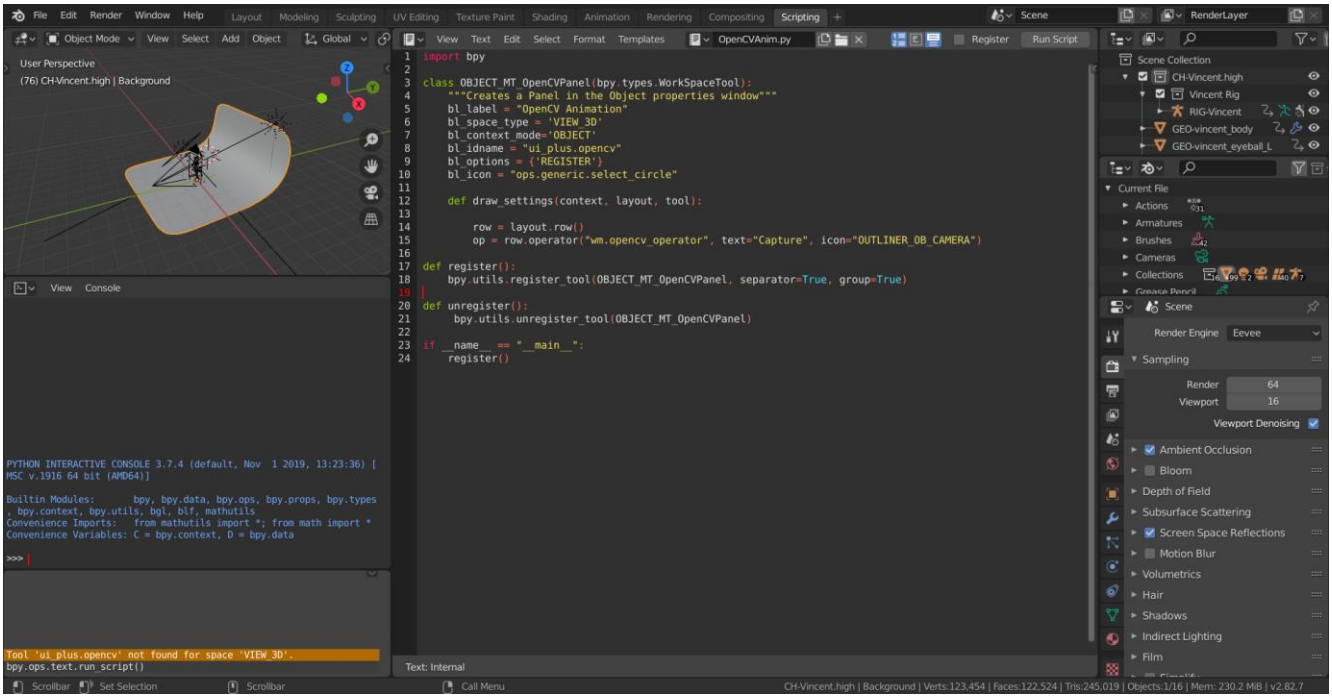
3. **Implementation –**
    a. The requisites mentioned above are downloaded and installed.
    b. The OpenCV libraries are installed using the following commands in the command prompt –
        i. cd C:\Program Files\Blender Foundation\Blender 2.82\2.82\python\bin
        ii. python -m pip install --upgrade pip
        iii. python -m pip install opencv-contrib-python numpy
    c. The character model is opened in Blender.

d. Under the Scripting tab in the Blender application, a new text code editor is opened using Text -> New and the python scripts installed for the AI-based model are opened using notepad.
e. The first code, OpenCVAnim.py, is pasted in the code editor in Blender. The editor is renamed OpenCVAnim.py and saved. The script is executed, and upon successful execution, the code is executed, and no error is shown on the console.
f. Similarly, the second code, OpenCVAnimOperator.py, is also pasted in the code editor in Blender and the file is renamed and saved.
g. The landmark_model_path is updated to the file location of lbfmodel.laml within the AI-based model folder.
h. The Register option is selected, and the second script is executed.
i. On the layout page, under the background header, the OpenCV Animator tab will appear along with a capture button.
j. Upon clicking on the capture button, the file will access the webcam and integrate the User's facial movements onto the model.

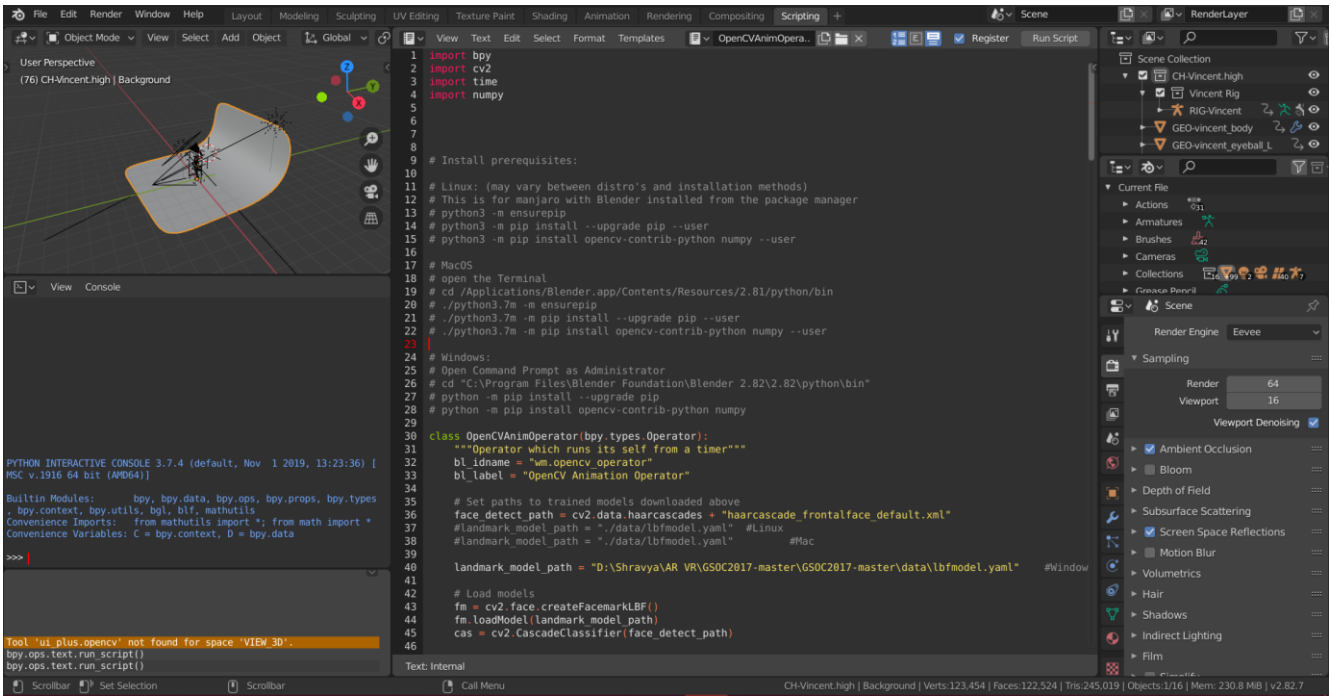
4. Output –

a. The character model is opened in Blender and modified.

b. The OpenCVAnim.py script is copied and successfully executed.



c. The OpenCVAnimOperator.py script is copied, modified and successfully executed.



d. The OpenCV Animation tab and capture button are activated.

e. The webcam captures the User's facial movements and integrates them with the Blender character model.

5. **Output Video Link –**
   https://drive.google.com/file/d/15VausVlrnps7ncMZ2UP27zDbEUZ1n9e9/view?usp=sharing

6. **Conclusion –** Thus, an AI-based model was successfully implemented on a Blender character model.

7. **Contribution –**
   a. Model implementation – Shravya Sharan (RA1911026010055)
   b. Documentation – Shravya Sharan (RA1911026010055) and Arpita Muleva (RA1911026010046)