Media Engineering and Technology Faculty
German University in Cairo

Implementing facial expressions in Maya using Python

Bachelor Thesis

Author: Amro Gamal
Supervisors: Prof. Dr. Rimon Elias
Submission Date: 1 August, 2021

Media Engineering and Technology Faculty
German University in Cairo

Implementing facial expressions in Maya using Python

Bachelor Thesis

Author: Amro Gamal
Supervisors: Prof. Dr. Rimon Elias
Submission Date: 1 August, 2021

This is to certify that:

(i) the thesis comprises only my original work toward the Bachelor Degree

(ii) due acknowlegement has been made in the text to all other material used

---

Amro Gamal
1 August, 2021

Acknowledgments

I would like to thank Dr. Rimon Elias for all the great help that was given to me in making this project. I am very grateful that I was given the opportunity to experience all that was taught to me for 3D animation. It is a new found passion that I want to be able to continue to work on in my future.

Abstract

Facial expressions can convey various sets of emotions in 3D characters. Without facial expressions and 3D facial animation many of the iconic characters that you see in cartoons would not be as lovable as they are.

The aim of this project is to be able to implement various facial expressions through the use of 3D Character Models. The user is given control over how he wants any certain facial expression to look like through the use of some GUI interfaces. The interfaces will be implemented through various commands using Python in Maya.

Contents

Chapter 1

Introduction

Everyday, we interact with several faces of our friends and families, whether it is looking, talking or making facial expressions. In the world of 3D graphics, creating facial expressions on 3D characters is not a simple task. It requires deep understanding of the complex system that lies underneath our faces and also very good animation principles and that is to be able to produce realistic facial expressions. In visual effect industries, facial animation and expression have become an important way to convey emotions to a 3D character and in this thesis project we will give an idea as to how facial expressions on 3D characters can come to be, and how they represent the life and soul of what our favourite cartoon characters came to be.

The software used to implement this project is Autodesk Maya which is a 3D animation software that is used by 3D artists and content creators. Maya can be used hand to hand with command-based scripting tools that offer the consumer a way to be able to build customized tools for his project.

1.1 Project approach

The approach as to how this project will be implemented involves mainly two steps:

- Two GUI windows will be implemented using Python, one that houses various facial expressions for the 3D characters, and the other window which is accessed from each facial expression houses the sliders that give you control over how you want that facial expression to look like and be able to modify it to your liking.
- Animating the facial expression either in a sequence or by itself using the key framing method that will be elaborated on.

1.2 How the thesis is organized

This thesis consists of five chapters, the first chapter being the introduction which gives an idea as to what the project is about.

Chapter 2 is where all the concepts that are used in this project are explained thoroughly and given an idea as to what they are.

Chapter 3 is the implementation of several facial expressions and their respective GUI slider windows.

Chapter 4 is the summary of the thesis and the possibility of future work and how this project could be improved upon.

Chapter 2

Background

In this chapter, all the concepts that are going to be used in the implementation chapter are thoroughly highlighted to give more exposition on what is to come, first we start off with an introduction on human facial anatomy and then we go on to the core concepts of both Maya and Python together.

2.1 Anatomy of a facial expression

Muscles in the face are the main driving force behind a facial expression. The part that innervates all the muscles is the seventh cranial nerve known as CN VII which is part of our nervous system.
Facial muscles are split into three distinct groups: orbital, nasal, oral. The most important groups being orbital and oral will be elaborated on [5].

2.1.1 Orbital section

The orbital group contains two muscles that are in control of the eye socket. They control the movement of the eyelids.

Orbicularis oculi

The orbicularis oculi highlighted in green in Figure 2.1 surrounds the eye socket and part of the eyelids and consists of three main parts. The two primary parts in control of the closing of eyelids are palpebral and orbital part.

Corrugator supercilii

This is the other smaller muscle highlighted in purple in Figure 2.1 that is located on top of the orbicularis oculi and its role is to draw the eyebrows together and create the wrinkles and together they make up the orbital group.

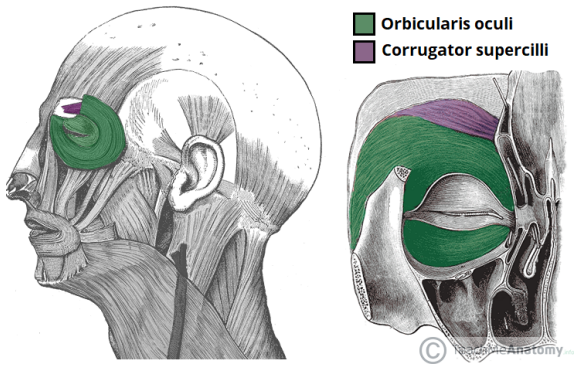


Figure 2.1: Orbital group consisting of two muscles [5]

2.1.2 Oral section

This group represents the biggest part in facial expressions, it is responsible for the movement of the lips and mouth. This group consists of several muscles, the most important being the orbicularis oris, depressor anguli oris and levator anguli oris [13].

Orbicularis oris

This muscle is located around the circumference of the mouth Figure 2.2 and the role of this muscle is to pucker the mouth and control the closing of the mouth.

Figure 2.2: Orbicularis oris highlighted in green [11]

Depressor anguli oris

Contracting this muscle highlighted in green in Figure 2.3 causes depression of the angle of the mouth and contributes to **frowning**.

Figure 2.3: Depressor anguli oris highlighted in green [9]

Levator anguli oris

This can be called the opposite of the depressor which is also highlighted in green in Figure 2.4 in which it elevates the angle of the mouth and contributes to **smiling**.

Figure 2.4: Levator anguli oris highlighted in green [10]

2.2 Maya

Maya is a 3D computer graphics application that is used to develop 3D applications, TV series, animated films, video games, and basically any visual effects [1].

It addresses the needs of a range and variety of digital content creators. The Maya software tools have been developed with the artist's needs in mind, it offers scripting using commands to build custom tools that suit more integrated production workflows [2].

2.3 The animation pipeline

For any scene to be made to life by animation and rendering, they have to go through a series of steps in the animation pipeline.

2.3.1 Modeling

This is the stage in which you use geometry to create a representation of the 3D objects and surfaces by using primitives that range from cubes to spheres to cones. The 3D primitives are then manipulated by vertices and edges to create the character or the scene you desire.
3D artists must also be aware of how the model will be used in the future, as in they should take care of how the character may bend without it getting too unrealistic. One of the most common ways of creating models is by starting with primitives, the primitive shape is then expanded or extruded. Primitives can either be made with NURBS or polygons [2].
In Figure 2.5, the model on the left started as a polygon cube that was then extruded, scaled and manipulated in a way that resulted in the shape on the right.

Figure 2.5: Extruded geometric shape [2]

The model used in Chapter 3 is made of polygon primitives as a primary starting point.

Polygon primitives

Maya has several polygon primitives shown in Figure 2.6 that can act as a starting point for a model you are trying to build. These primitives can then be manipulated by extruding, splitting and merging [2].

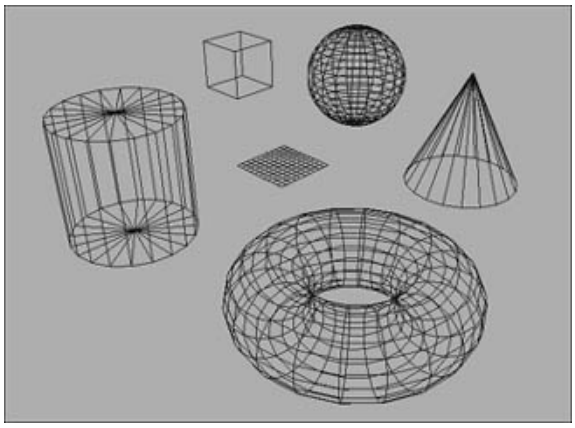


Figure 2.6: Primitive cube, cylinder, sphere, cone, torus and plane [4]

Modeling a head and facial features

The steps of creating a polygonal head for your model can be as follows [2].
In Figure 2.7, a cube was smoothed as the first step, it was cut and split at appropriate areas to result into the outline of the head, then the two rear bottom faces of the polygon were extruded downwards to create the neck and the front faces were moved down to create the chin.

Figure 2.7: Creating the chin and neck [2]

In Figure 2.8 the lower faces on top of the chin were split to create the nose. The eyes can also created by splitting faces in the appropriate area where you want your eyes to be.

Figure 2.8: Creating the eye area [2]

In Figure 2.9 the mouth was created by splitting the faces at the bottom of the polygon model in the mouth area and moving the vertices accordingly.

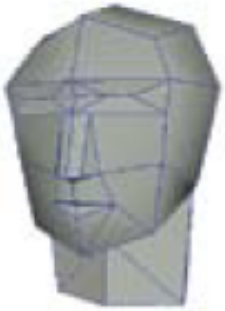


Figure 2.9: Creating the mouth [2]

2.3.2 Rigging

Rigging is a technique that succeeds modeling and texturing of a 3D model. It is a technique that is used to represent 3D models using a serious of so-called bones that are connected together to form a full skeleton.

The bone structure is then manipulated to perform any type of animation, and once any 3D model/object is rigged, it can be controlled and manipulated for any need [7].

There are many techniques as to how to be able to create facial rigging for animation, one of which is by using the Set Driven Key method, which groups several attributes of the face under one singular driving attribute. The driving attribute is then used to drive multiple attributes. These driving attributes are then attached to a control node that can be used to manipulate the driving attribute. In Figure 2.10 for example, three attributes of the index finger are grouped under one singular driving attribute that can move all parts of the index finger together by using a control node that is represented as a circle in the figure.

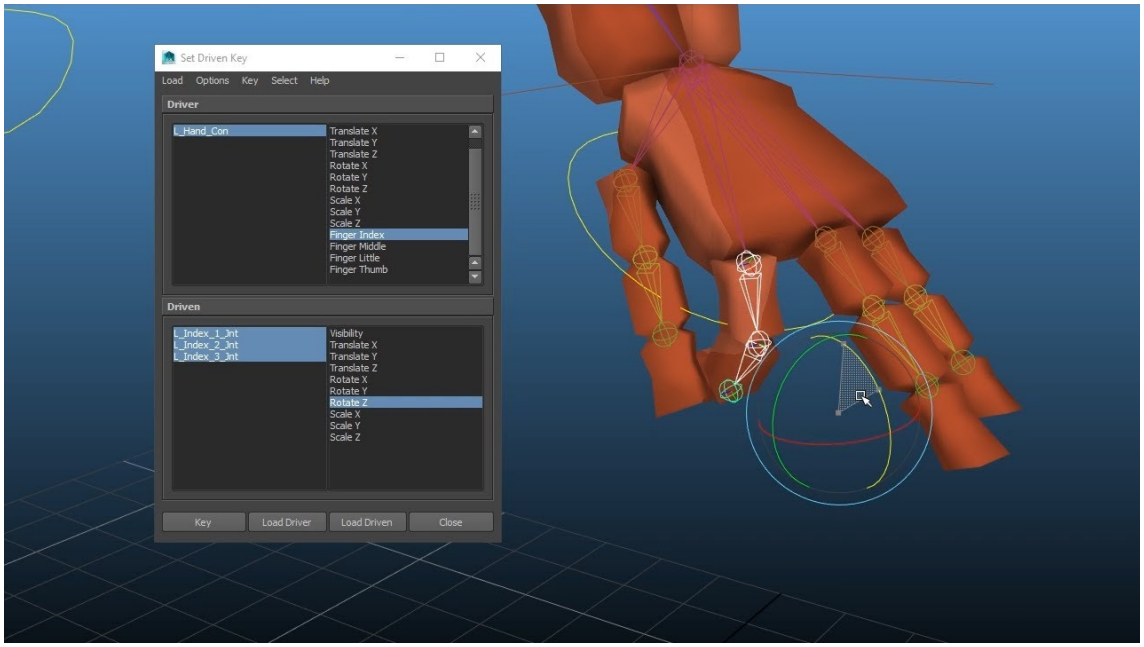


Figure 2.10: Rigging the hand by set driven key [8]

2.3.3 Animation

Once a model has been created and rigged, animation is bringing the characters to life. By changing its position or shape over time. You can create any specific motions you desire [2].
To be able to animate, a 3D artist needs to be able to understand the mechanics of the models they are working with and so there are a number of ways and techniques to animate an object, and in this project we will only use one of these techniques to animate facial expressions.

Setting keys

Setting keys is the easiest and most fundamental way to be able to animate any object to your liking. In this technique, you record attribute values as keys for an object at any points in time.
By settings keys on attributes at different points in time, you essentially define how the object will move in motion in Figure 2.11. In Section 3.3.1 keying attributes will be elaborated on and implemented thoroughly on the character rig named Dude [12]. The next step is playing back the animation, the object in question uses the keys and the values to then create the resulting motion by moving the time slider.

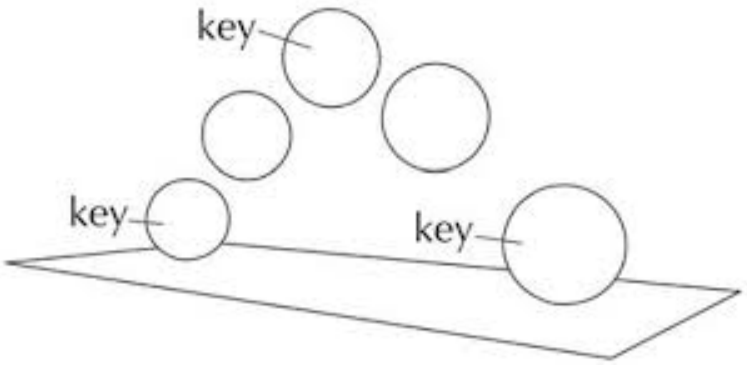


Figure 2.11: Keyframed ball [14]

Time slider

To be able to playback your animation, the time slider in Figure 2.12 gives you the ability to start the animation and see how the keys you set in the attributes flow in motion.

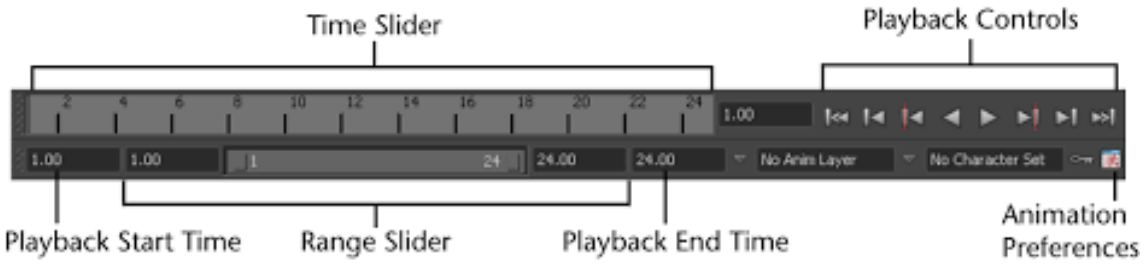


Figure 2.12: Time slider in Maya [3]

2.4 Python and scripting

In 3D Animation, visual effect companies were always looking for ways to speed up tasks, repetitive and routine tasks were a reason as to why Python was used in visual effects, e.g. copy pasting animation frames, making rigs for characters and adjusting numerous parameters. If a task were to be described as an algorithm, then it could be potentially be scripted. And the goal was to remove burden from authors and let them focus on art creation, and automate the workflow as much as possible [6].

Maya which is used in visual effects and animation has built-in scripting tools, one of which is Python.
Python can be used to configure basic functionality of Maya actions, a Maya user can record his commands as a script in Python to build a user friendly interface or a minimal application, it can also give access to features you would not be able to find in the Maya interface.
Python can also be used to make complex GUI's that you can use to manipulate 3D models and characters.

2.4.1 Accessing Python

To be able to use Python commands is through the built-in Script Editor in Figure 2.13 in Maya, which can be accessed by Window > General Editors > Script Editor. It consists of two main panels, History Panel located on top which provides feedback on commands and Input panel in the bottom which you can use to write commands.

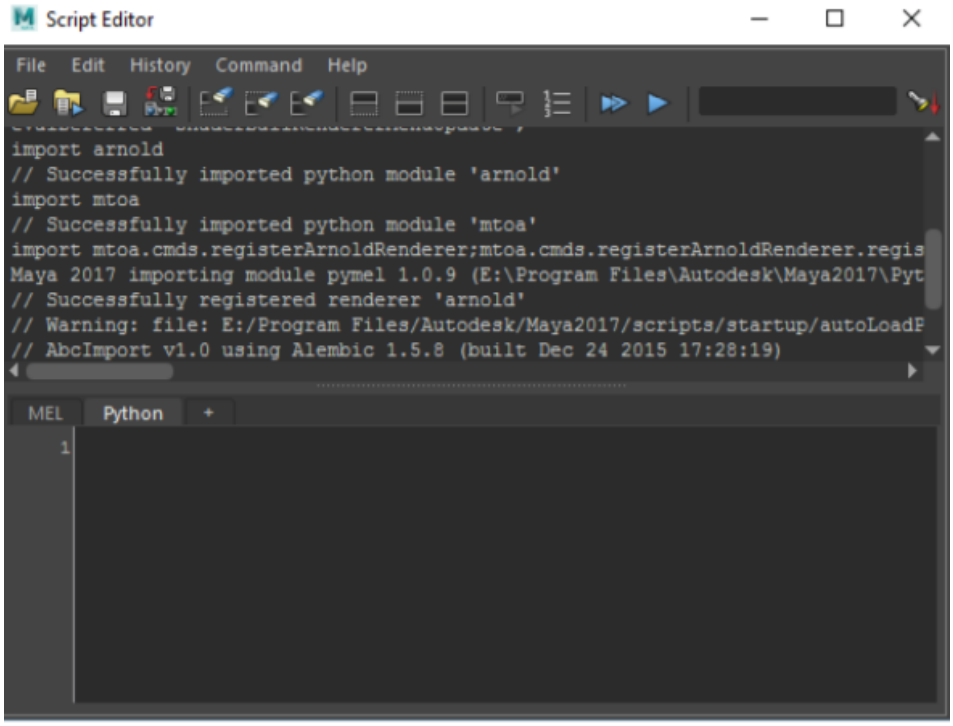


Figure 2.13: Script editor in maya

2.4.2 Python commands

There are various commands that will be used to be able to create both the slider GUI and the Facial Expression interface and these are the main commands that are used for this project.

- `setAttr`: This will be used to control the facial rig by manipulating any attribute of it through changing its numbers.
- `attrFieldSliderGrp`: This is used to create any slider and attach that slider to any attribute that is part of the rig.
- `Button`: This is used to create any button that will be mainly used to house the facial expression panel.
- `setKeyFrame`: This command sets a key frame on a facial attribute at a specific point in time which will be used in animation.

Chapter 3

Implementation

In this project, a GUI is to be created that will consist of many facial expressions to choose from along with a slider GUI that will represent the facial controls of a specific facial expression.
A character rig will also be used to demonstrate the facial expressions on the Dude model [12], and the various parts of the rig will be touched upon.

3.1 Facial expression controls

In this section, each part of the face in the character rig will be explained in detail and how the controls are represented on the rig.

3.1.1 Eyebrows

In Figure 3.1, for both eyebrows there are mainly two control nodes that will be used that are highlighted in the figure, these control nodes are what takes care of the attributes that are related to the eyebrows, e.g, `DudeLtBrow2Ctrl` which controls the middle part of the eyebrow and `DudeRtBrow2Ctrl` which controls the left end part of the eyebrow of the 3D character.

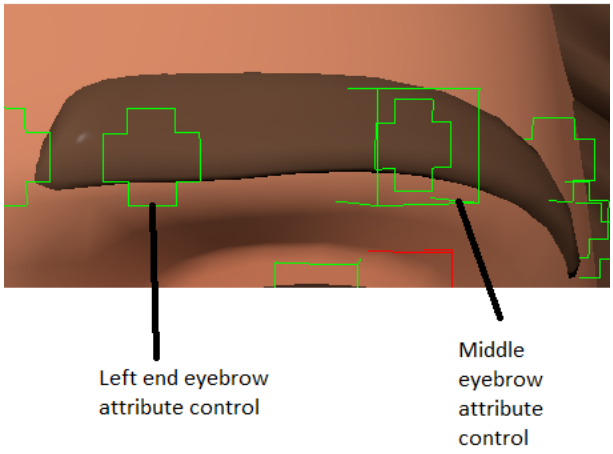


Figure 3.1: Eyebrows of 3D character

3.1.2 Eyeball and eyelids

The upper and lower eyelids are each handled by two different control nodes for both the attributes named `DudeLtEyeLowLidMstrCtrl` and `DudeLtEyeUpLidMstrCtrl` that are in both the eye areas, manipulating these attributes mimics the up and down movement of the eyelids. The eyeball movement is also handled and can be controlled by the bounding shape shown in Figure 3.2.

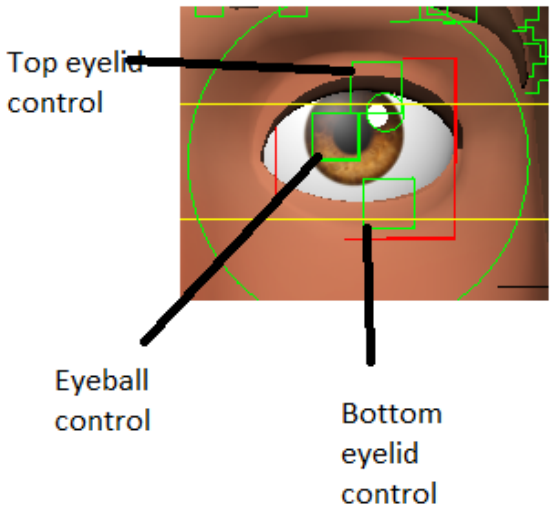


Figure 3.2: Eyelids and eyeball of the 3D character.

3.1.3 Mouth and jaw

For both the lips, there are multiple attribute controls for each lip that essentially control how the lips might look like during a facial expression in a human. The boxes on the ends of the lips control the degree of the smile the 3D character can potentially achieve. The two controls in the middle of the lips shown in Figure 3.3 control the opening of the lips.
The up and down movement of the jaw can also be mimic-ed in the 3D character by a single control that moves vertically in the 3D space.

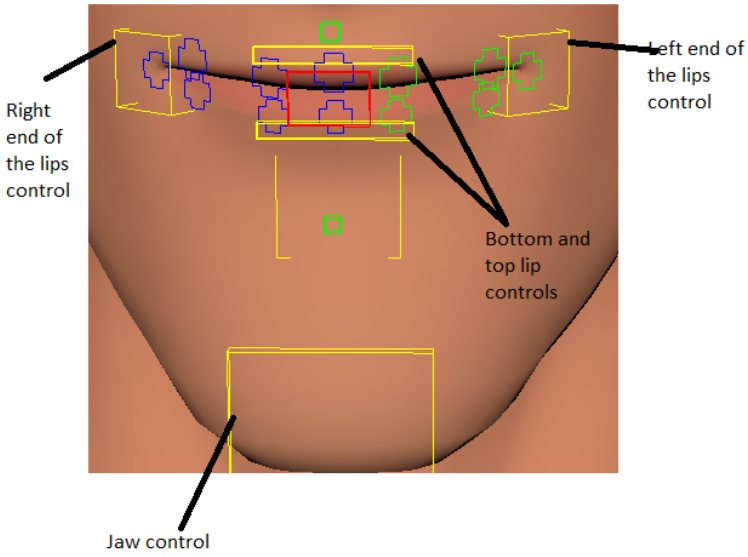


Figure 3.3: Mouth and jaw of the 3D character.

3.2 Implementation of the GUI

Two unique GUIs will be created by utilizing the commands mentioned in Section 2.3.2. One is the facial expression window which houses the various facial expressions, and the other GUI window is the slider window that is made separately for each facial expression, this window houses the specific sliders that are attached to each facial attribute and when manipulated results in that expression.

3.2.1 Facial expressions window

The facial expressions are enabled by several buttons. The function of the button is to create a certain facial expression on the 3D character rig. It will also pop out another window that showcases the various sliders that were used for the facial expression.
The following code sample creates the window in Figure 3.4 for the various facial expressions.

```
cmds.window(width=300)
cmds.columnLayout(adjustableColumn=True)
cmds.button(label='Smile', command=Smile,backgroundColor=(0.0,0.0,0.1))
cmds.button(label='Frown', command=Frown,backgroundColor=(0.0,0.3,0.1))
cmds.button(label='Surprise', command=Surprise, backgroundColor=(0.0,0.0,0.5))
cmds.button(label='Happy', command=Happy, backgroundColor=(0.0,0.5,0.5))
cmds.button(label='Paranoia', command=Paranoia, backgroundColor=(0.5,0.5,1))
cmds.showWindow()
```

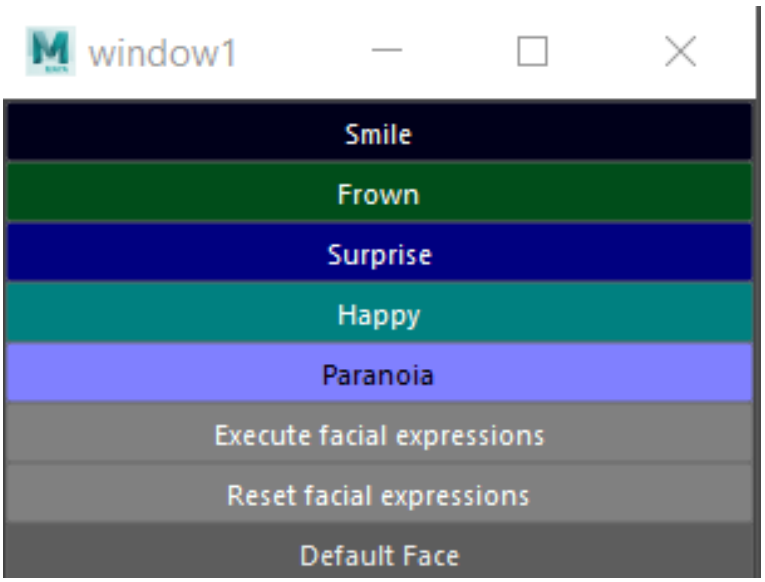


Figure 3.4: Facial expression window

Each facial expression in Figure 3.4 executes a command that calls its related function, e.g, pressing Smile will execute the function `Smile()` and in that function, another function called `smileAttributes()` is also called which automatically moves the attribute controls for the associated facial expression in Figure 3.5.

```
def smileAttributes():
        defaultFaceAttributes()
        cmds.setAttr("DudeLtMouthCrnrCtrl.tx",-1.735)
        cmds.setAttr("DudeLtMouthCrnrCtrl.ty",3.41)
        cmds.setAttr("DudeRtMouthCrnrCtrl.tx",-1.735)
        cmds.setAttr("DudeRtMouthCrnrCtrl.ty",3.41)
        cmds.setAttr("DudeLtBrow2Ctrl.ty",2.943559)
        cmds.setAttr("DudeRtBrow2Ctrl.ty",2.943559)
```

The facial expression was created by moving several attribute controls one of which is `DudeLtBrow2Ctrl.ty` and `DudeRtBrow2Ctrl.ty` which represents the ends of the eyebrows that were translated upwards in the Y direction a certain value by the `setAttr` command.

Figure 3.5: Smile expression

The function also opens the slider window that houses the attributes related to that facial expression.

3.2.2 Slider window

Any facial expression has a set of sliders associated with them as represented in Figure 3.6 which results in the smile facial expression in Figure 3.5, these sliders are basically attached to the attributes that result in the facial expression, like for example the

`DudeLtBrow2Ctrl.ty` and `DudeRtBrow2Ctrl.ty` attribute controls that resulted in the smile which were mentioned in Section 3.2.1 are attached to the first two sliders in Figure 3.6.
The sliders are given for the user to modify how the specific facial expression may want to look like instead of just using the default values for the facial expression that was implemented in Figure 3.5.

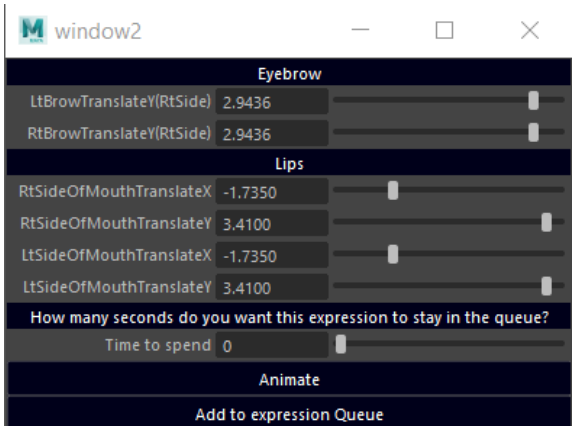
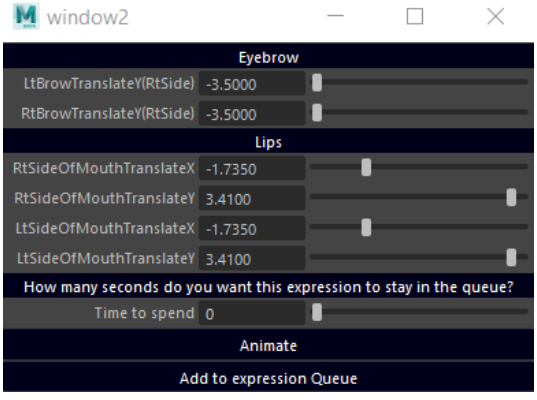


Figure 3.6: Slider window for the default smile facial expression that has several sliders for each related attribute control.

As for the units of each attribute control slider seen in Figure 3.6 the manipulation of those attributes is defined by **Translation Transformation**, translating some units happens in the XYZ direction of the 3D space.

If the user wanted to modify any default facial expression he is given the ability to do so and in Figure 3.7(b) the eyebrow attribute controls for the smile were modified by the user, which is reflected in Figure 3.7(a) as compared to the default slider values in Figure 3.6.

(a) Modified Smile



(b) Eyebrow sliders moved

Figure 3.7: Modified smile by manipulating the attribute sliders

3.3 Animating the facial expressions

When the slider window is opened in Figure 3.6, the user is given two options, he can either press the button Animate which gives the facial expression life or add it to an array named `expressions[]` which was initialized at the beginning of the code which is used to create a sequence of facial expressions.

3.3.1 Animate

Animating the facial expression will use the set key frames method that was explained in Section 2.3.3.
Key frames are automatically set by using the `setKeyFrame()` command, key frames are set for each attribute of the default face at frame 0, and then at frame 20 the attributes for the chosen facial expression are saved.

The modified facial expression like in Figure 3.7 that the user desires is animated or if the expression was not modified to the user's liking then the default facial expression like in Figure 3.5 is animated instead.

The modified smile in Figure 3.7 is animated by playing back using the time slider discussed in Section 2.3.3, enabling the time slider is by the following function which sets the frame to 0th frame and then starts the slider.

```
def startAnimation(*args):
    cmds.currentTime(0)
    cmds.play( forward=True)
```

Starting the playback sets the key frames in motion, and is shown in Figure 3.8.

(a) Frame 0

(b) Frame 10

(c) Frame 20

Figure 3.8: Modified smile at frame 0, frame 10 and frame 20 by moving the Time Slider

3.3.2 Creating a sequence of facial expressions

The user is also provided with a way to create a sequence of facial expressions by simply clicking the `Add to expression queue` button in any of the slider windows in Figure 3.6 that appear with any facial expression.

Each facial expression when added to the queue is appended to the array named `expressions[]`. Appending several expressions then pressing the button `"Execute facial expressions"` will key frame all the facial attributes that change every 20 frames for each expression and every 20 frames is a facial expression in the queue that was added by the user.

In Figure 3.9, smile, happiness and paranoia were added in this sequence. And each increment of 20 frames resides one of these expressions, in between every 20 frames is how the expression looks like when transitioning from one facial expression to the other.

(a) Frame 0

(b) Frame 20: Smile

(c) Frame 30: Smile-Happy

(d) Frame 40: Happy

(e) Frame 50: Happy-Paranoid

(f) Frame 60: Paranoid

Figure 3.9: Sequence of facial expressions by moving the Time Slider

Chapter 4

Conclusion and future work

Facial expressions on 3D characters are one of the main reasons as to why people love a lot of the well known cartoon characters.
Normally creating and animating facial expressions is a much longer and a non intuitive process, as you would have to manipulate attributes from scratch and animate each facial expression separately, and so this project is targeted towards making and animating facial expressions a much easier process, through the use of the provided GUI's the user is able to choose from a set of facial expressions from which he is able to modify, and with a click of a button he can create a sequence of facial expressions. The creation of the GUI's and the animations was through Python commands provided by Autodesk Maya.

4.1 Future work and limitations

The scripts provided in this project only work on the character rig named Dude [12], since each manipulation of any attribute depends on the name of the attribute for which each rig is different from another. The only way for this script to work on other characters is to have the same exact attribute names which is unintuitive.
More facial expressions can also be added, some facial expressions can also use the addition of the arms and shoulders to add more to the expression.
Adding an environment to which the character can be placed can also be an idea of an improvement.

Bibliography

[1] What is maya? https://www.educba.com/what-is-maya/.

[2] *Art Of Maya*. Autodesk, 2007.

[3] Autodesk. Getting started with maya >animation >lesson 1: Keyframes and the graph editor. https://download.autodesk.com/us/maya/maya_2014_gettingstarted/index.html?url=files/Keyframes_and_the_Graph_Editor_Setting_the_playback_range.htm,topicNumber=d30e12155.

[4] Flylib. About polygon primitives. https://flylib.com/books/en/4.422.1.28/1/.

[5] Oliver Jones. The muscles of facial expression. https://teachmeanatomy.info/head/muscles/facial-expression/.

[6] Vladislav Kazakov. The role of python in visual effects pipeline case: Talvi tools (thesis), 2016.

[7] Josh Petty. What is 3d rigging for animation character design? https://conceptartempire.com/what-is-rigging/.

[8] Winston Powell. Tutorial: Set driven keys. https://www.youtube.com/watch?v=3txXQNWE-xA.

[9] Gordana Sendic. Depressor anguli oris muscle. https://www.kenhub.com/en/library/anatomy/depressor-anguli-oris-muscle.

[10] Gordana Sendic. Levator anguli oris muscle. https://www.kenhub.com/en/library/anatomy/levator-anguli-oris-muscle.

[11] Gordana Sendic. Orbicularis oris muscle. https://www.kenhub.com/en/library/anatomy/orbicularis-oris-muscle.

[12] Ahmed Shalaby. Dude model rigged and textured. https://shalabology.wordpress.com/free-stuffs/free-rigs/dude_free-rig-_maya/.

[13] Jennifer Shennan. Muscles of facial expression. https://geekymedics.com/muscles-of-facial-expression/.

[14] what-when how. Animation — keyframe basics (essential skills) (3d animation using maya. http://what-when-how.com/3d-animation-using-maya/animation-keyframe-basics-essential-skills-3d-animation-using-maya/.