VFX T4

Scripting.VFXpipeline

Due:

Part 01 - Feb 14th 6:30pm (before the class) for the single node operation

Part 02 - Feb 25th 6:30pm (before the class) for multiple node operations

**20% of final grade**

**Assignment 01**

Hello World Python

*This assignment is to challenge each student in an individual way and a consolidate the foundations before jumping into more complicated coding.*

**Part One:**

Create a small script to be ran in Nuke’s script editor. Save this script into a .py file for the assignment delivery. Keep in mind of making your script clean and easy to understand. The following elements will be counted on the final grade:

* Variable Naming clarity
* Naming consistency within the code, e.g, use only underscores with lower case or only camel case
* Comments to explain your code

Each student will have a slightly different assignment, which intent to challenge you individually.

For the 1st part of the assignment, develop a small script to do the following:

1. Store the current selected Node in a variable
2. Read the node’s name and print it in the sentence that provides feedback in the console, for instance:  
   “Now processing Transform1”
3. For each one of the knobs listed on the table below, do the following:
   1. create an unique variable to store the current’s knob value.
   2. Print a single line text informing the knob name and knob value, e.g.:  
      “Original value of rotate is 90.0”
   3. **Set** a new value on the knob, following the operation described on the column “operation” on the table below
   4. Print a single line text informing the name and new value of the knob, e.g.:  
      “New value of rotate is 360.0”
   5. If the knob has multiple inputs, like translate x and y, you should modify and print both values.

Bellow, there is a list of names of the students and what class of nodes and knobs should be used for the assignment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Student | Node Class() | Knobs | Operation |
| 1 | Bruno | Transform | * Rotate * Translate * Scale | \*4 |
| 2 | Chun-Yao | Sphere | * Rows * Columns * Translate | /2 |
| 3 | Daniel Ivan | Card3D | * Lens-in focal * Translate * Uniform scale | \*\*2 |
| 4 | Dhruv | TransformGeo | * Rotate * Translate * Uniform scale | +5 |
| 5 | Divasees | ZDefocus | * Depth of Field * Focal Point * Focus plane | \*5 |
| 6 | Gayatri Jayant | Card | * Rows * Translate * Uniform scale | +2 |
| 7 | Harshal Sandeep | Camera 3D | * Focal length * Translate * Uniform scale | \*4 |
| 8 | Jessica | Light | * Color * Intensity * Translate | /2 |
| 9 | Jesus Alexander Miolan | HSV Tool | * Hue’s Range * Rotation * Adjustment | \*\*2 |
| 10 | Kevin | Ramp | * Point 0 * Point 1 * Opacity | +3 |
| 11 | Khloe Jade | Transform | * Rotate * Translate * Scale | \*2 |
| 12 | Marissa | Sphere | * Rows * Columns * Translate | \*2 |
| 13 | Mayank Devendra | Card3D | * Lens-in focal * Translate * Uniform scale | /2 |
| 14 | Naman Bhavesh | TransformGeo | * Rotate * Translate * Uniform scale | \*\*2 |
| 15 | Piyushkumar | ZDefocus | * Depth of Field * Focal Point * Focus plane | +5 |
| 16 | Rahul | Card | * Rows * Translate * Uniform scale | +3 |
| 17 | Sameer | Camera 3D | * Focal lenght * Translate * Uniform scale | \*3 |
| 18 | Sukhjit | Light | * Color * Intensity * Translate | /2 |
| 19 | Venkata Sri Ranganath | HSV Tool | * Hue’s Range * Rotation * Adjustment | \*2 |
| 20 |  | Ramp | * Point 0 * Point 1 * Opacity | \*2 |

**Part Two:**

You will continue to develop/modify your code from assignment part 01. The goal is to create a script that **creates copies of the selected nodes and modify the value of the knobs on each copy.**  
You will do that by adding:

* Nuke’s message and prompt dialogs
* For loops

Add the following functionality to your script from class 01:

1. Using the function nuke.getInput(“how many copies”, 1), ask the user how many copies of the node should be created. Doc:   
   <https://learn.foundry.com/nuke/developers/13.0/pythonreference/_autosummary/nuke.getInput.html>
2. Create a copy of the node using the function ***nuke.createNode("Class", inpanel=False)*** and store it in a variable  
   *Doc.:* [*https://learn.foundry.com/nuke/developers/13.0/pythonreference/\_autosummary/nuke.createNode.html*](https://learn.foundry.com/nuke/developers/13.0/pythonreference/_autosummary/nuke.createNode.html)
3. With the new node stored in a variable, set the values of the knobs listed on the table from Part One. The value should be set after calculating the required operation from the table above on top of the value from the previous node created or, if it’s the first copy, on top of the original node. E.G.:  
   Knob “intensity” on the original node: 2.0  
   Required operation: \*\*2   
   Value of the 1st copied node: 4  
   Value of the 2nd copied node: 8  
   Value of the 3rd copied node: 16  
   Value of the 4th copied node: 32  
   etc.
4. For each of the created nodes, print a feedback message in the console. E.g.:  
   *Node Transform 1 has been created. Knob values set:*  
   *rotate: 64*  
   *translate: 32, 32*  
   *uniform scale: 48*

For an example of expected result,

**see the GIF in the assignment’s folder**

Grade breakdown:

|  |  |
| --- | --- |
| Points | Task |
| 20 | Code organization and cleanliness   * Properly named variables * Naming formatting consistency * Comments * Not unnecessary code |
| 30 | Functionality Part 01   * Does the code work? |
| 30 | Functionality Part 02   * Does the code work? |
| 10 | Correct .py file delivered |
| 10 | Assignment due on time |
| **100** | **Total** |
| 10 | Bonus. Any extra functionality, for instance: - use nuke.message in the end of the loop  - Add some validation for data type after the users’ getInput  - set the position of the nodes after they had been copied (posy and posx knobs)  - Use an extra for loop , make your code to work with multiple selected nodes |

**Assignment submission:**

**Feb 14th, 2022 (6:30 pm - before the class)**

Please submit your python file name as: **StudentID\_VFX01T4\_Assignment01\_p1.py**  
No zipping/compressing

**Feb 25th, 2022 (6:30 pm - before the class)**

Please submit your python file name as: **StudentID\_VFX01T4\_Assignment01\_p2.py**  
No zipping/compressing