Getting Started – Bắt Đầu:

1. Cách cài đặt?

* Vào Link
* <https://download.com.vn/download/blender-15961?linkid=1>
* Click “Nhấn vào đây” + chờ nó tải + mở File vừa tải + tiến hành cài đặt
* Khi này, trong thư mục “C:\Program Files\” sẽ xuất hiện thư mục “Blender Foundation”, trong đây sẽ chứa các thư mục là các phiên bản Blender khác nhau bạn đã tải, ví dụ phiên bản 3.5 sẽ nằm trong thư mục “Blender 3.5”, trong đây sẽ chứa File EXE “blender-launcher”, chạy File này để bắt đầu sử dụng Blender mặc định, ngoài ra còn có File EXE “blender”, nếu bạn muốn Pass các tham số khởi đầu cho chương trình thì chạy File này trên CMD

1. Phần Mở Rộng File Blender?

* “.blend”

1. Cấu Trúc Cửa Sổ?

* Một lúc chỉ có thể làm việc trên 1 File Blender
* Gồm 4 phần
* Thanh trên cùng là thanh tiêu đề, ghi đường dẫn tới File đang làm việc
* Thanh bên dưới là thanh công cụ
* Phần to nhất ở giữa là cửa sổ làm việc, trong cửa sổ làm việc, ta có thể phân chia thành nhiều cửa sổ con khác nhau
* Thanh dưới cùng là thanh thông tin, ghi phiên bản Blender và giải thích chức năng các phím chuột khi bạn di lên 1 phần tử nào đấy trong cửa sổ làm việc
* 1 File Blender gồm nhiều Scene, mỗi Scene sẽ có các thuộc tính riêng như Scene Collection chứa các Collection, View Layers chứa các điểm nhìn khác nhau, ví dụ ở View Layer này Object bị ẩn, ở View Layer khác thì không, …
* Blender cùng lúc chỉ làm việc được trên 1 Scene được chọn

1. Thanh Công Cụ?

* Thanh công cụ gồm 3 phần
* Phần bên trái cùng, gọi là thanh hoa tiêu
* Phần ở giữa, gọi là thanh bố cục
* Phần bên phải cùng, gọi là thanh tầm nhìn
* Về thanh bố cục
* Mặc định nó sẽ gồm các Tab “Layout”, “Modeling”, … cho đến “Scripting” và kèm theo biểu tượng dấu cộng bên phải cùng, gọi là biểu tượng “Add Workspace”
* Mặc định bạn sẽ ở Tab “Layout”, mỗi Tab sẽ có cách bố trí riêng các cửa sổ con trong cửa sổ làm việc, bạn có thể chuyển qua lại giữa các Tab và thay đổi cách bố trí mặc định của các Tab
* Bạn có thể đổi tên các Tab bằng cách Double Click vào Tab + gõ lại tên + Click ra ngoài
* Các Tab không được phép trùng tên
* Bạn có thể thêm 1 Tab bằng cách Click vào biểu tượng “Add Workspace” + vào 1 trong 5 mục được liệt kê ra và chọn 1 cách bố trí nào đó hoặc Click “Duplicate Current” nếu muốn Copy Tab đang chọn + tiến hành đổi tên lại Tab nếu muốn và chỉnh sửa lại bố trí của Tab nếu muốn
* Về thanh tầm nhìn
* Nó sẽ chỉ định Scene làm việc hiện tại và View Layer hiện tại
* Để tạo 1 Scene mới, Click biểu tượng “New Scene” ở giữa + chọn 1 trong 4 tùy chọn + khi này 1 Scene mới được tạo ra và bạn sẽ được chuyển sang làm việc trên Scene này
* Để đổi tên Scene làm việc hiện tại, Click vào ô “Name” + chỉnh lại tên + Click ra ngoài
* Để xóa Scene làm việc hiện tại, Click vào biểu tượng “Delete Scene” hình dấu X
* Để chuyển sang làm việc trên Scene khác, Click biểu tượng bên trái cùng + Click vào tên Scene mong muốn
* Nửa bên phải thanh tầm nhìn là của View Layers, cách hoạt động tương tự như những gì đã liệt kê ở trên

1. Làm Việc Với Cửa Sổ Làm Việc?

* Nếu có 2 cửa sổ con cùng loại trong cửa sổ làm việc, thì chỉnh sửa cửa sổ con này sẽ ngay lập tức cửa sổ con kia cũng bị chỉnh sửa theo
* Để tạo 1 cửa sổ con cùng loại cửa sổ con A, di chuột lên 1 trong 4 góc của cửa sổ con A cho đến khi con trỏ chuột biến thành hình 4 gạch dấu cộng + kéo thả
* Để hợp nhất 2 cửa sổ con A và B kề nhau, nghĩa là xóa cửa sổ con B và khiến cửa sổ con A chiếm phần không gian của cửa sổ con bị xóa, di chuột lên 1 trong 4 góc của cửa sổ con A cho đến khi con trỏ chuột biến thành hình 4 gạch dấu cộng + kéo thả vào cửa sổ con B, hoặc bạn kéo vào cửa sổ con B rồi kéo lại vào cửa sổ con A rồi mới thả, thì A sẽ là thằng bị xóa còn B là thằng mở rộng
* Mỗi cửa sổ con đều có biểu tượng góc trái trên, đại diện cho loại cửa sổ của nó, để chuyển sang loại khác, Click vào biểu tượng này + vào 1 trong 4 nhóm cửa sổ “General”, “Animation”, “Scripting”, “Data” + chọn loại cửa sổ mong muốn

System Console:

1. Cách Mở System Console Để Hiện Mấy Cái Thông Báo Lỗi Với Mấy Cái Mình In Ra?

* Vào Tab “Window” + chọn “Toggle System Console”
* Nếu cửa sổ System Console đã mở thì nó sẽ đóng lại

3D Viewport:

1. Vào Chế Độ 3D Viewport?

* Click biểu tượng “Editor Type” góc trái trên cửa sổ + tại mục “General” + chọn “3D Viewport”

1. Zoom?

* Cuộn chuột

1. Quay Góc Nhìn?

* Nhấn giữ nút cuộn chuột rồi di

1. Chọn Object?

* Click trái chuột vào vật

1. Tạo Cửa Sổ Mới?

* Click giữ trái chuột vào góc phải trên cùng nơi không có biểu tượng của 1 cửa sổ rồi kéo

1. Cách Chọn Nhanh Cách Bố Trí Cửa Sổ Để Làm Việc?

* Nhìn vào thanh trên cùng, từ Tab “Layout” trở đi là các mẫu bố trí có sẵn, chọn vào 1 cái sẽ chuyển đến cách bố trí tương ứng, riêng “Layout” là cái bạn bố trí theo ý thích
* Để Reset tất cả mẫu bố trí
* Lưu File + vào Tab “File” + chọn “New” + chọn “General” + vào Tab “File” + chọn “Open…” + chọn File vừa mới lưu + Click nút răng cưa góc phải trên + bỏ Tick “Load UI” + Click “Open”

1. Phím Tắt Chọn Tất Cả Object Đang Hiện Trong Scene?

* “A”
* Click trái chuột ra chỗ trống để bỏ chọn tất cả

Properties:

1. Vào Chế Độ Properties?

* Click biểu tượng “Editor Type” góc trái trên cửa sổ + tại mục “Data” + chọn “Properties”

Outliner:

1. Vào Chế Độ Outliner?

* Click biểu tượng “Editor Type” góc trái trên cửa sổ + tại mục “Data” + chọn “Outliner”

1. Chế Độ Hiển Thị Scenes?

* Click biểu tượng “Display Mode” góc trái trên + chọn “Scenes”
* 1 File Blender gồm nhiều Scene, mỗi Scene sẽ có 1 Scene Collection chứa nhiều Collection con, mỗi Collection con chứa nhiều Object và Collection con khác
* Blender chỉ hiển thị 1 Scene được chọn
* Tạo 1 Scene mới
* Click biểu tượng “New Scene” ở thanh trên cùng góc gần phải cửa sổ Blender + chọn “New”

Text Editor:

1. Vào Chế Độ Text Editor?

* Click biểu tượng “Editor Type” góc trái trên cửa sổ + tại mục “Scripting” + chọn “Text Editor”

1. Phím Tắt Chạy Code?

* Để con trỏ chuột vào cửa sổ Text Editor + nhấn “Alt” + “P”

Mathutils:

1. Cách Import?

from mathutils import Vector, Matrix

1. Vector?

* Tạo 1 Vector

<Vector> = Vector(<Tọa Độ>)

* Ví dụ

foo = Vector((1, 2, 3))

* Tính tích có hướng của <Vector 1> với <Vector 2>

<Tích Có Hướng> = <Vectorr 1>.cross(<Vector 2>)

* Chuẩn hóa <Vector>

<Vector>.normalize()

* Ví dụ

foo.normalize()

* Ta có

|  |  |
| --- | --- |
| foo trước | [2, 4, 9] |
| foo sau | [0.199, 0.398, 0.896] |

* Trả về độ dài <Vector>

<Độ Dài> = <Vector>.length

1. Ma Trận?

* Tạo 1 ma trận

<Ma Trận> = Matrix(<Mảng 2D>)

* Ví dụ

foo = Matrix([

[1, 2, 3],

[5, 6, 7]

])

Context:

1. Cách Import?

from bpy import context

1. Trả Về List Các Object Đang Được Chọn?

<List Các Object> = context.selected\_objects

Operation:

1. Cách Import?

from bpy.ops import object as ob

1. Cách Chọn Và Bỏ Chọn Tất Cả Object?

ob.select\_all(action = <Kiểu Chọn>)

* Các cơ chế chọn tương ứng

|  |  |
| --- | --- |
| <Kiểu Chọn> | Cơ chế |
| "SELECT" | Chọn tất cả Object trong Scene hiện tại |
| "DESELECT" | Bỏ chọn tất cả Object trong Scene hiện tại |
| "TOGGLE" | "DESELECT" khi có ít nhất 1 Object được chọn, nếu không thì "SELECT" |
| "INVERT" | Chọn tất cả Object không được chọn và bỏ chọn tất cả Object được chọn trong Scene hiện tại |

Data:

1. Cách Import?

from bpy import data

1. Toán Tử @?

* Nhân 2 ma trận hoặc 1 ma trận với 1 Vector

1. Trả Về Object Có Tên Nào Đó?

<Object> = data.objects[<Tên Object>]

* Trả về kiểu của <Object>

<Kiểu Object> = <Object>.type

* Các giá trị

|  |  |
| --- | --- |
| <Kiểu Object> |  |
| "MESH" | Các Object có đỉnh cạnh mặt các thứ |
| "CAMERA" | Cái Camera |
| "LIGHT" | Cái bóng đèn |
|  |  |

* Trả về 1 mảng chứa tất cả đỉnh của <Object>
* <Các Đỉnh> = <Object>.data.vertices
* Trả về 1 đỉnh
* <Đỉnh> = <Các Đỉnh>[<Index>]

1. Đỉnh?

* Trả về Vector tọa độ của 1 đỉnh theo hệ qui chiếu Local

<Tọa Độ Đỉnh Local> = <Đỉnh>.co

* Trả về trọng số Bevel của 1 đỉnh

<Trọng Số Bevel Đỉnh> = <Đỉnh>.bevel\_weight

1. Trả Về Ma Trận Biểu Diễn Vị Trí, Độ Quay, Độ Phóng To Của Vật?

<Ma Trận Object> = <Object>.matrix\_world

* Dùng ma trận này để chuyển đổi hệ quy chiếu Local thành Global

<Tọa Độ Đỉnh Global> = <Ma Trận Object> @ <Tọa Độ Đỉnh Local>

Code – Viết Mã:

1. Cách Dùng Chat GPT Viết Hộ Mã?

* Sau khi hỏi Chat GPT nhờ nó viết mã cho Blender 2.7 thì nhấn nút Copy Code ở tiêu đề đoạn Code nó viết rồi Paste vào Blender

Modifier:

1. Tạo 1 Đống Hình Lập Phương Sao Cho Chúng Tạo Thành Hình Hộp Chữ Nhật?

* Dùng Array Modifier

Graph:

1. Offset 1 đường Đồ Thị Lên Xuống 1 Khoảng Nào Đó?

* Chọn tất cả Key Frame của đường đồ thị + nhấn phím “G” + “Y” + nhập giá trị Offset

1. Cách Chọn Tất Cả Key Frame Của 1 Đường Đồ Thị?

* Ẩn toàn bộ đường đồ thị = cách nhấn hình con mắt + hiện đường đồ thị muốn chọn Keyframe = cách nhấn lại hình con mắt của đồ thị đó + nhấn phím “A”

Material:

1. Áp Dụng Tất Cả Material Của 1 Đối Tượng Cho Nhiều Đối Tượng Khác?

* Chọn tất cả đối tượng muốn áp dụng Material sao cho đối tượng chứa Material gốc được chọn cuối cùng + vào Tab “Object” + chọn thẻ “Make Links” + Click “Materials”
* Khi này, bộ Material của tất cả đối tượng sẽ y đúc nhau

Object:

1. Cách Thế Nhiều Object Thành 1 Object Nào Đó

* Chọn tất cả Object muốn thế sao cho Object dùng để thế được chọn cuối cùng + vào Tab “Object” + chọn thẻ “Make Links” + Click “Object Data”

1. Cách Copy Vị Trí Của 1 Object Cho Nhiều Object Khác?

* Chọn tất cả Object sao cho Object có vị trí dùng để Copy được chọn cuối cùng + nhấn “Ctrl” + “C” + chọn “Copy Location”

Render:

1. Các Thông Số Cần Điều Chỉnh?

* Bỏ Tick “Open Shading Language” để Render nhanh hơn
* “Start Frame” và “End Frame” sẽ xác định ta Render trong khoảng nào
* Ví dụ
* “Start Frame” = 3, “End Frame” = 7, thì các Frame được Render là 3, 4, 5, 6, 7
* “Output File Path” sẽ xác định ta lưu File ở đâu, với tên gì
* Ví dụ
* “Output File Path” = “D:/foo” thì đường dẫn tới ảnh là “D:/foo.png”
* “Render Samples” xác định độ chống nhiễu, đặt thấp thôi, = 10 là được
* Bỏ Tick “Caustics” để Render nhanh hơn
* “Tiles” xác định 1 ô Render rộng bao nhiêu, đặt 512 x 512 cho nhanh

1. Cách Render 1 Phần Của Khung Ảnh Và Những Phần Còn Lại Trong Suốt?

* Nhấn “0” để chuyển sang Camera + nhấn “Ctrl” + “B” + kéo thả để xác định phần cần Render + vào Tab “Render” + Tick “Border” + tiến hành Render
* Để trở lại như cũ, nhấn “0” để chuyển sang Camera + nhấn “Ctrl” + “Alt” + “B”

Short Cuts – Các Phím Tắt:

1. Tạo Key Frame?

* “I”

1. Chọn Nhiều Đối Tượng?

* “B”

Quick Code – Mã Nhanh:

1. Tích Tổng Moment Động Lượng?

from bpy import context

from mathutils import Vector, Matrix

from random import random

[obj] = context.selected\_objects

vertex\_masses = []

total\_mass = 0

center = Vector((0, 0, 0))

for vertex in obj.data.vertices:

vertex\_mass = random()

total\_mass += vertex\_mass

vertex\_masses.append(vertex\_mass)

center += vertex\_mass \* vertex.co

center = center / total\_mass + Vector(<Offset>)

omega\_direction = <Vector Vận Tốc Góc>

omega = Vector(omega\_direction)

normalized\_omega = Vector(omega\_direction)

normalized\_omega.normalize()

angular\_momentum = Vector((0, 0, 0))

distances = []

for idx, vertex in enumerate(obj.data.vertices):

r = vertex.co - center

d = r - normalized\_omega \* (r.dot(normalized\_omega))

m = vertex\_masses[idx]

angular\_momentum += m \* r.cross(omega.cross(d))

distances.append(d.length)

axis\_angular\_momentum = angular\_momentum.dot(normalized\_omega) \* normalized\_omega

print("Method 1 :")

print("Raw Angular Momentum : ", angular\_momentum)

print("Axis Angular Momentum : ", axis\_angular\_momentum)

moment\_of\_inertia = Matrix([

[0, 0, 0],

[0, 0, 0],

[0, 0, 0]

])

axis\_moment\_of\_inertia = 0

for idx, vertex in enumerate(obj.data.vertices):

x, y, z = vertex.co - center

m = vertex\_masses[idx]

moment\_of\_inertia += m \* Matrix([

[y\*\*2 + z\*\*2, -x\*y, -x\*z],

[-x\*y, x\*\*2 + z\*\*2, -y\*z],

[-x\*z, -y\*z, x\*\*2 + y\*\*2]

])

axis\_moment\_of\_inertia += m \* distances[idx]\*\*2

angular\_momentum = moment\_of\_inertia @ omega

axis\_angular\_momentum = axis\_moment\_of\_inertia \* omega

print("")

print("Method 2 : ")

print("Raw Angular Momentum : ", angular\_momentum)

print("Axis Angular Momentum : ", axis\_angular\_momentum)

print(moment\_of\_inertia)

* <Offset>, <Vector Vận Tốc Góc> là các Tuple gồm 3 phần tử là tọa độ
* Đoạn Code trên sẽ tính trọng điểm của Object đang chọn = cách coi nó là hệ chất điểm với các đỉnh là chất điểm với khối lượng ngẫu nhiên, lưu ý hệ quy chiếu Local, sau đó tìm mốc để tính Moment động lượng = trọng tâm + <Offset>
* Giả định rằng tất cả các đỉnh của Object đang quay quanh mốc với trục có hướng của <Vector Vận Tốc Góc>
* Đoạn Code trên cũng sẽ tính Moment động lượng = phương pháp nâng cao là sử dụng Moment quán tính, với mốc trùng với khi tính Moment động lượng và cũng in ra Moment quán tính
* Đoạn Code trên cũng sẽ trả về Moment động lượng theo trục quay = 2 phương pháp

1. Dùng Ma Trận Biến Đổi Hình Dạng Của 1 Object?

import bpy

from mathutils import Matrix

ob = bpy.context.object

me = ob.data

M = Matrix(<Ma Trận 4 x 4>)

me.transform(M)

me.update()

* <Ma Trận 4 x 4> là ma trận tích hợp quay và tịnh tiến bạn muốn dùng để biến đổi hình dạng Object
* Ví dụ

M = Matrix((

(1, 0, 0, 1),

(0, 1, 0, 0),

(0, 0, 1, 0),

(0, 0, 0, 1),

))

1. Plot Đồ Thị 3D Của Hàm Bất Kì?

* Đoạn code này xóa tất cả vật thể hiện tại, rồi tạo Object mới là đồ thị

import bpy

from math import \*

bpy.ops.object.select\_all(action='DESELECT')

bpy.ops.object.select\_by\_type(type='MESH')

bpy.ops.object.delete()

mesh = bpy.data.meshes.new("graph")

obj = bpy.data.objects.new("GraphObj", mesh)

scene = bpy.context.scene

scene.objects.link(obj)

vertices = []

edges = []

faces = []

num\_points = <Độ Phân Giải>

lim\_x = <Giới Hạn Giá Trị Của X>

lim\_y = <Giới Hạn Giá Trị Của Y>

x\_step = (lim\_x[1] - lim\_x[0]) / num\_points

y\_step = (lim\_y[1] - lim\_y[0]) / num\_points

for i in range(num\_points):

x = lim\_x[0] + i \* x\_step

for j in range(num\_points):

y = lim\_y[0] + j \* y\_step

z = <Biểu Thức Hàm>

vertices.append((x, y, z))

for i in range(num\_points - 1):

for j in range(num\_points - 1):

v0 = i \* num\_points + j

v1 = v0 + 1

v2 = v0 + num\_points

v3 = v2 + 1

edges.extend([(v0, v1), (v1, v3), (v3, v2), (v2, v0)])

faces.append((v0, v1, v3, v2))

mesh.from\_pydata(vertices, edges, faces)

mesh.update()

* <Giới Hạn Giá Trị Của X> và <Giới Hạn Giá Trị Của Y> là List bao gồm 2 giá trị cận có tác dụng giới hạn hiển thị hàm số
* Ví dụ

lim\_x = [-5, 5]

lim\_y = [-10, 10]

* <Độ Phân Giải> càng lớn thì đồ thị càng mượt
* <Biểu Thức Hàm> là hàm bạn muốn Plot
* Ví dụ hàm của bạn là z = 2x + 4y2

z = 2 \* x + 4 \* y \*\* 2

1. Plot Đường Cong?

* Đoạn Code này sẽ xóa tất cả Object hiện tại sau đó tạo đường cong có dạng

z = f(x, y), trong đó y = g(x)

import bpy

import numpy as np

from math import pow

def create\_curve():

curve\_data = bpy.data.curves.new(name='My Curve', type='CURVE')

curve\_data.dimensions = '3D'

spline = curve\_data.splines.new(type='POLY')

x\_values = np.linspace(<Cận Trái>, <Cận Phải>, <Độ Phân Giải>).tolist()

y\_values = [<Biểu Thức Y Theo X> for x in x\_values]

z\_values = [<Biểu Thức Đường Cong> for x, y in zip(x\_values, y\_values)]

spline.points.add(len(x\_values) - 1)

for i, (x, y, z) in enumerate(zip(x\_values, y\_values, z\_values)):

spline.points[i].co = (x, y, z, 1)

obj = bpy.data.objects.new('My Object', curve\_data)

scene = bpy.context.scene

scene.objects.link(obj)

obj.select = True

scene.objects.active = obj

bpy.ops.object.select\_all(action='DESELECT')

bpy.ops.object.select\_by\_type(type='CURVE')

bpy.ops.object.delete()

create\_curve()

* <Cận Trái> và <Cận Phải> là giới hạn hiển thị đường cong theo x
* <Độ Phân Giải> càng cao thì đường cong càng mượt
* <Biểu Thức Đường Cong> là f(x, y)
* Ví dụ z = f(x, y) = 2x + 4y2

2 \* x + 4 \* y \*\* 2

* <Biểu Thức Y Theo X> là g(x)
* Ví dụ y = g(x) = 5x

5 \* x

1. Render Từng Khoảng Frame Rời Rạc?

import bpy

FRAME\_RANGES = [<Các Khoảng Frame>]

OUTPUT\_PATH = <Đường Dẫn Tới Thư Mục Lưu Ảnh>

for frame\_range in FRAME\_RANGES:

start\_frame, end\_frame = frame\_range

for frame\_number in range(start\_frame, end\_frame + 1):

bpy.context.scene.frame\_set(frame\_number)

bpy.context.scene.render.filepath = OUTPUT\_PATH + str(frame\_number)

bpy.ops.render.render(write\_still=True)

* <Các Frame Range> là các Tuple chứa 2 phần tử lần lượt là Start Frame và End Frame ứng với mỗi khoảng Frame
* Ví dụ

[(0, 10), (20, 30), (50, 70)]

* Chạy đoạn Code trên sẽ Render Scene hiện tại dưới nền theo cài đặt ở giao diện người dùng

1. Xóa Toàn Bộ Key Frame Liên Quan Đến Chuyển Động Của Tất Cả Object Được Chọn?

import bpy

SELECTED\_OBJECTS = bpy.context.selected\_objects

for obj in SELECTED\_OBJECTS:

obj.animation\_data\_clear()

1. Xuất Tọa Độ Của Tất Cả Các Đỉnh Trong Object Được Chọn Sang File CSV?

import bpy

CSV\_FILE = <Đường Dẫn Tuyệt Đối Tới File CSV Có Phần Mở Rộng>

VERTICES = [

bpy.context.object.matrix\_world @ x.co

for x in bpy.context.object.data.vertices

]

f = open(CSV\_FILE, 'w')

f.writelines([";".join([str(x) for x in coord]) + "\n" for coord in VERTICES])

f.close()

* Trong File CSV, mỗi dòng sẽ là 1 bộ 3 hoành, tung, cao độ ngăn cách nhau bằng dấu chấm phẩy, ví dụ

1.0;1.0;-1.0

1.0;-1.0;1.0

1.0;-1.0;-1.0