Assignment Project Exam Help

Add WeChat powcoder

Chapter 4 Part 1
CS-396
Jeff Bush

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Coordinates and Transformations

Assignment Project Exam Help

Scalars, Vectors, and Points

Add WeChat powcoder

- **Scalars** are a single value, either real or complex
- **Vectors** have direction and magnitude
- **Points** are a location in space
- Note: in GLSL and glMatrix.js vec2, vec3, or vec4 are simply the *storage* types and can hold anything with 2, 3, or 4 values (like points, vectors, colors, …)

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Scalars, Vectors, and Points

Add WeChat powcoder

- These form the minimum set of primitives necessary to build any object
- They all have well-defined properties in n-dimensional space but we only care about 3D versions
  - Actually 4D homogeneous versions

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Scalars

Add WeChat powcoder

- Fundamentally require:
  - Addition and multiplication
  - Associativity, commutativity, inverses
- We will use real and complex numbers with all their usual rules
- Scalars alone have no geometric properties

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Vectors

Add WeChat powcoder

- A quantity with two attributes:
  - Direction
  - Magnitude
- In the physical world:
  - Force
  - Velocity
- We usually draw them as a directed line segment

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

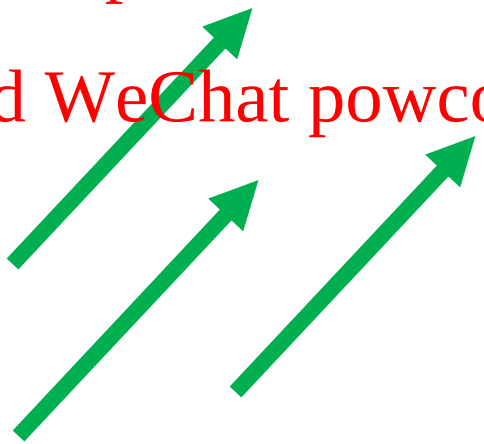
Assignment Project Exam Help

Vectors Lack Position

Add WeChat powcoder

- These vectors are all identical
  - Same direction and magnitude
  - Vectors do not include location information

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder



Assignment Project Exam Help

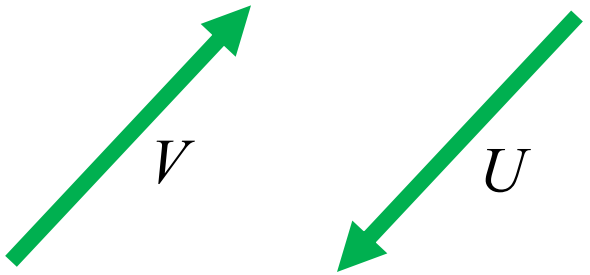
Vector Operations

Add WeChat powcoder

- A single zero vector: zero magnitude, undefined direction
- Inverse: same magnitude but opposite direction

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

$$U = -V$$



Assignment Project Exam Help

Vector Operations

Add WeChat powcoder

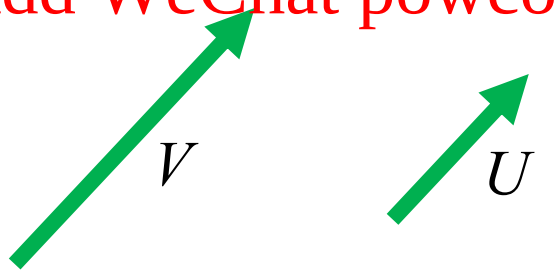
- scalar-vector multiplication: same direction and multiply magnitude by scalar

Assignment Project Exam Help

$$U = \alpha V$$

https://powcoder.com

Add WeChat powcoder



Assignment Project Exam Help

Vector Operations

Add WeChat powcoder

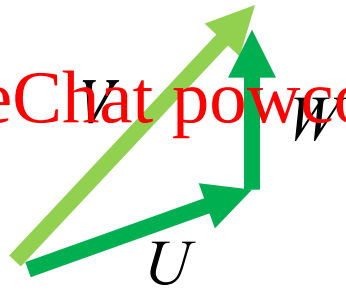
- vector-vector addition: use head-to-tail axiom

Assignment Project Exam Help

$$V = W + U$$

https://powcoder.com

Add WeChat powcoder



Assignment Project Exam Help

Linear Vector Spaces

Add WeChat powcoder

- Since we have the basics of scalar-vector multiplication and vector-vector addition we can use expressions like:

Assignment Project Exam Help
https://powcoder.com

$$V = U + 2W - 3R$$

Add WeChat powcoder

- Vectors spaces are insufficient for geometry since they lack location
  - Need points

Assignment Project Exam Help

Points

Add WeChat powcoder

- Represent a location in space
- Several operations allowed between vectors and points, but none between scalars and points

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Point Operations

Add WeChat powcoder

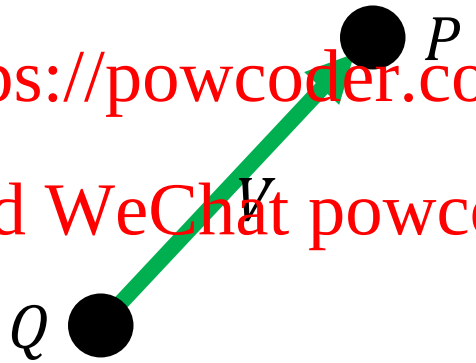
- Point-point subtraction yields a vector

$$V = P - Q$$

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder



- Point-vector addition is reverse of point-point subtraction

$$P = V + Q$$

Assignment Project Exam Help

Affine Spaces

Add WeChat powcoder

- Point plus a vector space is an ***affine space***
- Operations:

Assignment Project Exam Help

  - Vector-vector addition

https://powcoder.com

  - Scalar-vector multiplication

Add WeChat powcoder

  - Point-vector addition (and point-point subtraction)
  - Scalar-scalar operations
  - Homogenous coordinates operations

Assignment Project Exam Help

Cross (Outer) Product

Add WeChat powcoder

$$U \times V$$

- Results in a vector that is orthogonal to both vectors (if they are non-parallel vectors)
- Helps define linearly independent vector spaces (soon)
- Can be used to get the sine of the angle between two vectors:

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

$$\sin(\theta) = \frac{\|U \times V\|}{\|U\|\|V\|}$$

Assignment Project Exam Help

Dot (Inner, Scalar) Product

Add WeChat powcoder

$$U \cdot V = U_1V_1 + U_2V_2 + \cdots + U_nV_n$$

- Results in a scalar value
- If $U \cdot V = 0$ then the vectors are orthogonal
- $\|U\| = \sqrt{U \cdot U}$
- Can be used to get the cosine of the angle between two vectors:

$$\cos(\theta) = \frac{U \cdot V}{\|U\|\|V\|}$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Linear Independence

Add WeChat powcoder

- A set of vectors $V_1, V_2, \ldots, V_n$ is *linearly independent* if

Assignment Project Exam Help

$$\alpha_1 V_1 + \alpha_2 V_2 + \cdots + \alpha_n V_n = \mathbf{0} \; iff$$

https://powcoder.com

$$\alpha_1 = \cdots = \alpha_n = 0$$

- If a set of vectors is linearly independent we cannot represent one vector in terms of the others

Add WeChat powcoder

  - Conversely, if a set of vectors is linearly *dependent* at least one can be written in terms of the others

Assignment Project Exam Help

Dimension

Add WeChat powcoder

- In a vector space, the maximum number of linearly independent vectors is fixed and is called the *dimension* of the space
- In an *n*-dimensional space, any set of n linearly independent vectors form a *basis* for the space
- Given a basis $V_1, V_2, \ldots, V_n$ any vector V can be written as:

$$V = \alpha_1 V_1 + \alpha_2 V_2 + \cdots + \alpha_n V_n$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

The 3rd Dimension

Add WeChat powcoder

- In 3D our *basis* is formed from 3 linearly independent vectors
- If our basis is V_1, V_2, V_3 then every possible vector can be written as:

Assignment Project Exam Help

https://powcoder.com

$$V = \alpha_1 V_1 + \alpha_2 V_2 + \alpha_3 V_3$$

Add WeChat powcoder

  - The unique set of scalars $(\alpha_1, \alpha_2, \alpha_3)$ is the *representation* of v with respect to the given basis
- We can write the representation as an array of scalars:

$$\mathbf{a} = [\alpha_1 \quad \alpha_2 \quad \alpha_3]^T = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

  - We will use boldface letters for representations

Assignment Project Exam Help

The 3rd Dimension

Add WeChat powcoder

- The standard/natural basis for 3 dimensions is:

$$V_1 = \langle 1,0,0 \rangle$$
$$V_2 = \langle 0,1,0 \rangle$$
$$V_3 = \langle 0,0,1 \rangle$$

Assignment Project Exam Help
https://powcoder.com

  - We can check that they are linearly independent by using the cross product:

Add WeChat powcoder

$$V_1 \times V_2 \neq 0, V_1 \times V_3 \neq 0, V_2 \times V_3 \neq 0$$

- In this basis the representation of a vector is simply the x, y, and z components:

$$v = \langle \alpha_x, \alpha_y, \alpha_z \rangle = \alpha_x V_1 + \alpha_y V_2 + \alpha_z V_3$$
$$\mathbf{a} = [\alpha_x \quad \alpha_y \quad \alpha_z]^T$$

Assignment Project Exam Help

The 3rd Dimension

Add WeChat powcoder

- Other bases are possible though! Example:

$$V_1 = \langle 1,0,0 \rangle$$
$$V_2 = \langle 0,0,-1 \rangle$$
$$V_3 = \langle 0,1,0 \rangle$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

- What is the representation of the following vectors in this basis?

$\langle 1,0,0 \rangle$

$\langle 0,1,0 \rangle$

$\langle 0,0,1 \rangle$

$\langle 1,2,3 \rangle$

Assignment Project Exam Help

The 3rd Dimension

Add WeChat powcoder

- Other bases are possible though! Example:

$$V_1 = \langle 1,0,0 \rangle$$
$$V_2 = \langle 0,0,-1 \rangle$$
$$V_3 = \langle 0,1,0 \rangle$$

Assignment Project Exam Help
https://powcoder.com

- What is the representation of the following vectors in this basis?

Add WeChat powcoder

$\langle 1,0,0 \rangle \quad [1 \quad 0 \quad 0]^T$

$\langle 0,1,0 \rangle \quad [0 \quad 0 \quad 1]^T$

$\langle 0,0,1 \rangle \quad [0 \quad -1 \quad 0]^T$

$\langle 1,2,3 \rangle \quad [1 \quad -3 \quad 2]^T$

Assignment Project Exam Help

Coordinate Systems

Add WeChat powcoder

- Each coordinate system has its own basis
- Converting between coordinate systems is the same as transforming between bases
  - The last slide had the rotation 90° around the x-axis basis
- A coordinate system is an array of basis vectors:

$$\mathbf{v} = [V_1 \quad V_2 \quad V_3]^T$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Coordinate Systems

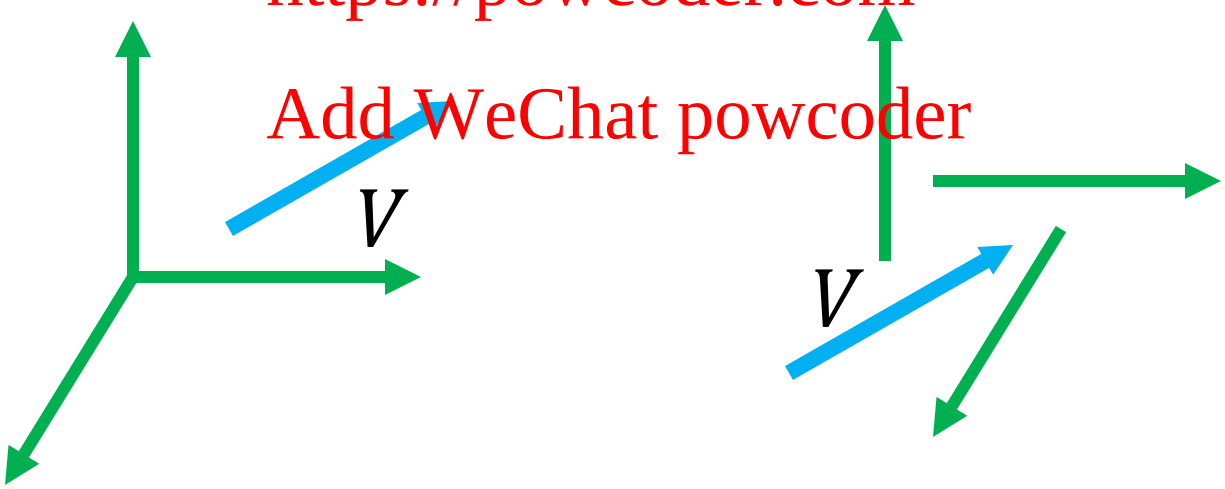
Add WeChat powcoder

If green vectors are the basis, which is correct?

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder



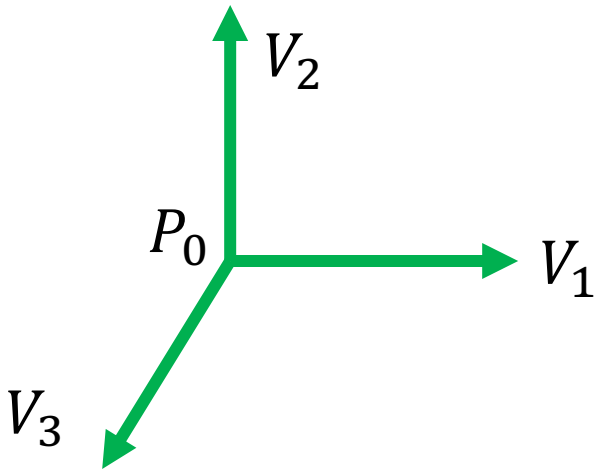
Assignment Project Exam Help

Frames

Add WeChat powcoder

- A coordinate system is insufficient to represent points
- If we work in an affine space we can add a single point, the *origin*, to the basis vectors to form a *frame*

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder



Assignment Project Exam Help

Representation in a Frame

Add WeChat powcoder

- We need an array of 3 basis vectors plus 1 point for the origin:

Assignment Project Exam Help

$$[V_1 \quad V_2 \quad V_3 \quad P_0]$$

- Within this frame:

https://powcoder.com

  - Every vector can be written as:

$$V = \alpha_1 V_1 + \alpha_2 V_2 + \alpha_3 V_3$$

Add WeChat powcoder

  - Every point can be written as:

$$P = P_0 + \beta_1 V_1 + \beta_2 V_2 + \beta_3 V_3$$

- They have identical representations:

$$\mathbf{v} = [\alpha_1 \quad \alpha_2 \quad \alpha_3]^T \qquad \mathbf{p} = [\beta_1 \quad \beta_2 \quad \beta_3]^T$$

But then how do we distinguish them? We need to be able to since vectors don't have a position…

Assignment Project Exam Help

Homogeneous Coordinates

Add WeChat powcoder

- If we define $0 * P = \mathbf{0}$ and $1 * P = P$ then:

$$V = \alpha_1 V_1 + \alpha_2 V_2 + \alpha_3 V_3 + 0 * P_0 = [\alpha_1 \quad \alpha_2 \quad \alpha_3 \quad 0]\mathbf{F}^{\mathrm{T}}$$
$$P = \beta_1 V_1 + \beta_2 V_2 + \beta_3 V_3 + 1 * P_0 = [\beta_1 \quad \beta_2 \quad \beta_3 \quad 1]\mathbf{F}^{\mathrm{T}}$$

Assignment Project Exam Help

where the frame is given by:

https://powcoder.com

$$\mathbf{F} = [V_1 \quad V_2 \quad V_3 \quad P_0]$$

Add WeChat powcoder

- And now we obtain our 4D homogeneous coordinate representation:

$$\mathbf{v} = [\alpha_1 \quad \alpha_2 \quad \alpha_3 \quad 0]$$
$$\mathbf{p} = [\beta_1 \quad \beta_2 \quad \beta_3 \quad 1]$$

Assignment Project Exam Help
Add WeChat powcoder

Homogeneous Coordinates and Computer Graphics

- Homogeneous coordinates are key to all computer graphics systems
- All standard transformations like rotation, translation, and scaling can be implemented with matrix multiplications using 4 x 4 matrices
- Hardware pipeline works with 4 dimensional representations
- For orthographic viewing, we can maintain $w = 0$ for vectors and $w = 1$ for points
  - E.g. $\langle x, y, z, 0 \rangle$ is a vector and $\langle x, y, z, 1 \rangle$ is a point
- For perspective we need a *perspective division*

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Homogeneous Coordinates Check

Add WeChat powcoder

- Do all of our basic formulas for vectors and points work out when using homogeneous coordinates?
  - Vector Inverse?
  - Scalar-Vector Multiplication?
  - Vector Addition?
  - Point-Point Subtraction?
  - Point-Vector Addition?
- How about other things not on the list that don't make sense – do they still not make sense?
  - Scalar-Point Multiplication?
  - Point-Point Addition?

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Change of Coordinate Systems

Add WeChat powcoder

Consider two representations of a the same vector with respect to two different bases. The representations are:

Assignment Project Exam Help
https://powcoder.com

$$\mathbf{a} = [\alpha_1 \quad \alpha_2 \quad \alpha_3]$$
$$\mathbf{b} = [\beta_1 \quad \beta_2 \quad \beta_3]$$

where

Add WeChat powcoder

$$V = \alpha_1 V_1 + \alpha_2 V_2 + \alpha_3 V_3 = [\alpha_1 \quad \alpha_2 \quad \alpha_3][V_1 \quad V_2 \quad V_3]^{\mathrm{T}}$$
$$= \beta_1 U_1 + \beta_2 U_2 + \beta_3 U_3 = [\beta_1 \quad \beta_2 \quad \beta_3][U_1 \quad U_2 \quad U_3]^{\mathrm{T}}$$

Assignment Project Exam Help

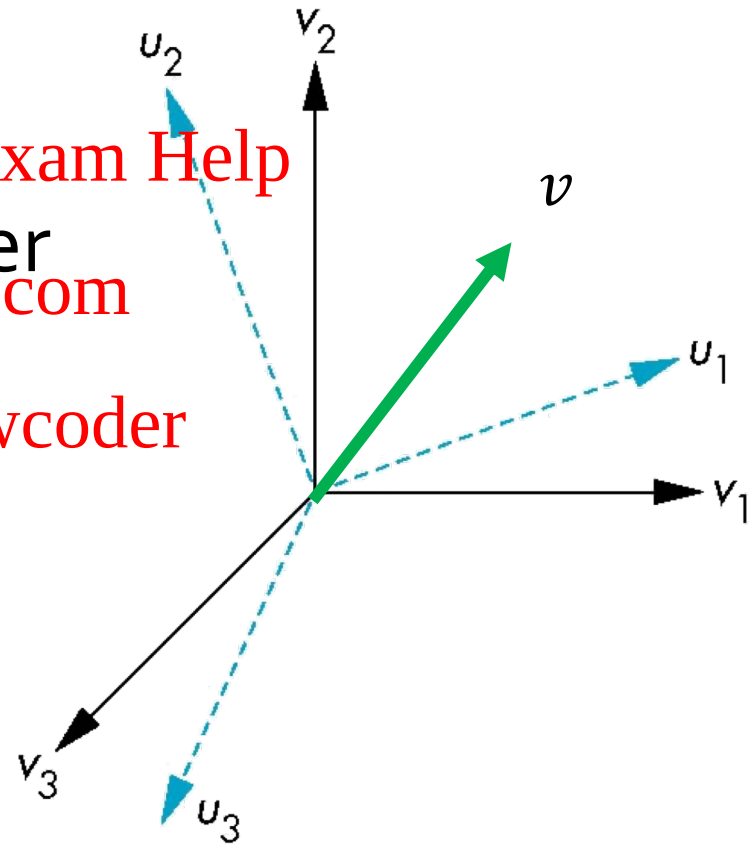
Representing one basis in terms of another

Add WeChat powcoder

Each of the basis vectors U_1, U_2, and U_3 are vectors that can be represented in another basis

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

$$U_1 = \gamma_{11}V_1 + \gamma_{12}V_2 + \gamma_{13}V_3$$
$$U_2 = \gamma_{21}V_1 + \gamma_{22}V_2 + \gamma_{23}V_3$$
$$U_3 = \gamma_{31}V_1 + \gamma_{32}V_2 + \gamma_{33}V_3$$



Assignment Project Exam Help

Matrix Form

Add WeChat powcoder

- The coefficients define a 3x3 matrix

$$\mathbf{M} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \end{bmatrix}$$

Assignment Project Exam Help
https://powcoder.com

- Then the equation for converting from one basis to another can be written as:

Add WeChat powcoder

$$\mathbf{a} = \mathbf{M}^T \mathbf{b}$$

Assignment Project Exam Help

Change of Frames

Add WeChat powcoder

- That only allows the coordinate system to be changed but not the frame
- We can apply a similar process in homogeneous coordinates to the representations of both points and vectors
- Consider two frames:

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

$$\mathbf{F} = [V_1 \quad V_2 \quad V_3 \quad P_0]$$

$$\mathbf{G} = [U_1 \quad U_2 \quad U_3 \quad Q_0]$$

- Any point/vector can be represented in either frame
- We can also represent $\mathbf{G}$ in terms of $\mathbf{F}$

Assignment Project Exam Help

Representing one frame in terms of another

Add WeChat powcoder

$$U_1 = \gamma_{11}V_1 + \gamma_{12}V_2 + \gamma_{13}V_3$$
$$U_2 = \gamma_{21}V_1 + \gamma_{22}V_2 + \gamma_{23}V_3$$
$$U_3 = \gamma_{31}V_1 + \gamma_{32}V_2 + \gamma_{33}V_3$$
$$Q_0 = \gamma_{41}V_1 + \gamma_{42}V_2 + \gamma_{43}V_3 + P_0$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Or in matrix form:

$$\mathbf{M} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} & 0 \\ \gamma_{21} & \gamma_{22} & \gamma_{23} & 0 \\ \gamma_{31} & \gamma_{32} & \gamma_{33} & 0 \\ \gamma_{41} & \gamma_{42} & \gamma_{43} & 1 \end{bmatrix}$$

Assignment Project Exam Help

Working with Representations

Add WeChat powcoder

- Within the two frames any point or vector has a representation of the same form

Assignment Project Exam Help

$\mathbf{a} = [\alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \alpha_4]$ in the first frame

$\mathbf{b} = [\beta_1 \quad \beta_2 \quad \beta_3 \quad \beta_4]$ in the second frame

https://powcoder.com

- Where $\alpha_4 = \beta_4 = 1$ for points and $\alpha_4 = \beta_4 = 0$ for vectors and

Add WeChat powcoder

$$\mathbf{a} = \mathbf{M}^T\mathbf{b}$$

performs an affine transformation on homogeneous coordinates using a 4x4 matrix $\mathbf{M}$

- Every point and vector is given by:
  - 3 scalar values, 0 or 1 for indicating type, and the frame it is represented in

Assignment Project Exam Help

The World and Camera Frames

Add WeChat powcoder

- Changes in frames are defined by a 4x4 transformation matrix which is the transpose of the matrix we have been working with ($\mathbf{T} = \mathbf{M}^T$)
- We start the world frame and eventually need everything in the camera frame
  - Accomplished by multiplying the points and vectors by the "model-view matrix"
- So far these frames have been the same and the transformation has been the identity matrix:

$$\mathbf{T} = \mathbf{M}^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Affine Transformations

Add WeChat powcoder

- Every linear transformation is equivalent to a change in frames
- Every affine transformation preserves lines
- Affine transformations have 12 degrees of freedom
  - Even though it uses a 4x4 (16 element) matrix, the bottom row is fixed and thus only a subset of all 4D linear transformations are possible
- Characteristic of many physically important transformations:
  - Rigid body: translation and rotation
  - Scaling and shear
- Importance in graphics is that we need to only transform endpoints of line segments and let implementation draw line segment between the transformed endpoints

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Frame of Reference

Add WeChat powcoder

- As scenes get more complex every object will likely have its own frame of reference and those will be grouped in other frames of reference, which are grouped into other frames, …
  - For example: this screen is 4 inches from the west wall of the room, this room is on the second floor in the south-east corner of the building, the building is in the south-east corner of the campus, the campus is in the middle of the city, the city is in the southeast corner of the state…
  - Called a *scene graph*
  - We will end up having several transformation matrices applied to every point

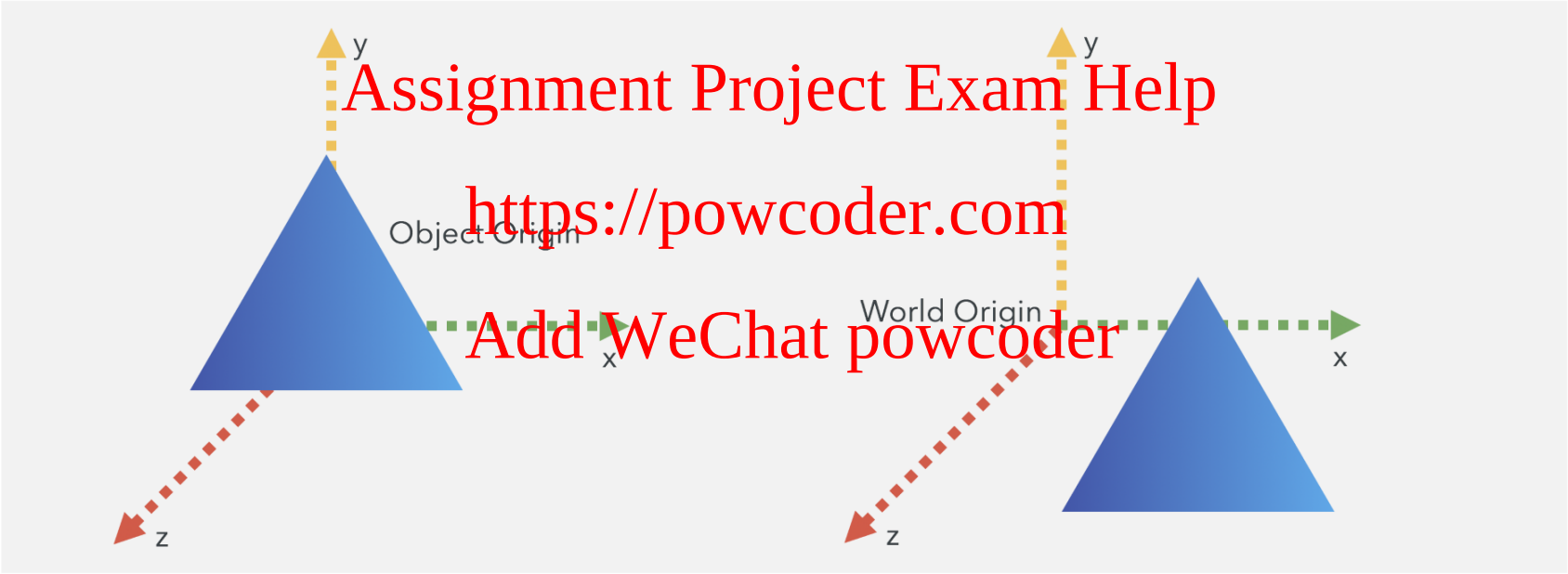
Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Model Transform

Add WeChat powcoder

Model Transform



Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

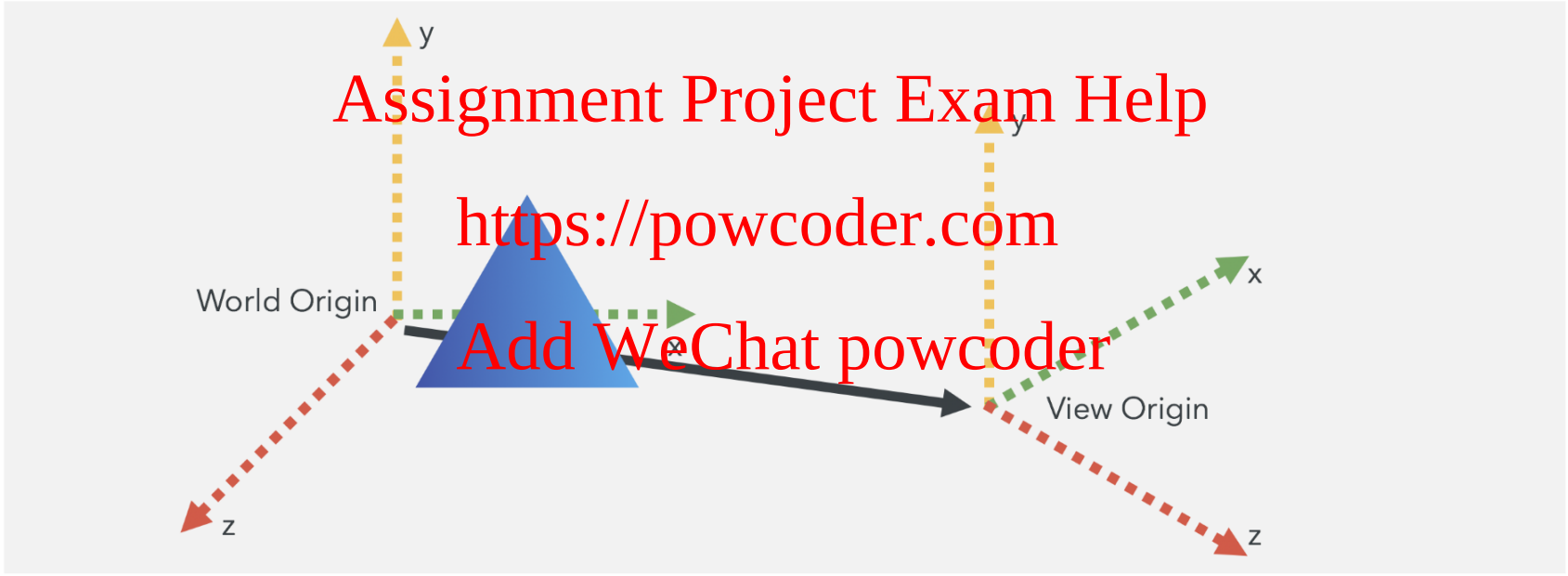
The object has not moved. The model transform assigns coordinates that are shared by all objects: the world coordinates.

Assignment Project Exam Help

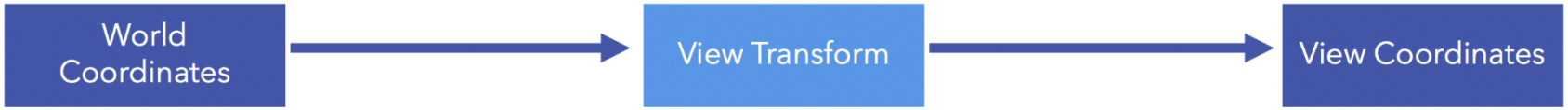
View Transform

Add WeChat powcoder

View Transform



Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder



The view transform moves the origin of the world to the coordinates of the view. This is where our *camera* is located.

Assignment Project Exam Help

Model + View Transform

Add WeChat powcoder

- What is the difference between moving the viewer and moving the object?

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Assignment Project Exam Help

Model + View Transform

Add WeChat powcoder

- What is the difference between moving the viewer and moving the objects/world?

Assignment Project Exam Help

  - Just reversed, moving objects/world:

https://powcoder.com

    - forward is the same as moving the camera backwards
    - right is the same as moving the camera left

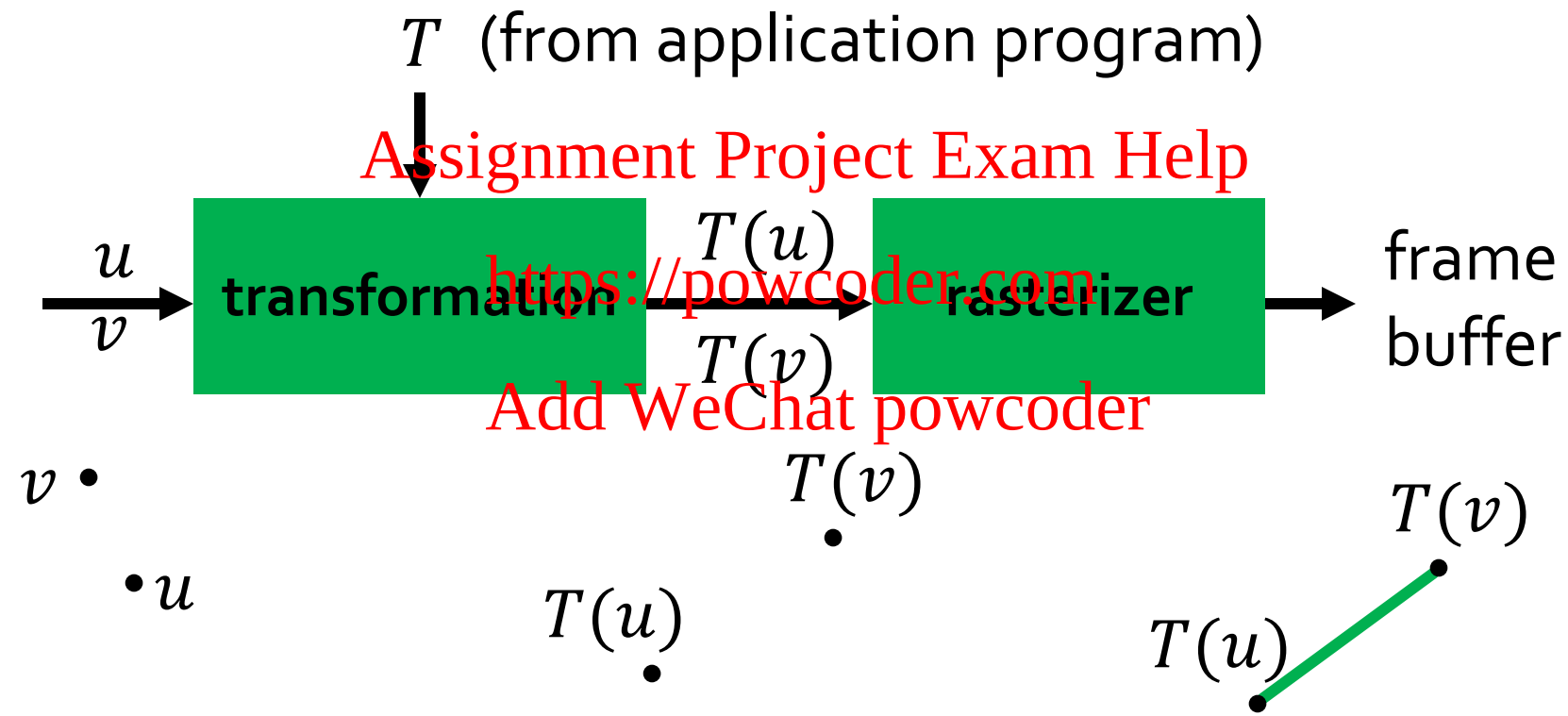
Add WeChat powcoder

    - up is the same as moving the camera down

Assignment Project Exam Help

Pipeline Implementation

Add WeChat powcoder



Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

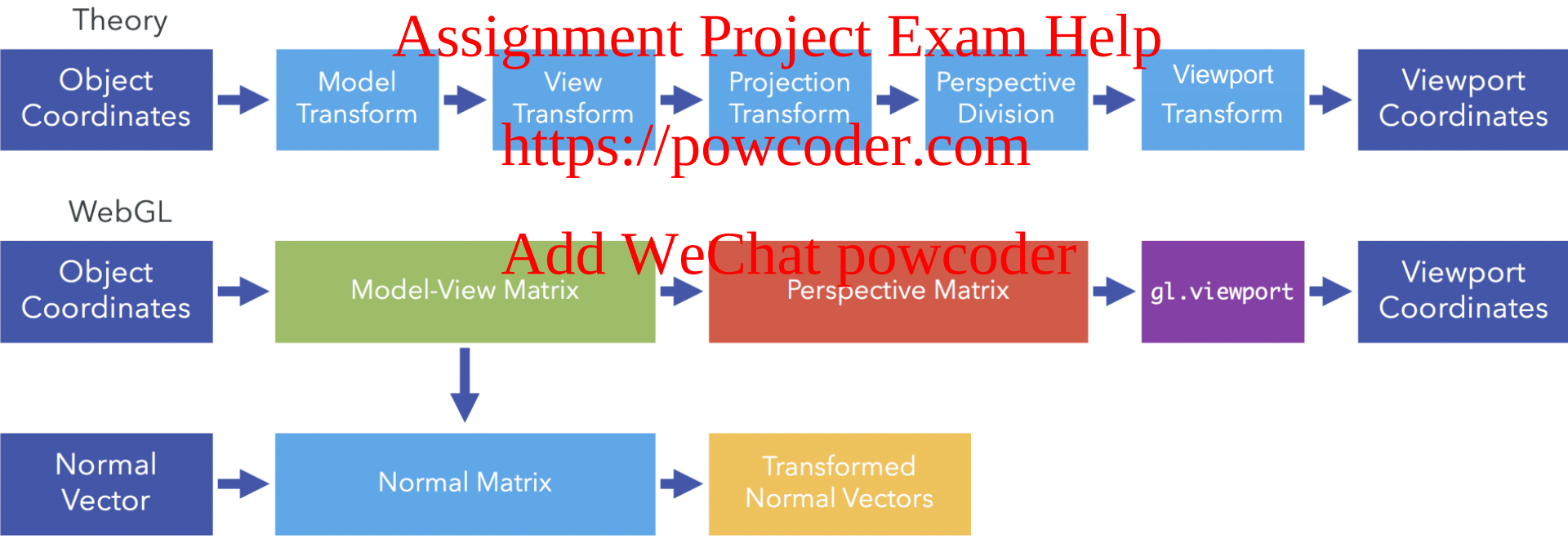
Pipeline Implementation

Add WeChat powcoder

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder



Assignment Project Exam Help

Homogeneous Conversion

Add WeChat powcoder

- Sometimes we want vectors and points in homogeneous coordinates
  - 4D with $w == 0$ is a vector and $w == 1$ is a point
- Sometimes we want vectors and point in Euclidean coordinates
  - 3D and have to guess at vectors vs points
- Can convert between the two. In GLSL, how do we get a…
  - homogeneous version of a Euclidean vector?
  - homogeneous version of a Euclidean point?
  - Euclidean version of a homogeneous vector?
  - Euclidean version of a homogeneous point?

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Homogeneous Conversion

Add WeChat powcoder

- Sometimes we want vectors and points in homogeneous coordinates
  - 4D with $w = 0$ is a vector and $w = 1$ is a point
- Sometimes we want vectors and point in Euclidean coordinates
  - 3D and have to guess at vectors vs points
- Can convert between the two. In GLSL, how do we get a…
  - homogeneous version of a Euclidean vector? `vec4(v, 0.0)`
  - homogeneous version of a Euclidean point? `vec4(v, 1.0)`
  - Euclidean version of a homogeneous vector? `v.xyz`
  - Euclidean version of a homogeneous point? `p.xyz`

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

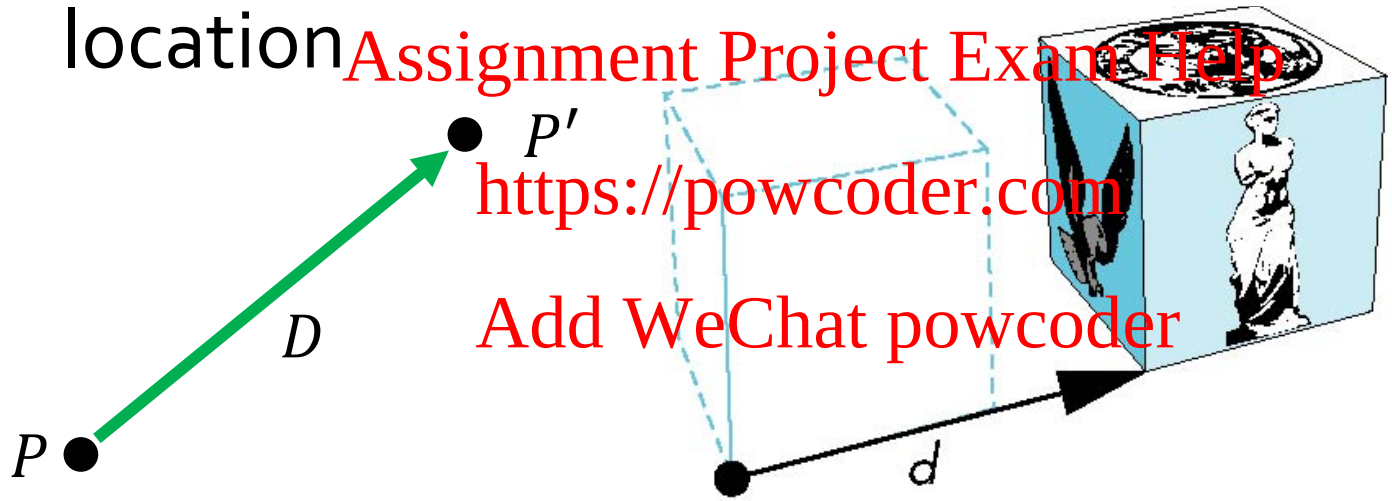
Assignment Project Exam Help

Translation

Add WeChat powcoder

- Move/displace/translate a point to a new location

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder



- Displacement determined by a vector V

$$P' = P + D$$

  - 3 degrees of freedom (x, y, and z displacement)

Assignment Project Exam Help

Translating using Representations

Add WeChat powcoder

- Use the homogeneous coordinate representation in some frame

Assignment Project Exam Help

$$\mathbf{p} = [x \quad y \quad z \quad 1]^T$$

$$\mathbf{p}' = [x' \quad y' \quad z' \quad 1]^T$$

https://powcoder.com

$$\mathbf{d} = [D_x \quad D_y \quad D_z \quad 0]^T$$

With:

Add WeChat powcoder

$$\mathbf{p}' = \mathbf{p} + \mathbf{d}$$

vector-point addition in 4D

Or:

$$x' = x + D_x$$

$$y' = y + D_y$$

$$z' = z + D_z$$

Assignment Project Exam Help

Translation Matrix

Add WeChat powcoder

Can also express using a 4x4 matrix so that:

$$\mathbf{p}' = \mathbf{T}\mathbf{p}$$

Assignment Project Exam Help

using the following transformation matrix:

https://powcoder.com

Add WeChat powcoder

$$\mathbf{T}(D_x, D_y, D_z) = \begin{bmatrix} 1 & 0 & 0 & D_x \\ 0 & 1 & 0 & D_y \\ 0 & 0 & 1 & D_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Using a transformation matrix here is technically more computations but the GPU has built-in 4x4 matrix multiplication and all affine transforms can be done in this form and can be concatenated together

Assignment Project Exam Help

Adding Model-View Matrix

Add WeChat powcoder

- Before adding translation we need to add the model-view matrix to our setup:

Assignment Project Exam Help

  - What data type is the variable?

https://powcoder.com

  - What kind of variable in which shader?

Add WeChat powcoder

    - What other things do we need to when adding this kind of variable to make it usable?
  - Which values do we need to apply it to?
    - How do we apply it to each of them?
    - Is it applied to each the same way?

Assignment Project Exam Help

Adding Model-View Matrix

Add WeChat powcoder

- In translation example:
  - Add the uniform `mat4 uModelView` to vertex shader
    - Apply it to `aPosition` and `aNormal`
      - For aNormal just get the 3x3 matrix by doing `mat3(uModelView)`
      - "Applying it" means which math operation?
        - The matrix ***must*** be the first operand
  - Make sure to get access to it from JS

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Adding Translation

Add WeChat powcoder

- To get a translation matrix:
  - Function is `glMatrix.mat4.fromTranslation()`
  - Takes two arguments:
    - The matrix we are saving to
    - The array with the x, y, z translation
  - To create a matrix use `glMatrix.mat4.create()`
  - Make sure to call your variable `mv` so my code after it works
- Update the uniform with your new model-view matrix:
  - `gl.uniformMatrix4fv(LOC, false, mv);`

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Play Around with Translations

Add WeChat powcoder

- What happens if you translate too far to the right or left?
- What happens if you translate too far up or down?
- What happens if you translate too far forward or backward?
- What doesn't the teapot change size when further away?
- Are we translating the camera or the model?
  - How would we translate the other one?

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Rotation – 2D

Add WeChat powcoder

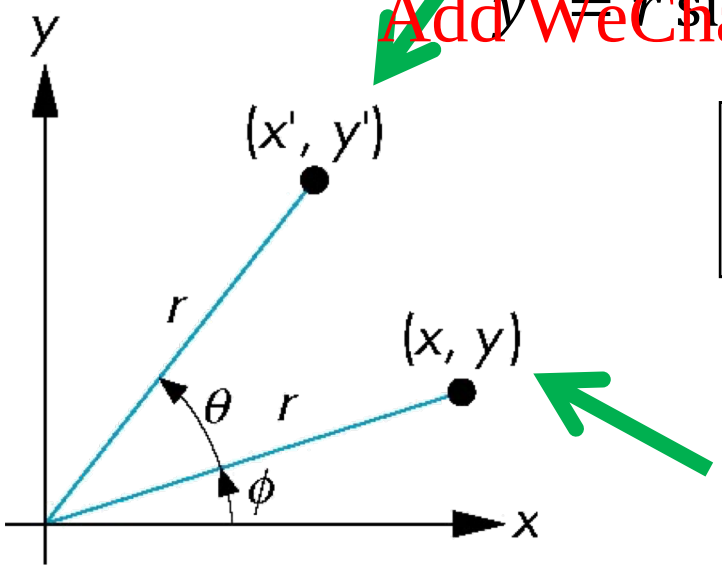
- Rotation about the origin by θ radians
  - Radius stays the same, angle increases by θ

Assignment Project Exam Help

https://powcoder.com

$$x' = r\cos(\phi + \theta)$$
$$y' = r\sin(\phi + \theta)$$

Add WeChat powcoder



$$x' = x\cos(\theta) - y\sin(\theta)$$
$$y' = x\sin(\theta) + y\cos(\theta)$$

$$x = r\cos(\phi)$$
$$y = r\sin(\phi)$$

Assignment Project Exam Help

Rotation Matrix –z-axis

Add WeChat powcoder

- Rotation about the z-axis in 3D leaves all points with the same z value
  - Equivalent to rotation in 2D keeping constant z

Assignment Project Exam Help

$$x' = x\cos(\theta) - y\sin(\theta)$$

https://powcoder.com

$$y' = x\sin(\theta) + y\cos(\theta)$$

$$z' = z$$

Add WeChat powcoder

  - Or as a transformation matrix:

$$\mathbf{p}' = \mathbf{R_Z}(\theta)\mathbf{p} \text{ with } \mathbf{R_Z}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Assignment Project Exam Help

Rotation Matrix – x/y-axis

Add WeChat powcoder

- Repeat process that we did for z-axis
  - For rotation about x-axis leaves x values unchanged
  - For rotation about y-axis leaves y values unchanged

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

$$\mathbf{R_X}(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R_Y}(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Assignment Project Exam Help

Rotation Matrix – Arbitrary Axis

Add WeChat powcoder

- We can perform any rotation as a single rotation around an arbitrary axis
  - Rotation matrix is fairly ugly…

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

$$\mathbf{R}(\theta, a) = \begin{bmatrix} a_x^2\dot{c} + c & a_x a_y \dot{c} - a_z s & a_x a_z \dot{c} + a_y s & 0 \\ a_x a_y \dot{c} + a_z s & a_y^2 \dot{c} + c & a_y a_z \dot{c} - a_x s & 0 \\ a_x a_z c - a_y s & a_y a_z \dot{c} + a_x s & a_z^2 \dot{c} + c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where a is the axis to rotate around (a unit vector), $s = \sin(\theta)$, $c = \cos(\theta)$, and $\dot{c} = 1 - c$

- Figuring out the axis to rotate around can be a bit tricky as well

Assignment Project Exam Help

Rotation of the Teapot

Add WeChat powcoder

- Do we need to adjust the shaders at all from the translation example to make this work?
  - Why or why not?

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Assignment Project Exam Help

Rotation of the Teapot

Add WeChat powcoder

- Let's first do rotation using the x, y, and z axes
- Use `rotate()`, `rotateY()`, and `rotateZ()` functions of `glMatrix` to compute the model-view matrix

Assignment Project Exam Help

https://powcoder.com

  - First argument is the output

Add WeChat powcoder

  - Second argument is the matrix to ***start from***
    - Your first and second arguments will be the same variable
  - Third argument is angle in ***radians***
    - We need to use and complete `deg2rad()` function
  - Need to build up the model-view matrix one rotation at a time

Assignment Project Exam Help

Play Around with Rotations

Add WeChat powcoder

- You probably setup your code to do X first, then Y, then Z
  - This actually performs them in the opposite order, but we will learn about that later
- What happens if you do them in a different order?

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Rotation of the Teapot

Add WeChat powcoder

- Let's do rotation around an arbitrary axis
  - Different examples then the last one
- Using the `fromRotation()` function of `glMatrix` compute the model-view matrix
  - First argument is the output
  - Second argument is angle in ***radians***
    - We need to use and complete `deg2rad()` function
  - Third argument is the axis as an array like [0, 1, 1]
  - Only one call this time!

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Play Around with Rotations

Add WeChat powcoder

- Can you get (non-trivial) same results with the XYZ rotation and the angle-axis rotation?
- What symmetries do you see in the model-view matrix?

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Scaling

Add WeChat powcoder

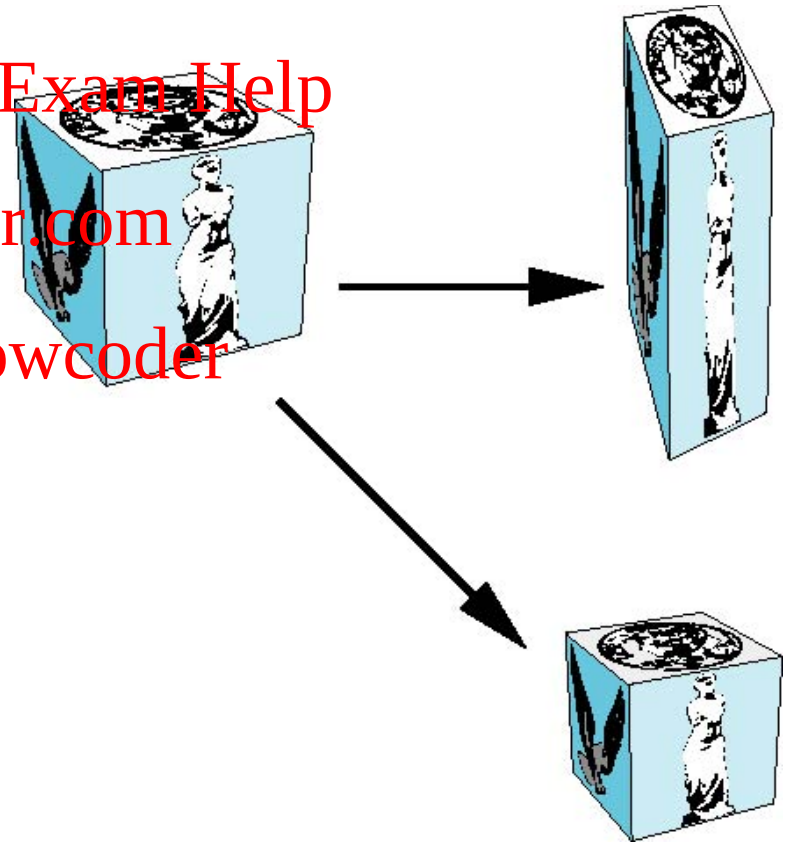
Expand/contract along each axis with a fixed origin point

$$x' = s_x x$$
$$y' = s_y y$$
$$z' = s_z z$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

$$\mathbf{p}' = \mathbf{S}\mathbf{p}$$

$$\mathbf{S}(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Assignment Project Exam Help

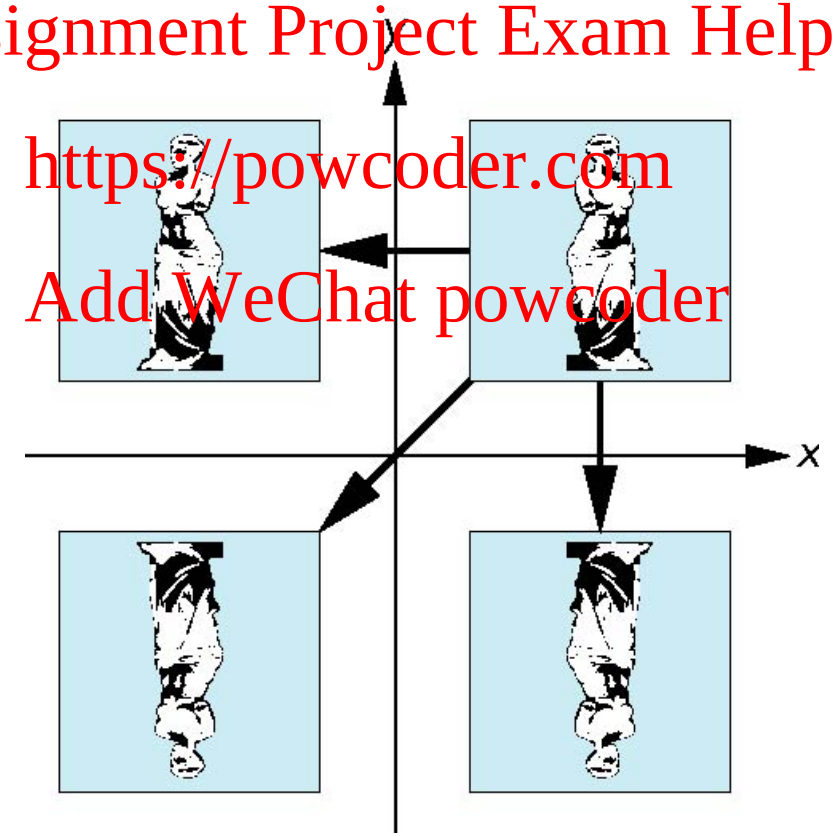
Reflection

Add WeChat powcoder

- Just scaling with negative parameters

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

$s_x = -1$
$s_y = 1$



$s_x = -1$
$s_y = -1$

$s_x = 1$
$s_y = -1$

Assignment Project Exam Help

Scaling and Reflection of the Teapot

Add WeChat powcoder

- Using the `fromScaling()` function of `glMatrix` compute the model-view matrix
  - First argument is the output
  - Second argument is the scaling vector (i.e. array)
- Play around with values including reflections

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Bonus: Shear Transformation

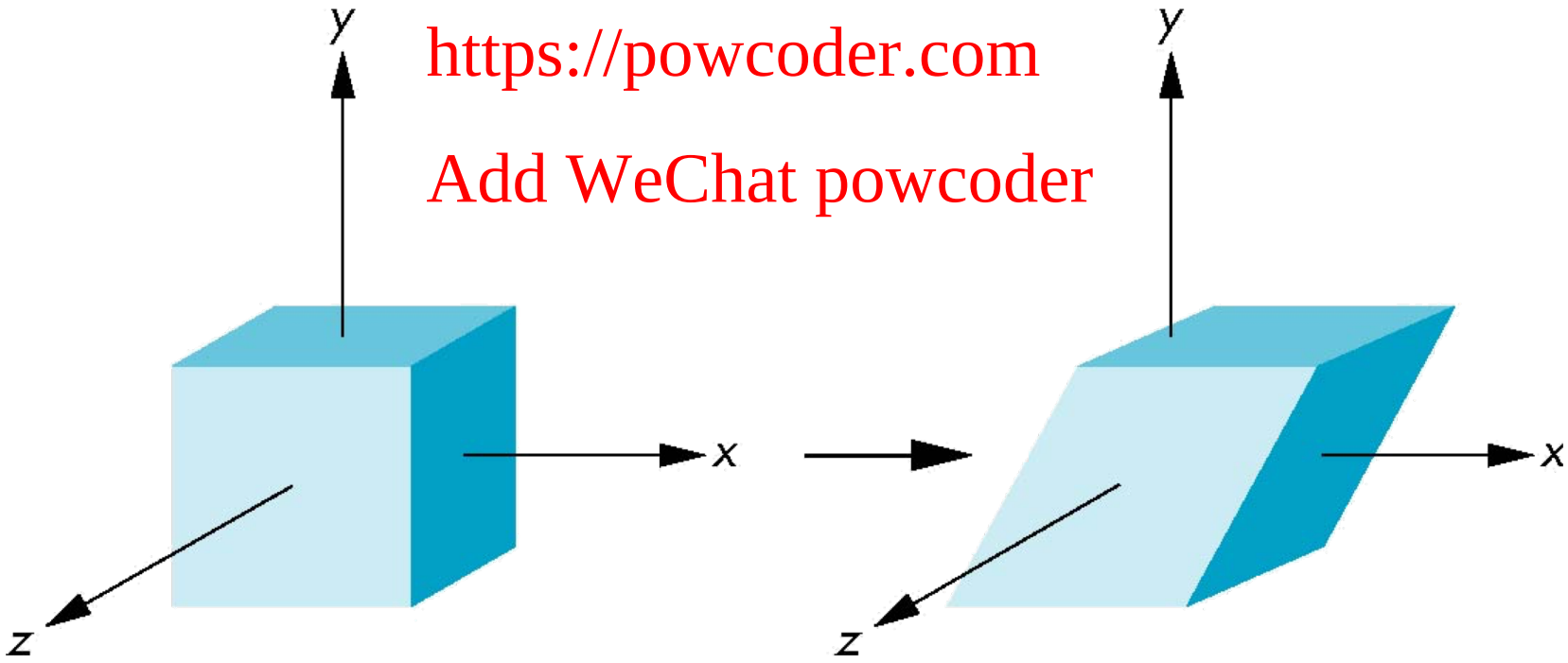
Add WeChat powcoder

Pulling faces of an object in opposite directions

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder



Assignment Project Exam Help

Bonus: Shear Transformation Matrix

Add WeChat powcoder

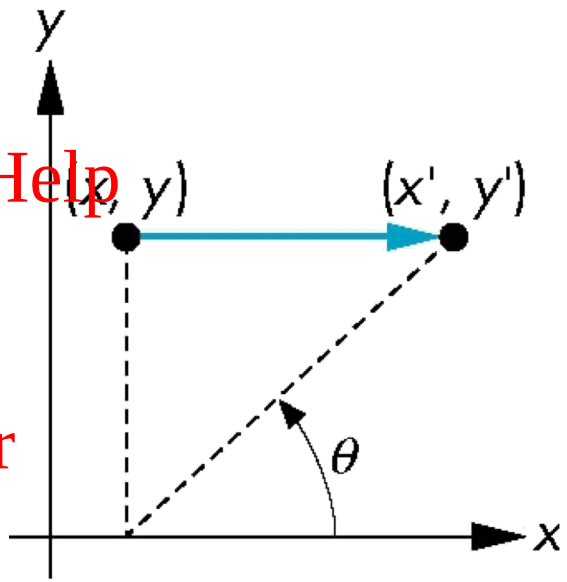
Shear along the x-axis:

$$x' = x + y\cot(\theta)$$
$$y' = y$$
$$z' = z$$

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder



$$\mathbf{H_x}(\theta) = \begin{bmatrix} 1 & \cot(\theta) & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Assignment Project Exam Help

Transform Matrix

Add WeChat powcoder

Rotation – Need 3x3 matrix to describe rotations in 3D

Assignment Project Exam Help

Scaling – Along diagonal, overlapping with rotation

https://powcoder.com

Add WeChat powcoder

$$\begin{bmatrix} a & b & c & x \\ d & e & f & y \\ g & h & i & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Translation – effects points but not vectors which is why it must be in last column

Fixed – last row always $0, 0, 0, 1$

Assignment Project Exam Help

Transformation Inverses

Add WeChat powcoder

- How to reverse effects of a transformation?
- We could compute inverse matrices, or we could use geometric observations:
  - Translation?
  - Rotation?
  - Scaling?
  - Shearing?

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Transformation Inverses

Add WeChat powcoder

- Translation: $\mathbf{T}^{-1}(D_x, D_y, D_z) = \mathbf{T}(-D_x, -D_y, -D_z)$
- Rotation: $\mathbf{R}^{-1}(\theta) = \mathbf{R}(-\theta)$
  - Works for rotation around any axis
  - Also $\mathbf{R}^{-1}(\theta) = \mathbf{R}^{T}(\theta)$ since $\cos(-\theta) = \cos(\theta)$ and $\sin(-\theta) = -\sin(\theta)$
- Scaling: $\mathbf{S}^{-1}(s_x, s_y, s_z) = \mathbf{S}(1/s_x, 1/s_y, 1/s_z)$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Concatenation

Add WeChat powcoder

- We can form arbitrary affine transformation matrices my multiplying together translation, rotation, and scaling matrices
- Because the same transformation is applied to several vertices the cost of forming the complete transformation matrix $\mathbf{M} = \mathbf{ABCD}$ is not significant compare to computing $\mathbf{Mp}$ for each $\mathbf{p}$
- The difficult part is how to form a desired complete transformation matrix from the what is going on in the application

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Order of Transformations

Add WeChat powcoder

- Matrix multiplication is not generally commutative so the order of application is critical
- Works from right-to-left:

$$\mathbf{p}' = \mathbf{ABCp} = \mathbf{A}(\mathbf{B}(\mathbf{Cp}))$$

- This assumes row-matrices to represent points, if using column matrices then needs lots of transposes and to reverse the order:

$$\mathbf{p}'^{T} = \mathbf{p}^{T}\mathbf{C}^{T}\mathbf{B}^{T}\mathbf{A}^{T}$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Compound Transforms

Add WeChat powcoder

- Two general "compound" transforms:
  - Rotation about the origin
    - Concatenation of rotations around each axis
  - Rotation about a fixed point
    - Concatenation of translation with rotation

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Rotation about the Origin

Add WeChat powcoder

- A rotation of θ about an arbitrary axis can be decomposed into the concatenation of rotations about the $x/y/z$-axes

Assignment Project Exam Help

https://powcoder.com

$$\mathbf{R}(\theta) = \mathbf{R_Z}(\theta_z)\mathbf{R_Y}(\theta_y)\mathbf{R_X}(\theta_x)$$

Add WeChat powcoder

- $\theta_x, \theta_y, \theta_z$ are called the Euler angles
- Note: the rotations do not commute, we could reorder them but would need different values for the thetas

Assignment Project Exam Help

Rotation Not about the Origin

Add WeChat powcoder

- The rotation transformations rotate the world around the origin
- How do we rotate around a different location?

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

Rotation not about the origin

Add WeChat powcoder

To rotate around a fixed point other than the origin:

1. Translate fixed point to origin
2. Rotate
3. Translate fixed point back

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

$$\mathbf{M} = \mathbf{T}(\mathbf{p}_f)\mathbf{R}(\theta)\mathbf{T}(-\mathbf{p}_f)$$