

Computer Graphics

- Camera (Questions)

J.-Prof. Dr. habil. Kai Lawonn

Possible Questions

Complement the sentence:

_____ z-axis is going through the screen/camera → moving the camera forwards means moving along the _____ z-axis

Possible Questions

Complement the sentence:

Positive z-axis is going through the screen/camera → moving the camera forwards means moving along the NEGATIVE z-axis

Possible Questions

How to ensure a consistent movement on various hardware?

Possible Questions

How to ensure a consistent movement on various hardware?

- Graphics applications and games usually keep track of a deltaTime variable that stores the time it takes to render the last frame
- Multiply all velocities with this deltaTime value

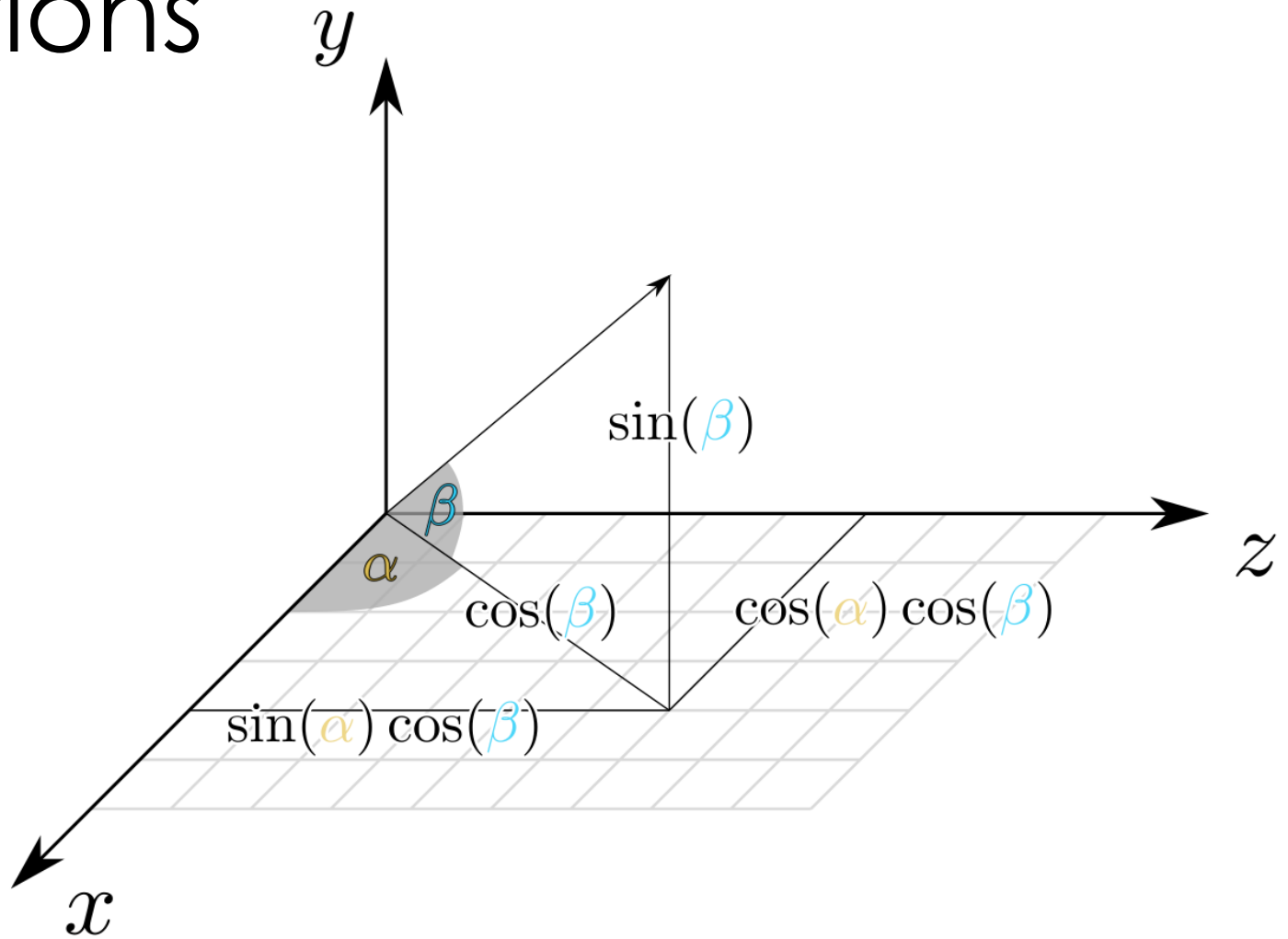
Possible Questions

Complete the equations:

$$\text{camera front}_y =$$

$$\text{camera front}_z =$$

$$\text{camera front}_x =$$



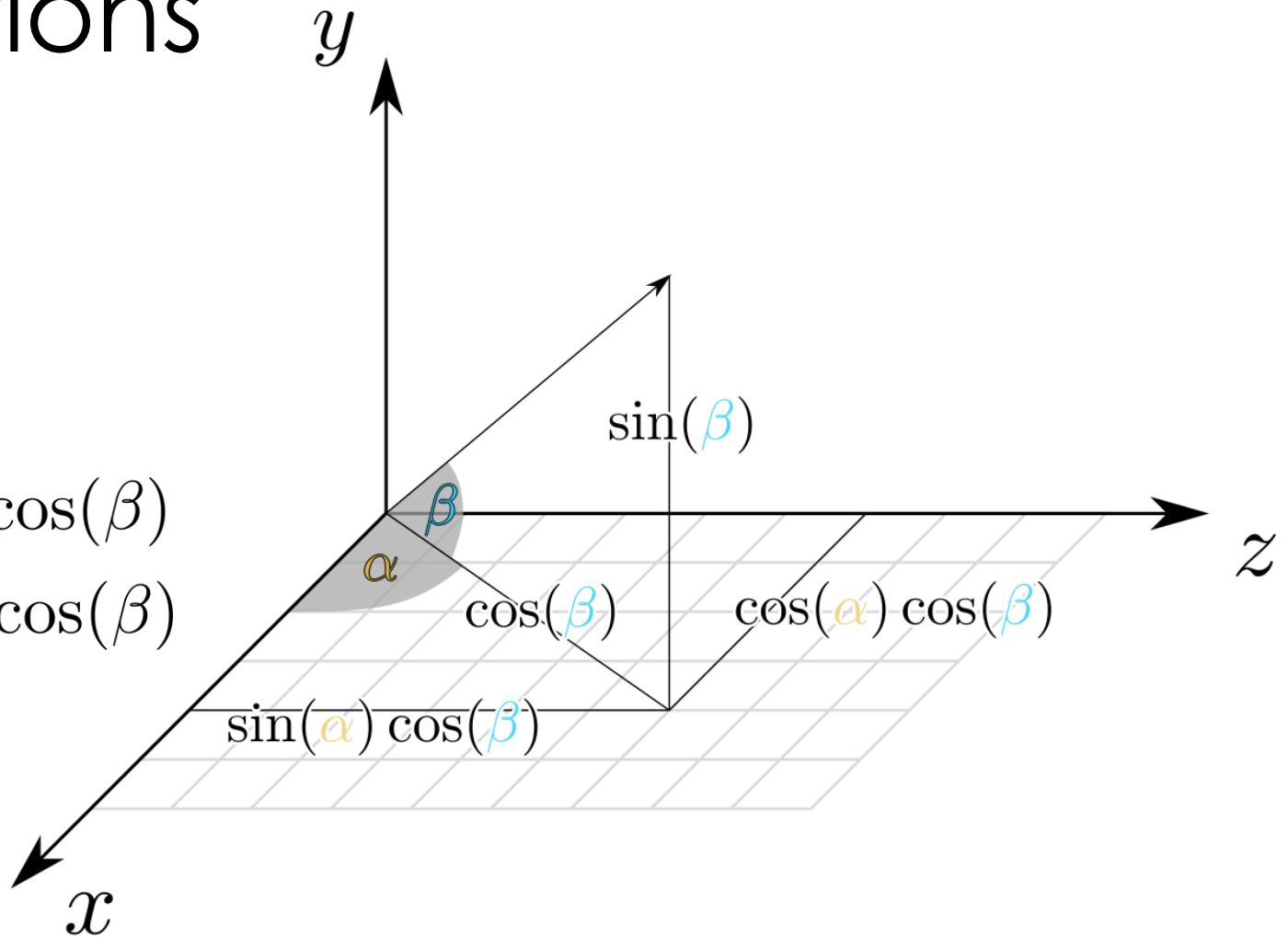
Possible Questions

Complete the equations:

$$\text{camera front}_y = \sin(\beta)$$

$$\text{camera front}_z = \sin(\alpha) \cos(\beta)$$

$$\text{camera front}_x = \cos(\alpha) \cos(\beta)$$



Possible Questions

What parameter can we change in the projection matrix to get a zoom effect?

Possible Questions

What parameter can we change in the projection matrix to get a zoom effect?

- Field of view or fov defines how much can be see of the scene
- Fov smaller → scene's projected space gets smaller (zooming in)

Possible Questions

Complete the code to obtain a ray from a mouse position :

```
double xpos = 0, ypos = 0;
glfwGetCursorPos(window, &xpos, &ypos);

glm::mat4 projection = glm::perspective(glm::radians(camera.Zoom), (float)SCR_WIDTH / (float)SCR_HEIGHT, 0.1f, 100.0f);
glm::mat4 view = camera.GetViewMatrix();

float x = (2.0f * xpos) / (float)SCR_WIDTH - 1.0f;
float y = 1.0f - (2.0f * ypos) / (float)SCR_HEIGHT;
float z = -1.0f;

glm::vec4 p_prime = glm::vec4(x, y, z, 1.0f);

glm::mat4 invProjMat = glm::inverse(projection);
glm::vec4 ViewModelp = invProjMat * p_prime;

ViewModelp.w = 0;
glm::mat4 invViewMat = glm::inverse(view);
glm::vec4 Modelp = ;
glm::vec3 ray = glm::normalize(glm::vec3(Modelp));
```

Possible Questions

Complete the code to obtain a ray from a mouse position :

```
double xpos = 0, ypos = 0;
glfwGetCursorPos(window, &xpos, &ypos);

glm::mat4 projection = glm::perspective(glm::radians(camera.Zoom), (float)SCR_WIDTH / (float)SCR_HEIGHT, 0.1f, 100.0f);
glm::mat4 view = camera.GetViewMatrix();

float x = (2.0f * xpos) / (float)SCR_WIDTH - 1.0f;
float y = 1.0f - (2.0f * ypos) / (float)SCR_HEIGHT;
float z = -1.0f;

glm::vec4 p_prime = glm::vec4(x, y, z, 1.0f);

glm::mat4 invProjMat = glm::inverse(projection);
glm::vec4 ViewModelp = invProjMat * p_prime;

ViewModelp.w = 0;
glm::mat4 invViewMat = glm::inverse(view);
glm::vec4 Modelp = invViewMat * ViewModelp;
glm::vec3 ray = glm::normalize(glm::vec3(Modelp));
```