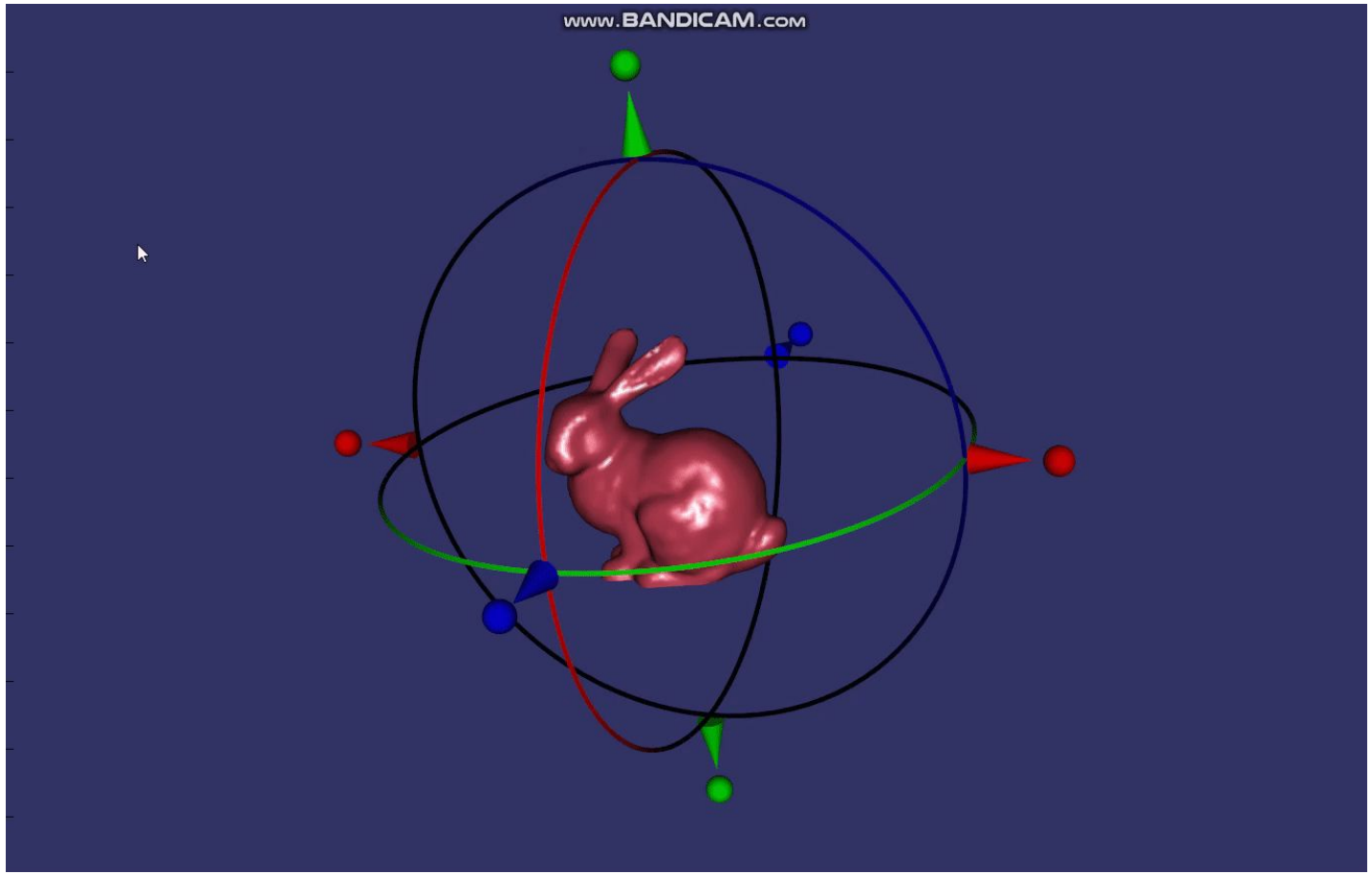
IZG - 3D transformations

Roman Čižmarik

Brno University of Technology, Faculty of information technology
Božetěchova 2, 612 66 Brno
icizmarik@fit.vutbr.cz

Introduction

- We will implement:
  - Translation
  - Scale
  - Rotation around arbitrary axis



- Quite old now
- Implements Scene Graph
- For math consider better options:
  - GLM
  - Eigen

- Column vectors
- Order of multiplication
- Indexing, normalization

```
osg::Matrix model, view, projection;
osg::Matrix MVP = model * view * projection;

osg::Vec3d vector;
osg::Matrix matrix;

osg::Vec3d transformedVector = vector * matrix;

vector .normalize(); //this vector will be normalized
matrix(2,3) = 1.0; //(row, column)
osg::Matrix matrixInv = osg::Matrix::inverse(matrix); //inverse
```

OSG Matrix documentation and class reference

$$T(v) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ v.x & v.y & v.z & 1 \end{bmatrix}$$

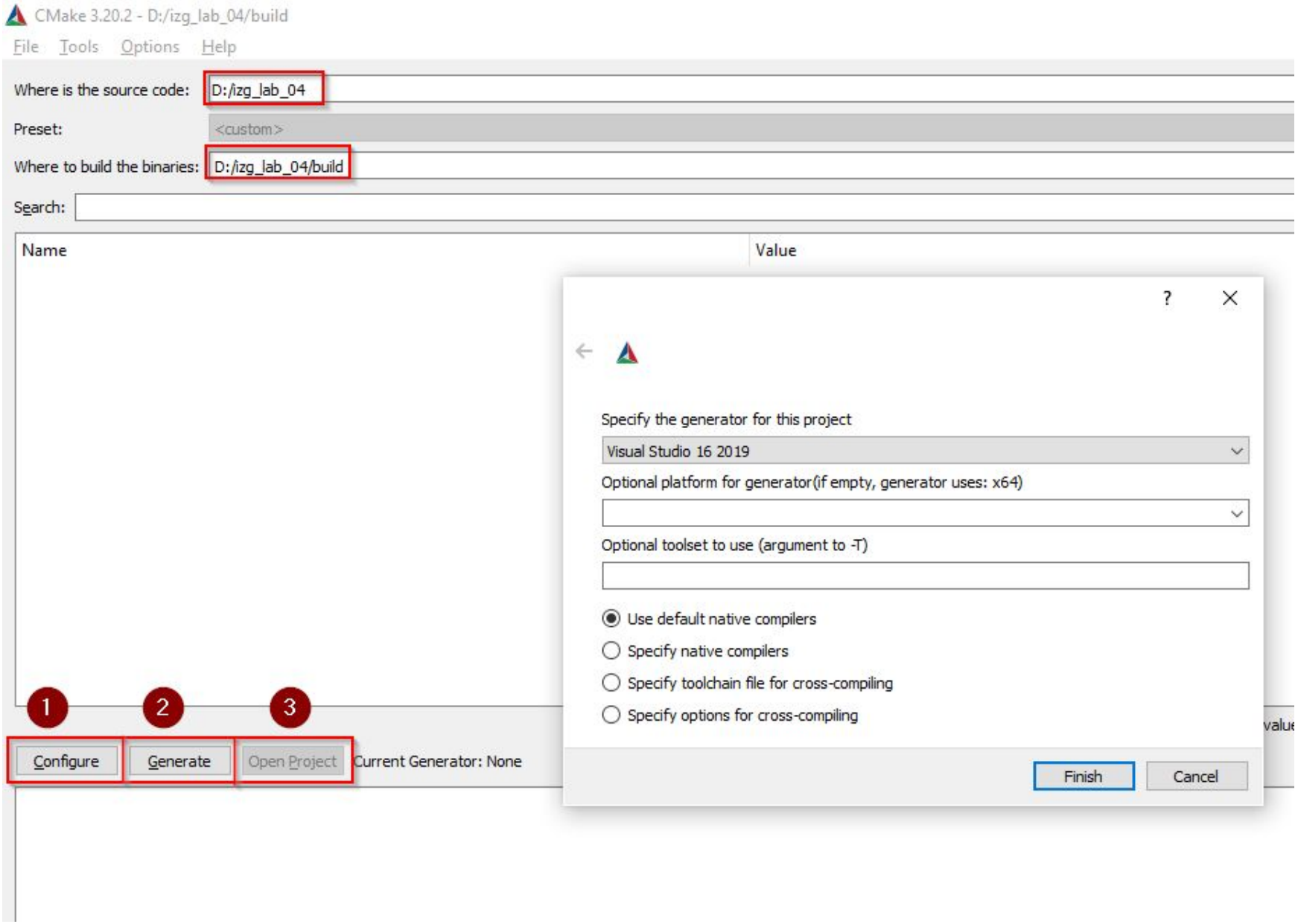
$$S(s) = \begin{bmatrix} s.x & 0 & 0 & 0 \\ 0 & s.y & 0 & 0 \\ 0 & 0 & s.z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\phi & -sin\phi & 0 \\ 0 & sin\phi & cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\phi) = \begin{bmatrix} cos\phi & 0 & sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -sin\phi & 0 & cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z(\phi) = \begin{bmatrix} cos\phi & -sin\phi & 0 & 0 \\ sin\phi & cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Build


CMake 3.20.2 - D:/izg_lab_04/build
File Tools Options Help
Where is the source code: D:/izg_lab_04
Preset: <custom>
Where to build the binaries: D:/izg_lab_04/build
Search:
Name
Value
Specify the generator for this project
Visual Studio 16 2019
Optional platform for generator(if empty, generator uses: x64)
Optional toolset to use (argument to -T)
Use default native compilers
Specify native compilers
Specify toolchain file for cross-compiling
Specify options for cross-compiling
Finish
Cancel
1
2
3
Configure
Generate
Open Project
Current Generator: None

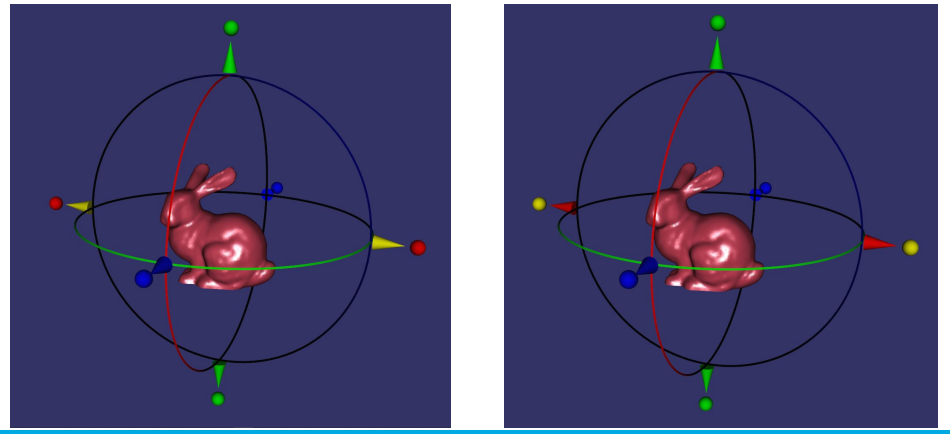
Task - Translate and Scale

- Modify Student.cpp **ONLY** !!
- Build matrices by hand, do **NOT** use built-in methods of osg::Matrix in tasks 1-6.
- Fill in methods:

```
osg::Matrix getScaleMatrix(const osg::Vec3d& scale);                    (1)

osg::Matrix getTranslationMatrix(const osg::Vec3d& translation);        (2)
```

- Return correct matrix representing *scale* and *translation*



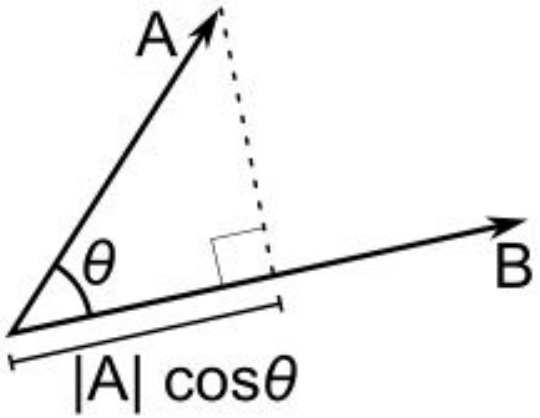
Task - Rotation

- Fill in methods

```
osg::Matrix rotateAroundX(double angle);                          (3)
osg::Matrix rotateAroundY(double angle);                          (4)
osg::Matrix rotateAroundZ(double angle);                          (5)

double angleBetweenVectors(osg::Vec3d u, osg::Vec3d v)            (6)
```

- Computing angle between two vectors
  - Normalize vectors! (method normalize)
  - Dot product (operator *)
  - Arcus cosine (std::acos)

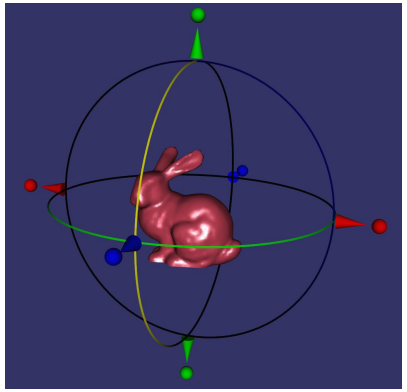


Task - Rotation from one vector to another vector

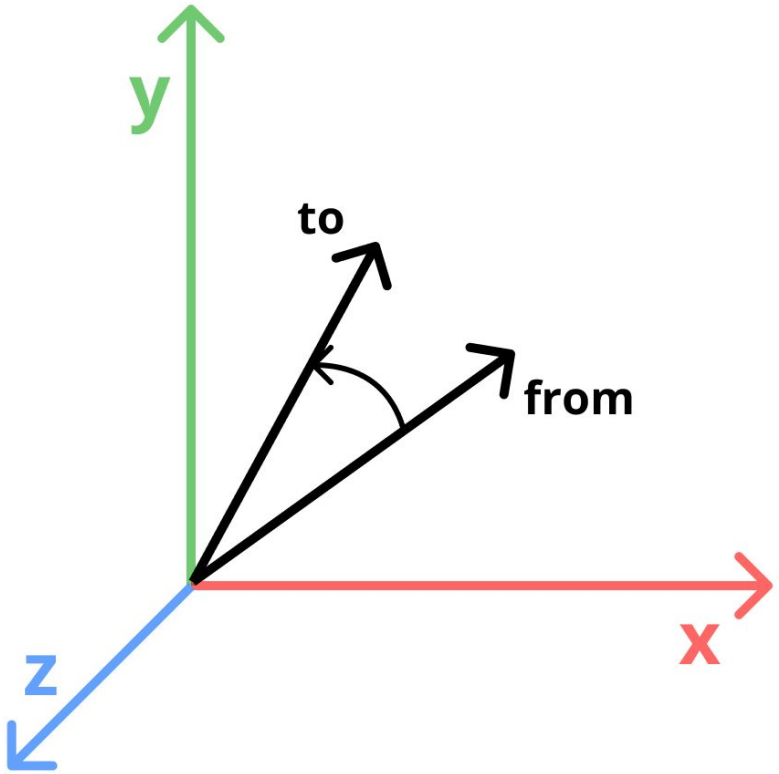
- Rotate around **X axis**
- Fill in method

```
osg::Matrix getRotationMatrix(const osg::Vec3d& fromVector, const osg::Vec3d& toVector);
```
(7)

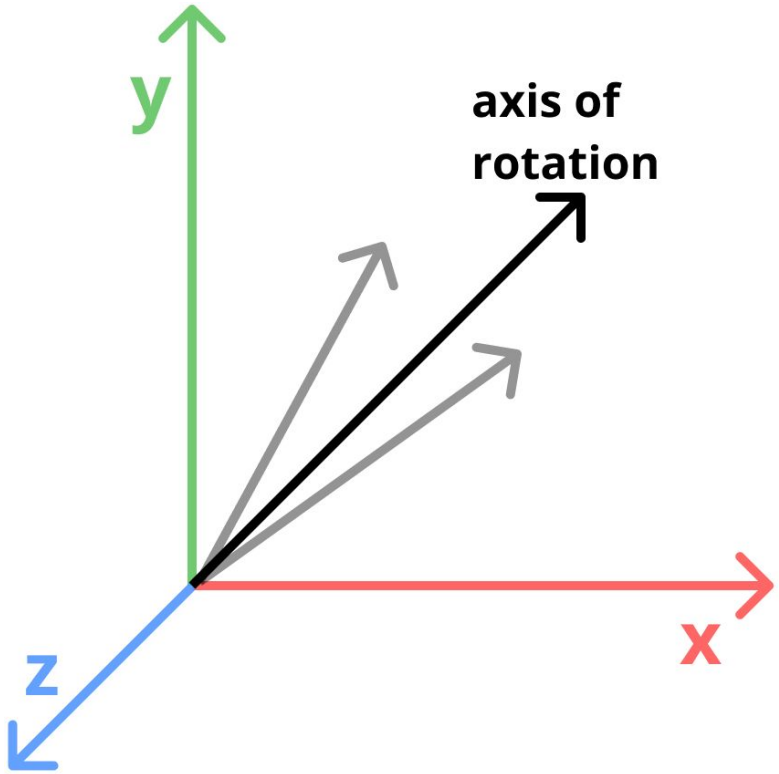
- Compute rotation axis using cross product (function cross) (7a)
- Project rotation axis into XY plane (function projectOnPlane) (7b)
- Compute the angle between projection and X axis (7c)
- Rotate around the Z axis (7d)
- Rotate to Y axis - carefully consider(negative) orientation! (7e)
- Rotate around X axis (7f)
- Compose the final rotation (function inverse) (7g)



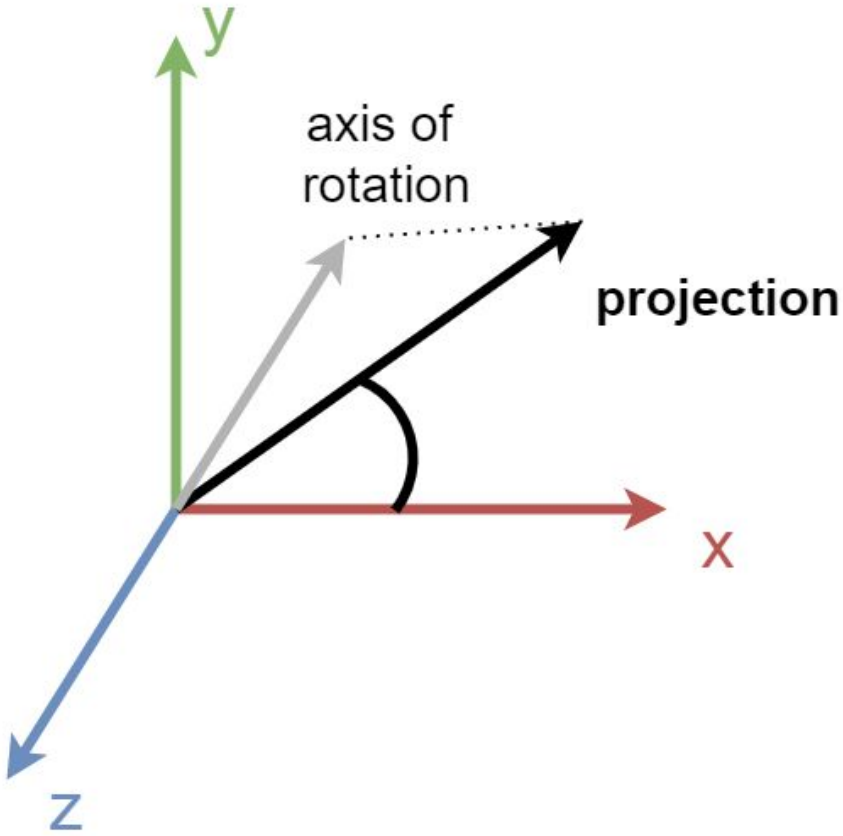
Rotation illustrated



Input

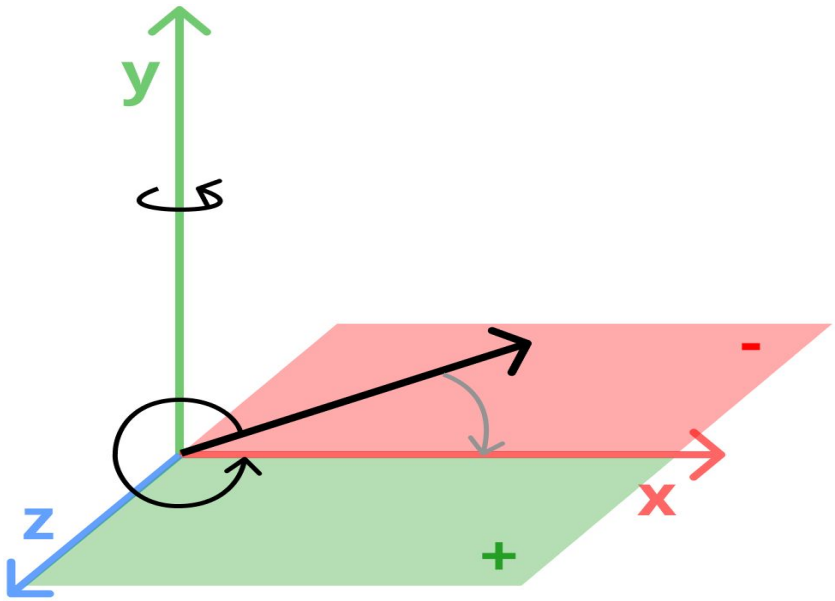


(7a)

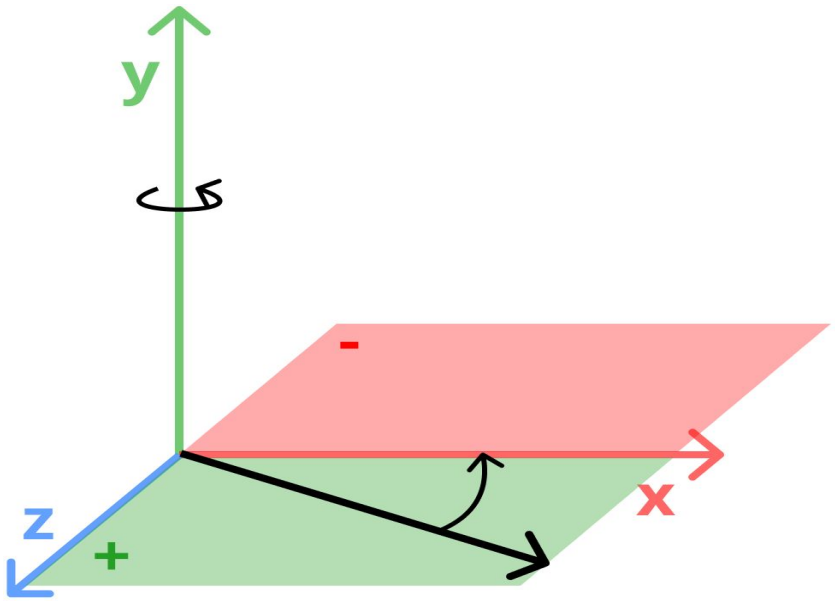


(7b, 7c, 7d)

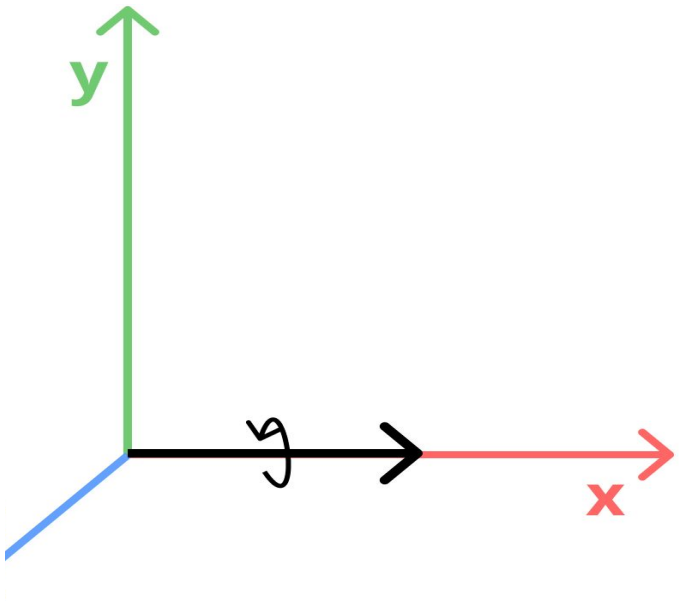
Rotation illustrated



7(e)

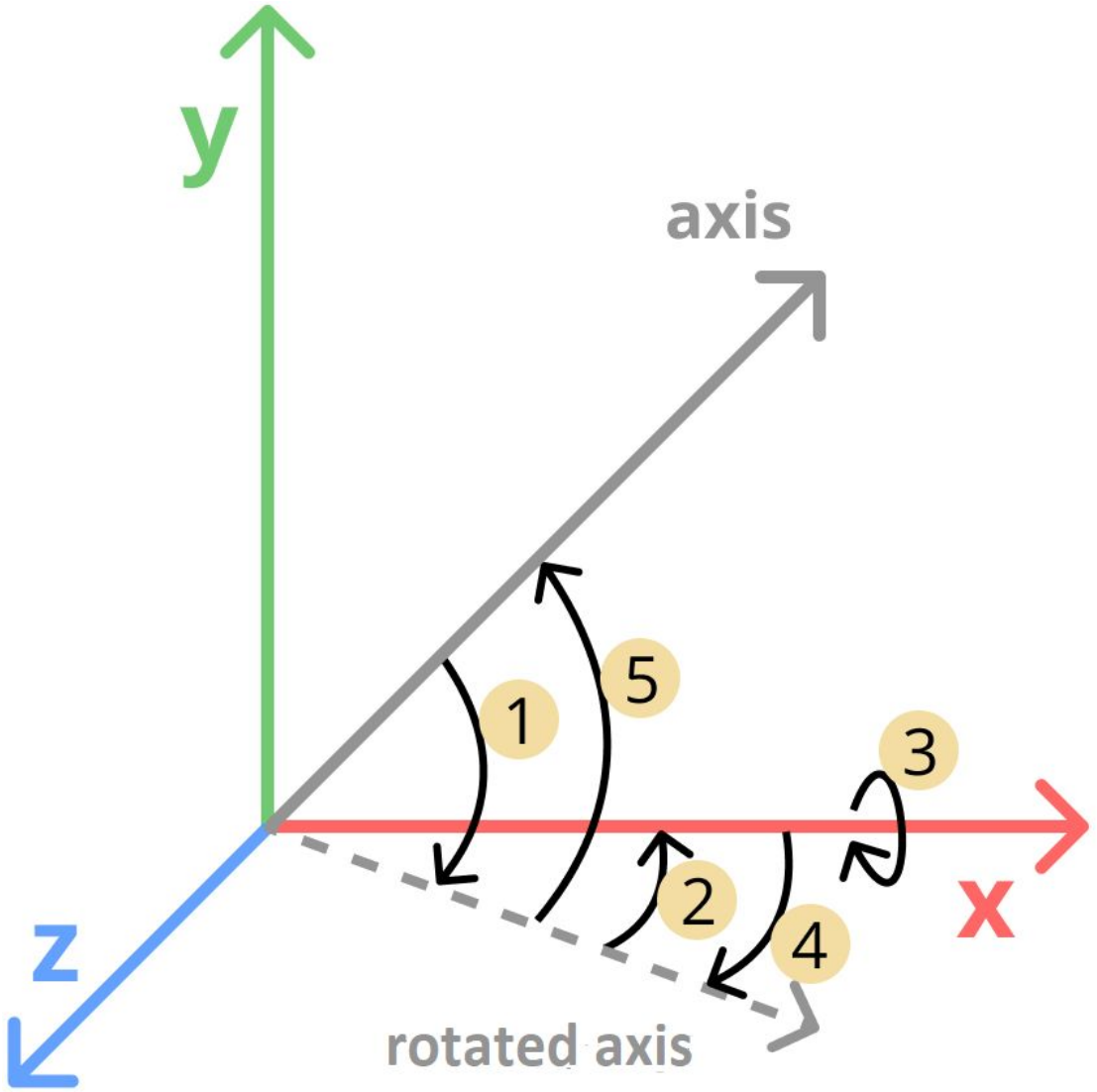


(7e)



(7f)

Rotation illustrated



(7g)

TIP: Input vectors (in task 7) have the Z-coordinate always 0

Thank you for your attention!