U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Computer Graphics
MIEIC

Project - Movement

Basic animation

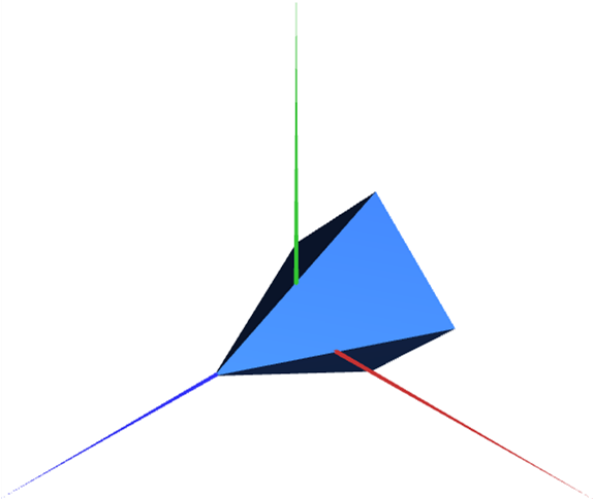
Teresa Matos
30 March 2021

MyMovingObject

Object in scene controllable by user through key presses

Default state of the object is:

- **Stationary** (no speed)
- **Centered** in origin (0,0,0)
- Front towards **positive Z axis**



Steps for moving object animation

The worksheet for the project delineates different steps to apply animation

The steps in MyMovingObject’s class can be broken down into three phases:

1. Define **default** state
2. **Update** current state
3. **Apply** state in object’s **display**

1 Define Default State - Variables

In the constructor, we add the variables that describe the default state.

Default state is:

- **Stationary** (no speed)
- **Centered** in origin (0,0,0)
- Front towards **positive Z axis**

```
constructor(...){
    …
    this.speed = 0;
    this.position = [0,0,0];
    this.orientation = 0;
}
```

2 Update Current State

The 3 variables that define the object’s state are updated/changed differently

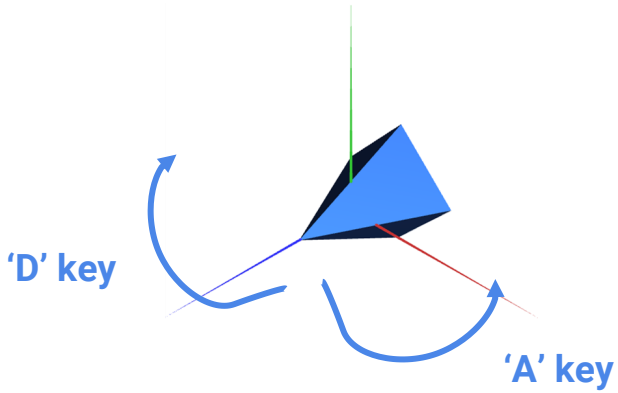
The **speed** and **orientation** are updated only with user input (key presses)

The **position** of the object periodically (according to scene’s update period)

2 Update Current State - Orientation

The object’s orientation is related to its rotation around the Y axis

The orientation changes when the ‘A’ or ‘D’ keys are pressed

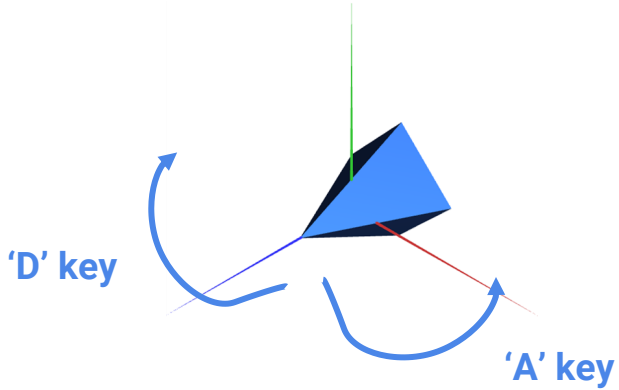


2 Update Current State - Speed

The object’s speed changes when the ‘W’ or ‘S’ keys are pressed

‘W’ increases speed every time it’s clicked, **‘S’ decreases** it until zero

It should move when speed is positive, **not only when ‘W’/‘S’ keys are pressed**

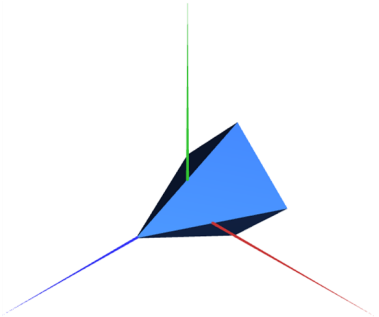


2 Update Current State - Position

The object only moves in X and Z axis (Y is always 0)

At any given moment, we want to move the object:

- from its current position
- In the direction it is facing (according to orientation)



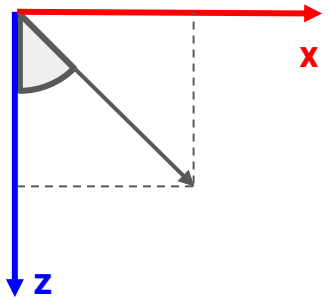
```
update(){

          position = position + directionVector;
}
```

2 Update Current State - Position

We calculate the **directional vector** using the current orientation

```
directionVector.x = sin(orientation)
directionVector.y = 0
directionVector.z = cos(orientation)
```



Directional vector, considering initial orientation = +Z direction

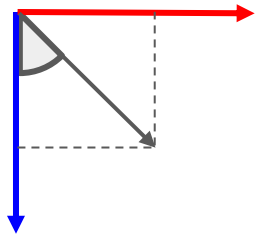
2 Update Current State - Position and Speed

The speed of the object influences the amount of movement (distance)

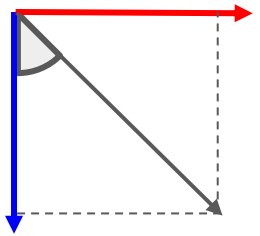
Multiplying a vector by a scalar changes the magnitude of that vector

```
update(){

  position = position + directionVector * speed
}
```



Directional vector (unitary magnitude)



Scaled directional vector (scalar > 1)

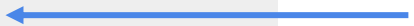
3 Apply state in display

In the object's `display()` function, we apply the geometric transformations to animate the moving object:

```
display(){

  translate(position);

  object.display();
}
```



Is it enough to apply translation?

3 Apply state in display

In the object's `display()` function, we apply the geometric transformations to animate the moving object:

```
display(){

  translate(position);
  rotate(orientation);
  object.display();
}
```

We need to rotate the object **before** positioning it.