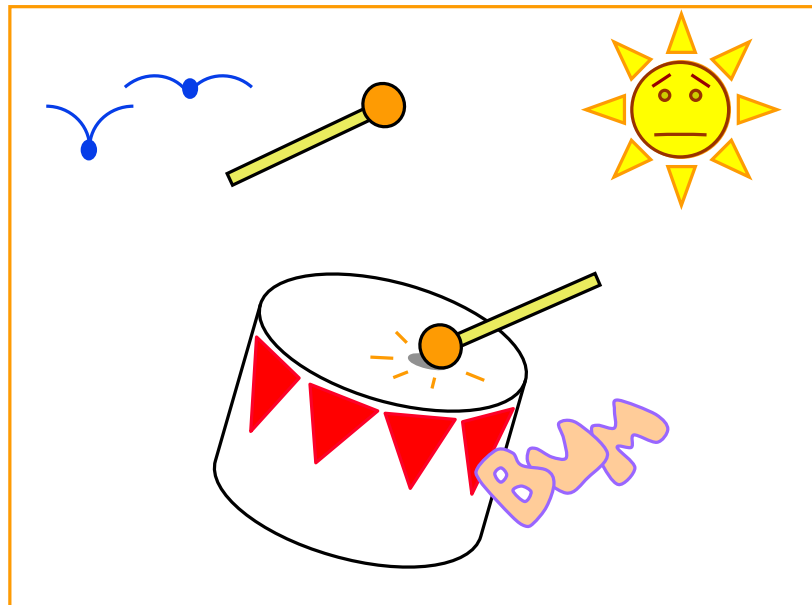


# Princípios de Animação

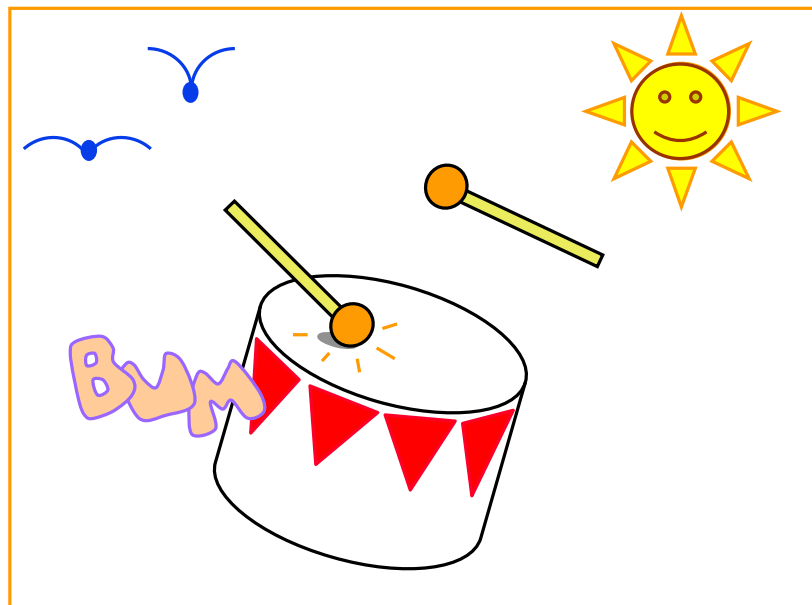
# Animação

## O PRINCÍPIO DA ANIMAÇÃO



+

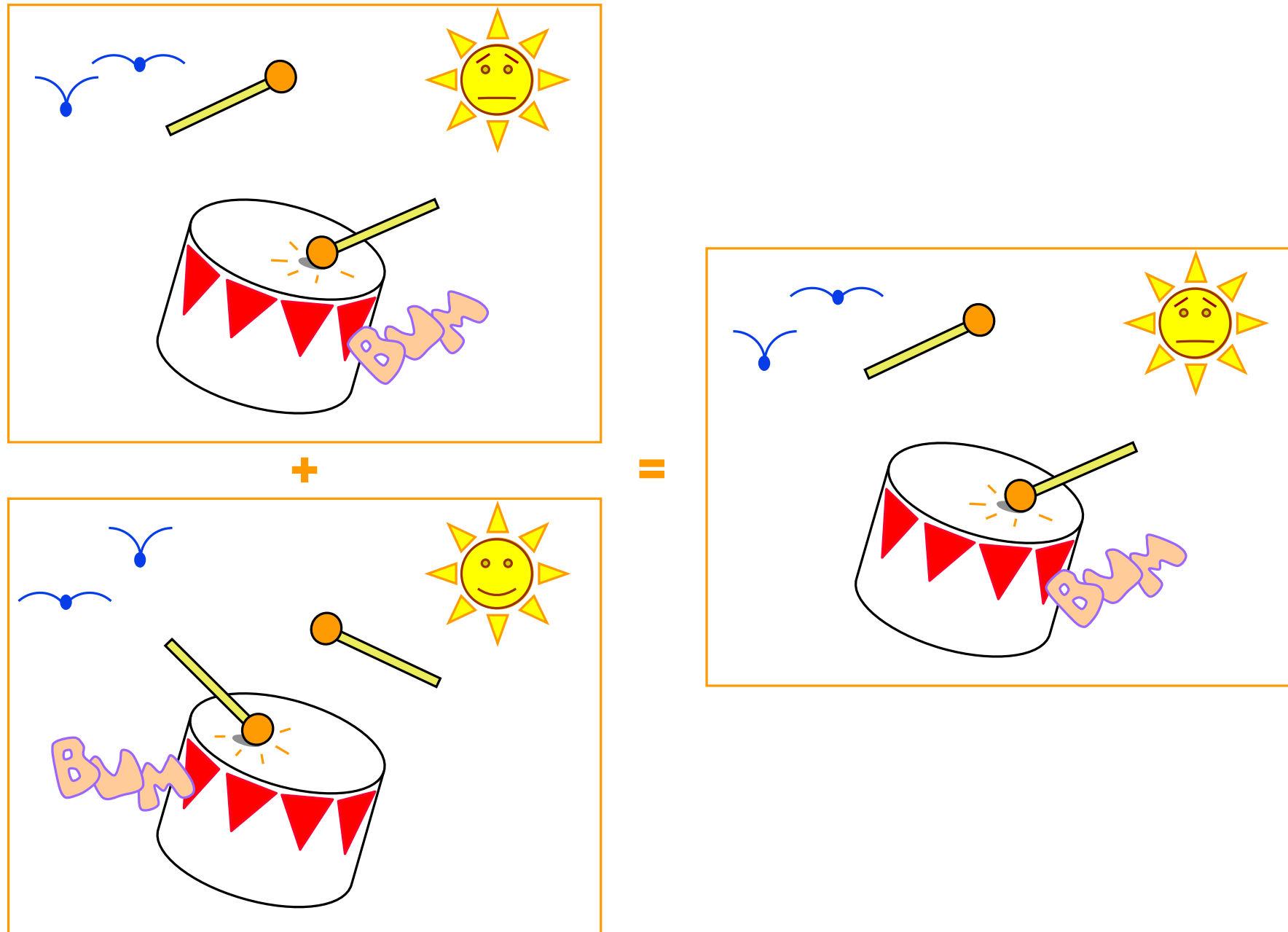
=



M.Próspero

# Animação

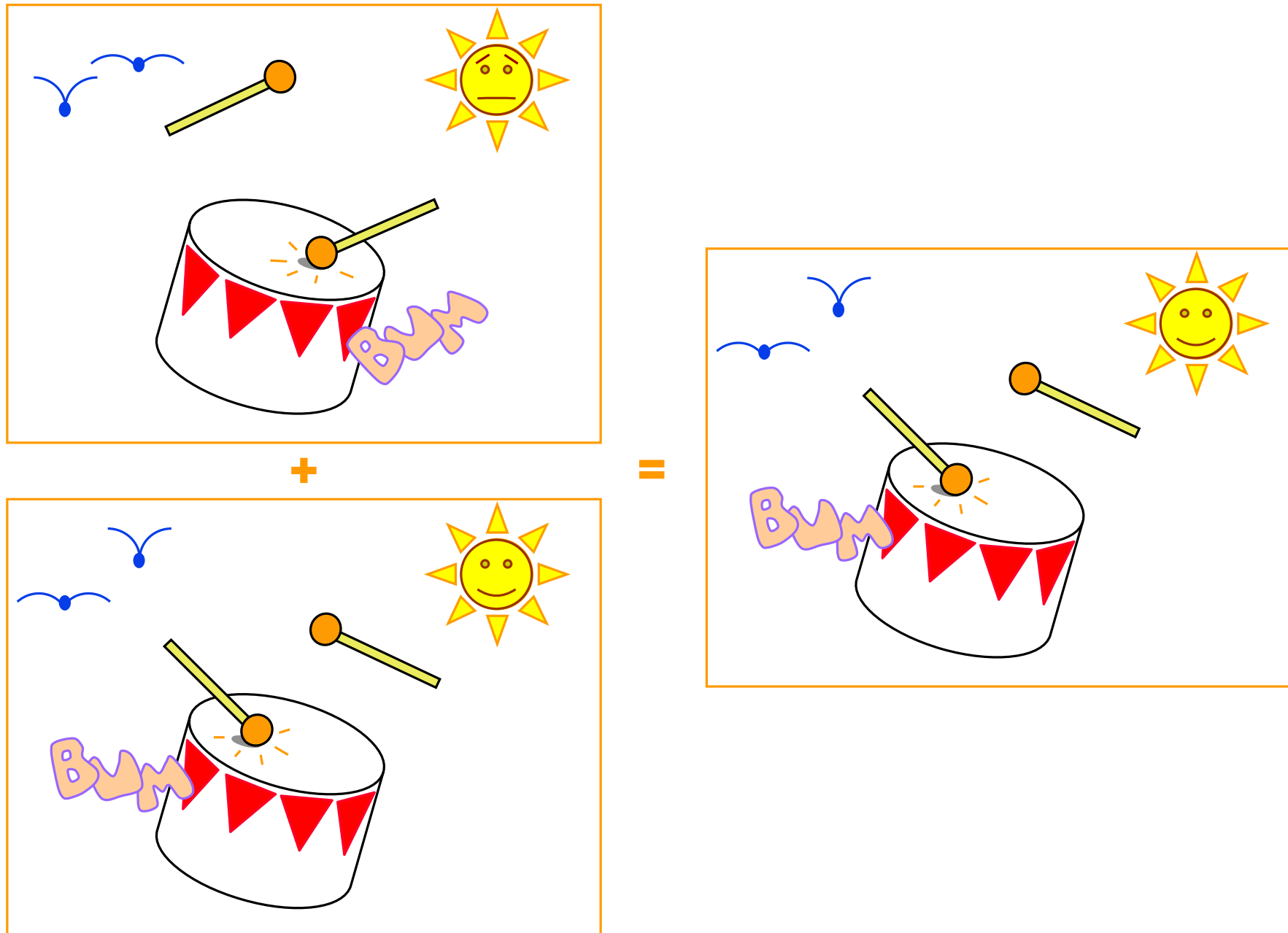
## O PRINCÍPIO DA ANIMAÇÃO



M.Próspero

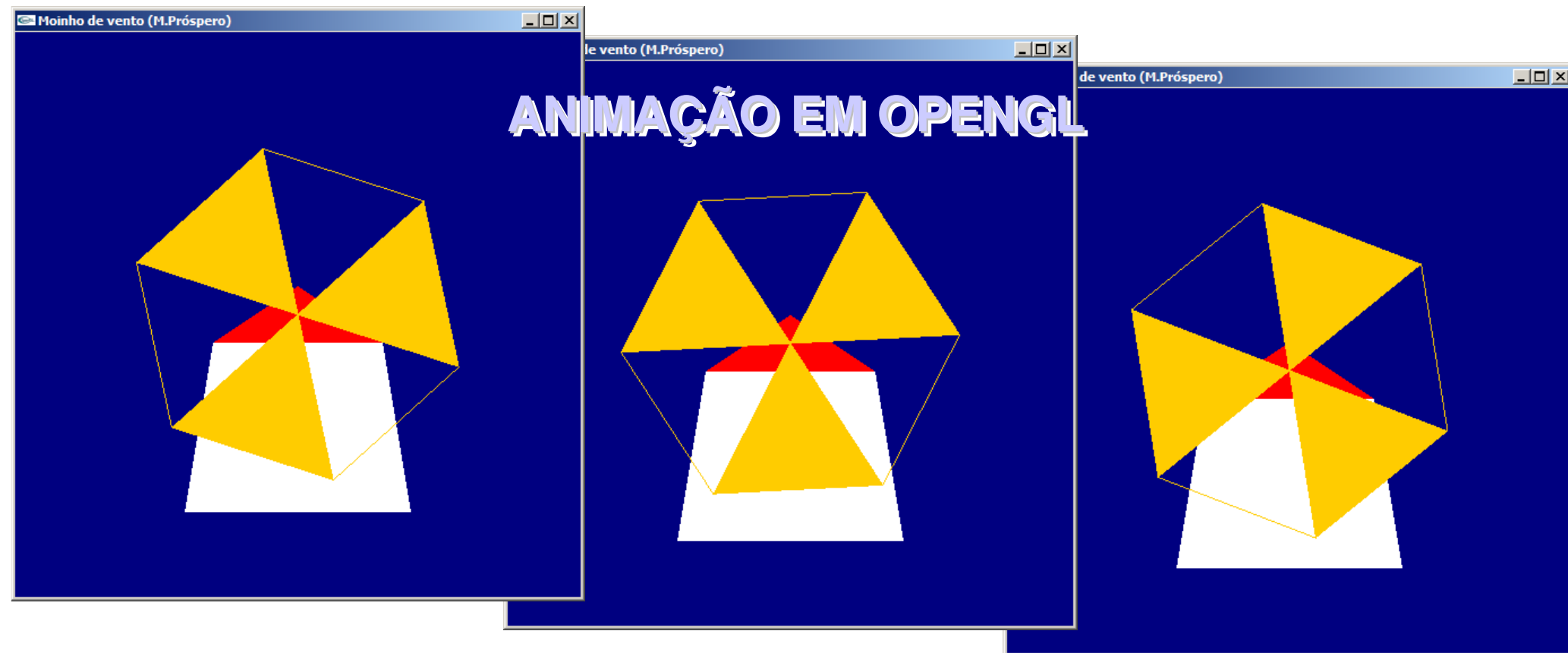
# Animação

## O PRINCÍPIO DA ANIMAÇÃO



M.Próspero

# Animação



Num programa em JOGL, é necessário executar o método `display()` em ciclo, tendo-se o cuidado de ir alterando o valor de cada parâmetro da animação:

```
...  
glRotated(alfa, 0.0, 0.0, 1.0);  
...  
alfa += 1.0; // valor para a próxima chamada de display()  
if (alfa > 360.0) alfa -= 360.0;  
...
```

*M.Próspero*

# Animação

## ANIMATION IN JOGL

`com.sun.opengl.util.Animator`

`com.sun.opengl.util.FPSAnimator`

An Animator ( or FPSAnimator) can be attached to one or more GLAutoDrawables to drive their `display()` methods in a loop.

**Frames Per Second, if supported by the hardware**

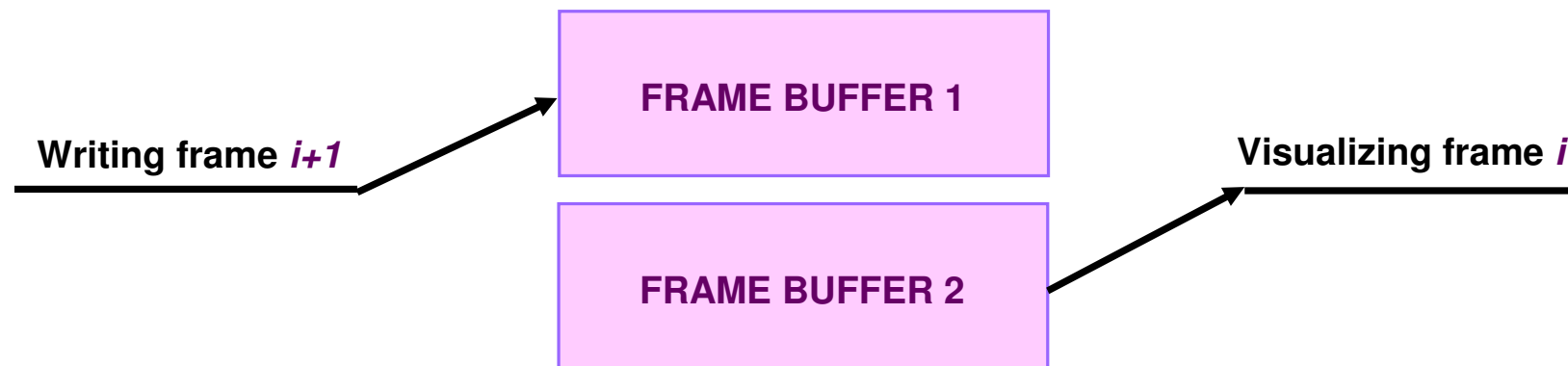
```
...  
    final FPSAnimator animator = new FPSAnimator(canvas, 60);  
    animator.start(); // start executing display() in a loop  
  
...  
    animator.stop(); // stop animation  
  
...
```

*M.Próspero*

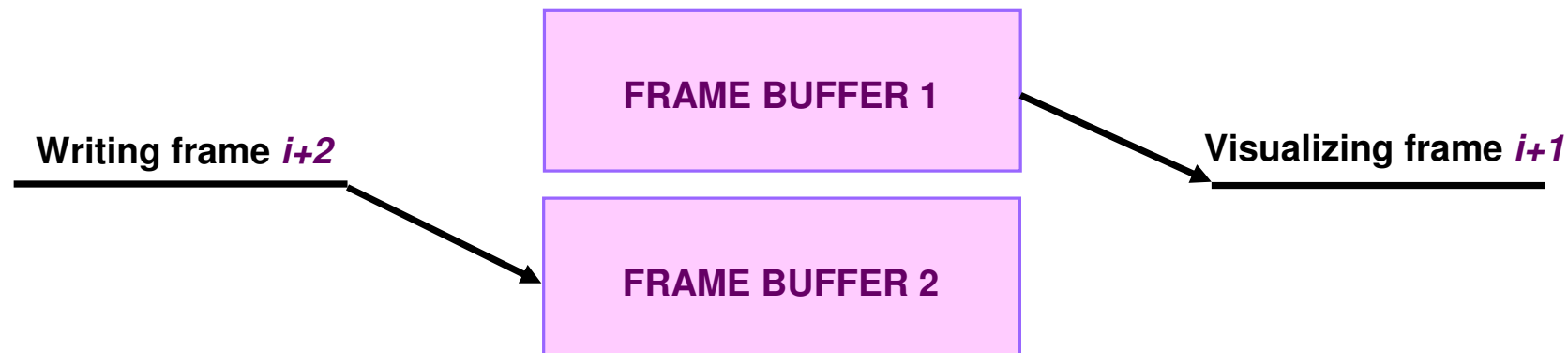
# Animação

## DOUBLE BUFFERING ( Frame Buffer duplo, usado em animação )

Instant  $t$  :



Instant  $t+1$  :



Default in JOGL: Double Buffer "on"

`setSwapInterval(1)` causes the wait for the next vertical refresh until swapping buffers

M.Próspero