

# Animação

# Animações - o dinamismo da forma!!!

- VRML permite um grande número de animações, que podem ser disparadas por aproximações e toques.
- Tais animações fazem as formas se movimentarem baseadas num dado tempo, que, é controlado por um relógio com tempos fracionados em números reais.
- As frações de tempo devem ser associadas a quadros que contêm as posições chaves da animação, dando a noção de movimento da forma.
- As animações podem ser estendidas também à aparência das formas e não somente à posição da forma.

## Animações - aspectos básicos:

- VRML provê diversas maneiras de descrição de animações. As mais comuns usam os nós:
  - PositionInterpolator
  - OrientationInterpolator
- O fundamento é a associação de posições chave com tempos chave - de forma fracional.
- Um nó importante no processo é o nó de controle de tempos fracionais: o nó TimeSensor:
  - TimeSensor{  
    enabled  
    startTime  
    stopTime  
    cycleInterval  
    loop ... }

## Animações - aspectos básicos:

- Sintaxe do nó PositionInterpolator:

- PositionInterpolator{

- key [ ... ] - lista de tempos fracionais chave

- keyValue [ ... ] - lista de posições chave - cada uma composta por X,Y,Z

- set\_fraction - eventIn

- value\_Changed - eventOut

- }

- Sintaxe do nó OrientationInterpolator:

- OrientationInterpolator

- key [ ... ]

- keyValue [ ... ] - lista de rotações chave - cada uma composta por X,Y,Z e um ângulo de rotação.

- set\_fraction

- value\_Changed }

# Exemplo de animação:

```
#VRML V2.0 utf8

Group {
    children [
        DEF Cube Transform {
            children Shape {
                appearance Appearance {
                    material Material {
                        diffuseColor 0.0 1.0 0.0 }
                }
                geometry Box { size 1.0 1.0 1.0 }
            }
        },
        # Relógio da animação - gerando tempos fracionais
        DEF Clock TimeSensor {
            cycleInterval 4.0
            loop TRUE
        },
    ]
}
```

```

# Caminho da animação do cubo - posições chave e valores de
# cada posição
        DEF CubePath PositionInterpolator {
            key [
                0.00, 0.11, 0.17, 0.22,
                0.33, 0.44, 0.50, 0.55,
                0.66, 0.77, 0.83, 0.88,
                0.99
            ]
            keyValue [
                0.0 0.0 0.0, 1.0 1.96 1.0,
                1.5 2.21 1.5, 2.0 1.96 2.0,
                3.0 0.0 3.0, 2.0 1.96 3.0,
                1.5 2.21 3.0, 1.0 1.96 3.0,
                0.0 0.0 3.0, 0.0 1.96 2.0,
                0.0 2.21 1.5, 0.0 1.96 1.0,
                0.0 0.0 0.0
            ]
        }
    }
# Observe a equivalência de quantidade de posições e tempos
# chave
}
]
}

ROUTE Clock.fraction_changed TO CubePath.set_fraction
ROUTE CubePath.value_changed TO Cube.set_translation

```

## Animações: sentindo ações do usuário

- Em VRML há a possibilidade de sentir o toque, a aproximação e os movimentos do usuário, através do mouse, claro.
- O conjunto de nós utilizados para este fim é constituído por:
  - Nó TouchSensor
  - Nó CylinderSensor
  - Nó SphereSensor
  - Nó PlaneSensor
- Tais nós podem ser incluídos em qualquer grupo e com rotas, podem disparar animações

- As ações do usuário podem promover movimentos diversos:
- **TouchSensor**: sentir ações do tipo “click” e “drag”
- **PlaneSensor**: sensível às ações do tipo “drag”, serve para computar distâncias de translação, gerando mudanças de posição das formas - usado para arrastar as formas através do cenário
- **SphereSensor**: sensível às ações do tipo “drag” - computa eixos de rotação e ângulos, com saídas que podem gerar rotações de formas
- **CylinderSensor**: também sensível às ações do tipo “drag”, capaz de produzir também rotações das formas, como, por exemplo, abrir uma porta, já que possibilita definir um máximo e um mínimo

# Exemplo de animação

- Este exemplo mostra um nome em animação, disparado por um sensor de toque - a delimitação de posições - inicial e final, se dá por meio do nó OrientationInterpolator, associado ao disparo feito pelo nó TouchSensor
- O nome pode ser arrastado, usando o nó PlaneSensor, que com o movimento do mouse, leva o mesmo para a posição que desejarmos na tela.
- Aqui, um texto é animado, mas, poderíamos fazer isto com qualquer forma do meu cenário.
- Observe que as rotas de mudanças devem ser expressamente declaradas.

```
#VRML V2.0 utf8
# Aqui um nome gira com o toque do mouse... basta pressionar o botão esq.
# do mouse para o movimento acontecer... Reparar na transparência do chão
# nas cores das letras e no toque.

Group {
    children [
        # Rotating Nome
        DEF Nome Transform {
            #translation -5.0 0.0 0.0
            children [
                Shape {
                    appearance DEF White Appearance {
                        material Material { }
                    }
                    geometry Text {
                        string ["Alexandre", "Cardoso"]
                        fontStyle DEF Fonte FontStyle{
                            size 0.9
                            style "BOLD"
                            family "SANS"
                            justify "MIDDLE" } } },
                    Shape { appearance Appearance { material Material {
                        diffuseColor 0.0 1.0 1.0
                        transparency 0.5 } }
                        geometry Box {
                            size 3.7 0.05 2.0 } } ]
                },
            ]
```

```
# Sensor
    DEF Touch TouchSensor { },
    DEF Arrasta PlaneSensor { },
# Animation clock
    DEF Clock TimeSensor { cycleInterval 5.0
                            loop FALSE},
# Animation path
    DEF CubePath OrientationInterpolator {
        key [ 0.0, 0.50, 1.0 ]
        keyValue [
            0.0 1.0 0.0  0.0,
            0.0 1.0 0.0 -3.14,
            0.0 1.0 0.0 -6.28
        ]
    }
}
]

ROUTE Touch.touchTime          TO Clock.set_startTime
ROUTE Clock.fraction_changed TO CubePath.set_fraction
ROUTE CubePath.value_changed  TO Nome.set_rotation
ROUTE Arrasta.translation_changed TO Nome.set_translation
```