

PARSER

MYSpriteSheet CLASS

base class that stores spritesheet texture and controls shader

constructor

texture CGFTexture
sizeM number of columns (int)
sizeN number of lines (int)
(scene) always necessary

The constructor function must:

1. Save data passed as arguments
2. Initialize shader (CGFTexture)

MySpriteText CLASS

class that uses base spritesheet class to represent string of text

constructor

(scene) always necessary
text string

MySpriteAnim CLASS

class that uses base spritesheet class to represent animated textures

constructor

(scene) always necessary
spritesheet id or reference to MySpriteSheet
duration seconds (float)
startCell index of first sprite (int)
endCell index of last sprite (int)

The constructor functions must:

1. Save data passed as arguments
2. Initialize base geometry (e.g. MyRectangle) and:

- MySpriteSheet MySpriteText
- animation-related variables MySpriteAnim

UPDATE

In scene: (reminder)

- SetUpdatePeriod() when graph finished loading
- update(): function that triggers update of all animations

MySpriteAnim CLASS

update(currentTime)

1. Calculate elapsedTime
2. Calculate which sprite cell is active → using
 · elapsedTime
 · duration
 · n° of cells
3. Save current state and other variables
 ↑ index of current sprite

APPLY

MySpriteText CLASS

display()

loop through text characters

1. Get character's sprite position
2. Activate sprite
3. Display base geometry

auxiliary function

getCharacterPosition(char)
Return index of character sprite in the font spritesheet

MySpriteAnim CLASS

display()

1. Activate sprite, using current index
2. Display base geometry

MySpriteSheet CLASS

auxiliary functions

activateCellP/activateCellMN

1. Pass cell parameters to shader
 shader.setUniformsValues()
2. Activate shader
 scene.setActiveShader()

Note: Don't forget to activate default shader when done!