HOW THREE.JS WORKS

- An object is defined by a Geometry and a Material. Both class are available in lots of specialisations.
    - Geometry is a set of vertices, disposed to represent a certain object.
    - Material defines object's properties (brightness, shadowing, texture, etc.).
- Moreover other libraries are available to help managing camera or movements (Orbit Controls, Tween, etc…).

All seems easy and amazing, but:

- Geometry and Material classes carry on a useless baggage of informations.
- Fortunately it's possible to define our own geometry and material using custom vertices sets and shaders.
- This strongly increase performances in a simple context like this (if all done correctly).

PROCEDURE

Shaders

- Vertex Shaders
  - Attributes
    - particle starting position
    - particle size
    - particle trajectory angle
    - time offset (for continuous generation)
  - Uniforms
    - time t
    - time life (before regeneration)
    - speed
    - opacity
- Fragment Shaders
  - Color
  - Texture

Three.js

- initialize scene
  - camera
  - light
  - box containing the scene
  - table
  - geometry
  - texturing
- load candle obj created with blender
- init of smoke and flame Geometry
  - setting first position e angle
  - linking attributes with shaders
  - setting and linking uniform variables
- Audio