# Concept Design / Storyboard

X and Y axes are horizontal. Z axis is vertical

**Z**

**Y**

**X**

N.B. The coloured triangles represent the lights. The black arrows coming off them are where they are pointing toward. The cube is the box that will contain all the elements to be displayed.

* Faces of the cube with their outer side pointing towards the viewer should be made semi-transparent (allowing the viewing of objects within).
* Box will rotate around the z-axis. Controlled from a pre-defined script, or using the left/right arrows.
* 8 spotlights situated at the corners, initially pointing towards the centre of the box.

**Frame 1 Frame 2**

N.B. The above picture shows the spotlight tracking an object.

* The colour, intensity and field of view of each of the eight spotlights should be definable in the script file
* Animated lights, being able to point at a particular object and follow it around the room
* Optionally, the light colour and intensity should change over time
* Physically, the spotlights should be represented graphically by a small cone, whose orientation changes as the spotlight follows an object

**Frame 1 Frame 10**

Particle ball

< null object >

Platonic Solid

glowball

Platonic Solid

Particle ball

glowball

< null object >

* One or more objects should be animated in the scene
* Objects must include: sphere, platonic solids (tetrahedron, cube, octahedron, dodecahedron, and icosahedrons), glowball, particleball, null object
* Size and material characteristics of each object should be included in the script file, along with a description of the animation path.
* Platonic objects should be loaded from an object file. This file should contain sufficient information to describe the geometry of the object, the related normals, and texture coordinates. In the materials description for an object should be included for texturing and therefore one or more filenames should be included for the textures used.

Glowball



Object no longer illuminated



Glowball

Illuminated object

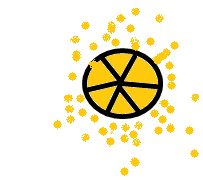
**Frame 1 Frame 10**

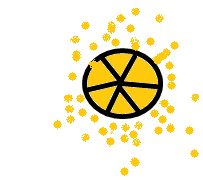
* The glowball is a special object that has 6 regularly space colour spotlights evenly distributed over a spherical surface
* When this object is in a scene, the 8 spotlights at the corner of the box should be automatically turned off
* The glowball should be able to follow an animation path, illuminating all other objects in the box

**Frame 1 Frame 10**

scorchmark.png

Scorch Mark





Emitting particles

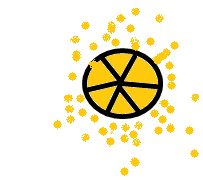
**Frame 20**

* The particle ball is similar to the glowball except each of the 6 lights is replaced with a particle emitter
* Supports at least two types of particle, points and textured quads
* Particles will be emitted from the surface of the sphere and either fade out, or impact onto the sides of the box
* On impact they should leave a scorch mark. This should also fade out over time
* Parameters of the particle should at least contain type, material/texture, rate of generation, lifetime, and scorch texture

scorchmark.png

New scorch mark

Scorch Mark Faded out



Frame 1

< null object >

< null object >

* The null object has no geometry and is therefore invisible. It is included as an invisible object for light sources to point at rather than a real object

**Frame 1 Frame 20**

**Frame 40**

* Animations should be definable for each object and included in the script file
* Your code should at least support animation paths based on the trigonometric ‘sin’ function
* Should include a keypress that starts and stops the animation
* Allow objects to tumble

Tumbling object as it moves through its animation path

Frame 1 Frame 20

Multiple animated objects

* Support multiple animated objects in a single scene
* Collision detection and collision response between objects is **not** required

**Wireframe Flat shaded**

**Smooth shaded gouraud Smooth shaded phong**

**Smooth shaded textured (gouraud)**

* Visualisation of the scene will depend on the current mode selected. Options must include:
  + Wireframe
  + Flat shaded
  + Smooth Shaded (Gouraud and Phong)
  + Smooth Shaded Textured
* Smooth shading using the Phong model should be programmed using GLSL and need only apply to spheres, cubes, and platonic solids. No texturing effects are required from your code

Object casting a shadow on the wall based on the light source direction

* Special Effects
  + Shadows. A simple implementation will allow shadows from objects to appear on walls of the cube. However a more complex implementation should allow shadows to appear on objects