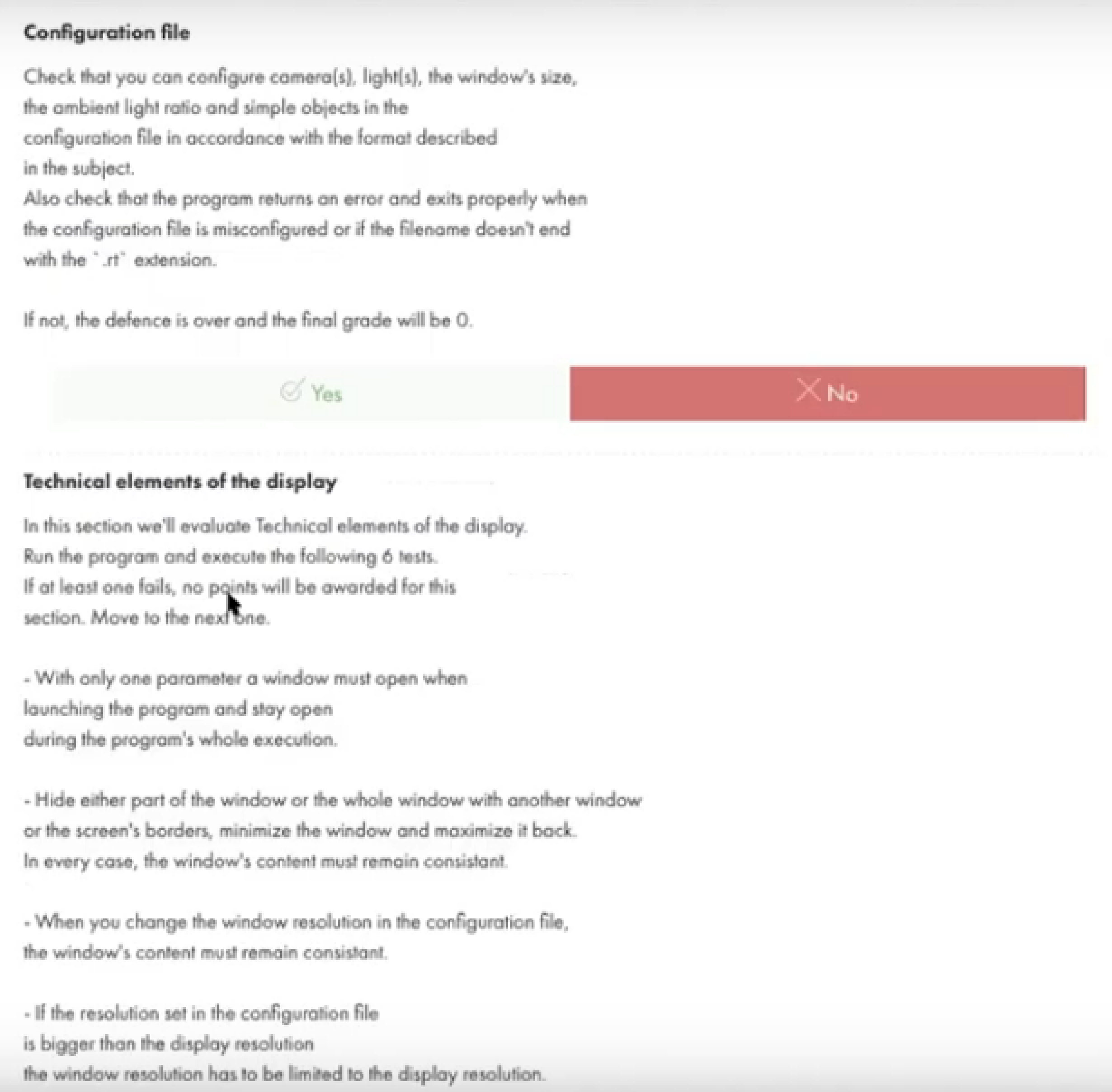


Configuration file

Check that you can configure camera(s), light(s), the window's size,
the ambient light ratio and simple objects in the
configuration file in accordance with the format described
in the subject.
Also check that the program returns an error and exits properly when
the configuration file is misconfigured or if the filename doesn't end
with the `.rt` extension.

If not, the defence is over and the final grade will be 0.

Technical elements of the display

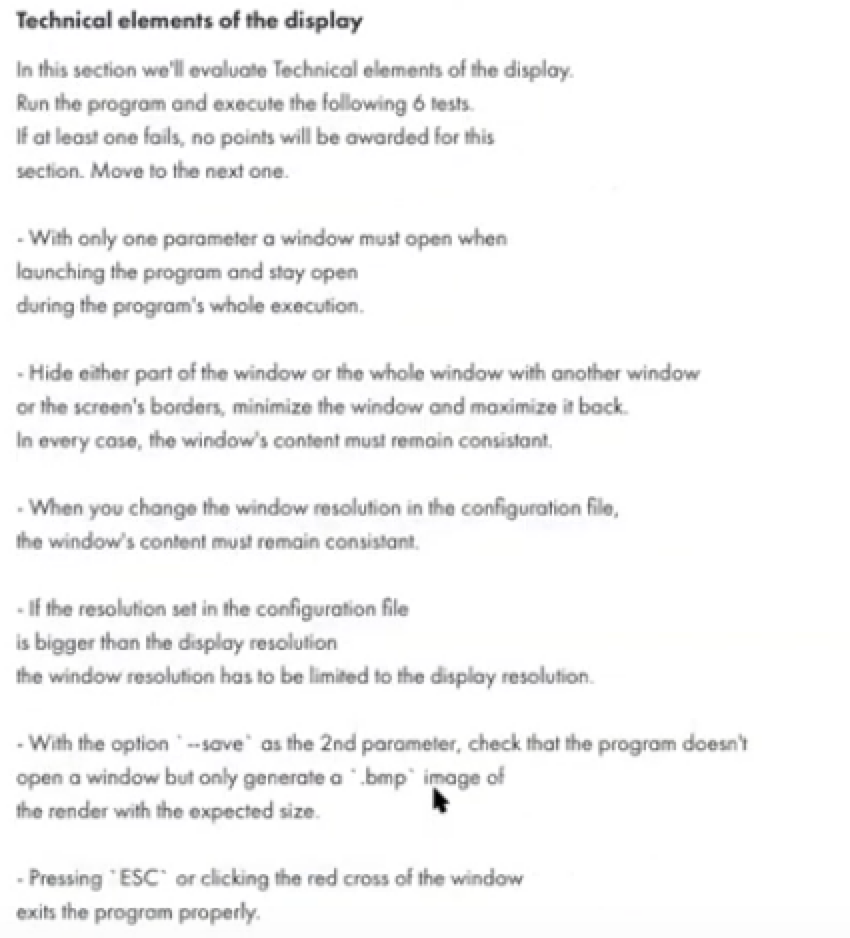
In this section we'll evaluate Technical elements of the display.
Run the program and execute the following 6 tests.
If at least one fails, no points will be awarded for this
section. Move to the next one.

- With only one parameter a window must open when
launching the program and stay open
during the program's whole execution.

- Hide either part of the window or the whole window with another window
or the screen's borders, minimize the window and maximize it back.
In every case, the window's content must remain consistant.

- When you change the window resolution in the configuration file,
the window's content must remain consistant.

- If the resolution set in the configuration file
is bigger than the display resolution
the window resolution has to be limited to the display resolution.



Technical elements of the display

In this section we'll evaluate Technical elements of the display.
Run the program and execute the following 6 tests.
If at least one fails, no points will be awarded for this
section. Move to the next one.

- With only one parameter a window must open when
launching the program and stay open
during the program's whole execution.

- Hide either part of the window or the whole window with another window
or the screen's borders, minimize the window and maximize it back.
In every case, the window's content must remain consistant.

- When you change the window resolution in the configuration file,
the window's content must remain consistant.

- If the resolution set in the configuration file
is bigger than the display resolution
the window resolution has to be limited to the display resolution.

- With the option `--save` as the 2nd parameter, check that the program doesn't
open a window but only generate a `.bmp` image of
the render with the expected size.

- Pressing `ESC` or clicking the red cross of the window
exits the program properly.

The 5 Basic Shapes

In this section we'll evaluate the 5 basic shapes. Run the program and execute the following 5 tests. If at least one fails, no points will be awarded for this section. Move to the next one.

- Place a sphere at the coordinates (0, 0, 0). With the camera facing the sphere, display the rendered image. The sphere should be visible and displayed without glitching.

- Place a plane with a 'z' value of null. With the camera facing the plane, display the rendered image. The plane should be visible and displayed without glitching.

- Place a square's center at the coordinates (0, 0, 0) with a side size of 10 and a blue (0,0,255) color extending along the y axis. With the camera facing the square, display the rendered image. The square should be visible and displayed without glitching.

- Place a cylinder extending along the y axis. With the camera facing the cylinder, display the rendered image. The cylinder should be visible and displayed without glitching.

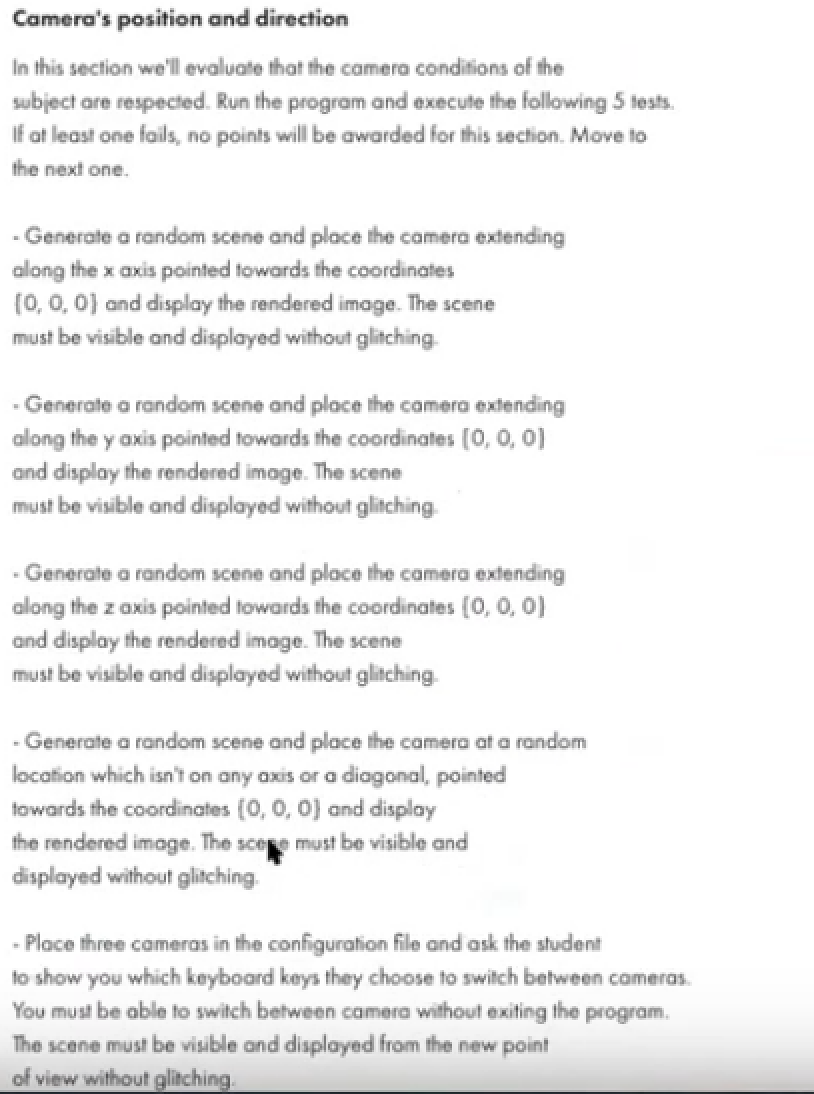
- Place a triangle with the following coordinates
point1 (0,20,0)
point2 (0,0,0)
point3 (0,10,20)
with a color set to (255, 255, 0). Display the image rendered. The triangle should be visible and displayed without glitching.

Translations and rotations

In this section we'll evaluate that rotation and translation transformations can be applied on the scene's objects. Run the program and execute the following 2 tests. If at least one fails, no points will be awarded for this section. Move to the next one.

- Place two spheres at the coordinates (0, 0, 0), the camera facing those spheres. Then put a translation on one of the two spheres oriented in a direction parallel to the camera's, of a greater distance than the sphere's diameter and display the rendered image. Both spheres should be visible and displayed without glitching.

- Place a cylinder extending along the y axis, the camera facing the cylinder. Then put a 90° rotation (PI/2 radian) along the z axis and display the rendered image. The cylinder should be visible and displayed without glitching.



Camera's position and direction

In this section we'll evaluate that the camera conditions of the subject are respected. Run the program and execute the following 5 tests. If at least one fails, no points will be awarded for this section. Move to the next one.

- Generate a random scene and place the camera extending along the x axis pointed towards the coordinates (0, 0, 0) and display the rendered image. The scene must be visible and displayed without glitching.

- Generate a random scene and place the camera extending along the y axis pointed towards the coordinates (0, 0, 0) and display the rendered image. The scene must be visible and displayed without glitching.

- Generate a random scene and place the camera extending along the z axis pointed towards the coordinates (0, 0, 0) and display the rendered image. The scene must be visible and displayed without glitching.

- Generate a random scene and place the camera at a random location which isn't on any axis or a diagonal, pointed towards the coordinates (0, 0, 0) and display the rendered image. The scene must be visible and displayed without glitching.

- Place three cameras in the configuration file and ask the student to show you which keyboard keys they choose to switch between cameras. You must be able to switch between camera without exiting the program. The scene must be visible and displayed from the new point of view without glitching.

Multi-objects

In this section we'll evaluate that it's possible to put several object in one scene. Run the program and execute the following 2 tests. If at least one fails, no points will be awarded for this section. Move to the next one.

- Place several intersecting objects on the scene, such as for example a sphere and a cylinder, and display the rendered image. Both objects should be visible and displayed without glitching. (especially where both object intersect)

- Execute the same test, but ensure it's possible to place the same object several times, for example two cylinders, two spheres and a plane.

Brightness 1/2

In this section we'll evaluate brightness on the scene's objects. Run the program and execute the following 2 tests. If at least one fails, no points will be awarded for this section. Move to the next one.

- Place a sphere at the coordinates (0, 0, 0), the camera facing the sphere, and put a spot left or right of the camera but positioned in such a way that the sphere will be lit sideways. Display the rendered image. The sphere should be visible, illuminated and displayed without glitching.

- Place a sphere at some coordinates resulting from a translation, the camera facing the sphere, and place a spot left or right of the camera but positioned in such a way that the sphere will be lit sideways. Display the rendered image. The sphere should be visible, properly illuminated and displayed without glitch. Properly means that the halo of light should be computed after translation not before.

Brightness 2/2

In this section we'll evaluate shadow management generated by the scene's objects. Run the program and execute the following 2 tests. If at least one fails, no points will be awarded for this section. Move to the next one.

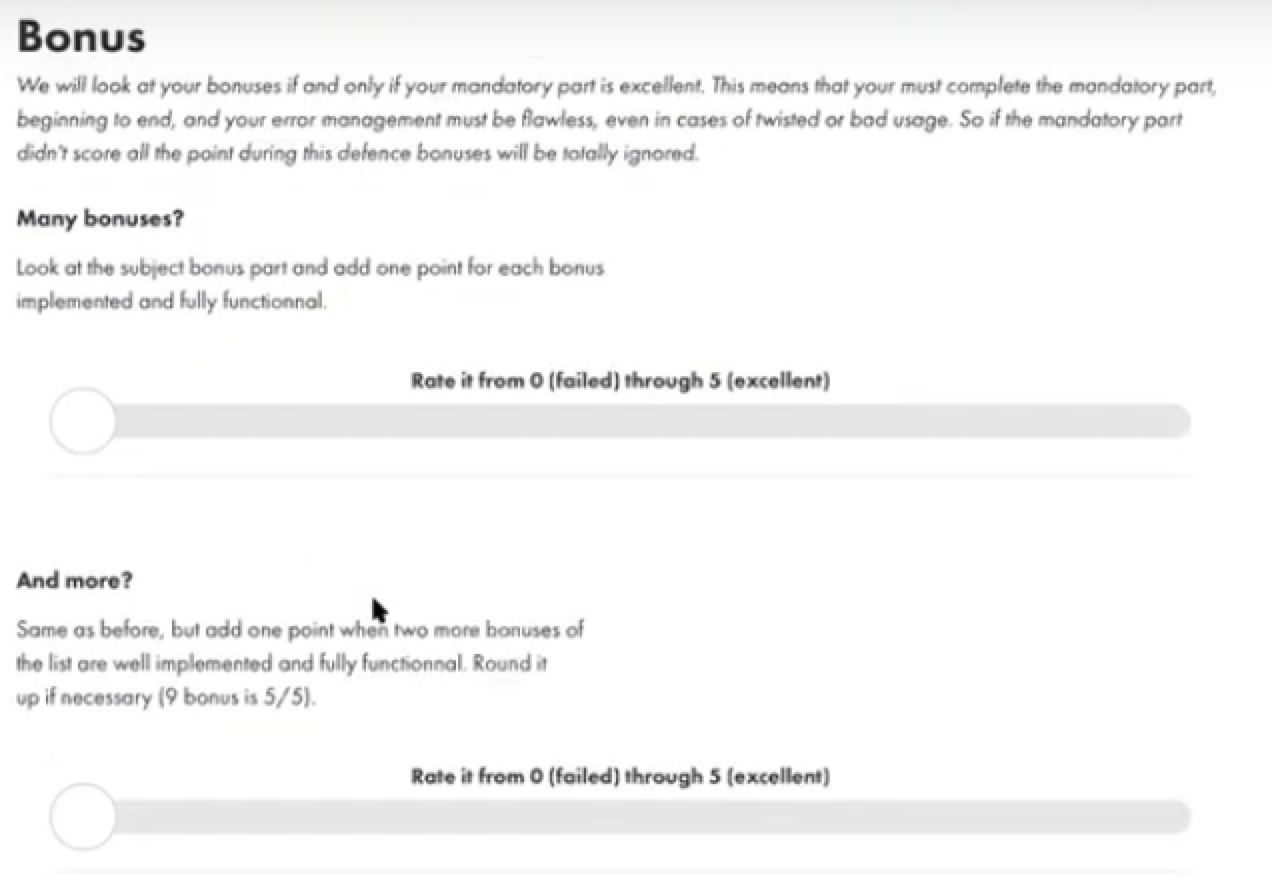
- Place a vertical spot, a sphere and a plane. The spot lighting the sphere's position to create a sphere shadow on the plane. Put the camera aside so we can see the sphere, the plane and the sphere's shadow on the plane. The shadow must be properly displayed without glitching.

- Put a complex scene together with several objects like on illustration V.6 page 10 of the subject. Shadows must be properly displayed without glitching.

Multi-spots

In this section we'll evaluate that it's possible to have several spots in the same scene. Run the program and execute the following test. If it fails, no points will be awarded for this section.

- Put together a scene with several objects including at least a plane on which shadows will be projected as well as 2 spots at the minimum. Check that brightness, shadows and shine effect (if implemented) work properly.



Bonus

We will look at your bonuses if and only if your mandatory part is excellent. This means that your must complete the mandatory part, beginning to end, and your error management must be flawless, even in cases of twisted or bad usage. So if the mandatory part didn't score all the point during this defence bonuses will be totally ignored.

Many bonuses?

Look at the subject bonus part and add one point for each bonus implemented and fully functionnal.

Rate it from 0 (failed) through 5 (excellent)

And more?

Same as before, but add one point when two more bonuses of the list are well implemented and fully functionnal. Round it up if necessary (9 bonus is 5/5).

Rate it from 0 (failed) through 5 (excellent)