CVI - Final project – Pierre Raimbaud

A WEBGL screensaver using random cubic Bezier curves (animation)

Subject chosen: the animation in WebGL / how to make a screen saver

Motivation

Objects in WebGL can be animated using translations and rotations. During this semester, we used this kind of movements to animate the helicopter for example. These movements can be done easily by giving a new position to an object and by calculating its linear translation to this new point. However, for this purpose, parametric curves such as Bezier curves can also be used. Some libraries allow building them more easily; in our case, we used ThreeJS and more precisely the class THREE.CubicBezierCurve3.

Explanations of the code execution

I used a rendering loop in my code; at each step of the loop, a new point on the curve is selected, then the difference between the current position of the objects and this point on the curve is computed and a translation is made to this point. When the final point is reached, a new curve is calculated from this point and the loop starts again, on this new curve. Additionally, the colour of one of the objects is randomly changed. Finally, note that four cameras are used constantly to show four different points of view, making better the screen saver illusion.

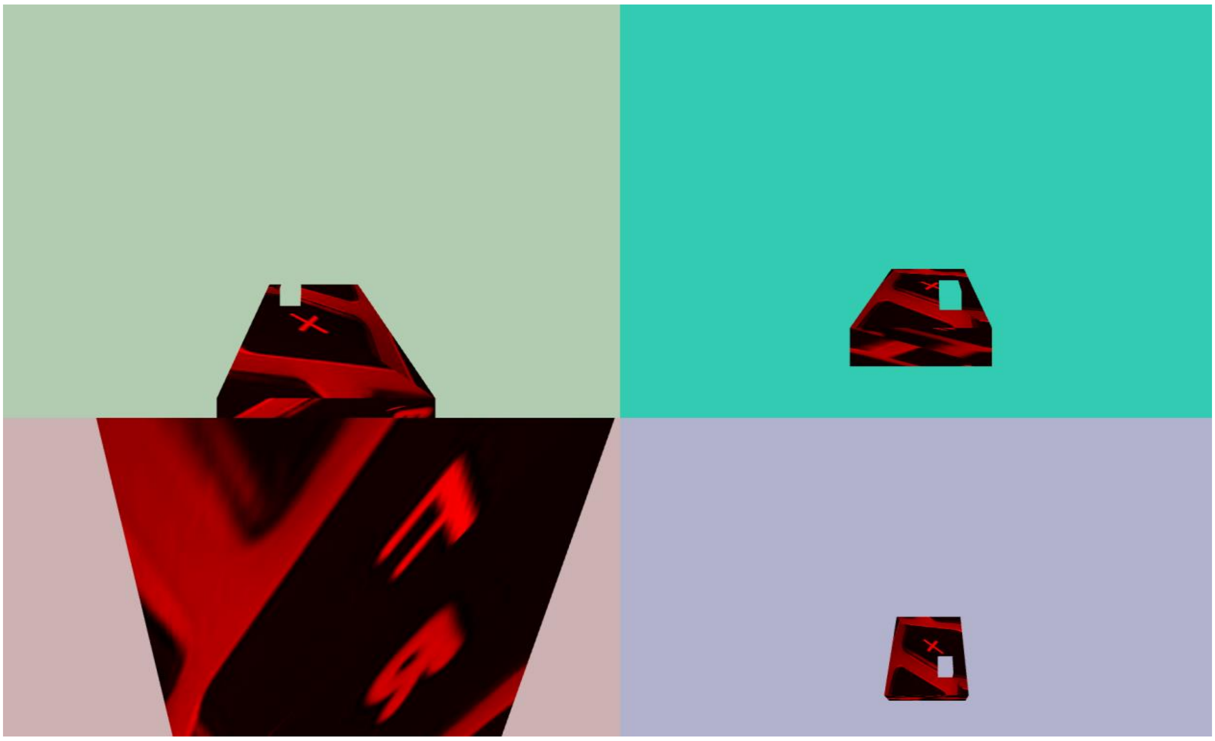
Issues solved

In order to be a great screen saver and compared to the previous work that I gave for the exercise in the class 6 of this curse, I faced two issues regarding the use of the Bezier curves. First, I needed to add a random part for my curves: for that I used Math.random() and multiply it to some fixed points. Then, I calculated the movements during each step when moving along the curve; when the last point is reached, the idea is to start again but on a new curve that starts on the last point obtained using the previous curve: that was my second issue.

Extras

As extras (not a direct contribution to the subject chosen), I present here two different objects coming from the same vertex and fragment shaders but with different vertices and textures (one with an image and one invisible). In addition, I combined the use of colours and texture for one object, allowing changing the colour of the object after each curve. Additionally, I used four different cameras, which show different points of view of the same scene.

A screenshot of the screen saver



Repo and link

The whole project can be found at https://github.com/pierreraimbaud/WebGLScreenSaver and a page where the project has been deployed is https://pierreraimbaud.github.io/WebGLScreenSaver/. Note that the project works locally without downloading libraries. Internet connection is not necessary; libraries are copied, to prevent issue when networking is not working. You only will not be able to download the texture in the screen saver if you are running the project without having Internet.

Bash (tested on Windows)

I wrote a bash script to open the project and to show it as a screen saver by opening it in a browser and putting it in full screen automatically. This feature was essential in order to provide not only a webpage made with WebGL but the illusion of a screen saver.

Future work

As future work, more objects could appear on the screen. Additionally, some actions from the user could be added – actually in the code, you could find some commented codes that allowed to modify the Bezier curves when the user presses some keys on the keyboards.