Bubble Trouble Game Development in WebGL

Pedro Emanuel Amaral Santos
José Pedro Almeida Duarte

***Resumo* – O presente artigo apresenta as etapas de elaboração do conhecido jogo 2D “Bubble Trouble” numa versão 3D em WebGL. Pretende-se mostrar as várias potencialidades do WebGL na área da Computação Visual elaborando-as num exemplo concreto.**

Usando uma abordagem de passo a passo, este artigo aborda os conceitos báscios de WebGL e tenta, essencialmente, torná-la (esta ferramenta) de fácil compreensão para qualquer novo estudante da abrangente área de computação visual. Para além desta abordage, este artigo irá também abordar aspetos de carácter específico e avançado de WebGL, proporcionando ao leitor uma melhor compreensão das potencialidade da Computação Visual combinada com WebGL.

***Abstract* – This paper presents the different phases concerning to the elaboration of the known 2D game “Bubble Trouble” into a 3D version developed in WebGL. It is intended to show the various potentialities of WebGL in what concerns to Computer Vision turn them explicit in a concrete example.**

Using a step-by-step approach, this article will address the basics of WebGL and it will essently try to make it clear to understand to any new computer vision student who want to learn this useful tool. Besides this approach, this article will also dive into some specific and advanced details of WebGL, providing to the reader a better understanding of what Computer Vision combined with WebGL is capable of.

I. INTRODUCTION

WebGL is a JavaScript tool, often thought as a 3D API, for rendering interactive 2D and 3D graphics into any web browser. It can be considered as a rasterization engine that draw primitives, such as points, lines and triangles, based on the code that the programmer suplly. WebGL runs over the Graphic Process Unit what means that the appropriated code that runs on that unit has to be provided. This is accomplished in WebGL using the shaders’ functions, vertex shader and fragment shader.

The fragment shader computes a color for each pixel of the primitive being drawn.

The vertex shader computes vertex positions and can rasterize various kinds of primtives including points, lines, or triangles.

WebGL only cares about clip space coordinates and color so it expects the programmer to provide these two things: a vertex shader that provides the clip space coordiantes and a fragment shader that provides the color.

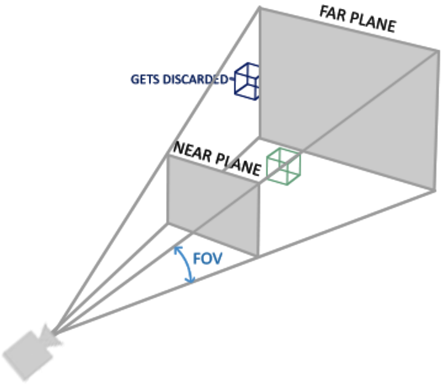

Fig. 1 - Representaion of the clip space

It is impotante to understand the concept of clip space. Clip space is the area that can be seen by the observer, the are where the primitives have really to be draw because they will not be hidden.

The WebGL will be deply discussed in the next chapter. For now, the Bubble Trouble game will be explained.

Bubble Trouble is a game that can be played by 1 or 2 players, a red devil and a blue devil. This game consists in having the players inside a square area that can move horizontally and are limited by the vertical walls at the end of each side. There are bubbles appearing and these bubbles bounce on terrain and if it touches the player, he loses. Players can shoot the bubles but only vertically. When they hit it, they hit one, that bubble will decompose itself in two smaller ones. The purpose of the game is that the player avoid and shoot the balls until them disappear, what means that the player, or the players, have to shoot the balls so many times as it needed for the balls to be so small that they are not able to decompose no more.

The game can be played online the Miniclip web page. There is some Desktop versions that can be found around the internet, some of them, fan made ones, are different from the original in many ways but the basic concepts are still there.

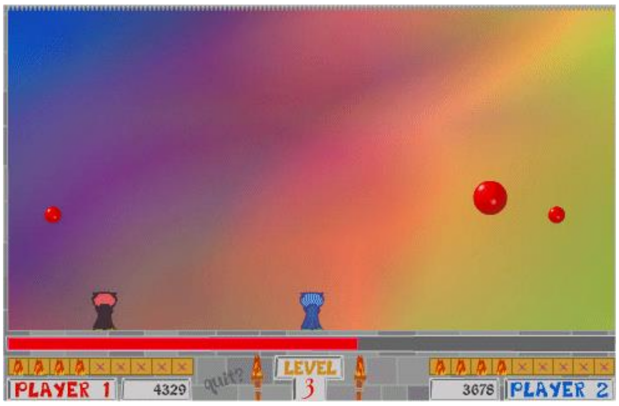

Fig 2 – Original Bubble Trouble game

II. TODO

TODO