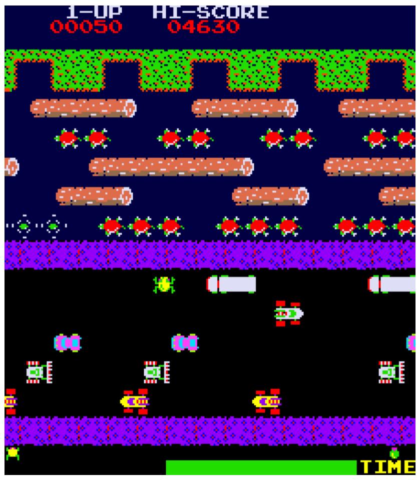
AVT 2019/20

Exercise 3

Introduction

The **Frogger** game was created in 1981 in a partnership between SEGA and Konami for arcade consoles. The player controls a frog that must, in limited time, cross a dangerous river and road. On the road the frog should avoid cars, trucks, buses and even cyclists, while on the river he should use the backs of crocodiles and turtles, as well as wood trunks to cross the river without falling into the water. In both situations, the objects are moving horizontally. The frog must start at the bottom of the screen and reach one of the "frog houses" at the top. Whenever it is hit by one of the obstacles on the road or in the river, the frog dies, i.e. the player loses one of five lives.

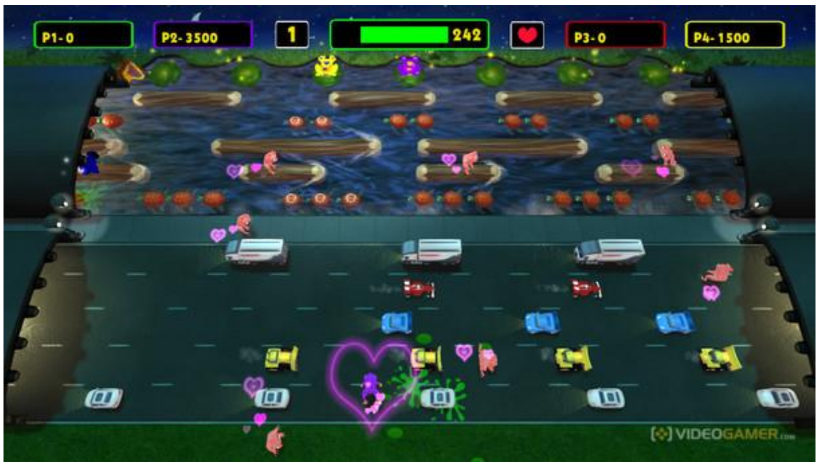


Recently several versions of the game have appeared, including some re-implementations with the original graphics, illustrated in the previous figure, with the cover of the game. Some of these versions can be played online[1].

[1] http://www.happyhopper.org/

Work to do

The main goal of AVT laboratory work this year is to recreate this classic in a 3D version, by using **C++** and **OpenGL version 3.3+**, in a first phase (2 assignments) and **WebGL** with **Google Cardboard** VR *headset*, in a second phase (1 assignment). The idea is to maintain the original gameplay but changing the graphic perspective for a 3D appearance. You can see an example for inspiration in the figure below.



The laboratory work corresponding to the **Frogger 3D** is divided into nine lab classes: 6 exercises and 3 assignments. <u>Each assignment will be evaluated during the semester according to a calendar provided in the Presentation class and corresponds to a certain percentage of the final grade</u>. In each of these assessments there are specific objectives and tasks so they can explore the various components of the AVT program.

The rest of this document refers to the developing tasks for Exercise 3.

Objectives

The objectives for the <u>lab work at 9th October</u> include to understand and to implement the architecture of an interactive graphical application and explore the basics of modeling.

Tasks

The tasks for this exercise are:

1. To model the river and its margins as well as the road and its sides by using a cube for each element. Model the frog, cars and wood trunks, by using simple three-dimensional geometric objects (cubes, cylinders, cones, spheres and torus). The frog and the car should be composed of more than a geometrical object, namely a minimum of two objects for the first and four objects for the second one. For now, you can use the **basic_geometry.cpp** library that was released in Exercise 2 referring to the demo Phong Lighting. Later, groups can improve the graphic quality of the models by using modeling tools and the respective loaders for Modern OpenGL.

2. Set three virtual cameras: a fixed camera to provide a top view of the scene by using an orthogonal projection (similar to a 2D view of the original game), another fixed but perspective one to provide also a top view, and a third moving perspective camera, which must be placed behind the frog and follow its movement (the frog should be visible). It must be possible to switch between the three cameras by using the keyboard keys "1", "2" e "3".

3. To orient the moving camera with the mouse movement and simultaneously pressing its left key.

4. To control the movement of the frog:
    - with the keyboard using the 'O' key to move it to the left, 'P' to move to the right, 'Q' to move forward and 'A' to move backward. The frog should start its movement after pressing the key and stop immediately when the key is released.
    - with the mouse by pressing its right key

5. To start implementing (partially) the movement of the cars, buses, turtles and wood trunks. This should be a uniform rectilinear movement, but the wood trunks should rotate as they are moving. Different elements should move with different speeds and their speed increase with the playing time, ie as the user is playing longer, faster elements move. After leaving the field of play (table)

game’s obstacles should disappear and reappear **randomly** after a while elsewhere in the table.