

CplusplusPie Vanilla: Road Map

Henry Steyer

January 2016

CplusplusPie concretly

stands for C++ exp

CplusplusPie is a 3D user interface and C++ development environment. It uses a two stage coordinate system and matrices of variable dimension and vector type: an object is constructed on a local 3D coordinate system positioned in an global one, which need not to be 3D cartesian but may be a 4D time space or a relativist coordinate system for example. The objects place themselves in a gravity field according to their volume, inertia and elasticity. A Surface with its position and orientation, a point of impression, is exported. A point(an ordinary point) of the surface is a linked list in which an object insert a pixel just after all the pixels which are before him relatively to the position of the point of impression and is responsible for its erasing when it moves or at its end of live. The graphical primitives commonly used are streith rodes with rounded ends which are assembled in K splines and nets. CplusplusPie Vanilla is shipped with a customisable variable font text editor with elastic tabstops, a pdf viewer, a spline editor and a 3D coordinate bases. CplusplusPie is C++ only, OS agnostic and can be easily ported, is its own development environment and is intended to be bare bone for customization and pedagogic purpose, hence the name Vanilla.

CplusplusPie introduces the concept of "organic programing"

and "immanance"

CplusplusPie distributs its points of impression and objects over processes and a computer network. An object can move while the position and orientation of a point of impression is invariant. CplusplusPie is an application program from which perspective the OS has two functions. It relays the sens and motor of the user and gives access to datas in general, in which the virtual objects exist. The demand to the OS is an unhindered access to the datas

as well as to all expressions and impressions channels of the machine to the senses and motor of the user. By "immanence" is meant that real objects are conceptualized together with the virtual ones. For illustration, reconstructing, maybe in form of an array, the keyboard as a data object with the only purpose to get an abstraction of it, would be as flawed as is an unnecessary duplication of any other data object. Also, the screen stands in relation to the pupil as the pupil to the retina and is conceived as an input device and the keyboard similarly as an output. There is also an inversion on the programming language level. In the same way something is taken out of a real file, an input stream would have been named an output stream.

CplusPie is semantic and experimental

the circle is closed again

CplusPie is experimental and does not attempt to maintain compatibility between versions. It is also experimental for its own sake: in analogy to the improvement of speed and memory displacing the needs to optimization towards maintainability, the improvement in the code expressiveness and readability attained among other by renouncing on compatibility means a displacement of the needs of standardization and convention to more negotiated interfaces. This relies on the semantic adequation of the user, which in return allows an improvement of the standards where they are needed.

CplusPie's peripherals

a call for reformation

A distinctive process of CplusPie is the one connected to the senses and motor of the user, so that it gives a triplet of the user, the machine and this particular process. Nevertheless, to emulate different machine user pairs on one machine, it is possible to switch this relation between processes on one machine. Also, to run the debugger on CplusPie, a clone of CplusPie can be put in a slave relation to the cloned process. It is projected to run CplusPie with the Linux kernel directly or through systemd or systemV as an alternative to x11. Care is taken that these peripherals projects or utilities do not infuse in CplusPie's architecture.

Copyright ©2016 Henry Steyer. All rights reserved

typesetting L^AT_EX