



Contribution ID: 48

Type: **not specified**

## **Bub-n-bros = Bubble Bobble + 10 players + general craziness from Python's flexibly**

*Monday, 3 July 2006 16:35 (30 minutes)*

Bub-n-bros is an Internet version of the original Bubble Bobble game: everyone play a dragon that fires bubbles with which he catches monsters. This version has a fast-paced crazy-bonuses feeling to it that is the true landmark of not only ten dragons jumping all around but a really flexible programming language that does not restrict inventiveness.

### **Summary**

I will present the various levels that make the game work. In doing so I will try to describe how this kind of game is made, starting from the basics, first focusing on the generalities and how a framework like bub-n-bros' own can be reused for other similar games (or for Europython presentation tools):

- On the client side, everything is just an image at a screen location, and the list of such images is updated by the server 20 times per second to create the animation. There are multiple minimal, game-independent clients: Pygame, X, Windows, PyGtk, Java, etc.
- The job of the server is to build and broadcast such a list of images. The server is not displaying anything by itself, only communicating with the clients via sockets.
- We will see the simple client-server architecture and a few highlights on the protocol, e.g. on using cross-packet UDP compression techniques originally developed for video data.

In the second part we focus on the bub-n-bros server and game logic:

- To write games, the next best thing to Stackless Python is generators. The server uses them all around: each sprite controls its own movements with a few generators that are advanced one step for each server frame, 40 times per second. The same approach is used by the bonuses, the special effects, and so on. This is a key feature to the everything-occurs-at-the-same-time feeling.
- We will see a few ways for sprites interact with each other, and various kinds of sprites, from walls to transient sprites to player-controlled sprites.
- We will see how the gameplay can occasionally be completely modified, with a minimal coding effort.
- From there, there are just too many big and little features that I could talk about – so I guess I will pick random ones until time runs out :-)

**Author:** Mr RIGO, Armin (University of Düsseldorf)

**Presenter:** Mr RIGO, Armin (University of Düsseldorf)

**Session Classification:** Games

**Track Classification:** Games and Entertainment