Artistic resizing: a technique for rich scale-sensitive vector graphics

Pierre Dragicevic, Stéphane Chatty, David Thevenin, Jean-Luc Vinot

To cite this version:


HAL Id: hal-01021746
https://hal-enac.archives-ouvertes.fr/hal-01021746
Submitted on 4 Sep 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Visual quality is given more and more importance in graphical user interfaces. This points out the need for new methods and tools to effectively involve graphic designers into GUI design and development teams. As a first step in this direction, IntuiKit [Chatty et al. 2004] showed how rich vector graphics could be exploited to combine the visual expressivity of authoring tools such as Adobe Illustrator with the behavioral expressivity of programming toolkits.

Converting static visuals into fully dynamic user interfaces still raises a number of issues. One of them is resizing. The only known technique for resizing vector graphics is uniform scaling. Such a method does not guarantee visual readability for all sizes and aspect ratios. Besides, user interfaces commonly show non-uniform scaling behaviors. When a button is resized for example, invariants are maintained such as label size and centering. Such behavior is traditionally modeled as a set of constraints which can be either coded by a programmer (as it is the case in most toolkits) or inputted to a constraint-solving system. But specifying constraints is hard, if not impossible, on rich and arbitrarily-structured vector graphics, even with declarative or visual approaches.

In spite of their familiarity with media more stable than interactive displays, graphic designers are nevertheless acquainted with the notion of adaptation. For example, a logo type can be used on different media, at different sizes, or on different color backgrounds. Designers often provide several variants of their designs to be either coded by a programmer (as it is the case in most toolkits) or inputted to a constraint-solving system. But specifying constraints is hard, if not impossible, on rich and arbitrarily-structured vector graphics, even with declarative or visual approaches.

In a typical scenario, the designer first uses her drawing tool to provide the static graphics of a GUI object or icon at an arbitrary size. She then specifies its resizing behavior by adding visual variants. This is done by duplicating a variant on the same document then moving and scaling its constituents (Figure 1, left).

The graphic document can be loaded into the IntuiKit interpreter and tested at any time, allowing it to be iteratively enriched until the designer is satisfied by the visual appearance at intermediate and extreme sizes and aspect ratios. Animated examples can be found at http://www.intuilab.com/artresize