

Document downloaded from:

<http://hdl.handle.net/10251/51387>

This paper must be cited as:

Linares Pellicer, JJ.; Micó Tormos, P.; Esparza Peidro, J.; Carrasquer Moya, MA. (2011).
Computer graphics: from desktop to mobile and web. IEEE Computer Graphics and
Applications. 31(4):94-97. doi:10.1109/MCG.2011.56.



The final publication is available at

<http://dx.doi.org/10.1109/MCG.2011.56>

Copyright Institute of Electrical and Electronics Engineers (IEEE)

Computer graphics: from Desktop to Mobile and Web

Authors: Jordi Linares-Pellicer * (jlinares@dsic.upv.es), Pau Micó-Tormos * (pabmitor@disca.upv.es), Javier Esparza-Peidro * (jesparza@dsic.upv.es), Empar Carrasquer-Moya ** (emparca@telefonica.net)

* Instituto Tecnológico de Informática - Universidad Politécnica de Valencia (Spain)

** Batoi Vocational Training School (Spain)

Abstract

Traditionally, computer graphics courses have been centered on desktop environments using very well-known frameworks such as OpenGL. That approach could have become obsolete in a new context where future developers must create applications in quite different environments, such as smartphones, tablets, and Rich Internet Applications (RIA). Touch screens, orientation and GPS sensors, and web integration are now essential when teaching computer graphics, animation and interaction. Teaching computer graphics in this new situation is becoming a very difficult challenge. Traditional tools do not address these new requirements in an integrated way. Here we present a new approach, based on a development environment, *processing*, which features provide high-end solutions in visualization, animation and interaction, with the additional advantage of allowing students to deploy, with minor changes, their programs on desktop computers, smartphones, tablets and websites. It becomes an ideal approach to introduce new devices and interaction paradigms in just a few hours, a fact especially important in our case: an introductory computer graphics course.

We have found that this new approach dramatically boosts the motivation of our students. Their work is not just visible in the laboratory, they can show it to classmates and friends on their own smartphones, tablets and websites. It's what we call the "*hey! look what I've done*" effect. Although the tool, *processing*, was relevant, it was that, showing their work on their websites and mobiles, what made a '*before and after*' in our course.

Keywords: Computer graphics, education, desktop, mobile, web, processing.org

Something new is happening

Computer graphics has traditionally been used as a motivational tool when teaching other topics. Now, we sometimes notice that our students are the ones who are not motivated when learning computer graphics. Three years ago, after finding really poor results when using the traditional OpenGL + GLUT approach in laboratory work and assignments, we discovered an interesting new tool: *processing* [1]. Easy to learn, open source, Java-based, scalable and inspired in OpenGL and PostScript, our students are now able to put into practice 2D, 3D, animation, and interaction techniques in just a few hours (see figure 1). *Processing* was designed for visual artists and its popularity is growing up very quickly in the computer graphics community [2][3].

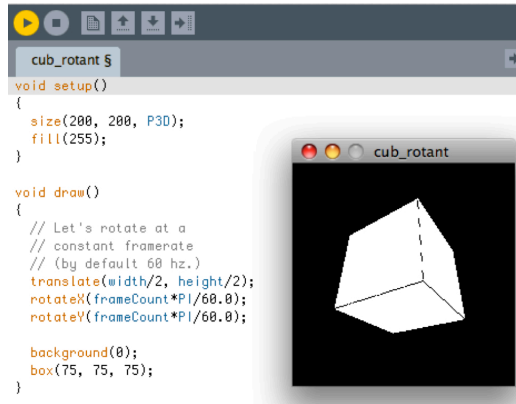


Figure 1. *Processing* and its development environment makes learning computer graphics really easy.

After three years working with it, we have noticed a significant increment in our student's motivation [4]. During the first few minutes, students are able to get results. Animation and interaction don't require additional tools or libraries and 3D, based on OpenGL, is seamlessly introduced from their 2D knowledge. Moreover, all their work is not a waste of time. Their applications can become pure Java programs using powerful IDEs such as Eclipse or Netbeans.

However, something new is happening. One just has to see market sales evolution regarding smartphones and tablets. Our students immediately ask us how their computer graphics knowledge and skills can be integrated into these new platforms. The same problem arises when they want to integrate their work into Rich Internet Applications (RIA).

There are different specific frameworks in order to develop computer graphics applications on desktop, mobile and web environments but they did not fit our requirements: an introductory computer graphics course, one semester, 2 hours per week for theoretical lectures and one and a half hours per week for laboratory work. There is no time to learn different tools for different environments.

Fortunately, the *processing* project has evolved to provide a unique development framework for all our objectives. Here we describe some of its advantages and why it is behind our successful current approach to our course.

Desktop development

As stated before, *processing* offers an important set of advantages in an introductory computer graphics course:

- Easy to learn but scalable. Interaction and animation are easy to learn. 3D topics like texturing and lighting can be covered in an introductory course.
- Multiplatform and open source.
- Powerful. Based on OpenGL. Students don't waste their time learning it, since they can eventually generate pure functional Java applications.

- Perfect tool when introducing theoretical concepts such as geometric transformations, by using it almost like an OpenGL interpreter in the classroom [5].
- It's a useful prototyping tool even in our own researching work.

Mobile development

Computer graphics cannot only be taught thinking of desktop environments anymore. Smartphones and tablets must now be considered and, with them, their specific features.

Ubiquitous computing gradual introduction forces us to allow students to put their knowledge into practice in all these new devices. Very important differences turn up regarding interaction and hardware capabilities. Bad keyboard interaction, no mouse, but multitouch and sensor-based interaction (GPS, accelerometers, magnetometer, light and temperature sensors and more).

The *processing* community has recently offered new possibilities in this field: a native port to the popular Android platform [6] (see figure 2) and indirect approaches to iOS devices (iPhone and iPad) using *processing.js* (a framework we will later describe).

With these new contributions to the *processing* framework, sensor-based and context-aware computing can be easily incorporated into the computer graphics skills of our students. Students can adapt their original desktop applications to fit new devices hardware restrictions but also to explore new interaction techniques based on their sensor-based capabilities. They just need to learn the characteristics of the sensors and add extra call-back functions to their original programs in order to get new specific events.

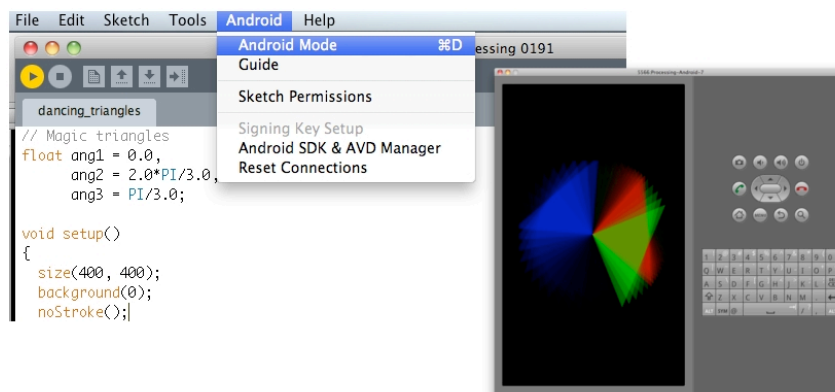


Figure 2. *Processing* has a native port to the popular Android smartphones and tablets.

Web development

From the very beginning, *processing* was designed to be able to become a web component. Exporting applications as Java applets, students can immediately integrate their work into any website. However, this approach needs a Java plugin in the client browser. This restriction is no longer necessary with *processing.js* [7].

Based on Javascript, *processing.js* allows any *processing* program to be interpreted and executed in an HTML 5 canvas, no Java plugin required. Students, overall those with additional web

development knowledge, discover a direct way of developing RIA products based on their *processing* desktop applications.

Our current approach

Our theoretical lectures have not changed. We are still using our traditional curricula.

The main differences are how our laboratory work and student's assignments are addressed (the full course materials can be consulted at [8]). Our objective is a step-by-step introduction to computer graphics, where 2D is firstly covered, 3D is then explored, and finally the new mobile and web possibilities are shown. *Processing* is the only tool we use from the beginning to the end.

Therefore we have two main objectives: a practical introduction to computer graphics using *processing* (with which students are centered on computer graphics topics from the word go), and boost their motivation letting them know that everything they are doing will be eventually working on mobile and web platforms.

The complete eight units are as follows:

1. Introduction to *processing*. A brief description of the development tool and its main advantages.
2. 2D basic primitives. 2D is still important in a lot of applications, therefore we introduce our students to points, lines, ellipses, arcs, curves, and complex shapes.
3. Images and text. A brief introduction to simple image processing techniques is also something our students appreciate. Creating, loading, and displaying them allows the development of more ambitious applications.
4. 2D geometric transformations are introduced. The concept of the model/view matrix, how to work with it and how it influences everything you draw in *processing* is essential.
5. Animation. If there is something that really motivates a student it is animation. And animation in *processing* is well-solved, just by adding a call-back function that is going to be called each frame.
6. Interaction. Despite other environments, using system variables or just adding some call-back functions, interaction can be covered in *processing* without the traditional complexity in other environments.
7. Introduction to 3D. Finally, 3D is covered. Highly influenced by OpenGL, *processing* allows a really smooth transition from 2D to 3D. Basic lighting, shading and texturing techniques can be covered with basic statements, and further OpenGL-based functions can be explained to those students with better capabilities.
8. From desktop to mobile and web. Although it is in the very last stage when students are formally invited to work with mobile and web environments, they are motivated from the very beginning. All the main topics are put into practice on a desktop platform. It is afterwards when mobile and web possibilities are shown with all their possibilities. Students discover then, with just a couple of hours, that their current desktop developments can be migrated to smartphones, tablets and websites. The consequence of this approach is that students directly

experiment how sometimes their original applications must be tailored to these new environments. Hardware or bandwidth limits must be considered in applications that work with huge images or geometries, forcing them to fine tune their models. Mobile sensors offer them new possibilities of interaction and students learn how to create full sensor-based applications from their original desktop applications, just by adding new call-back functions. With these new call-back functions, you can handle sensor events, get their values and map them into those ones that fit your interaction objectives.

Results

As mentioned before, we have previously described the advantages of using *processing* environment in laboratory work and assignments. Our objective now was whether the introduction of the new mobile and web ports of *processing* improved our previous results.

Three years ago, we started to measure motivation and marks of our students when we started using *processing*. Besides their final marks in the course, we started to use a questionnaire to measure their degree of motivation and satisfaction regarding their laboratory work.

The transition from OpenGL + GLUT to *processing* was very successful. Now we just had to evaluate whether the introduction of mobile and web functionalities were able to improve the results of our students.

The results have been really significant. Every year an average of 94% of our students agreed with the choice of *processing*. This last course we decided to introduce mobile and web development. When we asked after the course about what features made *processing* an interesting tool, most of them agreed that mobile and web integration were the features they found the most motivating.

What is more, voluntary assignments, those we suggest to our students to improve their final marks, have significantly increased. More than 50% of our students this year developed a voluntary extra assignment, compared to a previous average of 10%. Little games, image viewers, drawing tools, augmented reality applications and many more were some of the projects they developed (see figures 3 and 4). Talking with them, their actual motivation was the fact of being able to have their applications on their websites and pockets (smartphones and tablets).

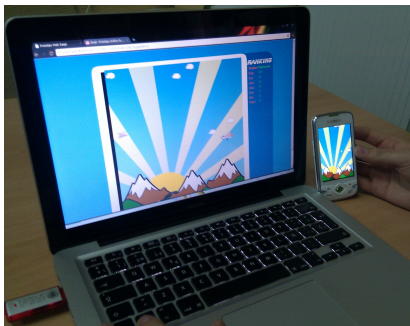


Figure 3. On the left, a game a student initially developed for a desktop computer. With some extra call-backs, the same game is now available on a website and on an Android mobile. Android version game interaction is controlled with the orientation sensor. On the right, a student's planetary system based on Augmented Reality.

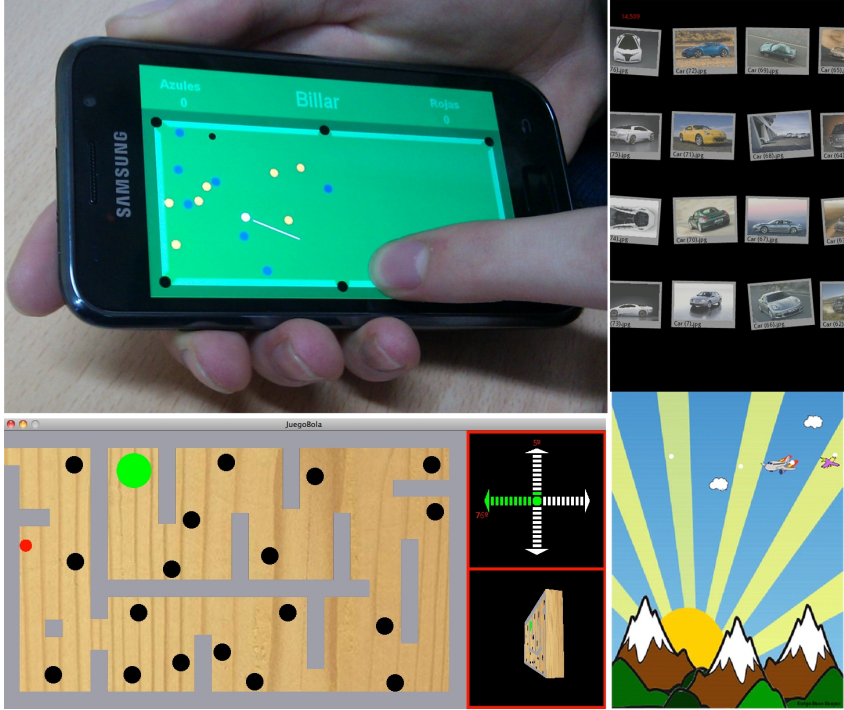


Figure 4. Games, image viewers, little drawing tools are just some examples of what our students have developed for desktop, mobile and web environments.

We had the opportunity of experimenting our approach in a very different context: a vocational training school, an alternative to university studies in our country. These students also cover computer science but from a more practical point of view compared to our university students. With just 8 hours, and with a less ambitious goal, these students were also developing computer graphics applications on desktop, mobile and web environments.

Conclusions

Smartphones, tablets, RIA are changing our traditional desktop computer graphics point of view. This is especially critical in laboratory work and student's assignments. Terms such as sensor-based or context-aware interaction are becoming popular and our computer graphics courses must consider them.

Hardware restrictions of mobile devices and bandwidth constraints are important in mobile and web development. Computer graphics courses should take into account these changes and the appropriate tools should be selected.

We have shown how *processing* has allowed us to increase our students skills, motivation, and overall satisfaction.

The results of our new approach allow three main advantages:

- A fast introduction to computer graphics main topics: 2D, 3D, interaction and animation; by using a very easy to learn tool and language.
- A straightforward adaptation of a desktop application to different media: web and mobile devices. Specific topics of these new channels, such as multi-touch screen interaction, accelerometers, magnetometers, GPS and others, and specific challenges in web development can be covered from a unique tool and programming framework, in a very short, introductory and yet comprehensive computer graphics course.
- The development of complete interactive applications in three environments: desktop, mobile and web. Consequently, students learn features, advantages and challenges of the three paradigms with a unique development framework, language and SDK.

Besides *processing* and its advantages, it was really clear compared to previous years, that the “*hey! look what I’ve done*” effect, i.e., using mobile devices and websites as the targets of their work, has become a ‘*before and after*’ in terms of student’s motivation.

Without any doubt, learning computer graphics becomes much more attractive if students can put into practice their skills in these new mobile devices and websites.

Bibliography

- [1] <http://www.processing.org>
- [2] Fry B.: Visualizing Data: Exploring and Explaining Data with the Processing Environment, illustrated edition ed. O’Reilly Media, Inc., January 2008
- [3] Reas C., Fry B.: Processing: programming for the media arts. AI Soc. 20, 4 (2006), 526 538
- [4] Linares-Pellicer J., Santonja-Blanes J., Mico-Tormos P., Cuesta-Frau D.: Using processing.org in an Introductory Computer Graphics Course. Domik G., Scateni R., (Eds.), Eurographics Association, pp. 23 28.
- [5] Chen B., Cheng H. H.: Interpretive OpenGL for computer graphics. Computers and Graphics 29, 3 (June 2005), 331 339.
- [6] <http://android.processing.org>
- [7] <http://processingjs.org>
- [8] <http://processing.tk>