Title

Student 1, Student 2, Student 3, and Student 4

Department of Informatics IfI University of Zurich UZH

Binzmühlestrasse 14, CH---8050 Zürich, Switzerland

{email1, email2}@uzh.ch

**Abstract.** The abstract should summarize the contents of the paper and should contain at least 70 and at most 150 words. It should be written using the abstract format. This example is for illustrative purposes only; for example, paragraphis in the outline below are much shorter than we expect you to have! Follow good writing style. We strongly encourage you to list up to 5 relevant keywords below; the five given below are simply examples. Use your own! If you are having trouble thinking of keywords, see the IEEE Computer Society keyword taxonomy [4].

1. Introduction

What does the project do? What was your main goal? Or what problem were you trying to solve? What were your learning objectives?

1. Solution/DApp Design

What was the project, what were the specifications. If you are doing a game, you would put the characteristics of the game, what guided your work (eg the rules? the graphics? the language? the platform(s)?) If this was a learning project, what did you hope to learn? What was significant about this problem to you? What criteria did you use to determine your "success"?

* 1. Use cases (If any)

What specific functions does your project accomplish for its users? A use case is an objective user(s) wants to achieve with a system. “Use cases are named with verb or verb + noun phrase. It is usually short yet descriptive enough to describe a user objective.” See Visual Paradigm’s brief tutorial for writing effective use cases [8].

* 1. Implementation choices

Details about your hardware and software choices (if you used pre-existing packages, for instance.) How you went about testing your project or what criteria you used in order to determine your outcomes.

If your project has an interesting or novel hardware configuration, you should include a diagram here, or in a separate subsection “Hardware Architecture”.

* 1. Software Architecture

How the code itself is organized into classes and/or functions. It's good to have an example that puts your project in context. Examples will vary depending on your project.

We also strongly suggestion you include a software architecture diagram. If your project has a database, we suggest you include a conceptual database model diagram. Remember, it is not enough to simply include the figure in your paper; you must also give it a caption and explain it in the text (referring to the figure in that text).



Figure 1. VISTAS Software Architecture

 Fig. 1 shows the VISTAS Software Architecture. VISTAS’ design (above Figure) makes it easy to add new input formats and visualizations. We separated the front- and back-ends, so our scientists could drive their models using visualizations. VISTAS currently uses a raster data model (grids), and does not convert coordinate systems, so height and elevation data must be input along with variable values for a given cell in the grid (in the same row).

* 1. Program Code

If relevant you could include a short piece of program code that you use to demonstrate some feature of your work. The LNCS template provides a way of listing your code in the proper format; in Word, this is the Prog.Code button. Use it!

Below is an example of some simple C++ graphics code that draws a robot arm. Note how the current ViewModel is pushed onto the stack prior to calling the translate and rotate functions; thus World Coordinates are preserved across those functions.

void drawArm(mat4 &mv)

{

 mvMatrixStack.pushMatrix(mv);

 mv = mv\*Translate(0,armLength/2.,0);

 mv = mv\*Scale(.2,armLength,.2);

 glUniformMatrix4fv( model\_view, 1, GL\_TRUE, mv );

 myCube.draw();

 mv = mvMatrixStack.popMatrix();

}

* 1. Front-End / Graphical User Interface
1. Conclusions/Critique

Identify both what succeeded (your positive results) and what didn't. If your work had a major shift in orientation you should have provided an indication earlier but you could discuss the consequences of that change here. Specify what design or programming issues you ran into and how you addressed them.

To what extent did the final version of your project resemble your original design for the project (as described in your project proposal and project plan.)

* Which parts of your project work went the way you expected them to? Which ones did not?
* How did keeping minutes affect your group's work? How did it shape your experience of your project work.
* If you were to do this project again, what would you do differently next time?
1. Future Work

What did you learn and what future plans do you have for this work. Or if you have no specific plans to continue the project, how the learning you accomplished here will inform your future studies/work.

1. References

Just list your references here (don’t include this paragraph). Use standard format (e.g., look at what follows). Be sure to include texts or web sites you used as references, as well as scholarly journals. Remember you are aiming to provide material that would help someone recreate your project. Alphabetize your references.

1. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. In: 10th IEEE International Symposium on High Performance Distributed Computing, pp.181--184. IEEE Press, New York (2001)
2. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)
3. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration. Technical report, Global Grid Forum (2002)
4. IEEE Computer Society Keywords, http://www.computer.org/
portal/web/publications/acmtaxonomy
5. May, P., Ehrlich, H.C., Steinke, T.: ZIB Structure Prediction Pipeline: Composing a Complex Biological Workflow through Web Services. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) Euro-Par 2006. LNCS, vol. 4128, pp. 1148--1158. Springer, Heidelberg (2006)
6. National Center for Biotechnology Information,
<http://www.ncbi.nlm.nih.gov>
7. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195--197 (1981)

Visual Paradigm, [http://www.visual-paradigm.com/
tutorials/writingeffectiveusecase.jsp](http://www.visual-paradigm.com/tutorials/writingeffectiveusecase.jsp)