PowerChalk: An adaptive E-Learning application.

Dan-El Neil Vila Rosado*, Margarita Esponda-Argüero y Raúl Rojas*****

Freie Universität Berlin, Department of Mathematics and Computer Science.

Takustrasse 9. 14195 Berlin.

* Tel.: +49 30 838 75148. E-mail address: vila80@inf.fu-berlin.de.

** Tel.: +49 30 838 75142. E-mail address: esponda@inf.fu-berlin.de.

*** Tel.: +49 30 83875130. E-mail address: Raul.Rojas@fu-berlin.de.

Abstract

This paper present a new interactive e-learning application called PowerChalk. PowerChalk arises as the result from the analysis of the evolution of Information Systems Design Theory for E-Learning; it was designed to resolve an important limitation of current design methods and e-learning systems: adaptability. Modular programming is the design technique used in PowerChalk to improve human computer interaction with the management of different types of data in order to have positive effect on both learning score and learning satisfaction. PowerChalk works like a Transaction Processing System in order to support collaboration, communication, creativity and learning through a collection of organized modules. The characteristics of PowerChalk facilitate developing of competencies for using multimedia technologies in any learning session taking into account the teacher and student perspective. The goal of PowerChalk is to provide a robust, reliable, usable and sustainable multimedia technology for collaborative learning.

Keywords*: Multimedia technology; e-Learning; information systems; modular programming.*

Introduction.

The importance of information, multimedia, communication and e-learning technologies in order of promote open, distance, flexible learning satisfaction is obvious; but an important problem not well-documented on e-learning is how and with what resources we need to develop an e-learning application to ensure usability and accessibility to the users.

Learning theorists justify that to reach an objective, acquire a skill or learn something; the learner must be actively involved through practice to cognitively incorporate it into long term memory. Hereby, the interaction while performing the objective helps the learner to reach the objective and recall the information, skill or behavior that was learned [1]. So, the most important characteristic of an e-learning application it is interactivity. Referring to interactive media tools, the adjectives “superior” or “best” depends on the specific context but the interactive whiteboard (IWB) report potential pedagogical benefits and less drawbacks for teachers and students [2,3,4]. In general, a formative application should: be interactive and provide feedback, have specific goals, motivate, communicating a continuous sensation of challenge, provide suitable tools; avoid distractions and factors of nuisance interrupting the learning stream [5]. On the other hand a set of features specific for e-learning systems interfaces are: they have to provide a comprehensive idea of content organization and of systems functionalities, simple and efficient navigation, advanced personalization of contents and clear exit.

Accomplish all this characteristics it is a difficult task, but through time many e-learning applications have been developed to satisfy some specific goals. Many institutions are adopting different kind of systems to support learning and teaching, however evidence suggest that the solutions are limiting the incentive to innovate and they are restricting the ability to integrate with other systems [6]. Herewith there is a need for theory to support the design and implementation of these e-learning systems. In this approach we analyzed the requirements of e-learning and the state of art in Information Systems Design Theory (ISDT) to propose Modular programming like a software design technique to solve the most important problem for programmers and users of actual e-learning applications: adaptability.

To support this design-science research, this work has been instantiated in an e-learning application that has been used by staff, students and teachers (PowerChalk).

This paper is structured as follows. We review ISDT information and the related work in section 1. Then, in section 2 we describe the PowerChalk system and in section 3 the different modules of the system. Finally in section 4 the implications of our findings and further research are suggested.

1. ISDT and related work.

The media richness theories argue that the communication efficiency between people is affected by the fitness of the media and the characteristics of the communication task. Also, we know that utilizing high richness multimedia materials for the curse unit with high uncertainly and equivocality has positive effect on both learning score and learning satisfaction [7]. Actually, the more efficient tool to use and create high richness multimedia materials is the electronic chalkboard.

Among electronic chalkboard applications we find: educational tools, intelligent work-spaces, group decision making tools, graphical visualizations tools, etc. Currently there are few electronic systems and projects that offer a combination of collaboration platforms, interactive chalkboards and displays that enhance any discussion session.

In the state of the art we can found the followings projects:

- NotePals. Ink-based, lightweight note sharing system that gives group members easy access to each other's experiences through their personal notes on personal digital assistants. PhD. project developed by UCLA-Xerox [8].
- E-Cognocracy. Democratic system conceived for the purpose of extracting and diffusing the knowledge derived from a group of people. Developed by Univ. Of Zaragoza [9].
- K-Sketch. Interface design for creating informal animations from sketches. Developed by University of Washigton - University Of California [10].
- BumpTop. Virtual Desktop. Developed by University of Toronto [11].
- PADDs. Digital documents that can be manipulated either on a computer screen or on paper. University of Maryland [12].
- SMART systems. Company of electronic whiteboards [13].
- E-Chalk. Electronic chalkboard developed by FU-Berlin [14].
- Cabri software. Interactive media tool to create content faster to accompany text books or provide activities as resources in 2 o 3 dimensions [15]

The above systems are specializing in a very specific task but focusing on availability and usability, they have different limitations depending on stakeholders, among which we mention: costs, hardware or software limitations, can only work with certain types of data and difficult to update to different kinds of hardware. Like conclusion, they have difficulty to adapt on different kind of stakeholders or circumstances.

The ability to integrate with other systems and evolve are not well satisfied, but Information Systems Design Theory (ISTDT) provides a sight on structures and processes to effectively implement technology for learning activities and improve actual e-learning systems.

David Jones defines three generations of ISDT formulation [16].

- First Generation (1996-now). Typified by the generation of requirements, use of templates, software wrappers and commercial off-the-shelf products. Advantages: ability to adapt to change, platform independence. Disadvantages: High level of technical skills required for the users and developers. Results: A growing number of stakeholders feeling limited by the approach, therefore this kind of platforms are abandoned for the users.
- Second generation (1999-now). Delineated by increase use of the system by modifying the development and support processes with insights from diffusion

theory, design patterns and pattern mining. Advantages: Instantiations more acceptable to users leading to greater adoption. Disadvantages: The systems have a pro-innovation bias that, amongst other effects, can decrease flexibility and increase difficulties to implement changes. Results: The systems are not able to evolve so quick like hardware technology, requirements or needs.
- Third generation (2000-now). Differentiated by increase the agility of the system to change by encompassing features from emergent and agile development methodologies and use of OO and patterns. Advantages: Simple design, coding standards and collective code ownership. Disadvantages: Need to coordinate an efficient and disciplined development team. Results: Significant increase in systems use, a notorious decrease of developer's teams and therefore less capability to evolve.

Actual design methods rely on some of the ISDT generations, but in general rely on development being performed by a development team whereas the original template system provided methods by which end users can develop new templates. This characteristic could improve the system's ability to support faster response to change. In this sense PowerChalk is the platform that solve many of the limitations of current systems in order to support diversity, easy development, adaptability and improve human-computer interaction for the management of different types of information.

2. PowerChalk structure.

PowerChalk is a collaborative e-learning system for a new kind of electronic chalk, where we can combine the advantages of the traditional chalkboard with the functionality of multimedia, electronic devices and modern distance educations tools. PowerChalk will transform any working session into a visual and reliable communication tool.

Goals of PowerChalk:

- To make the system robust, reliable, usable and sustainable with an efficient software structure.
- Provide the platform with a set of tools to build new modules that let the end-users to analyze complex miscellaneous information quickly, insert relevant notes, access maps and integrate specialized simulation modules with ease (Rich Client Platform).
- Provide the PowerChalk with a communication module via internet to share information and have real-time collaborative sessions.
- Easy to adapt to different kind of hardware.
- Easy to update.

With this, we can reach a high-performance system for teaching and learning. For this purpose, PowerChalk was build through a distributed development model based on modularization.

A modular application like PowerChalk is composed of smaller, separated chunks of code that are well isolated. Those chunks of software can then be developed in separated teams with their own life cycle, their own schedule. The results can then be assembled together by a separate entity [17]. This modular architecture has the followings advantages:

- It simplifies the creation of new features. These features can be Macros or useful objects for a multimedia lesson-planning session.
- It makes it easy for users to add and remove features. With this characteristic the user can modify the different tools that are using, for example: the electronic ink, the pdf viewer, the image reader and more between the availables for PowerChalk.
- It makes it easy to update existing features.

With these benefits PowerChalk becomes a modern, flexible, technology-friendly approach to e-learning and teaching.
The architecture of PowerChalk is showed in Figure 1.

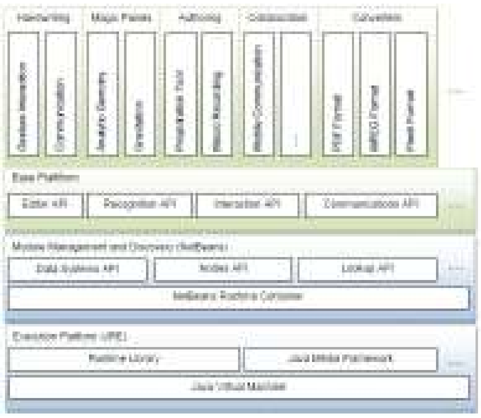

Figure 1: PowerChalk structure

We have a platform and architecture for distributed development with a modularized architecture in Java NetBeans platform [18]. We solved the design problems of another analyzed chalkboard applications through design patterns like navigation, composition, semantic zooming, lookup, etc. [19]. The software structure allows us to give more functionality such as affine transformations over the strokes or images, zooming in all canvas section, layering, distributed development, etc.

3. PowerChalk modules.

As applications become more complicated and we need high sustainable software, they are more frequently assembled from modules that were developed independently. In the PowerChalk system, the modules work together to improve our experience as learner or teacher.

Main Editor module.

The base of an electronic board is an electronic ink. We have developed a prototype of an Editor in Java to add electronic ink components, others objects (images, keyboard input, etc.) and its edit functionalities (Figure 2). This editor is the base of a handwriting recognition system. Also, the editor includes pen-based applications to process the different kinds of objects through annotation, correction, condensation, organization, zoom abilities, print options (normal and pdf converter) and building of slides.

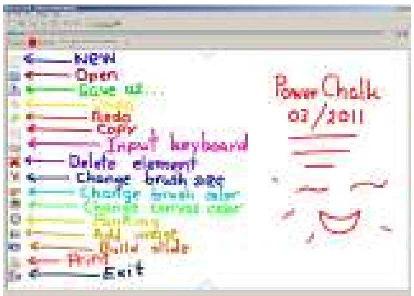

Figure 2: PowerChalk Main editor.

Teaching a lesson with interactive media involves not just looking at information on a screen or electronic chalkboard, but also underlining text, highlighting, commenting or adding another objects on the fly. This combination of showing the information with critical thinking is useful for interactive media learning or active reading [20]. For this purpose, we have an annotation layer mechanism.

In the Main Editor module, PowerChalk have a hierarchical mechanism for supporting multiple overlapping layers of annotations for data (images, text, strokes, etc.). Identification and processing of the different layers in the document structure of a PowerChalk session is a great tool for end-users (Figure 3).

It should be noted, that every object in the PowerChalk canvas, has its own timestamp and the complete session can be stored; therefore, archived sessions can be played on-demand as conventional videos, fast-forwarding or rewinding the file. On the other hand, students do not have to copy the board content any more, since PowerChalk can generate a printable version of the board content via printer or pdf file.

Pen and Digitizer Tablets module.

PowerChalk have a library for accessing pen/digitizer tablets and pointing devices using Java. Its key features are Event/Listener architecture and the fact that device access is implemented through providers in different operating systems and hardware devices (Linux, Windows, Wacom tablets, Hanvon tablets, etc.). Editor tools that interact with the pen tablet have also been developed; example of this is to open a color chooser through a click with pressure or a selection tool (Figure 3).

Figure 3: Annotation layers and digitizer tablet module example.

PowerChalk has been used in PC computers and tablet-PC systems for classes in FU-Berlin by teachers and students to test the functionality and practicality.

Multi-screen Manager module.

Rapid adoption of digital devices leads to use several screens to perform the same activities. Every kind of screen has unique benefits; therefore, together they enhance the user experience. In general, the user wants relevant, consistent and connected information across their screens. This module allows to use Power-Chalk in several end-users hardware configurations giving accessibility and increasing productivity in any collaborative learning session (Figure 4).

The principal experiments with the Power Chalk Editor are in a Data-wall with four screens that was builded in a classroom for testing. We have gathering information from teachers and students to improve the PowerChalk Editor as a part of the usability test [17].

Handwriting Recognition module.

Adding handwriting recognition to the system makes the electronic board a great tool for a learning session because obtained signal is converted into letter codes which are usable within computer and text-processing applications (Recognition of mathematical expressions, diagrams, etc.) or user interface prototypes.

Figure 4: PowerChalk in a multi-screen system.

With the handwriting module, we are capable to:

- Build bridge modules between PowerChalk and MATLAB, MATHEMATICA, etc. With this functionality, the user just need to type user interface prototypes for calculations in Mathematica´s language as well as commands or actions for graphical output and the result can be displayed on the PowerChalk canvas (Collaboration modules for applications).
- Recognize user interface prototypes. Pen, finger, and wand gestures are increasingly relevant to many new user interfaces for tablet Pcs or PDAs. Designing and implementing gesture recognition was our goal to give to PowerChalk a highly human computer interaction tool [18].

The problem of gesture recognition is divided in 2 cases: one-stroke recognition and multi-stroke recognition. In the one-stroke recognition we have analyzed and implemented methods for use in rapid prototyping; these methods have the advantage that do not require numerous training examples [19]. The recognition results for the best configuration for one-stroke recognition reached 1.15% recognition errors (S.D. = 3.69) with 12 training examples. Equivalently recognition rate was 98.85%. For multi-stroke recognition the method worked with 9 training examples and reach a recognition rate of 91.02% and equivalently reached 8.98% recognition error (S.D. = 6.62) [20].

Communication module.

PowerChalk is a system that helps users communicate and collaborate for a certain project. In a PowerChalk session the conversation and information on the chalkboard are equally important. We want the user to be able to communicate and work together with richly formatted text, photos, videos, maps and more. For this purpose a review in new technologies like advanced protocols of communications (Google Wave Protocol) and open source platforms (SkypeKit, Java Media Framework, transmitting audio and video over RTP) were developed.

The structure of communication is through 3 channels: Audio, video and PowerChalk-canvas data. The system was tested transmitting audio and video over RTP and JMF but due to limitations of the JMF 2.1 implementation, audio and video are not in tight synchronization and have problems of RTP-streaming. The system was improved with an external synchronization.

The Java Media Framework API (JMF) enables audio, video and other time-based media to be added to applications and applets built on Java technology. With this module in PowerChalk we can capture, playback, stream, and transcode multiple media formats providing a powerful toolkit to develop scalable, cross-platform communication technology.

The hierarchy class of the objects or data (based on Piccolo 2D) let us to send any object through internet and to show the same information in any other PowerChalk canvas [21]. This technology will allow us to transmit the PowerChalk-canvas data with high efficiency and synchronization with the other channels.

Otherwise, designing a web framework, web services and an update center for the PowerChalk platform will allow the users to achieve distributed software development for increasing the applications of PowerChalk platform and also, review, edit and build shared sets of recorded sessions. A recorded session could be a lecture, a discussion session over a set of data for group decision-making, an animation, or a set of data processed for gathering information. The communication module for developers is responsible for supporting these activities.

PowerChalk have a design for the web framework design and the beginning of a framework support module for the “Wicket framework” (Apache Wicket). With proper mark-up/logic separation, a POJO data model, and a refreshing lack of XML, Apache Wicket makes developing web-apps simple and enjoyable [22]. With this kind of application, the end-user is able to publish any work session to a general public, for example: the classroom notes or homework.

Web services are software systems that are externally available over a network. You can use them to integrate computer applications that are written in different languages and run on different platforms. Web services are independent from language and platform because vendors have agreed on common web service standards [23]. PowerChalk works with RESTful Web Services and SOAP-based Web services.

Running the Update center will check if there are new modules or new versions of already installed modules available. From new or updated modules, the user could select, download, and install them. We developed an update center module for the PowerChalk structure over the HTTP protocol.

Collaboration Modules.

A collaboration module for applications is a bridge module between PowerChalk and software like MATLAB, OCTAVE, MATHEMATICA, Gnuplot, etc. With this kind of modules we can send instructions for plotting functions, expression evaluation, solve equations, run a script, etc. and see the output on the PowerChalk canvas. We have designed an API for this kind of modules. PowerChalk has a collaboration module for OCTAVE and Gnuplot implemented.

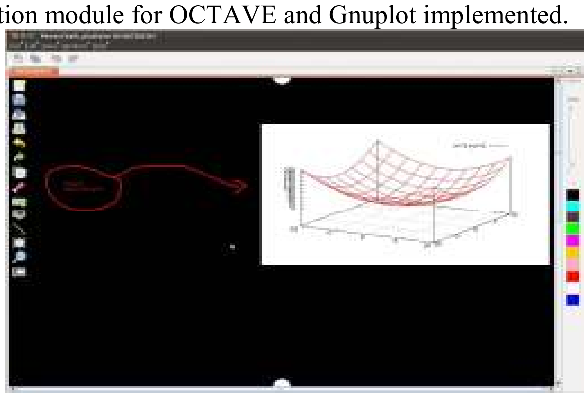

Figure 5: Example of Gnuplot collaboration module.

The call to Gnuplot, or another application is made trough a keyboard input or handwriting information in the PowerChalk canvas (Figure 5).

Magic Panels.

A magic panel module has an area on the PowerChalk canvas, where the users can embed their own objects, tools or applications. An example is the animation module.

An animation it is one of the best ways to express moving visual images because it can represent dynamic concepts and it can make information more attractive and engaging [24].In general, animation is a popular medium for entertainment, communication and therefore in education; however to make animations it is out of reach for common user. Most animation tools are complex and time consuming to learn and use.

To endow PowerChalk with an informal animation tool we merge the structured 2D graphics framework Piccolo2D [25] with an efficient lookup system. The design of the object structure with the lookup system in PowerChalk let us to implement commands to animate objects over the PowerChalk-canvas, for example, we can animate affine transformations over an object and composite a set of elements.

As a result, PowerChalk has an API for animations and the platform to developed. The user will be able to write his own magic panel. For example: A physics teacher, would like to explain the law of gravitation and Kepler's laws. He draws a circle (a solid mass) and a second circle (a second mass). The second mass receives an initial velocity (specified by a small vector). The magic panel recognizes the objects and starts a simulation of the motion of a planet around a solid mass. Since the gestures and object recognition are provided by PowerChalk, the user must just give the details of the simulation, and linked them to the objects. For this purpose PowerChalk provide an API.

Macros module.

We use the term Macro to make available to the user, a sequence of quick notes (notes written in a prior time) or information (images, text, videos or pdf files) to use in a class session. The user interface of PowerChalk allows having the information for a class, available in an organized and efficient window system (Figure 6).

Figure 6: PDF macro.

It should be noted that we are able to add to the macro notes or a page of notes written in a normal paper. This module can export a scanned file of the notes, binarize the image, make the segmentation of different annotations (if the annotations are splitted for a horizontal line) and make it available into the macro.

4. Conclusions and future work

PowerChalk preserves the pedagogical benefits of the traditional chalkboard and provides the possibility to show not only results or isolated ideas but the train of thoughts. Communication was established with some universities, companies, researchers and students for testing the capability to develop and perform teaching sessions, and also the usability and functionality of PowerChalk. The consensus is that PowerChalk it is a robust, reliable, usable and sustainable score learning platform and also a friendly tool to review and share lecture notes.

To support in a measureable way the advantages of PowerChalk, a complete usability test it is in progress. Also, we are increasing the efficient of every module. For example:

- We are developing new algorithms to improve the recognition rates in the handwriting recognition module.
- Improve the communication module. A disadvantage of the RTP approach detected is that the sender and receivers should not be behind firewalls because RTP is not allowed by most corporate firewalls. For this purpose a configuration protocol is needed.
- New tools for sketching are developed for a complete animation tool in the magic panel structure. The objective it is to provide a complete platform for quick sketching to the end-user.

The modular structure of PowerChalk let us to amend quickly any detail of the system since PowerChalk has the distributed development characteristic. Because of this advantage new modules are being developed. Among the future modules we can find: Voice recognizer module, image processing module and a java compile module, etc. The idea is to transform the PowerChalk system (e-learning system) in a complete platform to share and process general information.

6. References

[1] Dick, Walter, Lou Carey, and James O. Carey (2005) [1978].The Systematic Design of Instruction (6th Ed.). Allyn & Bacon. pp. 1–12. ISBN 0205412742.

[2] Twiner, Alison; Coffin, Caroline; Littleton, Karen and Whitelock, Denise (2010). Multimodality, orchestration and participation in the context of classroom use of the interactive whiteboard: a discussion. *Technology, Pedagogy and Education*, 19(2), pp. 211–223.

[3] Cutrim Schmid, Euline (2010). Developing competencies for using the interactive whiteboard to implement communicative language teaching in the English as a Foreign Language classroom. In *Technology, Pedagogy and Education*, Vol. 19, No. 2, pp. 159-172.

[4] Cutrim Schmid, Euline (2008). Potential pedagogical benefits and drawbacks of multimedia use in the English language classroom equipped with interactive whiteboard technology. In *Computers and Education. Vol. 51, no. 4, pp. 1553-1568.*

[5] Norman D (1993) Things that make us smart: defending human attributes in the age of the machine. Perseus Publishing, Cambridge, MA.
[6] Anonymous (2004) Mixed skies ahead: What happened to e-learning and why. The Learning Alliance for Higher Education. Change March-April 2004.
[7] Pei-Chen Sun, Hsing Kenny Cheng (2007). The design of instructional multimedia in e-Learning: A Media Richness Theory-based approach. Computers and Education, Volume 49, Issue 3, November 2007, Pages 662-676, ISSN 0360-1315.
[8] Richard C. Davis et al. (1999) Proceedings of the SIGCHI conference on Human factors in computing systems (1999).
[9] Alberto Turon Lanuza et al. (2008) REDALYC: Magazine Computación y Sistemas, Vol. 12, Num. 2, diciembre 2008, pp. 183-191.
[10] Richard C. Davis, B. Colwell et al.(2008) K-sketch: A “kinetic” Sketch Pad for Novice Animators. 2008.
[11] Agarawala, Anand. Ravin Balakrishnan (2006). Keepin' it Real: Pushing the Desktop Metaphor with Physics, Piles and the Pen. Proceedings of CHI 2006 - the ACM Conference on Human Factors in Computing Systems. pp. 1283–1292.
[12] Francois Guimbretiere (2003). Computer Human Interactions Letter. Volume 5, Issue 2. 2003 ACM pp. 51-60.
[13] http://smarttech.com/
[14] Kristian Jantz, Gerald Friedland, Raul Rojas (2007): Ubiquitous Pointing and Drawing, International Journal on Emerging Technologies for Learning (iJET), ISSN: 1863-0383, Wien, Austria 2007
[15] http://www.cabri.com
[16] David Jones and Shirley Gregor (2006). The formulation of an Information Systems Design Theory for E-Learning. In Proceedings of the First International Conference on Design Science Research in Information Systems and Technology 2006.
[17] Tim Boudreau, Jaroslav Tulach, Geertjan Wielenga(2006): Rich Client Programming, Plugging into the NetBeans Platform. Prentice Hall 2006. ISBN: 0-13-235480
[18] www.netbeans.org
[19] Gamma et al. (1994) Design patterns: Elements of reusable object oriented software. Addison-Wesley professional computer series.1994.
[20] Adler, M.J and van Doren, C. (1972) How to Read a Book. Simon and Schuster, New York, NY.
[21] Karwowski W, Soares M M, Stanton, N A. (2011) Human Factors and Ergonomics in Consumer Product Design: Methods and Techniques (Handbook of Human Factors in Consumer Product Design): Needs Analysis: Or, How Do You Capture, Represent, and Validate User Requirements in a Formal Manner/Notation before Design” (Chapter 26 by K Tara Smith) , CRC Press. 2011
[22] Dirk Bäumer, et al.User interface prototyping—concepts, tools, and experience. Proceedings of the 18th international conference on Software engineering. ICSE '96.
[23] Wobbrock,J.O., Wilson, A.D. and Yang Li (2007). Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. Proceedings of the ACM Symposyum on User Interface Software and Technology 2007.
[24] Anthoy, L. And Wobbrock, J.O. (2010). A lightweight multistroke recognizer for user interface prototypes. Proceedings of Graphics Interface 2010.
[25] Bederson, B., Grosjean, J. & Meyer,J. (2004) Toolkit Design for Interactive Structured Graphics. IEEE Transactions on Software Engineering 2004
[26] http://wicket.apache.org/
[27] http://www.w3.org/TR/ws-arch/
[28] Park, O. & Hopkins, R. (1993). Instructional Conditions for Using Dynamic Visual Displays: A Review. Instructional Science, 21, 427-448.
[29] http://www.piccolo2d.org