Game Wrench:
Design of a user-centered Interactive Game Engine

Eduardo Costa Jacober
Escola Politécnica da Universidade de São Paulo /
Centro Universitário Senac São Paulo

Romero Tori
Escola Politécnica da Universidade de São Paulo /
Centro Universitário Senac São Paulo

Abstract

Being a complex computer system, targeted for multidisciplinary users and multimedia tasks, although low-level programmed in nature, a game engine is a strong candidate to have its user interface designed according to CHI methodologies. This article describes the process of applying Interaction Design techniques to achieve an interface for a game engine targeted at typical Game Designers and other potential and actual users of these systems.

Keywords: Interaction Design, Game Engine, Persona Modeling, User Goals, Computer-Human Interface.

Authors' contact:
{eduardo.jacober,romero.tori}@poli.usp.br
{eduardo.cjacober,romero.tori}@sp.senac.br

1. Introduction

Throughout its relatively recent history, computer games have been one of the most challenging and innovative forms of software projects. Always breaking technology barriers, these game projects involve nowadays groups of multidisciplinary professionals, million-dollar budgets and very tight schedules.

A form of software reuse and code modularity adopted by this industry has been called game engine, where parts of the coding of a game is utilized as starting point for other game projects. Although it is a programmer's tool in nature, its main users – the game designers - are coming, each time more, from different areas and have other important competencies, such as arts, communication and design, but no computer engineering skills.

This document describes the process of applying the techniques from Interaction Design to broaden the user base of a game engine and help them reach their goals, by creating an interactive game engine.

The methodology is composed by known interface design techniques, such as Persona Modeling, Scenario Creation and Heuristic Inspection.

2. Related Work

Description and analysis of well-established computer game engines, their interaction styles and characteristics [Bianchini 2005];

Serious games and pedagogic aspects of games [Nakamura et al 2003b];

enJine: an open source, object-oriented, 3D game engine written in Java [Nakamura et al 2003a].

3. Concepts and Definitions

Game Engine: a group of software modules of a computer game that can be reused on other software projects (normally targeted for a class of computer games);

Computer Game: a game-product based on computer systems;

Game-product: A commercial product or activity that has game elements;

Game or Game state: A second order transgression upon reality (through play), providing special structure and rules to play state [Walther, 2003];

Play or Play state: A first transgression upon reality, freeing an object, person or place from its normal rules for a specific period of time [Walther, 2003];

Usability: Measure of the easiness provided by the interface of a product on fulfilling the needs of its users. This measure can be done with usability quantifiers, such as learnability, efficiency, memorability, error rate and satisfaction [Shneiderman and Plaisant 2005].

4. Problem

4.1 Target Audience

Until the middle of the nineties decade, hardware restrictions were the major concerns of game studios. Today, the industry tends toward other competencies such as design arts, plot writing, resource management, outsourcing, team motivation, etc. As a result, more and more professionals from other areas than the computer and technology fields are confronted to use this game development tools, and less and less do game developers have time (even if they like it) to learn a new set of functionalities that a modern game engine has to offer.

4.2 Increasing Complexity

With time, what once were hardware restrictions become affordable and common resources. Nonetheless, the gaming industry continues to base its production on constant innovation. This innovation comes either from new technological challenges (such as multi-player interactivity, physics simulation, artificial intelligence, etc) or from innovation on content (new interaction forms, twisting plots, open ended narratives, interactive cinematics, sound design, etc). Either way, this leads to one thing: higher complexity. What once was an activity well performed by a small group of hardware specialists, or even a multidisciplinary single individual, now needs many and diverse teams, on one or more companies in intense communication throughout the project.

5. Design Tasks and Artifacts

The interaction design tasks described here were applied to the matter of making a more interactive tool of a game engine. They are based on various methodologies for interaction design, favoring the ones that would consume less effort in terms of financial and time resources, such as the design process described in Cooper and Reimann [2003] and Jakob Nielsen´s *discount usability engineering* [1994].

The goal behind these design tasks is to achieve a list of requirements to optimize the programming phase. The first task – Identify the user universe – tries to avoid forgetting an important yet not obvious user, mapping all possibilities. With this, it is possible to make a better selection of target users. Next, these identified real users are mapped through models called personas, what will help in determining a set of user goals. Knowing user goals will help listing user tasks – good tasks will be the best steps towards their goals.

With all these design tasks done, some prototyping and evaluation can take place. To choose which set of tasks will be tested, a table of persona by task is filled with information that helps this decision. Then, a scenario is constructed including specific users, specific tasks and specific context.

All these steps are detailed in the following sections. In order to fit this format, only a description of the tasks will be presented, for illustration purposes only. The resulting set of specification listed on Section 6, can be useful to the programmer of a game engine that is either beginning a new project, or needs to implement an interface (game development tool) to the functionality of an existing one. The former is preferable and more unlikely to be the case (just a few present game companies devote resources on building a new game engine from scratch [Fristrom 2004]).

It is important to note that there are decisions made during this design process that would target the solution to a specific case of game engine. The researcher is invited to adapt the results to his/her special needs.

5.1 The Game Engine User Universe

This task uses a user classification in primary, secondary and tertiary levels of contact with the product, according to Holtzblatt, K.; Jones, S. (1993) *apud* [Preece et al 2002]. Primary users are the ones that have direct contact with the product. Normally, they are considered as the only users. Secondary users are either the ones that access it indirectly (through the primary users), or those that use it so sporadically that need to climb part of the learning curve each time. Tertiary users do not use the product. But still they are very important on design time: they are the ones that are affected by its use in some way, or they have decision power for its purchase or acceptance. This user universe for a game engine product can be very large. It was identified three different user bases, the corporative world, the serious games world and the amateur world.

5.1.1 *Corporative World (Game Studios)*

The first ambient to come to mind through the subject game engine is the Corporative World, since it is from where this tools have emerged. It is composed by the companies that assume game projects in a commercial approach and have been using game engines and a plethora of other tools to meet their goals on time and on budget.

In order to illustrate the user base for game engines in corporative world, chapter 5 of Bethke [2003] is a good starting point, providing a list of job positions in the game industry. Then, it is possible to associate the following characteristics with them:

- the type of company would have such a position;
- some competencies needed and desired for it;
- their classification towards the use of a game engine.

A detailed table resulted from this activity. Through this table it was possible to select the following group of users from the corporative world whom are most important for the intended game engine.

5.1.2 *Serious Games and Amateur/Hobbyist Worlds*

Once the game engines became commercial products, they were available to other groups of users than the ones in companies that produce game-products. Since these game engines, in the process of keeping pace with the top technologies computer games need, can resolve many problems other software applications strive to achieve, researchers are studying possibilities for their use outside the entertainment domain. This software that makes use of

game engines is called serious game. The first fields to benefit from this approach are arts, education, training and psychology. There is still another user base with specific characteristics, which are the ones that use a game engine for personal game projects – the amateur / hobbyist world. Information on the users from these non-commercial worlds is not as easily available.

For the interactive game engine, it was decided that the users from amateur world would not be important for development, but could still benefit from any advance in its usability. Though, it was also decided that the serious games world should be represented, so the ones with education needs were chosen for design purposes.

5.2 Persona Modeling

Personas are models that can represent human groups. They are an excellent design asset since they can avoid the unpredictability and imprecision that human behavior can present. For instance, it is easier to identify user goals from a persona than from a real group of people [Cooper and Reimann 2003]. All that is needed is a good, representative model.

For this task, it was used a collection of formal and informal methods, such as:

- Physical characteristics;
- Gender and age;
- Cognitive style[Riding and Rayner 1998];
- Basic archetype[Philips and Huntley 1996];
- Social and cultural characteristics;
- Skills and abilities.

The (type of user)-to-persona mapping do not have a one-to-one relationship. Characteristics of a specific type of user can be scattered among various personas and as much personas as needed to represent all types of users should be created. Once ready, these personas were used in order to identify the user goals. These goals can be classified as:

- **Life goals:** the personal aspirations of the user;
- **Experience goals:** how the user want to feel using the product;
- **End goals:** what the user wants to accomplish through the product;

5.3 Task vs. Persona Table

Having acknowledged the goals of the users, their tasks, the ones they actually achieve through the product, and where they spend their work and time resources, are the features the system should have and should be steps toward these goals. No identified goal should be left outside the tool (or, at least, the product must not go against them), and features that do not take the users toward their goals should be cut from the design (if it does not help, it will probably be getting in the way).

5.4 Scenario Description

Examining this table listing tasks, their criticality and frequency, it is straightforward to assign higher priority to most common and critical tasks for the development stage (scope). Scenarios are narratives that should combine a specific user (persona), performing a series of tasks with the product (the ones chosen above), in a specific context (period of time and environment information). They should describe the "best scenario", where none of the innumerous problems that could happen occur (which is alright, since they will appear further through evaluation).

5.5 Heuristic Evaluation

With the aid of a list of usability heuristics, a small group of evaluators (3 to 5 would be enough), in pairs, can perform the usability inspection [Nielsen 1994].

Assuming the role of the persona (which means, on each interaction step, thinking how that user would react to the system cues), the evaluator tries to detect all interaction problems that emerge. These problems then are associated with one or more of the heuristics it is infringing along with its criticality (how frequent, persistent and the impact a problem causes). The course of one scenario is performed at least two times by each evaluator. At last, the lists of problems of all evaluators should be compiled to one consensual list.

6. Game Wrench Specification

This list of requirements was compiled based on: user goals (through persona modeling); competitive analysis (through heuristic inspection); which were explained before, and were added to a set of (functional and non-functional) requirements already raised on other related works [Nakamura 2003a, 2003b].

6.1 User requirements:

- All range of programming knowledge (basic to expert);
- Corporative or serious games world;

6.2 Usability requirements:

- Priority on learnability (no need for external tutorials or manual) and efficiency (average user should create a simple game – such as pacman or tetris – in one programming session);
- Identify user's end goals through "wizards" and keep track of the project;
- Suggest choices among next steps towards user's goals;

6.3 Functional requirements:

- Scope-sensitive visualization "General / Producer", "Project / Game Designer" and "Task / Developer"
- multi-user support;

- Use the game definition as model for development;
- Control game elements and resources;
- Keep game logic and presentation separated, but linked and visible;
- Automatic documentation targeted to developer, producer and end-user (player), through "Wizards";
- Easy configuration of state machines, and scripts for game objects behavior;

6.4 Non-functional requirements:

- Integrate reliable auxiliary tools such as map design and 3D editing;

6.5 Data requirements:

- Auto-save user´s progress in projects;
- Use XML for storing game projects' data, in order to allow quick development of parsers to act between the framework and third-party engines and tools;

6.6 Environmental requirements:

- Standard computing working place.

7. Conclusion

A game engine is a complex tool for creating games, that has recently become available as a commercial product. Therefore, there is still a long way to the usability level where it will become accessible to users other than hardcore programmers. This study presents some first steps towards this goal.

The results, presented through this list of requirements, were only possible through the process of Interaction Design. It is very improbable that they could be detected or inferred just by speculation, imagination or common-sense. Some are novel within traditional game engines, and others definitively could not be defined by traditional requirement analysis, simply because of different focus of the methods.

The methodology described in this article is not definitive or fail-proof, but is rapid and effective, and could be applied similarly to other software products. It contributed greatly to avoid problems that would be detected in further stages of development - if so - saving time and resources.

Other Interaction Design tasks, such as: quantitative research through real-user interviewing, Thinking Aloud testing and creating interactive prototypes, will be executed in further iterations of development, when coding will already be occurring. This constant evaluation and feedback can improve the design and avoid detection of problems that need too much re-work and waste of resources. The set of requirements obtained in such way is being applied to enJine, an open-source Java game engine, developed by Interlab – Interactive Technology Laboratory.

References

BETHKE, E., 2003. *Game Development and Production*. Plano, Texas: Wordware Publishing, Inc.

BIANCHINI, R., 2005. *Uma Arquitetura BDI para Comportamentos Interativos de Agentes em Jogos Computacionais.* Doctor Thesis for Escola Politécnica da Universidade de São Paulo.

COOPER, A. AND REIMANN, R., 2003. *About Face 2.0: The essentials of interaction design*, Wiley Publishing, Inc

FRISTROM, J., 2004. *Manager in strange land: reuse and replace* [online] Available from: http://www.gamasutra.com/features/20040109/fristrom_01.shtml <free login required> [Accessed 10 June 2006]

NAKAMURA, R.; TORI, R.; BERNARDES, J.L.; BIANCHINI, R.; JACOBER, E.C. 2003a. A Practical Study on the Usage of a Commercial Game Engine for the Development of Educational Games In: *Proceedings of SBGames WJogos 2003.*

NAKAMURA, R.; TORI, R.; JACOBER, E.; BIANCHINI, R.; BERNARDES, J., 2003B. Development of a Game Engine using Java In: *Proceedings of SBGames WJogos 2003.*

NIELSEN, J., 1994. *Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier* [online] Available from: http://www.useit.com/papers/guerrilla_hci.html [Accessed 28 may 2006]

PHILIPS, M. AND HUNTLEY, C., 1996. *Dramatica: A New Theory of Story*, Third Edition, Screenplay Systems Incorporation

PREECE, J.; ROGERS, Y.; SHARP, H., 2002. *Interaction Design: beyond human-computer interaction*, IE-Wiley

RIDING, R. AND RAYNER, S., 1998. *Cognitive Styles and Learning Strategies – Understanding style differences in learning and behavior.* London: David Fulton Publisher.

SHNEIDERMAN, B. AND PLAISANT, C., 2005. *Designing the User Interface: Strategies for Effective Human-Computer Interaction* – Fourth Edition. Addison-Wesley.

WALTHER, B. K., 2003. Playing and Gaming: Reflections and clarifications. *In: Game Studies: the international journal of computer game research - volume 3, issue 1, May 2003.* [online] Available from: http://www.gamestudies.org/0301/walther/ [Accessed 28 May 2006]