

Case Study: Development of a Corporate Learning Game

Author:

Sharon L. Gander, Senior Learning Analyst, Cerner Corporation, Cerner Virtual University, 2800 Rockcreek Parkway, W0721 Kansas City, MO 64117 (phone: 816-201-2623), (fax: 816-201-8623)
sgander@cerner.com

Author Biography

Ms. Gander is a Senior Learning Analyst with Cerner Corporation where she was Project Manager and Learning Analyst for the *HNAM DataQuest* Learning Game Development Team. She has her M.Ed. from Montana State University, Bozeman, MT. Ms. Gander has over 20 years experience providing education to both adults and youth and specializes in the development of alternative learning practices such as games, goal-based scenarios, and group-process learning events.

Abstract

This paper provides a case study perspective on the development of a learning game for adults in a corporate environment. The game, Cerner Corporation's *HNAM DataQuest: The Millennium Architecture Knowledge Adventure*, teaches Cerner's specific information systems architecture to associates. The process of creating a game for educational rather than entertainment purposes is not well documented. This paper provides a window on one educational game's development process within the corporate education/training environment.

Educational game design and development projects like this one have different hurdles to manage than do entertainment games. Creating educational games within a corporate environment adds other challenges. Therefore, other corporate game development project managers may find this window on one educational game's project useful.

Background

In 1998, Cerner Virtual University, (CVU), developed a computer-based learning game to teach core processes and knowledge of Cerner's three-tiered client/server architecture and relational database. The resulting product, *HNAM DataQuest: The Millennium Architecture Knowledge Adventure* (a.k.a. *HNAM DataQuest*), was an experimental venture for CVU. At the same time, CVU was creating more traditional CBTs using standard project management methodologies. In many ways the game development project was a "skunk-works" type project.

Project Overview

The Vision

The vision was to develop a learning game for adults on the topic of Cerner's architecture. The game needed to be engaging and fun while educational. That is, players not only needed to learn Cerner's architecture, they were to have fun learning.

The Challenges

There were three major challenges in developing this game:

- (a) Cerner's architecture was thoroughly defined at a deep technical level and not at all documented at the intermediate level needed for content in the game. This meant that extensive translation was needed to move that knowledge from deep technical language and abstract knowledge stored in experts heads into the concrete language needed by those not familiar with the technology.
- (b) Cerner Virtual University's game development team was staffed with individuals who were committed to the concept of educational gaming but who had previously created a computer-based learning game.
- (c) The project was staffed as a skunk-works with most of the project's team members participating in multiple other projects as well as this one.

Project Champion

One of the first steps, as with any project, was to get one or more executive level champions for the project. The primary champion was Cerner's Vice President of Learning, Dr. Robert Campbell, E.D., who had the original vision of game on this topic for several years before it came into being. He not only provided the vision; he financed and staffed the game development team and guided educational decisions. In addition, he provided invaluable feedback embedded in an abiding commitment to the concept of a learning game throughout its development.

The second champion, Steve Oden, was a senior manager in engineering who provided subject matter experts as resources and whose commitment to the concept created a demand within the organization. He understood the game format and its value in creating learning for the intended audience. He provided a long-term stability throughout many changes and effectively represented potential audience. In addition, he often translated the deep technical knowledge into lay-language, which is more accessible to novices and more effective in gaming. His commitment to this learning product and to the skunk-works type project management process ensured that the needs of the audience were continually bonded into the game's content and processes while game development continued over an atypically long time period. These were invaluable assets throughout the design and development process.

The Game Project Team

Cerner Virtual University (CVU) started the project with a team of three – a multi-media programmer, an internal project manager/instructional designer and a multimedia/instructional design consultant. The multi-media programmer was the only full-time member of the team throughout the life of the project. The internal project manager/instructional designer was assigned half time to the project (and half time to several other projects). The consultant was also part-time.

There were five key subject matter experts. Subject matter experts had extensive workloads outside of this project. Their time commitment was less than two hours once a month. Due to work demands, it was not possible to bring together all subject matter experts in one room. Nor

was it possible to meet with any one person on successive days or even weeks. This 'hit or miss' relationship with these key actors impacted the project management process significantly. The knowledge needed had to be captured from them and translated into more concrete language. Their limited availability added several challenges for the project team. As a result, they mostly worked through the project manager/instructional designer and seldom participated in design discussions with the rest of the development team. However, their long standing patience and willingness to keep working on the project made it possible to complete the project.

As the project progressed and deliverables demonstrated that the concept could become reality, three more part-time specialists were added: a graphic artist/illustrator, part-time multi-media architect and a part-time writer. Like the rest of the team, each of these new team members was only partially assigned to this project while working on other major projects. The game development project eventually had six people working on it even though only one of them was assigned to the project full-time. However, each team member brought strong major personal commitments to the idea of gaming as a learning method. Without their commitments the game would not exist, as other key projects would simply have moved in and taken over their time. They deserve considerable applause for their commitment and dedication.

As may be expected under the circumstances, significant overtime was needed by all team members in order to make the game a reality. Even then, the project's deliverables were pushed back many times. In part this was due to the part-time nature of the team. Some of this was also due to the skunk-works nature of the project as well as the lack of experience the team had in developing gaming and equally to the low availability of subject matter experts. Timelines reported here are the actual timelines not the projected ones.

Educational Needs Assessment

Before starting the project, the project manager/instructional designer spent significant time developing contacts and subject matter experts. The project officially started only after it was determined that the content while available, was not clearly documented but mostly stored in subject matter experts' heads and that this was in fact the content needed by others throughout the organization.

The actual Needs Assessment was very primitive. The project manager/instructional designer asked individuals in assorted roles throughout the organization to 'describe the architecture' or 'explain the architecture.' Roles of individuals interviewed include programmers, instructional designers, managers, product specialists, certification analysts, sales people. Their answers were either a negative response (I can't) or a pat-answer that they could not elucidate further. **The need was very basic – to create a memorable and describable version architecture.**

Deliverables

Few deliverables actually met their projected timeline. Accuracy of time estimates continues to be a challenge for this team. Timelines reported here are the actual timelines not the projected ones.

However, unlike other kinds of projects at Cerner Virtual University, this game had no previous precedent within the organization and was kept relatively obscure within Cerner Virtual University during its development stages. In many ways it was a 'skunk-works' type

project. Therefore, sponsors were willing to reset timelines and continued to encourage refining and improving this product.

Project Process and Deliverables

Phase I: Development of the First Game

1. Content gathering and determination of whether there was a game – November 97–February 98
 - This project had been started at least twice previously. The difficulty in gathering information with sufficient breadth and depth had hindered previous development efforts. Previous efforts had included definitions of high-level concepts and high-level organizational visions of the abstract architecture. These had not provided sufficient breadth or variety to allow for a game format. Uncovering the process/flow of the conversation finally opened the design process providing sufficient breadth and depth for this format.
2. Instructional Design and Engineering/Functional Designs–completed March 98
 - Needs assessment and design documents were somewhat cursory. The need was clear and basic – new Cerner associates needed to understand Cerner's architecture and be able to articulate it. The available documentation did not clearly describe the architecture in lay language for the non-technical associates who are often the interface with clients. Describing and teaching Cerner's three-tiered client/server architecture and relational database became the key need.
 - The design documentation provided the basic educational descriptions of needs, outcomes, goals and objective and only very rudimentary descriptions of the projected game and game play. Textbooks on game development helped some but did not provide the experience to know what needed to be document and what did not. Therefore, the functional technical designs evolved through skunk-works style proof-of-concept and prototyping.
3. Determining Game Format–completed March 98.
 - The project sponsors recommended creating a drag and drop game similar to the children's game, The Incredible Machine, for at least one portion of the content. The team reviewed The Incredible Machine and considered several other options including quest-style adventure games. In the end, a combination of both a drag and drop game and a quest-style adventure game format were merged to create one game with several levels of play.
 - Defining the intended format allowed actual game development to begin. The actual decision was more 'hunch' and 'faith' and 'vision' than fact-based as there is little research on use of different game formats for educational purposes.
4. Paper-based prototype (or Proof of Concept)–completed April 98.
 - A paper-based prototype and usability test was used to define and test game playability – rules, images, some textual content, and some action. A side benefit of this

methodology was that it allowed Cerner Virtual University to be more confident that the end product would be both appropriate and useful. This confidence would be important during the coming months of development and revision.

- The key to the paper-based prototype was the involvement of an illustrator who created graphics that were both fun and informative. The paper-based prototype was a large white-board sized playing area on a magnetized surface. Each paper game piece could be moved around the playing area as though one were dragging and dropping it on a computer screen: it provided the look and feel of a drag-and-drop interface. Game pieces were small graphics (2–3 inches square) printed on a colour printer with small magnets attached. On the back of each game piece was additional support material in the form of textual messages about the role and function of the game pieces.
- During paper-based play, players were asked to think out loud in order to help the testers understand the player's unique perspectives and issues. Playing the game during the paper-based prototype phase involved picking up game pieces and placing them on the playing surface where they 'stuck'. If a player needed help, they could pretend to press the HELP button to receive additional hints. HELP was free-form discussion used to find out what kinds of questions people had about playing the game as well as what kinds of answers helped players move forward with the play.
- When the players were ready to run the game, they simulated pressing a run button and one of the testers played the role of computer moving graphical game pieces. If the game pieces were not correctly organized, the computer role-player would provide an error message explaining only that this was the wrong game piece for that location. Game players then had to correct that error and press run again to find out whether there were additional errors. It was clear even in the paper-based testing stage that with every error players learned more about architecture. It was also clear that most players had fun with this format.
- Some players worked hard to get the 'right' answer the first time and were very frustrated when they made any error; they became visibly more tense at each feedback. Others played the game by trial-and-error. They would make a reasonable guess about which piece to place next, press run and learn from the next error message – they became visibly more concentrated and relaxed with each feedback.
- Using a paper-based prototype was both an effective test of the game play and a learning experience for the game developers. In addition, there were several side benefits of the paper-based prototype. Having a paper-based prototype allowed us to:
 - (a) Show progress at a stage of development, which is otherwise extremely abstract.
 - (b) Test our ideas with subject matter experts, sponsors and a small group of volunteers without significant investment in the concept of a game as a learning tool.
 - (c) Generate enthusiasm and additional commitment for the project.
 - (d) Show subject matter experts where additional content needed to be developed.
 - (e) Define how the computer should interact with players.
 - (f) Define how different learning styles/approaches would need or want to use support materials and hints.

5. Computer-based prototype–June 98

- Once a paper-based prototype was available work began on a computer-based prototype. As the computer-based prototype emerged it became clear that the chosen development language/tool would significantly influence the resulting game. There was a hot debate around this issue. We changed languages mid-stream several times in order to find one that would provide quick development and a pool of programmers who could work with the language.
- We tried Java, VisualBasic and ToolBook. Each had advantages and disadvantages. Eventually, the multi-media programmers settled on ToolBook as the tool of choice. Once chosen, the team revisited game rules and imagery issues. Key issues of debate were:
 - (a) Did the images provide both tacit and explicit clues to support their function within the game's architecture? (Was it better to use images specific to computer hardware and software such as CDs and system towers or to use metaphorical images such those chosen around transporting the message? Was it necessary to follow a theme or could players handle working with unrelated images?)
 - (b) How many times would a player be required to play each level? Would 'fun' be enough to draw people back to play it over and over?
 - (c) How could we measure learning? Was it possible to get the 'right' answer without learning anything?
 - (d) How much text was needed? When and where was it needed? How obvious did the text need to be the player (e.g., should it appear automatically or require a mouse click, etc.)?
 - (e) Did we need to provide text explaining how to play or simply allow users to discover the rules of play as well as the educational content built into the game pieces and their relationships?
 - (f) How technical did the language need to be? Could we move non-technical players into understanding and comfort using technical terms without defining and using those terms?
 - (g) How easy or difficult should it be to get to supporting documentation such as a glossary, explanations of the technical process, etc.? How many steps might make it too difficult for the player to get supporting information? What automatic presentation of supporting information would be obnoxious to players?
- The 'play' that was tested in the prototype was not exactly how the game could be played in a computer version. Many discussions ensued as each element of play was hashed out and as each image was evaluated and justified or changed.
- In addition, as game play changed imagery changed. For example, at one point the game pieces for one level of the game were assigned 'in' and 'out' areas on the left and right of each game piece respectively. Over time, these areas moved up or down the game pieces due to programming issues or changes in the definition of how to play the game. In turn, as the design of the game pieces changed the rules of play changed. It

was an iterative change process fraught with strong emotion as everyone had their own perspective of what the players would need.

- In addition, there were many ideas that seemed reasonable in the paper-based format or document stages but were difficult to re-create in programming. The challenge here was to determine what concepts, interactions, educational theories, content, user interface requirements, and programming language/tools limitations were at stake and find ways to balance them and retain or improve the engaging, fun nature of the game while teaching certain content. For example, one portion of the play had a background of many lakes connected by rivers where the lakes represented data tables and rivers represented relationships between tables based on keys. This playing area was actually nine times larger than the visible playing area on screen. Considerations included how to make a canoe (the person 'fishing' for data) manoeuvre the rivers. The team considered issues like the use of a world map and close-up maps, how to provide an overview so that the player would know which lakes (tables) they might need and where currently outside their visible play areas, etc. It worked well in the paper-based prototype. However, computer prototyping demonstrated the severe limitation of this model on-screen. After much reworking, the resulting play area fit within one play area window, had a hydroplane fly to any lake on a double click, and provided detailed information about lakes on (single click) menus available for each lake.

6. Final game: Beta version – October 98

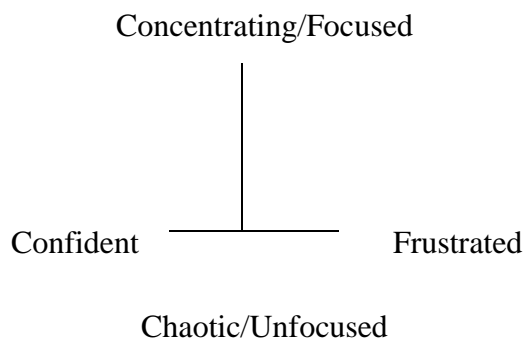
- Creating the final product meant coordinating images with action and text. Much of the text-based content development was from 'scratch', as it needed to be in lay language not technical language. Of course, this new content had to be validated by subject matter experts, revised, revalidated, etc. Since subject matter experts were difficult to reach, had competing demands on their time and, when available, found the translation of their technical expertise into lay language difficult, it meant that instructional designers wrote the first draft – and the second and third drafts until they got the translation correct. Then textual content was often revised and revalidated at least once more, as available space was not sufficient for the original content. As the project neared completion, the multimedia writer took over fine-tuning and finalizing the text to be sure that it both fun and understandable.
- During this stage, interface imagery changed almost daily as button designs, button placement, text placement, themes, borders, colors, and game piece images and relationships between game pieces and interface changed and changed again. At every iteration images, text and action were continually tested against usability.
- With strong owners of imagery, text, and action pitted against each other and against the usability and education theorists, this stage was fraught with tension. However, it was also filled with zest and enthusiasm for the emerging product. At each iteration, it was clear that the resulting game got better and better.
- As a team we also came to agreements over time on key theories which we felt needed to be consistently handled the same way throughout the game. The battle cry became, 'Is it fun? Will they come back and play it again and again?' We had finally decided that we could overcome the issues inherent in the massive, complex and difficult content if we could get players to come back and play the game at least three times.

- Usability and content testing continued throughout this phase. However, subject matter experts, learning specialists and sponsors were the major audience along with few new technical associates who had also participated in earlier alpha testing.
- However, finalizing the game did not end the development.

7. Testing–November/December 98

- Once the beta version was available, it became imperative to test both usability and learning. We wanted to know from the intended end-user audience whether or not the game (a) taught them what they needed to know, (b) was playable without classroom-based/instructor support and (c) was fun.
- In addition, since the game is a form of orchestrated immersion and since orchestrated immersion often overwhelms learners, it became important to define players learning and emotional states during play.
- Orchestrated immersion learning provides such massive content to learn and multiple paths through which to learn that most learners shift from modes they use in more traditional learning to deeper learning modes. However, a few individuals will ‘downshift’ during the early stages of orchestrated immersions. They freeze. If they have appropriate intervention, they can turn around the anxiety, fear and frustration and begin to learn. However, in a CD-based game, there may be no one available to intervene. This downshift and freeze may result in these individuals simply quitting and never returning. We needed to know what caused the downshifts, when and where in the game play.
- We had provided a semi-active agent who was intended act as coach and to prevent the downshift. However, the rules for this agent were limited. In addition, the programming necessary for its active (or rather semi-active) status had it behaving in a nearly passive mode. That is, the active agent popped up with some fairly generic text comments appearing in a small window and providing some coaching hints. There are three problems with the agent, which will need to be resolved in future productions:
 - (a) The text for coaching was not robust.
 - (b) The rules activating the agent are limited in scope.
 - (c) The agent does not intervene as much as it appears temporarily in a small window.
 - (d) The agent's behavior is mouse-sensitive – if the player is moving their mouse when the agent appears, it immediate disappears. This means that some players never see the agent.
- In order to find out how to make the active agent more effective in the orchestrated immersion-game format, we needed to find out what individuals did during play which would indicate that they were experiencing downshift or that they potentially heading for a downshift.
- The test process included a pre/post test and observation of players. In order to observe fifteen to twenty players, we needed to mobilize a team of observers. Observer training was provided.

- Our test audiences were new associates still in their first two weeks with Cerner. Observations were done in one large room with many players to one observer.
- Observers reported observed learning states and usability factors at five-minute intervals. Players self-reported their own learning state at twenty-minute intervals. Learning state was reported on a grid whose X axis was emotional state and Y axis was attention behaviors. The X axis ranged from frustration to confidence. The Y-axis ranged from chaotic/unfocused behavior to concentrating/ focused behavior.
- Learning State Observation Grid



- Results were mixed. Observers were inconsistent in their observations. Few observers managed to report 12 data points per player. However, some useful information did appear. Players seem to fluctuate learning states on 15–20 minute intervals and as they changed levels of play. The majority (80%) of players started out each level feeling somewhat to highly confident and somewhat to highly concentrating and focused. They became frustrated and more chaotic during the next 5–10 minutes. Most were able to turn that chaos and frustration around. A small percentage (15%) remained in that state of frustration – they got stuck in a downshift and could not get out.
8. Refined game based on test results: Full-release version – January 99
- Usability observation results identified areas for quick changes that could be made between a beta release and a full-release. These changes included a new strategy for a tutorial, fixing some bugs, providing the active agent with a few more hints, modifying the action slightly in the hopes of preventing downshift and frustration at one key point and fine-tuning some of the architecture for improved performance.
 - The first full-release version was provided to users in March of 1999. It is still being tested. Changes are planned for the next version.
 - Over 350 individuals are using the beta-release version of HNAM DataQuest with more about 700 more individuals waiting semi-patiently for the full-release to come through the CD product house's production cycle.

Phase II: What is next?

Complicating Factors

Many factors increased the inherent difficulty of the task. The innovativeness of the product is only the outward sign of many innovations in project management, development processes and teamwork. Everything from team resourcing to availability of subject matter experts to previous experience with gaming to choice of development language added complexity to this project. In addition to the traditional project management challenges, this project experienced some unusual challenges in deciding on the development language and in deciding on uses of imagery and text.

Development Language(s)

The choice of development languages also created challenges. The full-time multimedia programmer started with Java, which was a new language for him and for which he had no in-house support system. Eventually, the project moved to Visual Basic where the project consultant had expertise. In order to move to Visual Basic, the multimedia programmer learned a second new language. Development still was not as flexible as desired. Eventually the team moved to ToolBook where prototyping went faster and in-house expertise was available from a multimedia architect. However, this meant that the multimedia programmer learned a third language in order to move to ToolBook.

Imagery

Imagery was always a hotly contested topic. Everyone has their own way of perceiving reality and nothing was more controversial than the graphics of the game. The team acquired an illustrator who saved the day by producing fun images at an unbelievable speed. The game would not have had quite the same impact with clipart graphics.

Interface issues often ended-up in graphical redesign. Graphics were changing right up to the final days before full-release.

Text

Words move the tacit knowledge of imagery and game action back into the realm of explicit knowledge. Text was the under-valued aspect of educational gaming. Since there is little research available on the design aspects of education gaming, the team under estimated the value and importance of the text both for content and for game-playing guidance. The fun, precise and clear lay language describing each of the game piece and the matching fun, lay language videos were as vital to the overall impact of the game as was the game action.

Hints and tutorial were a hotly contested issue throughout the game design. Eventually we provided a combination of hints including suggestions of functions that should be found in the next game piece. The tutorial was re-written between beta release and the first full release.

Support information was of less interest even though it is necessary for some learners whose learning styles prefer text-based factual documentation. It had to be available even though the majority of game players may never find it much less read it. Support information included a glossary, FAQs, overviews of each level of play and access to video segments.

Phase II of the Project—What's next for the game development team

The beta-release and first full release of *HNAM DataQuest* have three levels of play embedded in three scenarios. In other words, the play is fairly limited. Additional levels of play need to be developed. A process for generating content and images has been drafted. The team needs to test that process. In addition, a scenario generation engine has been built which should increase the speed at which scenarios and levels of play can be built. In fact, instructional designers and technical writers can now do the majority of the work in designing scenarios. Programming will still be needed to integrate all the elements as well as to improve the quality of play. Programmers will be focusing on improving responsiveness, making the active agent more effective, building an external reference tool that allows content to be found outside of gaming, and building development engines to allow other instructional designers to develop more games in this format.

Summary

The *HNAM DataQuest* project was experimental in both its use of gaming for an educational methodology and its use of resources. The *HNAM DataQuest* game project required nearly one and a half years to complete due to constraints of part-time team members, limited access to subject matter experts, and team members who were new to computer-based game development.

Gaming as a learning delivery tool is not fully accepted as an adult education methodology, which contributed to the experimental nature to the project. However, the project was completed within a skunk-works type environment that allowed for experimentation and adjustments of timelines. Through the process of developing the game, team members learned a significant amount about the application of learning theories and their own personal learning experiences to a new methodology.

The *HNAM DataQuest* game is providing learning opportunities for nearly fifteen hundred Cerner associates. Testing proved that associates could develop the mental model needed to articulate Cerner's three-tiered client/server architecture and relational database.

Educational games can be effective learning tools. As their acceptance grows, project management methodologies can be fine-tuned and standardized. At this time education game projects in corporate education are the exception rather than the rule. Therefore, the structures and processes needed to work an educational game design project are often hit-and-miss. The *HNAM DataQuest* team used traditional project tracking tools and invented many communication tools. However, in the spirit of innovation and entrepreneurship, technical design documentation never did get created. Decisions about content, use of text, use of imagery, and use of development language were team decisions and, as with team decisions in inexperienced teams, took longer to finalize than they may have with a more experienced team.

As many corporations develop learning games for adults where the content and game methodology are complex enough to require complex project management methodology, educational game project managers and project teams will need to develop specific methodology for game projects. Specific methodology that includes design tools, technical documentation of functionality, and clarification decision processes will decrease the time required for game development.

References

Caine, R. M. and Caine, G. (1994). *Making Connections: Teaching and the Human Brain*. Alexandria, VA: Innovative Learning Publications, Addison-Wessely Publishing Company.

Dempsey, J.V., Lucassen, B. A., Haynes, L.L. and Casey, M.S., (1996). *Instructional Applications of Computer Games*, paper presented at 1996 annual meeting of the American Educational Research Associate, (ERIC Document Reproduction Service No. ED #394500.)

Dombrower, E. (1988). *Dombrower's Art of Interactive Entertainment Design*. City, State: Publishing company.

McMullen, D. (1987). Drills vs. Games - Any Differences? A Pilot Study. (ERIC Document Reproduction Service No. ED #335355.)

Merrill, M. D. (1999). Article Name. [On-Line] Available: <http://www.coe.usu.edu/it/id2>

Wolfe, J. (1997). The effectiveness of business games in strategic management course work. *Simulation and Gaming Journal*, v28 (4) p360-76.