EDC&I 510: History of Educational Technology Final Paper
Mark Chen
December 9, 2003

Histories of Computer Games and Educational Software

Introduction

I’ve been a “gamer” all my life. At first, in grade school, it was *Dungeons and Dragons* and a hex-based sci-fi war game called *Battledroids* (later renamed *Battletech*). Most of the time I played with my brother; organizing games with friends was difficult since we were kids and were never organized. Then one day in fifth or sixth grade, my dad brought home an IBM XT, a friend brought over some early games, and I was sold. When I went into high school, I discovered a new set of peers who were also into gaming. During lunch break, we would talk about the computer games we were playing at home, and on the weekends we would get together to play pencil and paper role-playing games or a board game called *Diplomacy*. What amazes me is how much time we spent *talking* about the games. We would share epic stories, ask for advice for areas or problems that had us stuck, and rant about poorly implemented or lacking features in the computer games we were playing. We would also spend a great deal of time talking about our weekend campaigns and would draw maps of the world we were exploring and sketches of the characters we played. For us, school outside of class was a place for us to meet and discuss our gaming life; our actual class time was a completely different world distanced from our lives outside of class. During this whole experience, my parents did not approve. They believed I spent too much time playing and not enough time studying. And so I’ve always felt a little guilty when playing these games.

When I entered college, I could see first-hand how dangerous game addiction can be. I had a group of gamer friends in freshman year, and we would go down to the computer labs and play all night, going to sleep when the sun came up and missing morning classes. For me, this happened once in a while; I was able to balance my gaming life with my schooling life (albeit rather poorly considering my grades that year). My two friends, on the other hand, dropped out of school by the time the year was over.

After I graduated, I was preparing myself for a career as a computer game artist when I was offered a job to create a virtual tour of the college for the admission department to give to incoming students. Thus began my multimedia/web career, and somewhere along the line, I became convinced that I would never work for a commercial entity and that I had a duty to my community. So, a few years later, I worked for a science museum, first on the website and then on exhibits software. The exhibit creation process includes an exhibit developer who is the primary investigator and does all the research into what science concepts would be appropriate for the visitors and works with educators and designers to figure out how these concepts should be presented. What I found most interesting was the fact that my gaming knowledge played one of the most important roles in designing the software for some of these exhibits (not all of the exhibits are software based). I found myself realizing that all the time I spent playing games was actually paying off and that my knowledge from my past experience was actually a valuable commodity. In fact, it was clear to me that without someone in the group with a history of playing games, the educational computer activities would have turned out very poorly. I was flabbergasted. I could not believe that my personal experience was helping the museum in such a way and kept thinking that there must be some legitimate academic research on games that we should be drawing upon rather than my anecdotal knowledge. My experience with the museum has completely changed my outlook on gaming, and I now no longer feel guilty whenever I play something; I'm just doing research!

What's the point in telling my personal history with games? My story, like many gamers' stories, is an example of cultural and social phenomena which, so far, have mostly been considered anecdotally and illuminates the following issues: A disparity exists between what is taught in the classroom vs. what children (and adults) are doing outside of the classroom. Gaming is a social thing; even when playing games alone, people tend to talk about their experiences with other people playing alone. Gaming addiction as a serious problem, but whether the addiction itself is the problem is another question.[1] There is also a large problem with some educational software: that sometimes it isn't designed by gamers or by people who know about games and that also sometimes it fails to draw the same kinds of interaction (and presence and passion) that games do. Perhaps we will be able to get a clearer idea of why educational software doesn't appeal to gamers if we examine the history of educational software in comparison with the history of computer games.

Definitions

The categories of educational software and computer games need to be defined first. I'm taking on a relatively narrow definition of educational software: stand-alone software created for standard personal computers for use either in the home or at school

[1] Unfortunately, covering some social aspects of gaming such as addiction, game violence, and gender bias in games is out of the scope of this paper which is too bad since I had to cut some juicy bits on those subjects…

for instructional purposes. Most of these software applications make use of game-like elements including imaginary or fantasy based narratives and the use of goals or levels of goals for the user to strive to achieve. This definition mostly includes things like *The Oregon Trail* and other single-player, story-based activities. An example of a non-narrative based activity is the familiar memory game where one has to flip cards over to make matching pairs. For the purposes of this paper, however, I'm only going to look at educational software which models itself after computer games. The distinction of the software being used on standard PCs is important only in that applications using specific hardware, like virtual reality goggles, do not qualify, at least for now; I am comparing software which home consumers are likely to use.

When talking about computer games it is important to note that I am not talking about video games. Generally speaking, a video game is an electronic program of entertainment which allows for user interaction via specific game controllers and which is presented on a video monitor. Today this usually manifests itself in the form of a game console (Microsoft's Xbox, Sony's PlayStation and PlayStation 2, and Nintendo's GameCube) which plugs into a television and uses hand-held controllers (gamepads) to move the on-screen action. Computer games, by contrast, are a form of electronic entertainment presented through a personal computer usually using standard PC controllers like the keyboard and mouse. The kinds of interaction found in video games and computer games are different due, at least in part, to the way people interact with the controllers. Other factors include the immediacy of a game console (no long boot-up time, for example). As a result there is a difference in the types of information presented visually and aurally. Video games tend to be more "twitch" and have simpler controls and game mechanics. Computer games tend to be more “strategy” and may feature very complex controls and complex game rules. Video games, however, also tend to have a lot more flash bang visual treatment, so there might not be a difference in the amount of information presented on-screen, just a difference in what is displayed and how.

I think, traditionally, academics and society at large have lumped video and computer games into one group. Even designers of computer games often describe themselves as “video game designers.” This might be valid when studying certain social aspects such as game addiction, but other social aspects cannot be looked at when the two are lumped together. For example, the debate on “video game violence” has actually traditionally been over computer game violence, and this becomes clear when one looks at the specific games which people have issues with; the ones cited have mostly been computer games (*Doom*, etc.). (More recently, however, correct terminology has been used to describe *Grand Theft Auto 3* as a video game.) The inappropriateness of treating the two platforms as the same is further made clear when one realizes that video game history is deeply rooted in Japanese culture while PC game history is distinctly Western. Furthermore, it is not valid to lump the two platforms together when one tries to look for aspects of games to use in educational software.

Generally speaking, educational software, by virtue of its medium, needs to be compared with computer game software, at least until educational software designers create products for video game consoles. Computer games and educational software, as defined here, both interact with the user via a standard PC with standard PC controllers, and it is in the computer gaming world where educators are likely to find the versatility needed to make educational software competitive (more on this later).

A History of Computer Games

Some students at MIT, known as “hackers” due to their constant tinkering of hardware and software, created a video/computer game, *Spacewar* (1962), using a PDP-1 computer from DEC.[2] Computers at that time did not have microprocessors; the brain *was* the computer itself. The PDP-1 had a video monitor and programming visual effects required actually controlling the beam of light hitting the screen; the programming itself was done on reams of paper tape. *Spacewar* featured two spaceships controlled by two players firing torpedoes at one another. Here we have the basis of what a game should be: it had a basic narrative (sci-fi fighting in space), it gave players control and immediate feedback (two make-shift “joysticks” were created for the game to control the PDP-1’s mechanical levers), and it challenged the players with a goal (kill the other player). Whether this was a computer game or a video game is debatable, as is whether it was the first (an earlier game, *Tennis for 2*, was made in 1958, using an oscilloscope and a custom-built circuit board), but the development of *Spacewar* and the environment which created it was definitely a strong influence for an aspect of computer culture and computer gaming culture—the free-spirited solo and small group endeavors to “hack” into a system and push the boundaries of what a computer could do while presenting an enjoyable game playing experience.

The next major milestone for computer games is a game called *Adventure*, first appearing in 1972. The originator, William Crowther, wrote it as a way to revisit the caves he and his ex-wife would go spelunking in, to connect with his children, who

[2] ClassicGaming (http://www.classicgaming.com/) is the source for most of this information on early computer games, especially their series of articles titled *The History of Computer Gaming*.

became disconnected when he and his ex-wife got a divorce, and to incorporate ideas from a new game called *Dungeons & Dragons*. The game featured caves, taken from Crowther's memory and careful cataloging, which the player moved through and explored. Although other "maze" type games like *Wumpus* existed, *Adventure* (also called "Advent" since that was the shortening of its name for computers to use) was the first game in the "interactive fiction" category, populated by non-player characters and puzzles the player negotiated with and overcame. The source code for Adventure was written in Fortran, making it platform independent, and was freely distributed via Arpanet, a precursor to the Internet. This caused it to spread quite quickly to all the other computers on Arpanet (only 23 at the time) and then spread to local computers at each node and so on. Since the source code (rather than a pre-compiled binary executable file more commonly found today) was distributed, people started to hack it and create many versions of *Adventure*. The most famous one, written by Don Woods in collaboration with Crowther, is *Adventure 2.0*, and it incorporated, for the first time, many fantasy-based rooms and creatures. The concepts of good game design are again apparent. *Adventure* presented an engaging world to explore, featured a natural language parser to give the player a sense of complete freedom of control, and set goals and puzzles for the player to achieve or overcome.

The first major example of using computers to help teaching and learning is the Automated Teaching Machine otherwise known as PLATO created by the Computer Education Research Laboratory (CERL) at the University of Illinois at Urbana-Champaign in the 1960s. Although it started out as a single-user system, it was conceived as a multi-user teaching and exploration environment, using a timesharing system where a mainframe computer could process many applications, handling many users at the same time via individual terminals.[3] PLATO went through several iterations and hundreds of programs were written for it covering math, science, and engineering to nursing, music, and foreign language acquisition,[4] and ranging from primary school to higher education and even workplace training. PLATO's multi-user architecture was also very conducive to gaming. In the early 1970s a succession of multiplayer games, all inspired by its predecessors, were introduced, the most famous being *Empire* which has evolved into the game *Netrek*.[5] Another game on the PLATO system was *Spasim* (1974 and originally meant to be pronounced "space sim", but instead was called "spasm" by all the players) which was the first "first-person shooter" game, being a 3-dimensional version of *Empire*. The code for this game was modified and led to the creation of the first flight simulator, *Airace* (short for "Air Race") which in turn inspired the *Microsoft Flight Simulator* series of games. Amazingly, the legacy of PLATO for gaming also had a major influence on "role-playing" games. If one takes the idea of *Adventure* and adds in number values, which may improve as the game plays on, to represent attributes of the character one is pretending to be then one has the makings of a role-playing game. By the late 1970s these dungeon-exploration games were being represented graphically from the first-person perspective on the PLATO system. The exemplar was a game called *Avatar* (various releases from 1979 to early 1980s) which featured a completely 3-dimensional first-person multi-user dungeon. This is astounding considering it took at

[3] This concept was pioneered by Project MAC (Multiple Access Computing *and* Machine Aided Cognition) in 1963 at MIT by the people of the same group who did *Spacewar*.

[4] One wonders if the foreign language teaching was effective since learning a foreign language traditionally requires social, verbal interaction..

[5] This happens to be the game I played in undergraduate school along with my to-be drop-out friends. It features a 2D space map using the Star Trek setting. Multiple players control starships and can shoot each other and drop armies down onto enemy planets. The goal is to take over the galaxy dropping armies down on each planet.

least another decade for a multiplayer game of similar sophistication to be introduced for the Internet or other networked play.[6]

In the non-PLATO world, games like *Adventure* continued to be developed. The most influential is probably *Zork* (1977) which also incorporated role-playing elements of character attributes and spawned a whole series of "text adventures." This genre would eventually see graphical representation in games such as *Wizardry: Proving Grounds of the Mad Overlord* (1981) and *Ultima* (also 1981), both to become successful series. The role-playing genre is best represented today with titles such as *Baldur's Gate* (1998), *Baldur's Gate II* (2000), and *Planescape: Torment* (1999)[7], all using the Infinity Engine, a 3D game engine, developed by Bioware, and the *Fallout* series (1997-1998). All of these games feature intuitive interaction schemes, very complex plots and goals interlaced with sub-plots and sub-goals, robust and consistent game rules, rich visual detail, and deep NPC interaction through dialog trees.

Another branch of the *Adventure* type of game emerged in the 1980s, too. These would be known (aptly) as "adventure" games and branched further into third-person graphical adventures and first-person adventures. The former are represented by the *King's Quest* series (1984-1998 Sierra On-Line) and the *Monkey Island* series (1990-2000 LucasArts)[8], though there are many, many good examples in this genre. The first-person type of adventure is probably best exemplified by the game *Myst* (1994 Cyan),

[6] Another astonishing concept is that many non-game applications created for PLATO were also extremely innovative, including precursors to Internet Relay Chat, a forum or discussion board, and a notes system.

[7] For an in-depth account on *Planescape: Torment*'s use of a dialog tree, read Charley Price's "Evolution of Interaction Schemes in Computer Role-Playing Games."

[8] Price also wrote an in-depth account of *The Secret of Monkey Island*'s verb-noun control scheme which was a direct graphics/mouse evolutionary step from preceding text adventure games.

which is a good example of how very attractive visuals can sometimes make a game successful.[9]

The two other major categories of computer games (role-playing and adventure being the first two) are "action" games and "strategy" games. During the 1990s, a plethora of innovative games caused these four categories to sub-divide so far that it seems almost all games of today are not easily categorized. In general, however, the action games mostly featured "first-person shooter" (FPS) games (*Doom*, *Half-Life*, etc.) and the strategy games were overrun by "real-time strategy" (RTS) games (*Command & Conquer*, *Starcraft*, etc.).

In the early to mid 1990s, through the very popular Bulletin Board Systems[10] and the growing Internet, independent game developers devised a new distribution and payment method for their games. Most mainstream games being sold at the time were distributed in electronic stores or video game stores like Radio Shack and Electronics Boutique. This distribution method was getting more and more costly and only viable for larger game publishers like Electronic Arts or Sierra On-Line. Small game development companies/publishers, most notably Apogee, began what is now known as "shareware." Software put in the public domain, free to anyone who wanted a copy, was known as "freeware." Shareware, on the other hand, was based on a sort of honor system where a user would pay if they continued to use the software. To enforce this, the common practice was to offer a stripped down version of the software. The common method for a

[9] *Myst* spawned a whole slew of first-person adventure games, the good ones refining the *Myst* experience and updating it with new technology (I believe *Myst* was a *Hypercard* stack, after all), the bad ones…well let me just say that there are definitely poor examples of computer adventure games.

[10] A "BBS" was basically a computer equipped with a modem ready to receive incoming calls and act as a discussion board and file repository for anyone who called in with another computer and had a valid username and password. It was also the distribution method of choice for software pirates.

game would be to split a game up into several episodes and only offer the first episode available for download. When one paid for the software, one was given access to the full version.

This distribution method, mostly available to teenagers on BBSs and college students at universities, proved to be very successful and cost effective for a company called id Software, who used it to sell over a million copies of their game *Doom* (1994). *Doom* is the most influential FPS ever, even though it was not the first. It amazed people at the time with a very convincing 3D maze-like environment and extremely fast, shooting action, transfixing and engaging many gamers into a Flow-like[11] state. Without innovative game-play and the means to distribute it, *Doom* would never have been as popular as it was. Every member of the development team became overnight millionaires. The success of *Doom* paved the way for a long succession of first-person shooters and the genre is still going strong today with the introduction of “stealth” shooters, WW2-themed shooters, and other refinements in computer games theory and practice on immersion, control, and story-telling.

Many other categories of computer games also emerged in the 1980s and 1990s. Here’s a summary[12] which attempts to detail the categories and cite early and late examples which define the genres:

- **flight simulators:** the *Microsoft Flight Simulator* series (1982-2003) and *IL-2 Sturmovik* (2001)

[11] For more information on Flow theory, read Csikszentmihalyi’s *Flow*.

[12] My god, have I actually played all of these? Yes, and many, many more. Also, I know I’ve left out many worthy additions, but each player has his own list. One notable omission: puzzle games, but only because I don’t have much knowledge of them…

- **computer role-playing games (CRPGs):** the *Ultima* series (1981-1999), the *Wizardry* series (1981-2001), *Wasteland* (1987), the Infinity Engine games (1998-2002), and the *Fallout* series (1997-1998)
- **adventure games:** the *King's Quest* series (1984-1998) and the *Monkey Island* series (1990-2000)
- **"rogue-like" games:** *Rogue* (1983) and *Nethack* (versions from 1985 to 2001)
- **action-adventure games:** *Castle Wolfenstein* (1984), *Dark Castle* (1987), *Starcontrol II* (1992 featuring the same kind of combat in *Spacewar!*), and *In Cold Blood* (2001)
- **sports games:** Epyx' *Games* series (1985-1990), *4-D Boxing* (1991), and *Tiger Woods PGA Tour 2003* (2002)
- **hex-based war games:** *Romance of the Three Kingdoms* (1989) and the *Steel Panthers* series (1995-2000)
- **tactical real-time war games:** *The Ancient Art of War* (1984), the *Myth* series (1997-2001), and the *Total War* series (2000-2003)
- **"god" games:** *Populous* (1989) and *Black and White* (2001)
- **"sim" games:** *SimCity* (1989) and *The Sims* (2000)
- **"4X" (eXplore, eXpand, eXploit, eXterminate) turn-based games:** the *Civilization* series (1991-2003) and *Stars!* (1996)
- **real-time strategy (RTS) games (also sometimes known as 4X):** *Dune 2* (1992), the *Command & Conquer* series (1995-2003), and the *Warcraft* series (1994-2003)
- **turn-based tactical squad games:** *X-Com: UFO Defense* (1994), *Jagged Alliance* (1994), and *Silent Storm* (2003)
- **"side-scrollers":** *Jumpman* (1983), the *Commander Keen* series (1990-1991), and *Jazz Jackrabbit* (1994)
- **space simulators and other simulation games:** the *Mechwarrior* series (1989-2002), the *Wing Commander* series (1990-1995) and *Freelancer* (2003)
- **first-person shooters (FPSs):** *Wolfenstein 3-D* (1992), the *Doom* series (1994-2004), and the *Half-Life* series (1998-2004)

- **third-person shooters:** the *Tomb Raider* series (1996-2003)
- **stealth shooters:** *Thief* (1998), the *Hitman* series (2000-2002), and *Splinter Cell* (2003)
- **action-role-playing games:** the *Diablo* series (1998-2003) and *Dungeon Siege* (2002-2003)
- **first-person role-playing shooters:** *Ultima Underworld* (1992), the *Gothic* series (2001-2003), and *Morrowind* (2002)
- **racing games:** *Grand Prix Circuit* (1988) and the *Need For Speed* series (1995-2003)
- **massively multiplayer on-line games:** various MUDs (multi-user dungeons—text-based games like *Zork* except with lots of players) many of which are still played today, *Ultima Online* (1997) and *EverQuest* (1999)
- **and a multitude of mixed genres!** *Archon* (1983), *Darklands* (1992), *Grand Theft Auto 3* (2002), etc.

All of these games have things in common which make them great engaging experiences. They all feature clear goals and consistent game rules, they all frame the main game-playing activity around a story, and they all present intuitive game controls, giving the impression that the player has control and self-determination over the outcome of the story without breaking the immersion. There is some overlap here with console video games, but, in general, computer games present a much richer gamut of games, featuring more variations in control schemes and more complex game mechanics.

A Brief Analytical History of Game-Like Educational Software

The first cases I've found[13] of software created for personal computers with the expressed intent for education appear in the early 1980s. Titles such as *Rocky's Boots* (1982, The Learning Company), *Granny's Garden* (1983, Mike Matson), *Robot Odyssey* (1984, The Learning Company), *The Oregon Trail* (1985 Minnesota Educational Computing Consortium (MECC)), and *Where in the World is Carmen Sandiego* (1985 Brøderbund) all fall under this period. What is unique about this period is that most of the educational software titles were created by educators who at the time had the leeway to explore innovations in teaching. For examples: The Learning Company[14] was founded by Warren Robinett (now working on VR at the University of North Carolina) and educators Ann McCormick Piestrup, Leslie Grimm, and Teri Perl. Mike Matson was (possibly still is) a teacher in the UK and was in contact with other teachers working on other educational titles under the organization Sherston Software. MECC was created by teachers with federal grant money to supply classrooms with educational software on the Apple II, and it later became a business entity.

These teachers/computer hobbyists knew first-hand what their classrooms were about and designed what they felt would be within their students' capabilities and what would help supplement, replace, or rework their classroom activities. I think they caught on early that computer activities could be used to interact with students' imagination.

[13] It is relatively difficult to find papers and articles documenting the history of educational software. Most of my searches for "educational software history" returned results about educational software which featured historical events as its content. Instead, I've looked up individual titles and years. This is what I've been able to piece together.

[14] The Home of the Underdogs (http://www.the-underdogs.org/) has this to say about The Learning Company: "The Learning Company's hallmark is their knack in combining fast-paced platform action with logic, trivia, and mechanical puzzles to create addictive edutainment for kids of all ages. User-friendly interface and comprehensive help files round off these classics." The word "platform" in this case means a category of game similar to *Super Mario Brothers*, which (frustratingly) features shelves or platforms to jump on in order to reach goals. It does not refer to different kinds of hardware, e.g. game consoles vs. PCs vs. Macs.

Imaginary worlds with specific rules could be explored in an involving way, one in which the students' would learn the rules and then create their own narratives within the rules. Also, during this time, it was easy for teachers to create activities for the computer which were visually and aurally compelling when compared to computer games of the same time. The PC was relatively limited in what it could do and games and educational software were pretty much equal by this limitation and by the novelty of computer use. By the late 1980s and into the 1990s parents and the government bought into the idea that just having computers in the classroom was enough to give children the leg-up needed to be effective students, and there was a renewed focus that teachers needed to use a standardized curriculum measured by standardized testing. Teachers had to change their teaching styles, some for the better, but others for the worse, and experimenting with new ideas for teaching was put aside for lack of time and no guarantees of effectiveness. Educational software development migrated to corporations who sold their products as a valid business model. Some game companies, like Sierra On-Line, Inc., decided to take on the development of educational software. They produced such titles as *Mixed Up Mother Goose* (multiple versions: 1987-1995) and the *Dr. Brain* series of games (1991-1998). Meanwhile, some existing teacher-based companies were able to adapt and grow into relatively large commercial businesses filled with "industry" people, modeling themselves after traditional commercial software companies. Other teacher-based organizations could not compete, and the 1990s can be seen as a series of smaller software houses being bought out or absorbed by larger more business-like corporations. The Learning Company, for example[15], acquired MECC in the late 1990s. Then The

[15] The history of The Learning Company is put together using accounts from The Home of the Underdogs (http://www.the-underdogs.org/) and Moby Games (http://www.mobygames.com).

Learning Company was bought by Brøderbund in 1998 and became its Education Division. Mattel then bought Brøderbund in 1999 and then, after financial failure, sold it in 2000 to Gores Technology Group. In 2001, Riverdeep Interactive Learning acquired the Education Division of Brøderbund (The Learning Company), and at least one employee[16] still feels like it is going through a state of turmoil. In 2002, Riverdeep Interactive Learning acquired the rest of Brøderbund (its Entertainment Division). The software developers in these new business entities were not necessarily trained as educators nor were they necessarily professional game designers. Many of the programmers came from other software industries while the artists and musicians may have come from other non-software related markets. Educational divisions of larger companies sometimes did not tap into game designers in the companies' own entertainment divisions. What this means is that sometimes an educator helped design the software, and sometimes a game designer helped design the software, but very rarely was there both working together, and often there was neither.

It was also during the 1990s when it became clear that an educator working on software was not enough to create something which would be compelling when compared to computer games. The speed and capabilities of computers dwarfed the ability of single people or small groups of people to create something which would take advantage of the technology. This is very apparent if you look at the end credits of applications created in the early 1990s compared to the end credits of applications created in the late 1990s or now. The sheer number of people needed to produce a visually and aurally impressive, or even just a competitive, title now is staggering. Another problem was the fact that these big budget projects had to be justified as valid business models,

[16] My wife's brother.

and typically, an educational title was simply not allotted the same budget amount as a game, which was perceived as a better investment. Ironically, I think a lot of effort was put into making educational software titles competitive in terms of multimedia that companies were forgetting to include sound educational practices. In other words, it was important to have an educator on the design team to ensure educational integrity *and* a game designer on the design team to ensure a use of intuitive controls and sound game design, and, as already pointed out, this was rarely the case, let alone, *teams* of educators and game designers.

The question arises as to how educational software survived, and the answer is in the market. The big money for educational software was in schools, not home consumers. This allowed some lee-way; the graphics and gameplay did not have to be on-par with current games. Instead, graphics for educational software simply needed to be more sophisticated than *previous* educational software titles. Games, on the other hand, were continually pushing the boundaries of hardware capabilities and presenting cutting-edge graphics and gameplay all the time.

Today, with the relative ease in creating material delivered via the World Wide Web, we see educational software being created by educators again. The problem, however, lies in the fact that while it may be easy to produce something for the Web, it's still difficult to produce something *well*. Additionally, non-educators (graduate students, the lay public, etc.) are also producing material for educational purposes, and most of these products are not compelling in many ways. A good example of having no game designer on the development team (or at least appearing to have no game designer) is the educational activity *Agent Guy Simplant* (http://www.guysimplant.org/) created by the

University of Washington Engineered Biomaterials (UWEB) Engineering Research Center and the Hope Heart Institute in 2003. Here's an example which features a clunky interface, hardly more than a glorified Powerpoint slideshow. The difference is that each "slide" features some animation. I do not think animation alone will capture students' attention, let alone transfer knowledge. A game designer could easily point out that interactivity is needed to give players a sense of involvement and goals are needed to motivate players to use their involvement in a directed fashion.

Sometimes, an educational software title does have sound educational backing and does have game designers on the development team, but, as I mentioned before, it may lack enough funding or resources to make the activity successful. An example is *The Clue Finders' 4th Grade Adventures* (1999 The Learning Company). The off-the-shelf CD-ROM title tries to directly compete with computer games of the time, presenting itself as an adventure game. Four (very politically correct) kids, their dog, and a strange, floating, portable, anthropomorphic computer-thing are solving a mystery in modern day Egypt concerning an archeological theft. To get help they must go on a quest to collect scarabs which they can then trade in to an informant/guide/mystic. To collect the scarabs, the player directs the gang to different shops and cafes within (I assume) Cairo where they must solve word and math problems before the (Western-looking and English-speaking) proprietors of the shops will give them their scarabs. The story is actually quite compelling and I found myself drawn in and could see how children would find this game engaging. Unfortunately, I could also see the shortcomings of a small budget which manifested itself in clunky, stop-and-go animation and periods of unresponsiveness. I am not sure children would sit through some of the more stuttering, delayed parts of the game. Additionally, game design has moved away from long drawn-out animation sequences with no player interaction which *The Clue Finders' 4th Grade Adventures* has in abundance[17]. Whether this is just a sign of it being made four years ago is debatable. I think if we look at the general range of educational software titles, we'll find that most of them trail behind successful commercial games by several years.

Overlapping Genres/Purposes and Convergence

Some educational software is blatantly uninfluenced by games; one can tell by their lack of interaction or immersion. Sometimes influences from a game designer would be the same as influences from an educator. Both might place emphasis on interactivity within a story-based world, for example. The educator would deem it important for constructivist and contextual learning. The game designer would intuit its importance for making a game fun. There are, however, some educational software titles that definitely cross over to games and vice versa. *The Oregon Trail* in its latest incarnation features many game-like settings, for example. Also, games like *The Incredible Machine* series[18] (1993-2001) and *SimCity* can be used and *are* used in schools.

By targeting schools, educational software publishers have defined a genre limited to children, when in fact they may have an enormous market in creating lifetime

[17] According to Witmer and Singer, interaction/involvement as well as immersion are necessary conditions for presence within a virtual world, and I believe this concept can be and has been (though maybe unconsciously) extended to gaming environments.

[18] *The Incredible Machine* challenges the player to construct Rube Goldberg contraptions out of a set of items to meet certain goals (get the basketball in the hoop, get the mouse some cheese, etc.). Each puzzle becomes successively more difficult, making the game continuously present new challenges to overcome. The game covers concepts such as gravity, mass, heat, wind, pulleys, levers, gears, switches, momentum, and elasticity using common everyday objects and the kinds of relationships they can have with each other.

learning activities targeted for home use and tap into the adult audience. Unfortunately, there’s something about software created for education which is qualitatively different than software created for entertainment. Children can pick up on this with great savvy, and furthermore, the majority of game players, *adults*, can also see this difference. Perhaps, something created for education intrinsically has a transparent agenda, whereas something for entertainment is not so overt. It may be better to focus on creating an entertaining experience and making sure it is founded on solid educational learning theory, only adding specific educational content when possible. In other words, educational software for the home needs to separate itself from software for schools and transform itself from an awkward subset of computer games into a pervasive underlying feature of computer games.

These new games would feature very rich worlds with consistent rules and allow players to explore the worlds and their rules with a fine level of control. Whether the world is modeled after “real-life” natural laws depends on the nature of the game and what kinds of learning experiences are desired. A war-time first-person shooter which tries to simulate real-world physics, for example, could impart some knowledge on momentum and stopping power, whereas a humorous one could take place on a fantasy world where our laws do not apply. What’s important is to define the world’s laws or game rules and then keep them constant, as the name “laws” implies.

Potential of Multiplayer Games

Even with constant rules in place, computer games might be used to greater effect by emphasizing non-curricular kinds of learning. The rise of multiplayer games, for example, has created environments where players have found the need to establish social rules of etiquette and structure. Often, more experienced gamers help new players (newbies) in a sort of expert-beginner relationship.

There are two main types of multiplayer games, ones that are limited to a few players and ones that are open to thousands of players at the same time, also known as “massively multiplayer on-line games” (MMOGs, the most common being MMORPGs). The smaller ones were made popular by *Doom* and are usually action games featuring some sort of “deathmatch” play where every player is out to kill the others. Sometimes these games have cooperative modes of play; either players go through the single-player gaming experience cooperatively together or they team up in special multiplayer maps against other players or computer controlled AI. Cooperative modes are actually my favorite kinds of gameplay. When done right, it allows specialization in player roles (sniper, machine-gunner, driver, etc.) and encourages players to talk to each other to overcome obstacles and solve problems. These smaller multiplayer games tend to limit the number of players, with a maximum number being anywhere from 4 to 64 and are common for LAN (local area networks) parties.

MMOGs, by contrast, can have thousands of players on at the same time and offer persistent worlds which players can log in to and out of at any time. The constancy of these virtual worlds has allowed many aspects of our real world to emerge and take form. Many of these worlds have virtual economies, for example, which allow players to trade, buy, and sell goods, property, and services. Social structures and protocols are even more defined in these massive games; some allow players to direct their characters to perform gestures and emotions to help with the communication already supported by built-in chat systems. The role multiplayer games might have in conveying knowledge about societal norms and morals may be even greater than its potential to present an environment where players can learn about natural laws.

Computer Games for Learning

As computer games become more and more pervasive in our society, it is important to realize generations of people are becoming more and more *digitally* literate and less traditionally so. There's now a disparity between what goes on in the classroom and what children are doing in real life, down to a disparity between the educational software in classrooms and what children are playing at home. It is important when creating a game *or* an educational software title to capture and keep the attention of its players. By modeling educational software after computer games, we may be able to leverage the captured-audience effect and teach something to the captured, but only if the educational software is able to compete graphically and, more importantly, interactively and immersively as good computer games. Otherwise, it may be more efficient to change gaming culture to take into account educational theories while designing games. Another possibility is to take successful computer games and modify them in such a way so that the entertainment qualities are not replaced but, instead, supplemented by educational qualities.

An extremely important and still-emerging aspect of computer games, one that is unique to computer games as opposed to video games and one that also has its roots in the first-person shooter *Doom*, is the "modding" community. *Doom* used a file format for its episodes which was made open to the public, allowing anyone with the right tools to make new episodes and levels for the game. In other words, players could *modify* the game. Each game that came from id Software since then included easier and easier ways for hackers to modify, or make "mods," for the game. *Half-Life* (1998 Valve), one of the first FPSs emphasizing story-telling, seemed to do the "right thing" at the "right time," and the modding community grew tremendously with hundreds, if not thousands, of mods available for the game, ranging from tweaks to weapon characteristics to "total conversions" using the same game engine to present a completely different game world with new art and a new story. Computer game players of today now have the option to be computer game *designers*. This phenomenon has spread to many categories of computer games such as role-playing games (*Dungeon Siege* and *Morrowind*) and turn-based tactical squad games (*Silent Storm*).

Modding a known successful game might be a very effective way for educational theories to be incorporated with the features of good gameplay. There would be some limitations to the tools you could give the players but the content would be easily modified to include more educational topics. This would be well suited for single-player games, but may also work for multiplayer games, adding a social dynamic to the mix. The idea I used earlier in this paper about using a first-person shooter to teach concepts of momentum might be more effective if one were to take an existing game with good physics modeling and create a new map or module for it featuring a test firing range with various firearms available for use. Each weapon would have different stopping power and accuracy based on the mass of the bullet, the speed at which it leaves the barrel, and the length of the barrel. Other ideas include using a racing game to demonstrate momentum and forensics through the analyses of traffic collisions. One could also demonstrate car racing physics (engine balance, gear ratios, etc.) by letting a player experiment with tweaking different aspects of his or her car (adding and removing spoilers, using softer tires, etc.) one at a time, possibly exaggerated to make the effects clear.

Some situations in the classroom present a great opportunity for the use of games. Using games to simulate patients in a health clinic is a good example. A specialized field such as this might call for software made specifically to address its issues. There is still value in making specific educational software titles. Outside of the classroom, however, people want to be entertained first and foremost. For these cases, my proposals are that game designers make their games more educational by just being aware of educational theories or having an educator on staff (we don't *have* to put in blatant educational content if we're interested in learning of social interaction skills) and that modifying current games may be a good way to encourage learning.

References

Anderson, Marshall. (1999). Re: Mission Statement. *Simple End User Linux/edu (SEUL/edu) listserv*. seul-edu@seul.org. Found on-line at http://archives.seul.org/seul/edu/Aug-1999/msg00030.html (December 2003).

The author presents a good history of educational software in the United Kingdom, starting with titles like *Granny's Garden* and briefly describes the trends of standardized curriculum, etc.

Goldberg, Marty. (2001). “The History of Computer Gaming.” *ClassicGaming.com.* Found on-line at http://www.classicgaming.com/features/articles/ (December 2003).

This is a very informative history of the early beginnings of computer games, keeping particular attention to the difference between computer and video games.

Gustavson, Stefan. (2001). “Multi-player 3-D graphics computer games: history, state of the art and development trends.” Found on-line at http://ttt.teleco.upv.es/~pabgarqu/prog/Generic/Multiplayer%203D%20Games%20History.pdf (December 2003).

The title is a slight misnomer as this paper only covers first-person shooters. Gustavson does a good job of showing the importance of *Doom* to the multiplayer and modding communities.

ClassicGaming.com. http://www.classicgaming.com/

The Home of the Underdogs. http://www.the-underdogs.org/

MobyGames. http://www.mobygames.com/

These three sites are full of information on older, sometimes forgotten, games including educational titles.

Price, Charley. (2003). “Evolution of Interaction Schemes in Computer Role-Playing Games.” Undergraduate thesis, Carnegie Mellon University. *Gamasutra*. Found on-line at http://www.gamasutra.com/education/theses/20030929/price_01.shtml (December 2003).

Price does a good job of describing key examples of good interface design for RPGs as a means to transfer the paper and pencil experience, but he is focused solely on single-player games and doesn’t address the social aspects of traditional non-computer RPGs which typically are played by a *group*.

Squire, Kurt. “Video Games in Education.” To appear in the *International Journal of Simulations and Gaming*. Found on-line at MIT’s Games-to-Teach site at http://cms.mit.edu/games/education/research.html (December 2003).

The author does a good job of citing video games for educational purposes and covers theories on engagement. Included are the choice, fantasy, and challenge ideas of Cordova and Lepper (1996) and Malone's ideas on the definition of a good game (1981): clear meaningful goals, multiple goal structures with feedback on progress, multiple difficulty levels, random elements of surprise, and emotionally appealing fantasy that is related to game skills. Squire does not specifically deal with computer games as separate from video games.

Witmer, Bob G. and Singer, Michael J. (1998). "Measuring Presence in Virtual Environments: A Presence Questionnaire." *Presence*, *Vol. 7, No. 3*, 225-240.

The authors define presence as a result of the combination of immersion and involvement.

Suggested Further Reading

Appelcline, Shannon. (2003). "A Brief History of Role-Playing." *Trials, Triumphs, and Trivialities.* Found on-line at http://www.skotos.net/articles/TTnT_131.shtml (December 2003).

Csikszentmihalyi, Mihaly. (1991). *Flow.* New York: HarperCollins Publishers.

Herman, Leonard, Howitz, Jer, Kent, Steve, and Miller, Skyler. (2001?). "The History of Video Games." *Gamespot.* Found on-line at http://www.gamespot.com/gamespot/features/video/hov/ (December 2003).

Games

Adventure. http://www.rickadams.org/adventure/

The Clue Finders' 4th Grade Adventures. http://www.cluefinders.com/

Doom. http://www.doomworld.com/classicdoom/info/

Half-Life. http://www.planethalflife.com/half-life/

The Incredible Machine. http://www.the-underdogs.org/game.php?gameid=537

The *Monkey Island* series. http://www.scummbar.com/

The Oregon Trail. http://www.classicgaming.com/rotw/otrail.shtml

Planescape: Torment. http://www.mobygames.com/game/sheet/gameId,648/