Investigating the Use of Computer Games and Virtual Worlds for Decentralized Command and Control

Final Report
Grant #N00244-10-1-0064
from the Center for the Edge, Naval Postgraduate School

***** DRAFT for review and comment *****

Walt Scacchi, Craig Brown, and Kari Nies
Institute for Software Research
and
Center for Computer Games and Virtual Worlds
University of California, Irvine
Irvine, CA 92697-3455 USA
wscacchi@ics.uci.edu

Final version Due: 30 June 2011

Current version: 1 July 2011

Executive Summary

This report serves to document results from a one year research study (July 2010-June 2011) that investigates how computer games and virtual world (CGVW) concepts, techniques, and tools can be employed to create an online virtual world (VW) that supports experiments in decentralized command and control (DCC). We refer to this project and the CGVW we have prototyped collectively as the DECENT project and system platform. DECENT is a platform for exercising and assessing the potential of a VW-based approach to decentralized C2, as well as to compare our efforts with others closely related. Overall, we find this effort gives rise to very promising results that point to additional opportunities and system extensions for new ways to consider the potential of decentralized approaches to C2 that merit further systematic investigation and experimentation.

DCC is envisioned as a new approach and model for how to organize and experience command and control systems, mission planning and scheduling processes, and physically decentralized user practices using low-cost or free (no-cost) software technologies. Our choice to employ CGVW technologies is in part influenced by the growing pervasiveness of such technologies, their availability as open source software and/or in user modifiable forms, and their widespread use by a new generation of online computer users who may see/anticipate that such technologies will become ubiquitous in future enterprise settings.

Next, our interest in not to simply replicate or mirror existing C2 systems, nor their traditional patterns of usage. Such usage generally assumes both centralized, hierarchical organizational authority and centralized location of users. Instead, our interest is to explore the alternative space where decentralized approaches to organizational decision-making and workplace location (e.g., top-down but physically dispersed versus peer-to-peer and physically dispersed) may be subject to experimental variance and study.

We also find that a number of computer games also suggest system features applicable to C2, as well as distributed individual/group problem-solving, using popular card games. Again, these may be candidates for further systematic study that can shape how next-generation C2 systems can be designed to support/embrace future workforce interests, skills, and capabilities.

We similarly identify and compare a small set of related technologies that could be compared to the efficacy of the VW technologies that we employ (OpenSim, an open source software toolkit for building, navigating, and socially interacting). OpenSim

provides many interesting affordances, some which are common to most VWs. But it is these affordances that merit further study. Understanding the potential for how VWs may be designed, built, deployed, and evolved seems to be a significant opportunity area for further study. In addition, there is still need to determine how best to evaluate and compare the efficacy of VWs that seek to mirror physical sites or physically-located human problem solving and social interaction. There is also need to evaluate and compare the efficacy of alternative VW and computer game development technologies, whether open source software, or proprietary, closed source software.

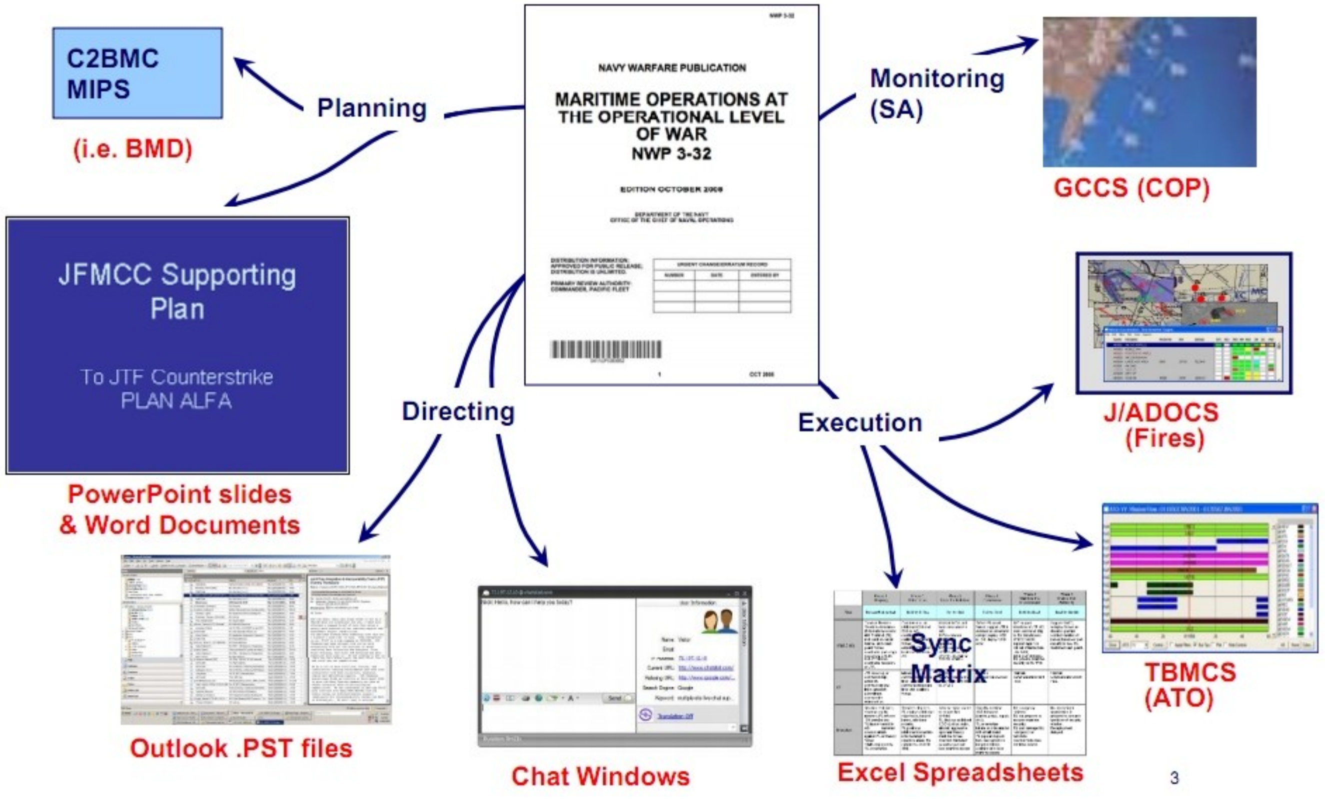
Last, we also report on topics that essentially have little/no published research study or results that are central to the continuing development of VW technologies for C2 applications. Principal among these are topics surrounding the security of VWs intended for experimental studies or future applications to the C2 domain. VWs are still very early in their technical development, and far from ready for deployment in actual application settings. Nonetheless, these technologies do merit ongoing study and investment as they offer new categories of affordances that can enable/support both centralized and decentralized approaches to C2 in ways that legacy C2 technologies and approaches cannot. Decentralized VW-based approaches may offer the potential to substantially reduce the cost and dramatically shorten the time to design, build, and deploy C2 systems that embrace new generations of low-cost, mobile technologies that future C2 workforces may expect, whether for use in physical or virtual/cyberspace worlds. So much remains to be studied, and time for appropriate and realistic research investments is at hand. In the near-term, such research is likely to still be considered risky, but the longer-term benefits may most quickly arise and be demonstrated through such near-to-mid term research investments. This is the future opportunity now at hand.

Accordingly, we welcome any comments or questions you may have from description and documentation we present in this Final Report.

Table of Contents

Introduction and purpose

In the effort described here, we have prototyped a computer game and virtual world (CGVW) environment we call the DECENT project. The purpose of this system is to provide an alternative to traditional, centralized methods used for Command and Control (C2) systems, processes, and practices. Our efforts here represent a substantial departure from current practice, where attention is directed to technologies that enable network-centric, decentralized "edge" approaches to C2 [Albert and Hayes 2003], and thus does not seek to primarily provide an incremental improvement to centralized C2 practice. Our efforts seek to be informed by such efforts, like the C2 Rapid Deployment Continuum (C2RPC), as they are critical to enhancing and demonstrating upgrades to current C2 operations which have high consequence [Garcia 2010, Gizzi 2011]. In such settings, C2 operations/tasks entail the creation, update, and sharing/presentation of information reports for C2 decision-making purposes, which may include resource schedules expressed as timelines or spreadsheets, for example.



Common information objects and software applications that may be involved in C2 operations/tasking, as identified in publicly accessible C2RPC [2010] materials.

However, as our efforts represent a basic research investigation, we can pursue more risky pathways and edgy alternatives that may or not yield significant advances. Our goal is to advance scientific and technical knowledge of how decentralized C2 might be put into practice in the future, especially with regards to future workforces who may have grown up playing computer games and/or exploring virtual worlds.

To help motivate our approach, we first choose to characterize a sample of computer games that have influenced our efforts for what might be possible with future C2 systems, and how such systems might enable people using the C2 system to be physically dispersed, for different reasons, yet still be capable of performing C2 missions, processes, and practices. We begin with our sample of games.

Computer game concepts

Computer games are computational systems for individual, group, or team interaction with game elements or other players. As computational systems, they may be designed to model, simulate, or "mirror" activity or other systems found in the physical world, or in imaginary/fantasy worlds. Game play activity may be sedentary, with seemingly little physical activity beyond computer mouse or keyboarding actions, or physically demanding, where computer play requires physical activity in diverse physical settings, or with complex devices. Most depictions of C2 activity may be closer to sedentary activity by individuals and work groups/teams, than to highly physical activity.

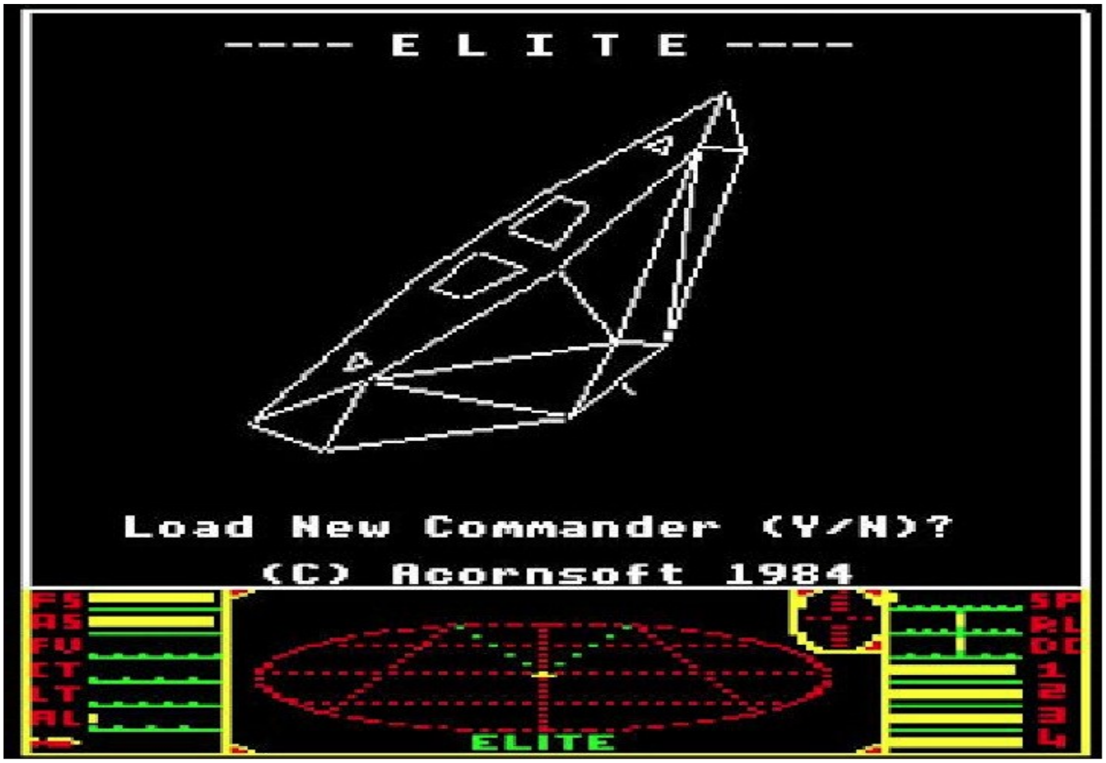
Modern computer games are complex systems (sometimes a system of systems) involving many components, sometimes from different vendors or contractors. Popular 3D action games represent software systems involving hundreds of thousands of lines of source code, with others larger or smaller. Common game system software components include game engine (the run-time environment controlling and coordinating all computational aspects of the game), the user interface (providing the ways and means for the user/player to interact with the system), networking (for wide-area game play with remote players), databases/repositories (accommodating persistent game play across multiple play sessions), graphics and animation, artificial intelligence, and other components. Game hardware components vary from personal computers, game consoles, hand-held game devices (e.g., PlayStation Portable, Nintendo DS), smart-phones, wired/wireless networking, backend servers, and more. A growing number of commercially successful game systems also are deployed for consumer use with full software development kits (SDKs) that allow the adventuresome and technically skilled user/player to modify the game, game object, rules of play, and more, including total conversion of a commercial game into another not recognizable from examination of the source game [Scacchi 2011].

Computer game play takes place within a visually (or textually) rendered game space that depicts a the world of objects and players (as in-game avatars) that can be manipulated, navigated, or engaged according to the rules of play built into the game system. The game space establishes game look and feel, as well as an overarching storyline and/or backstory that serve as a narrative backdrop intended to increase the immersive experience of the players. The game's rules of play controls game engine, mediates user interface display/interaction, as well as database transactions, and propagation of data/events to others in the game space. The rules of play may be pedantic, requiring completion of preliminary levels or play skill mastery before moving on to more challenging play scenarios; may be goal-oriented towards attaining some specified or implied final game state/objective; or may be fully emergent and open-ended to indicate that the purpose of the game is unbounded play or pursuit of fantasy. Game play rules are computationally realized and designed as “game play mechanics” to indicate the progressive play entails mastery of skills and achievements that recur across extended periods of game play . Putting these elements into an examples, in a card game like draw poker, the card table/surface denotes the play space for card dealing, holding, reviewing, and returning; the play mechanics entail how/when cards are dealt to players and how/when betting occurs; and game play rules determine how card holdings are scored and ordered across players. Finally, as the number and diversity of computer games is broad, it is common to refer to similar games as falling into widely recognized game “genres” like first-person shooters, (land/sea/space) combat simulators, real-time strategy, card/board games, role-playing games, etc. with 10-25 popular computer game genres.

Sample Games

In this section, we briefly describe five computer games that have influenced our view and conceptualization of how game concepts may be adapted for inclusion into decentralized C2 systems. The games include, Elite, Wing Commanded, EVE Online, Industry Giant II, and Texas Hold'em.

- *Elite* (1984), *Wing Commander: Privateer* (1993), *EVE Online* (2007)
  - *Genre*: Space flight combat simulator and resource trading games
  - *Core play mechanics*: Plan routes to remote planets, deal with obstacles or enemy threats (e.g., pirate vessels) in route, acquire resources and skill points through trading or combat, monitoring and managing resource utilization.
  - Wing Commander adds an explicit narrative backstory to help motivate game play, mission planning and resource management.

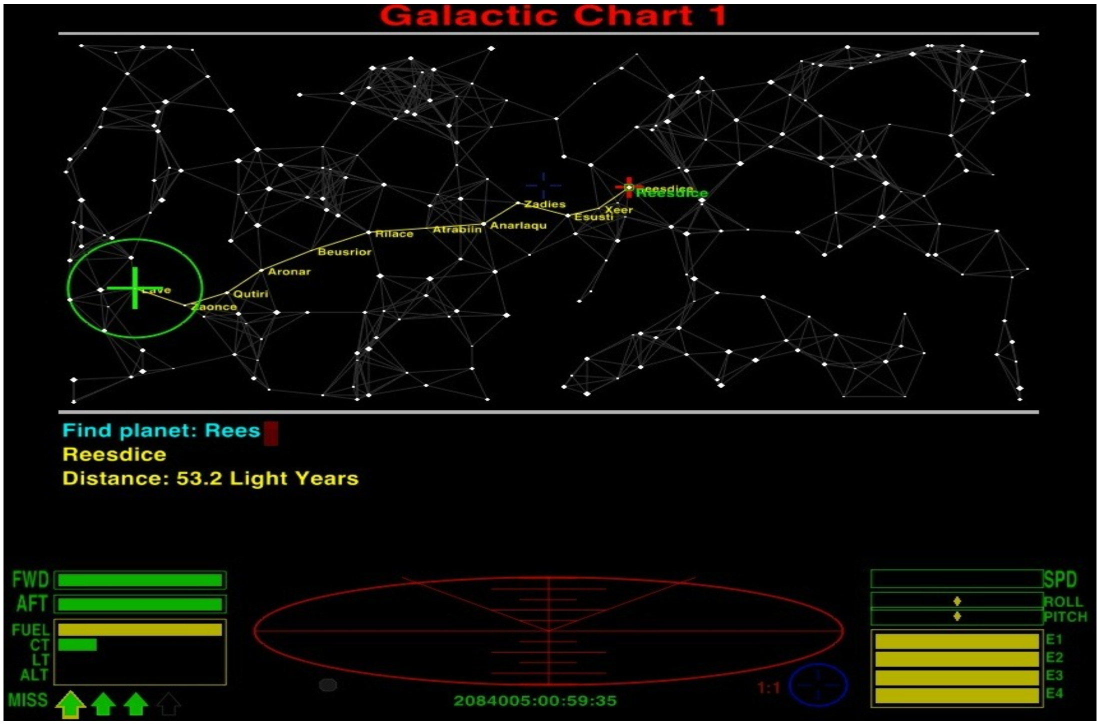


---- ELITE ----

Load New Commander (Y/N)?

(C) Acornsoft 1984

Elite game, with basic non-windows style user interface displaying resource status dashboard



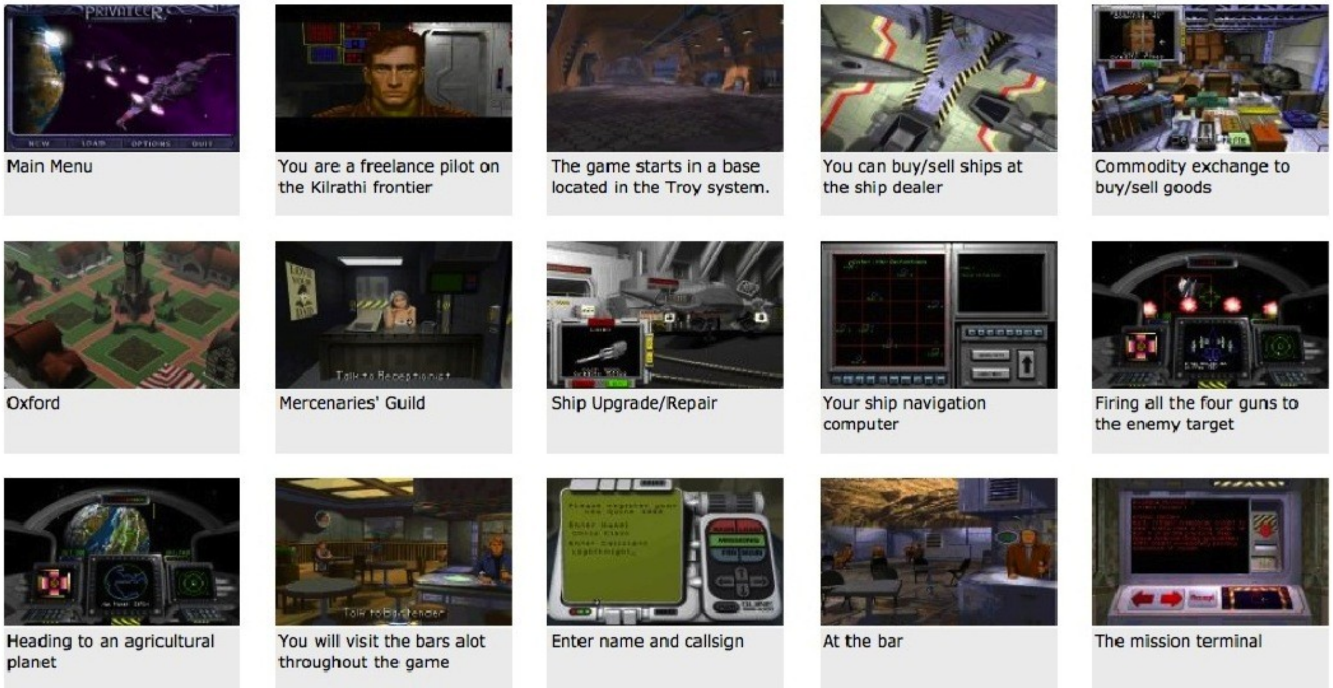
Find planet: Rees

Reesdice

Distance: 53.2 Light Years

2084005:00:59:35

Elite game, displaying mission route planning task interface view



Main Menu

You are a freelance pilot on the Kilrathi frontier

The game starts in a base located in the Troy system.

You can buy/sell ships at the ship dealer

Commodity exchange to buy/sell goods

Oxford

Mercenaries' Guild

Ship Upgrade/Repair

Your ship navigation computer

Firing all the four guns to the enemy target

Heading to an agricultural planet

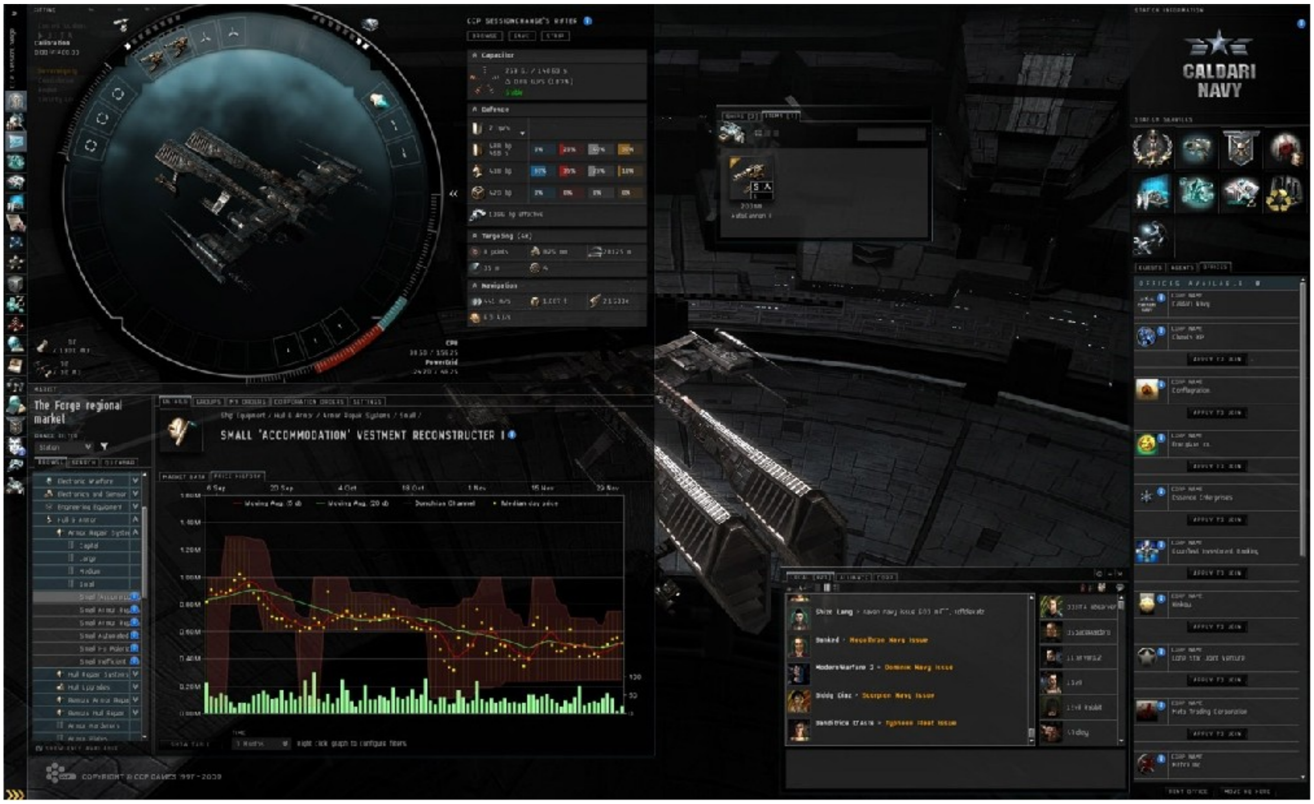
You will visit the bars alot throughout the game

Enter name and callsign

At the bar

The mission terminal

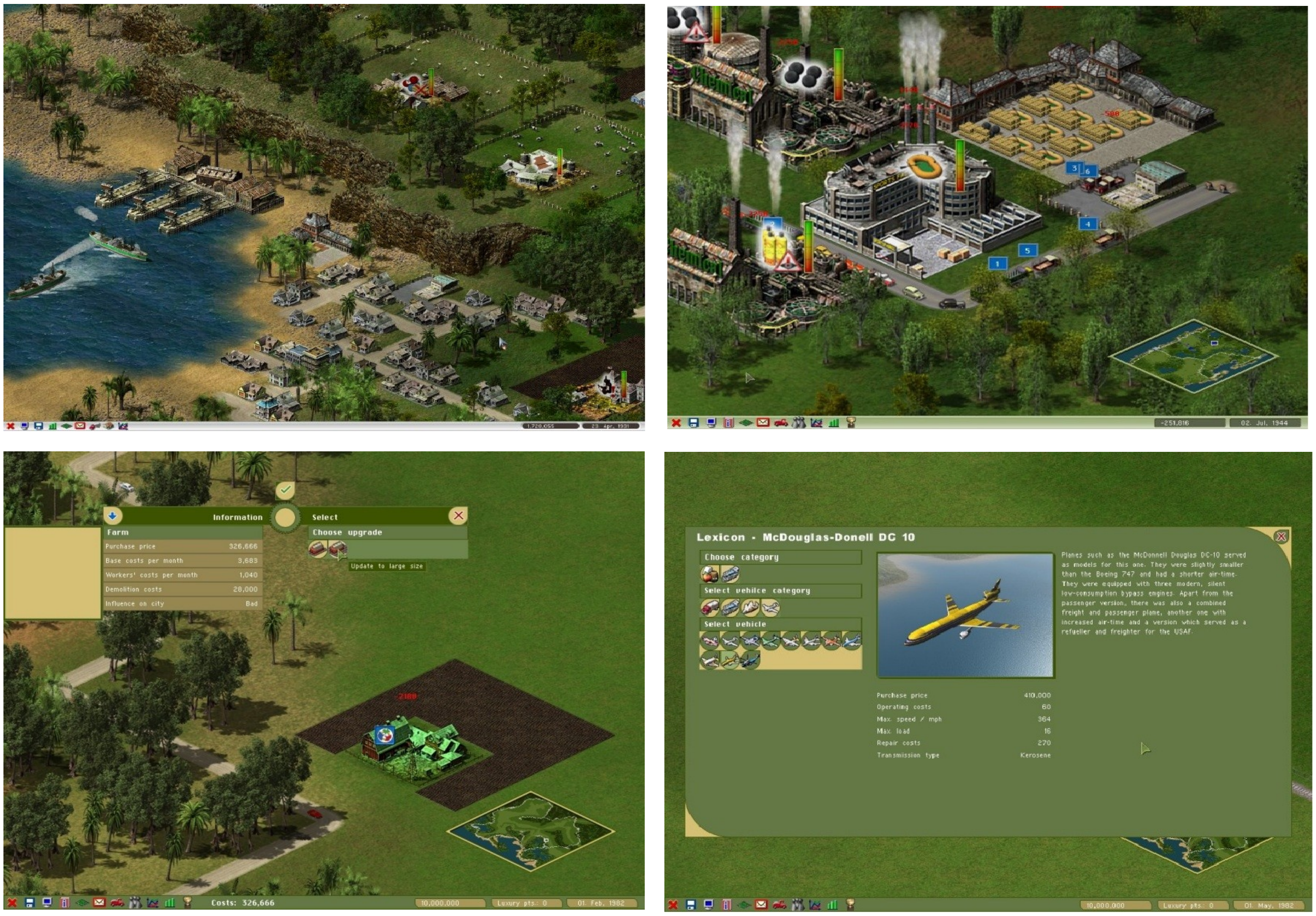
Multiple views of *Wing Commander:Privateer* game revealing overall narrative structure of game play mission.



CALDARI NAVY
The Forge regional market
SMALL 'ACCOMMODATION' VESTMENT RECONSTRUCTER I
COPYRIGHT © CCP GAMES 1997-2009

Multi-tasking user interface dashboard from *EVE Online*, including ship status, current resource allocation/consumption status, team status and chat system.

- *Industry Giant II* (2003)
    - *Genre*: Resource trading and accumulation game simulating industrial development within an computationally evolving game world
    - *Core play mechanics*: Travel to destinations, assess resource harvesting opportunities and barriers, acquire resources through skill investments and resource trading.



Information

Farm	
Purchase price	326,666
Base costs per month	3,683
Workers' costs per month	1,040
Demolition costs	28,000
Influence on city	Bad

Select

Choose upgrade

Update to large size

Lexicon - McDouglas-Donell DC 10

Choose category

Select vehilce category

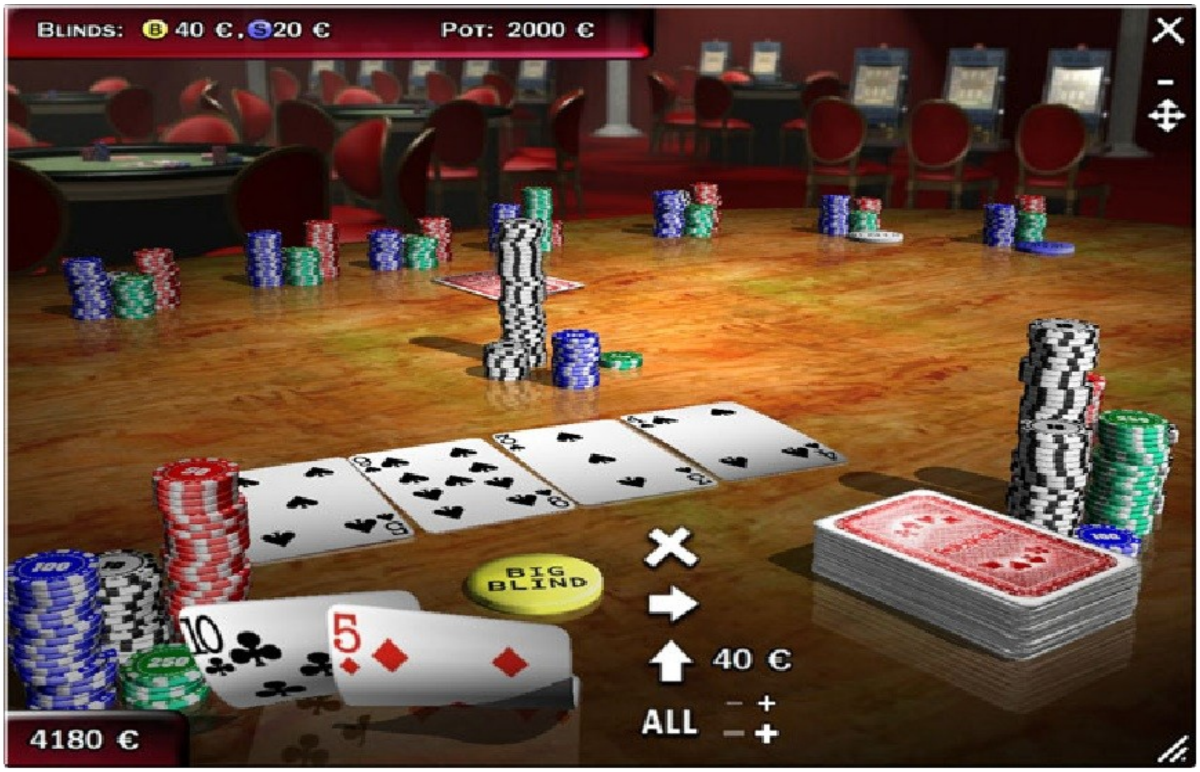
Select vehicle

Planes such as the McDonnell Douglas DC-10 served as models for this one. They were slightly smaller than the Boeing 747 and had a shorter air-time. They were equipped with three modern, silent low-consumption bypass engines. Apart from the passenger version, there was also a combined freight and passenger plane, another one with increased air-time and a version which served as a refueller and freighter for the USAF.

Purchase price	410,000
Operating costs	60
Max. speed / mph	364
Max. load	16
Repair costs	270
Transmission type	Kerosene

Multiple views from *Industry Giant II* game displaying situation awareness (*T*op *L*eft), resource awareness overlay (*TR*), target selection and status overlay (*BL*) and resource availability status (*BR*).

- *Texas Hold 'em* (traditional card game)
  - *Genre*: 7 card Poker game
  - *Core play mechanics*: incremental resource investment (betting), hedging, and bluffing in presence of emerging private view of resource position (cards held) and shared viewed of community resources (card values on table, resource wagers/investments), with the value of shared resources and private resources subject to uncertainty and subject to change with updates, competition for final resource allocation (winning the resource wagers). Versions of this game are extremely popular, with millions of players at online sites like Facebook and elsewhere, so its play mechanics are widely understood on a global basis.

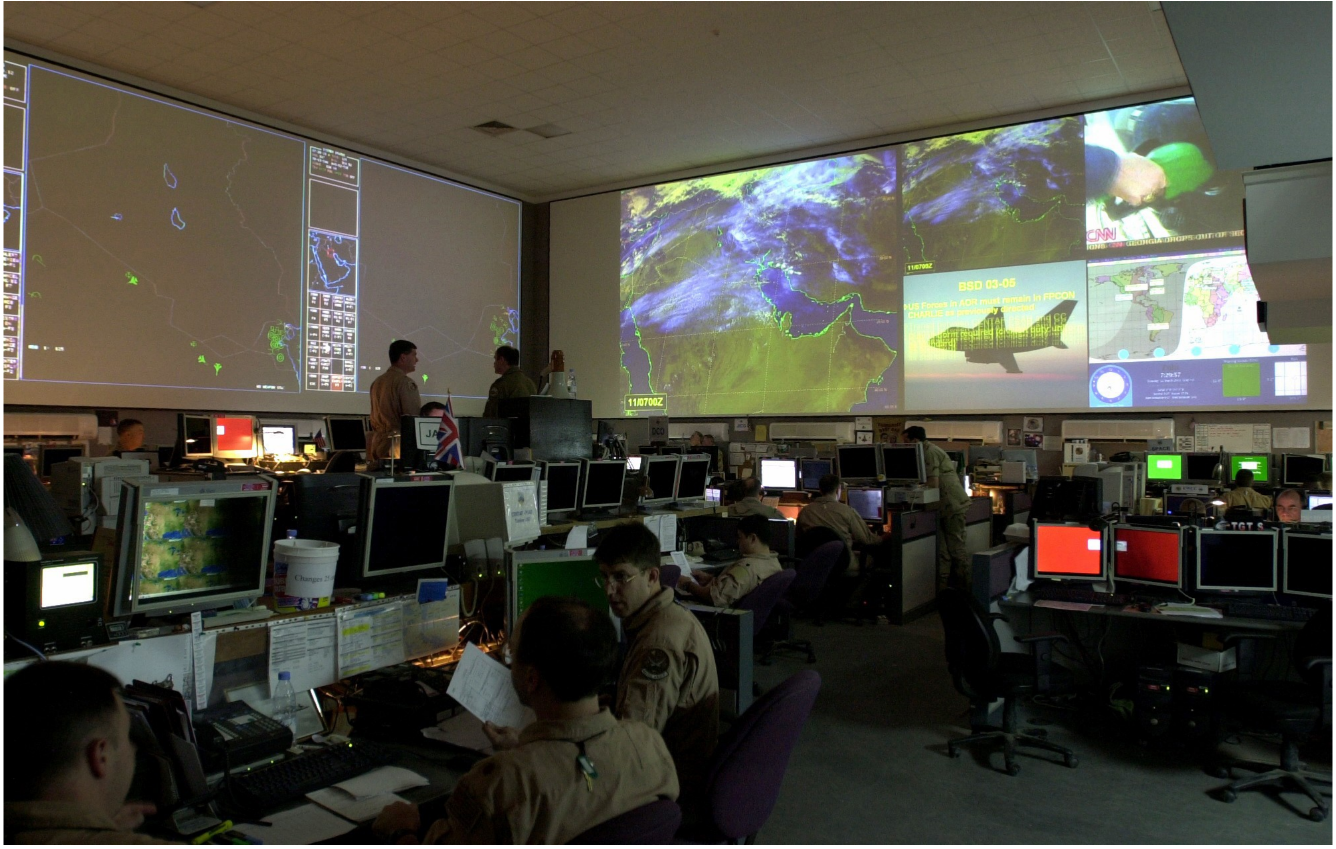


View of *Texas Hold'em* computer game with shared and private resource positions (community cards displayed on table, and cards turned up, respectively), and competitor resource allocations (and reserves) intended to sway outcomes in their favor.

Developing Virtual Worlds

Physical space

In order to achieve the highest quality for DECENT, development began by examining existing C2 structures and identifying key features to replicate. C2 facilities usually have several large screens on several walls, used for the display of two types of information: public information that must be available to several personnel simultaneously, and key information that is either higher priority or more topically relevant. While these large public displays are viewable by anyone in the command room, most C2 workers also have their own personal computer (sometimes with multiple displays), housed on desks with assorted papers, files, schedules, etc. Systems used in C2 facilities require a range of software, and information from one user may affect the relevance of another's information. Personnel are thus typically organized in a way that optimizes communication, with the most frequent communication being between neighbors. Nevertheless, spaces in C2 facilities are often seem crowded and cluttered, despite the need for efficiency.

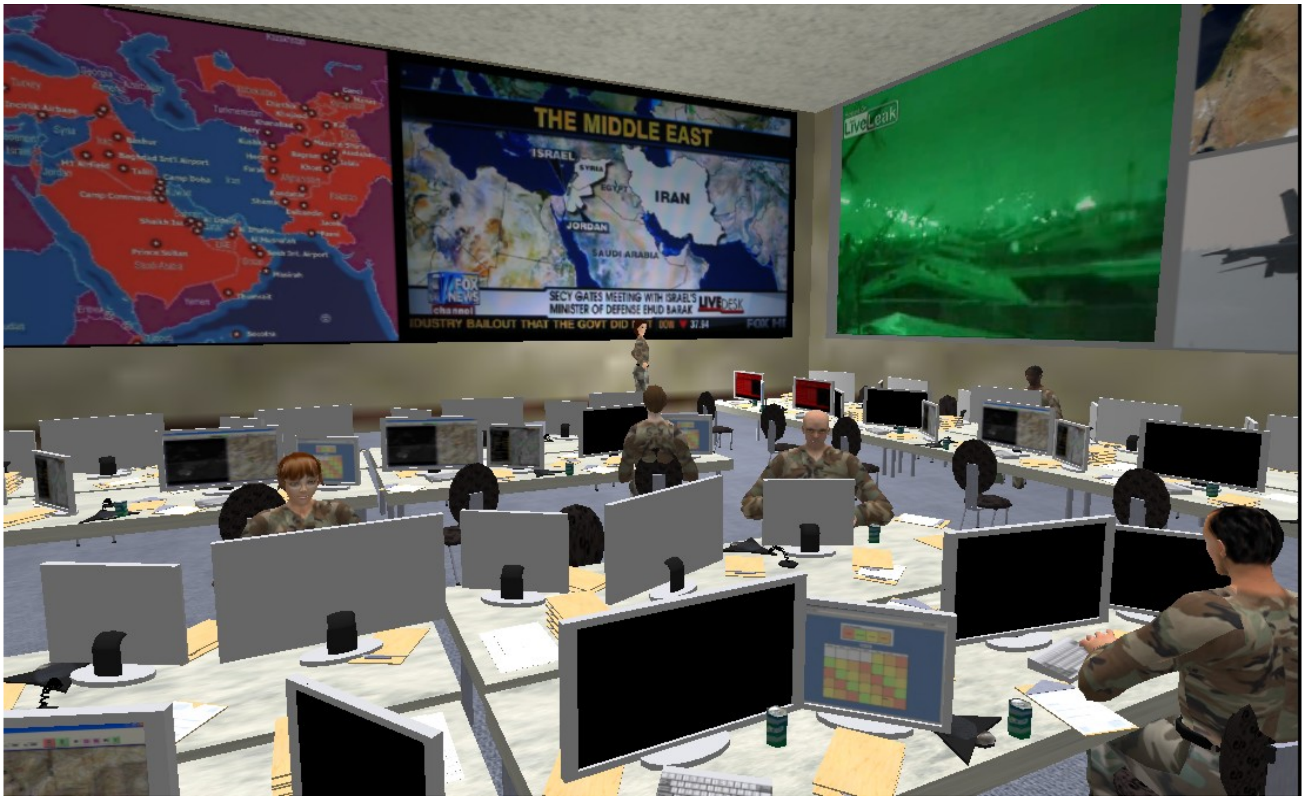


Photograph of a physical C2 facility. Both shared (wall screens) and private resources (tabletop displays) can be seen, with other mission tasking resources.

Virtual world space

Using images like the one above, we have created a virtual C2 room. Taking advantage of the malleable nature of virtual worlds, we can make C2 rooms with more room and a less cluttered or cramped appearance. All the important aspects and features common to C2 facilities have been faithfully and dutifully recreated. Simple tables contain monitors, input devices, paperwork (which can be modified or made interactive), and speakerphone boxes. Varying levels of clutter can be used, as well, by adding or removing clipboards, pens, soda cans, coffee mugs, and other cosmetic/non-functional objects. This variability allows each DECENT implementation to be customized by whomever is running the training system. Chairs anchor the user's avatar to a given workspace, and two monitors act as that user's private information displays, such as DECENT training game data specific to that user.

Two of DECENT's walls have large displays for the display of public information, including the main screen for the DECENT training game. These images may be easily changed, and can be used to display streaming video, as well as static images and the DECENT training game. Due to the nature of virtual worlds, bulky items necessary for actual C2 centers, such as servers, PC boxes, and cabinets can be ignored entirely.



Perspective view of user-controlled avatars in a C2 mirrored VW. Green wall display is an embedded video stream from a remote server.

Comparing Physical and Virtual Spaces

The strategy we have investigated in the DECENT environment is to prototype a mirrored virtual world for C2 that resembles and may operate like the physical world C2. In this way, we seek to explore and examine when/how the similarities and differences between the two can reveal potentially significant insights, opportunities, or advantages that one may pose over the other. For example, in C2 studies with VW at the Naval Postgraduate School's Center for the Edge (also the sponsor of this research), there has been sustained study examining how hypotheses about different models of team organization or theories of management might affect the course and outcome of play in the ELICIT multiplayer online counterterrorism intelligence game [Bergin, et al., 2010, Hudson and Nissen 2010, Winn, Ruddy and Nissen 2010]. Among other things, these studies seek the investigate the efficacy of organizational form, team play, and outcome in the ELICIT game when played in a physical setting (a large unadorned meeting room) in comparison to a virtual setting (seen in the picture below).

On-screen view of a virtual meeting room that may mirrors common meeting rooms, used to study team-oriented ELICIT game play [Hudson and Nissen 2010]

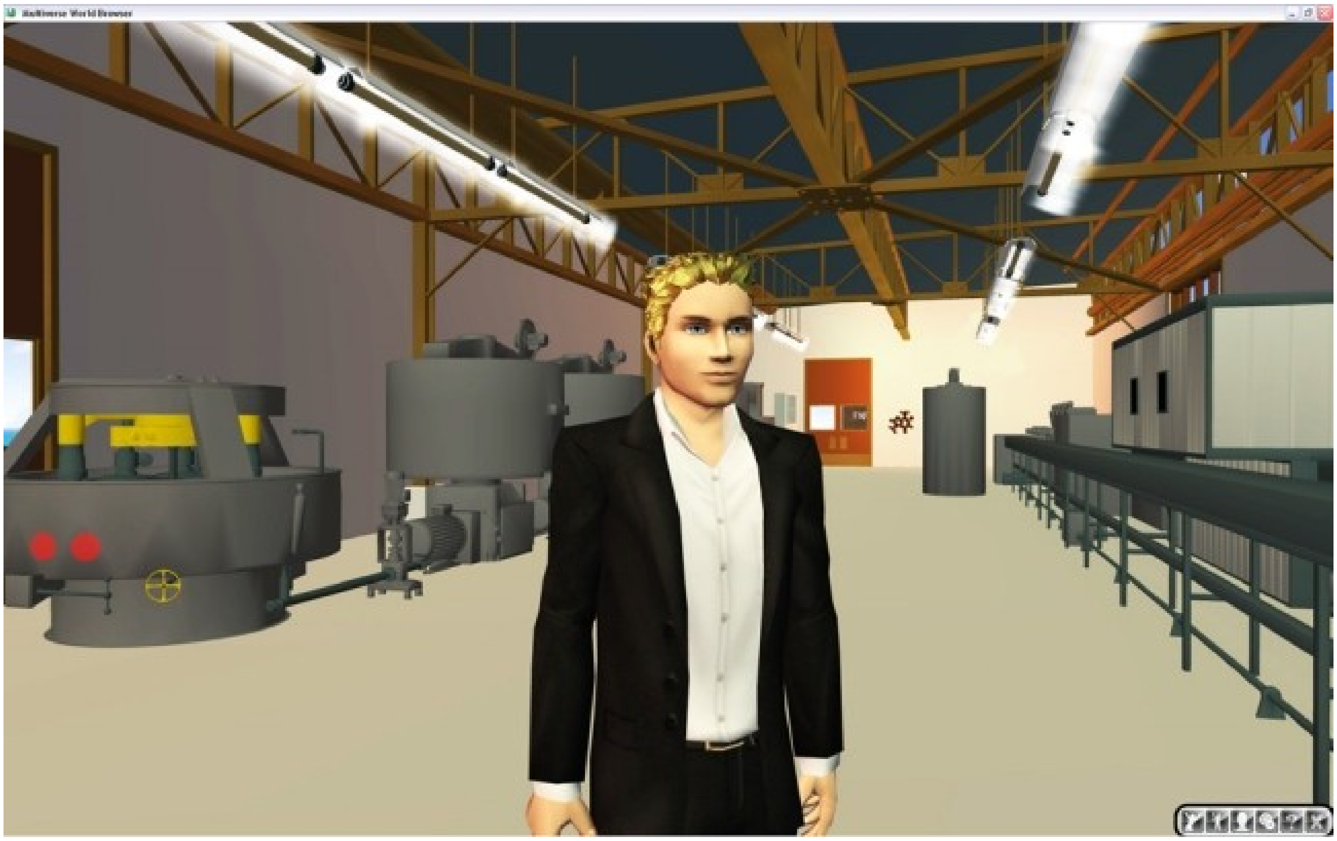
Studies by Bergin et al., [2010] suggest that decision-making performance in physical and virtual worlds can favor the physical. This may be due to the environmental richness and tacit knowledge affordances that familiar work spaces and co-worker gestures/gazes provide, compared to the paucity of similar affordances in a VW. However, DECENT VW may offer other benefits like low-cost, appropriateness for large-scale training, and

absence of a centralized (potentially vulnerable) C2 physical center.
Elsewhere, other research groups have been experimenting with the creation of mirror worlds that intermix physical world sites, with VW interfaces, navigational and interactive controls to devices in the physical site. One noteworthy example is the effort by Back, Kimber, et al. [2010] who have modeled a physical chocolate factory (the TCHO factory located in San Francisco, http://www.tcho.com/where-to-buy/tour/, that is open to the public), as part of their efforts at Fuji Xerox Palo Alto Laboratory (FXPAL).

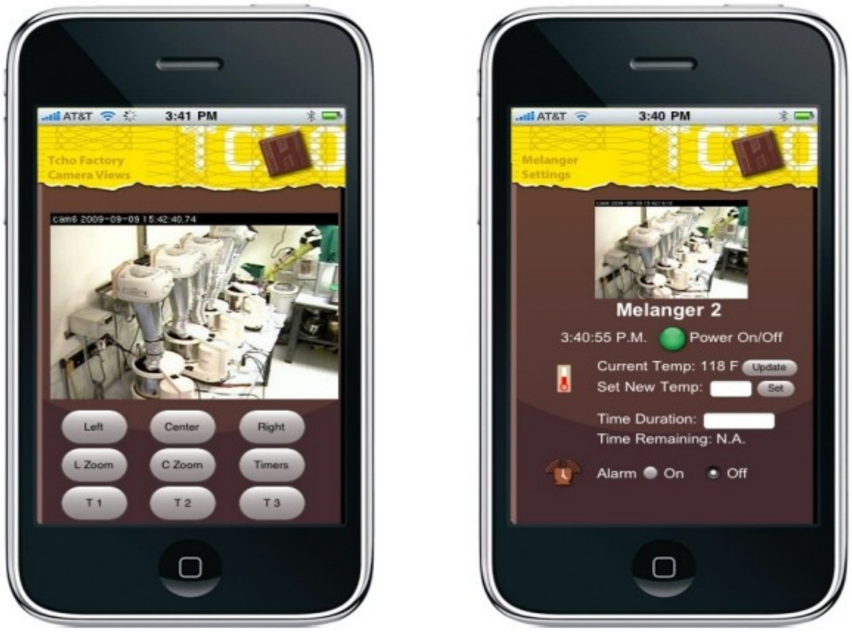
Photograph of the TCHO physical factory site (during construction), from Back, Kimber, *et al.* [2010].

Their VW system includes desktop PC and smart-phone based software clients that allow a user to navigate the VW space, and to enable/disable designated sensors (web cameras) located in the physical site, and thus demonstrate the potential to remotely control or monitor devices in the physical site though the VW client interface.

The FXPAL system thus demonstrates the potential for mixed reality applications that span and interlink physical world sites that are mirrored in a VW. Their efforts in turn can be compared with one of our earlier efforts at UCI which focused on the modeling and simulation of semiconductor fabrication processes and diagnosis of manufacturing devices for training technicians [Scacchi 2010]. However, this UCI effort was based on abstractions of semiconductor and nanotechnology fabrication facilities on site at UCI, but generalized into configurations that were suggested by the project sponsor at Intel.



View of VW that mirrors the TCHO factory, from Back, Kimber, *et al*. [2010].



cam6 2009-09-09 15:42:40.74

Melanger 2

3:40:55 P.M. ● Power On/Off

Current Temp: 118 F

Set New Temp:

Time Duration:

Time Remaining: N.A.

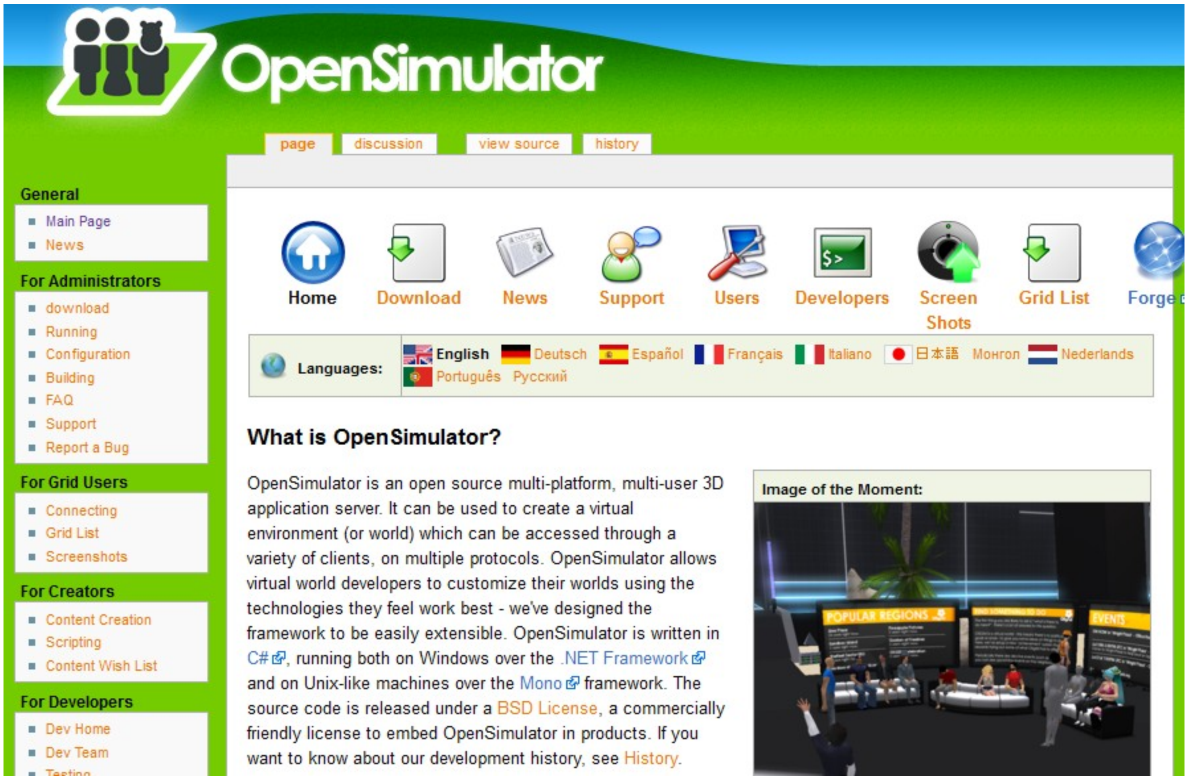
Alarm ● On ◉ Off

Smart-phone (iPhone) based views for monitoring and controlling devices in the physical TCHO factory, from Back, Kimber, *et al*. [2010].

Platform for VW development: OpenSim

Due to its ease of use and rapid development capabilities, DECENT is currently implemented in OpenSim, a open-source version of the SecondLife platform. SecondLife is the current leader in rapid virtual world development, with a high level of design flexibility and built-in tools for easy environment creation and maintenance. This makes it and OpenSim ideal for the creation of prototypes. Using OpenSim has allowed us to rapidly create a functional C2 VW analog, and populate it with users for testing. The degree of design freedom OpenSim has allowed DECENT to evolve from the promising concepts it began as, into a fully functional training environment.

While in OpenSim, DECENT has the potential to seamlessly interact and crossover with other, currently existing military projects, such as the Military Open Simulator Enterprise Strategy (MOSES) combat training environment and the Naval Underwater Warfare Center (NUWC) campus. Adding DECENT to to either MOSES or NUWC is a simple matter of adding a new region and importing the DECENT assets. This could be done as many or as few times as needed, and each instance of DECENT would act independently. The man responsible for MOSES and NUWC, Douglas Maxwell, has stated “All of [OpenSim's] features are desirable for the new virtual trainers needed to meet the changing situation demands on modern warfighters” [Neville 2011].

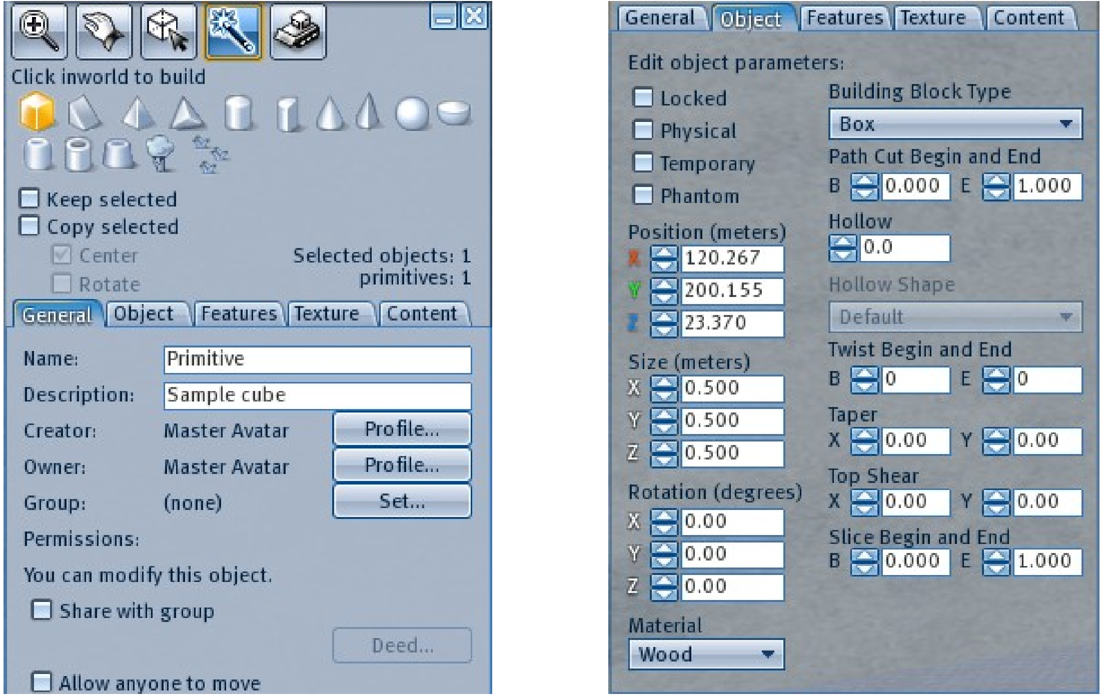


What is OpenSimulator?

OpenSimulator is an open source multi-platform, multi-user 3D application server. It can be used to create a virtual environment (or world) which can be accessed through a variety of clients, on multiple protocols. OpenSimulator allows virtual world developers to customize their worlds using the technologies they feel work best - we've designed the framework to be easily extensible. OpenSimulator is written in C#, running both on Windows over the .NET Framework and on Unix-like machines over the Mono framework. The source code is released under a BSD License, a commercially friendly license to embed OpenSimulator in products. If you want to know about our development history, see History.

OpenSim [2011] homepage.

SecondLife/OpenSim's major downside, however, is a reduced capacity for high-fidelity graphics. In SecondLife/OpenSim, items and buildings in the world are created by combining may different “prims” (simple geometric objects) into shapes that reflect the target object, then adding skins to make the shape appear more robust. A table, for instance, is made by overlapping a thin box with four cylinders, then adding textures to make the cylinders appear metal and the box appear to be wood (or plastic). While this technique is sufficient for building a simple version of any virtual object, and can be easily learned by users of any skill level, it carries with it a restriction on absolute graphic quality. Many other virtual worlds make use of meshes, complex geometric shapes which carry with them the potential for photo-realistic rendering and related visual effects/details. However, VW s like SecondLife and OpenSim lack the native ability to incorporate external application programs like office productivity tools (e.g. spreadsheet application programs), but can incorporate embeddable media objects like presentation slides, graphic images, and text/video streams.



Click inworld to build

□ Keep selected

□ Copy selected

☑ Center

□ Rotate

Selected objects: 1
primitives: 1

Name: Primitive

Description: Sample cube

Creator: Master Avatar Profile...

Owner: Master Avatar Profile...

Group: (none) Set...

Permissions:

You can modify this object.

□ Share with group

Deed...

□ Allow anyone to move

Edit object parameters:

□ Locked

□ Physical

□ Temporary

□ Phantom

Building Block Type: Box

Path Cut Begin and End: B 0.000 E 1.000

Hollow: 0.0

Hollow Shape: Default

Position (meters): X 120.267 Y 200.155 Z 23.370

Size (meters): X 0.500 Y 0.500 Z 0.500

Twist Begin and End: B 0 E 0

Taper: X 0.00 Y 0.00

Top Shear: X 0.00 Y 0.00

Rotation (degrees): X 0.00 Y 0.00 Z 0.00

Slice Begin and End: B 0.000 E 1.000

Material: Wood

Prim creation and modification in the OpenSim client, Hippo.

One solution to this would be to migrate DECENT to another platform with a higher level of graphical fidelity, but this would necessitate either the development of a cross-platform communication system, or a stand-alone version of the DECENT training world without access to other similar military ventures.

Last, using these unremarkable modeling tools, it is possible to create relatively complex objects, as well as associating behavioral scripts (using LSL) to enable rich animated behaviors to be associated with different objects.

SecondLife versus OpenSim

While OpenSim is an open source project based on the SecondLife platform, each has its own strengths and weaknesses, as described by Maria Korolov [2011]:

Cost: The cost of running a SecondLife server is $295 each month, as it must be hosted on a LindenLabs server. OpenSim can be run for free if it is put on a dedicated server. We currently host five dedicated OpenSim servers at UCI.

Users: SecondLife is a community, so it comes with a large established base of users from all conceivable backgrounds. OpenSim has access to smaller groups of existing users if connected to existing OpenSim servers, but is more often run as a stand alone server, only used by those whom the administrators give access to.

Stability: SecondLife maintains high stability for functions used by the largest majority of its users, but are unreliable for mission-critical operations due to problems with voice-chat and reoccurring connectivity issues. OpenSim tends to have much more stable connectivity and voice-chat, (and now integrated instant messaging support within and across worlds) due to the smaller number of users and lower required bandwidth.

Asset Ownership: LindenLabs retains the rights to all assets created in SecondLife (but not content uploaded to it), regardless of who created it; users pay for a license to use SecondLife and modify the contents of a region, but gain no ownership of actual content. Furthermore, Linden Labs reserves the right to revoke access to SecondLife. In contrast, owners of OpenSim servers determine the use policies and ownership of their servers, as well as control access to their servers.

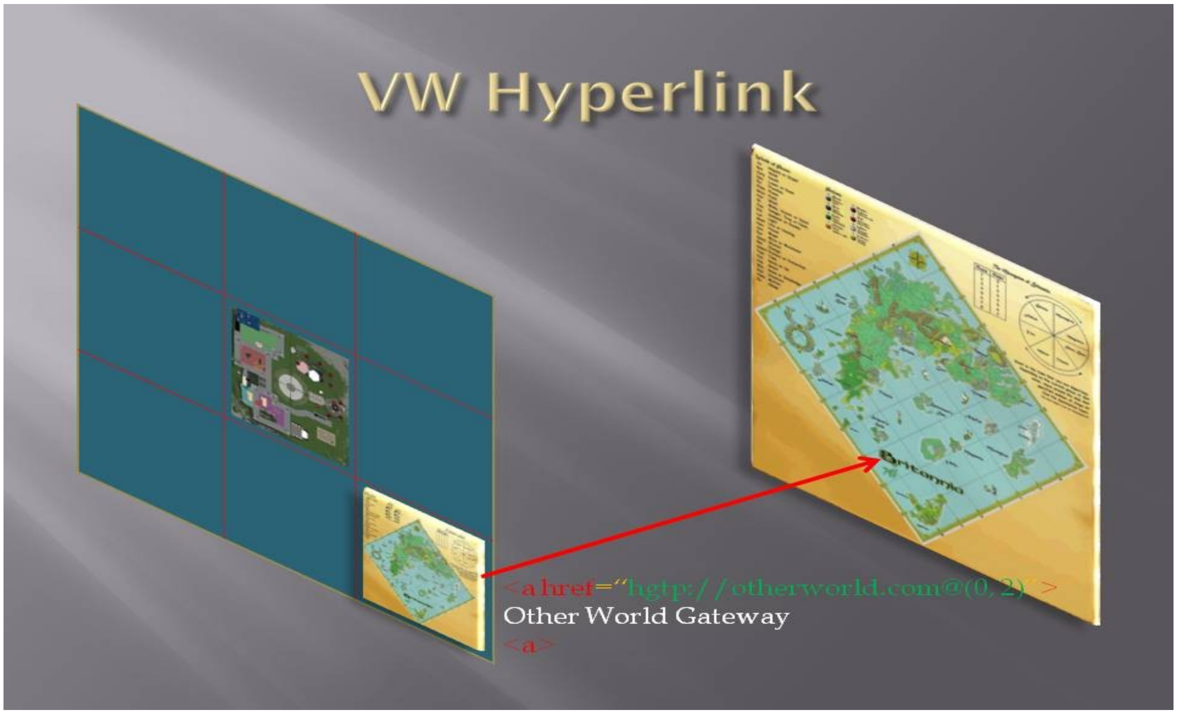
Scripting: SecondLife uses the Linden Scripting Language (LSL), a domain-specific scripting language. OpenSim supports LSL, but can be modified to support many other scripting languages, including JavaScript and Lua (which is a popular scripting language employed by many computer games, including *World of Warcraft*).

OpenSim developer relations

One advantage to using OpenSim is the presence of lead OpenSim developers currently at UC Irvine, including the lead developer and creator of the main shared OpenSim server (Diva), Professor Cristina Lopes (http://metaverse.stanford.edu/agenda/crista-lopes/crista-lopes). Having such immediate access to Lopes and her colleagues working on the OpenSim platform developing experimental VW applications [Lopes 2009] puts UC Irvine in a unique position to leverage expertise few others can claim. Lopes has already modified and updated versions of her Diva distribution to help meet the emerging requirements of current UCIrvine OpenSim-based projects, such as for DECENT.

The OpenSim Hypergrid

In addition to the reasons laid out about by Korolov [2011], we have decided to work with OpenSim because of its connections to the *OpenSim Hypergrid* [2011].

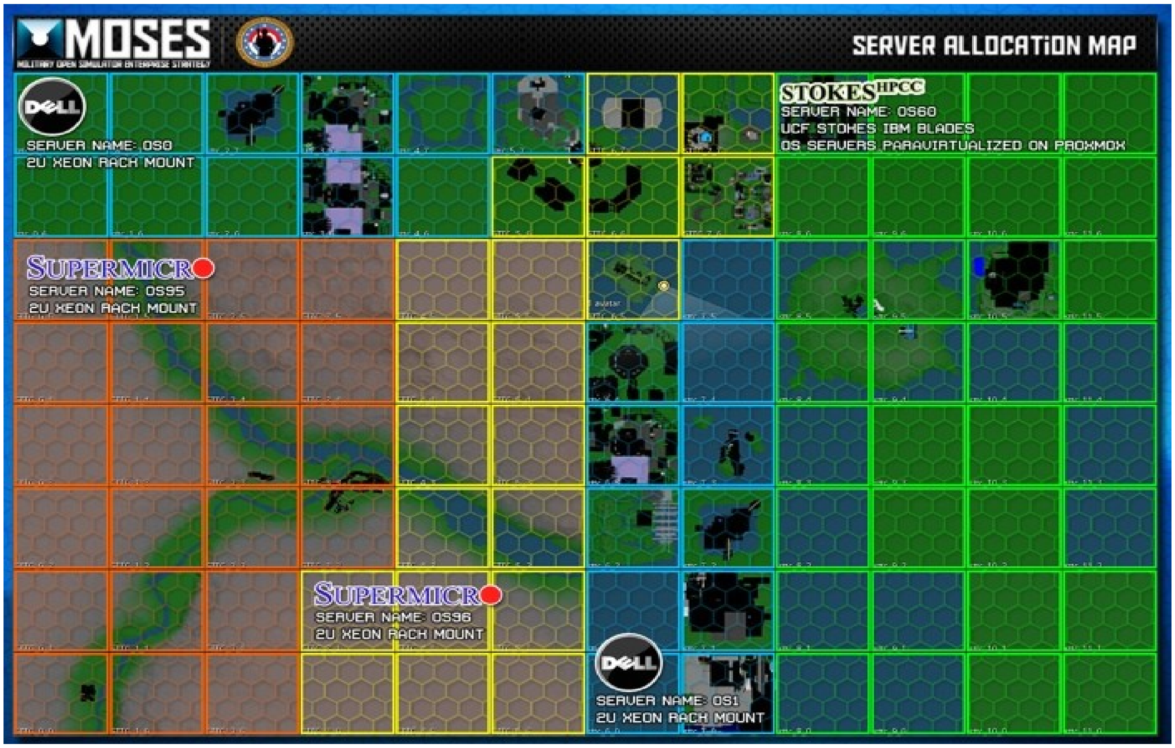


VW Hyperlink

```
<a href="hgtp://otherworld.com@(0,2)">
Other World Gateway
<a>
```

OpenSim Hypergird (2011) supports user/avatar teleportation to move from one VW region to another VW region, possibly on a different networked server.

The Hypergrid is a system used to connect one OpenSim server to others, and allows for the seamless transfer of avatars between any of these interconnected OpenSim worlds. Some large worlds span multiple servers. The virtual worlds are connected to each other via virtual world hyperlinks, similar to the links between web pages. The virtual world hyperlinks are places in the virtual worlds that act as doorways or entry/exit points for other worlds. To access one, a user simply moves his avatar to the VW hyperlink and activates it. VW hyperlinks could be used to connect DECENT to existing OpenSim-based military training systems, such as the MOSES Hypergrid [2011] facility.

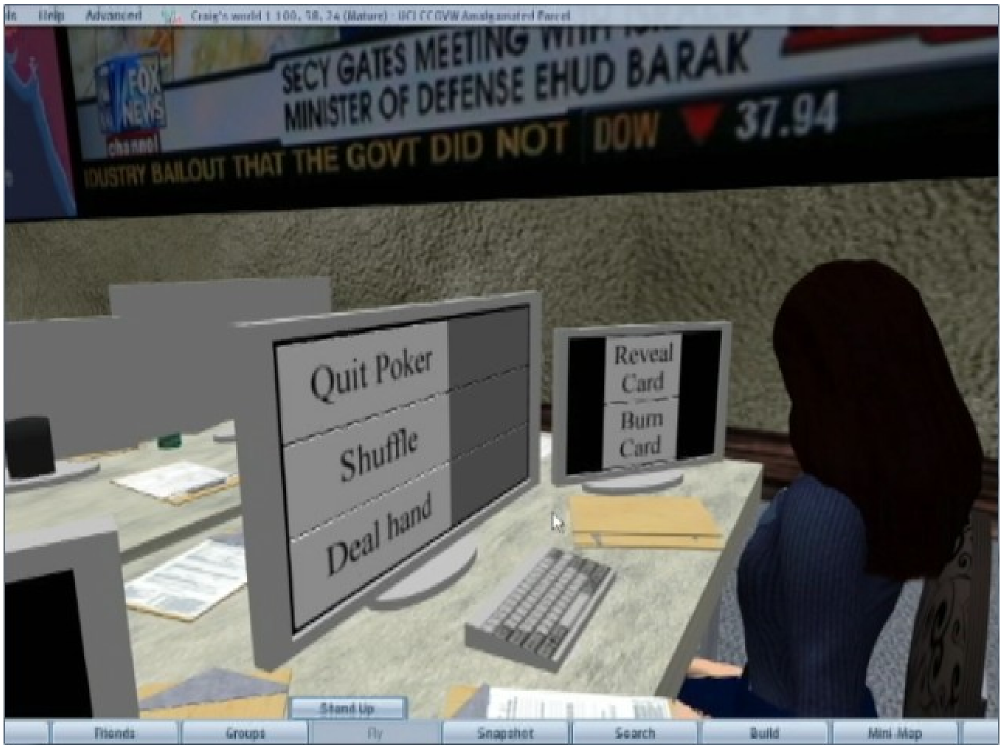


Overview of the *MOSES Hypergrid* [2011] and server assignment map used in the Military OpenSim Enterprise Strategy by the US Army. Some hypergrid cells are empty indicating available capacity for future development of new virtual worlds.

C2 Gameplay in DECENT

Instead of only simulating possible C2 situations and placing users in unprepared, we have chosen to use a gameplay metaphor that most users will have at least some familiarity with: Texas Hold'em Poker. This metaphor was chosen from amongst many options due the surprisingly large number of similar choices required by both.

In DECENT, one proctor controls the bulk of the gaming simulation. A simple interface allows the proctor to act as the dealer, and is responsible for shuffling the deck between hands, dealing each team their cards, and manipulating the public information cards. The proctor should be someone versed in the rules of Hold'em, and must maintain objectivity to ensure fair play. At the end of each hand, the proctor is also responsible for resolving the pot and ensuring all side-pots are dealt out appropriately.



Proctor at the dealer seat.

Two to six teams, representing separate military groups, compete in DECENT to see who is best suited to undertaking the current mission. Each team is completely autonomous, and can compete from any physical distance as long as they are connected to DECENT through a computer with an internet connection.

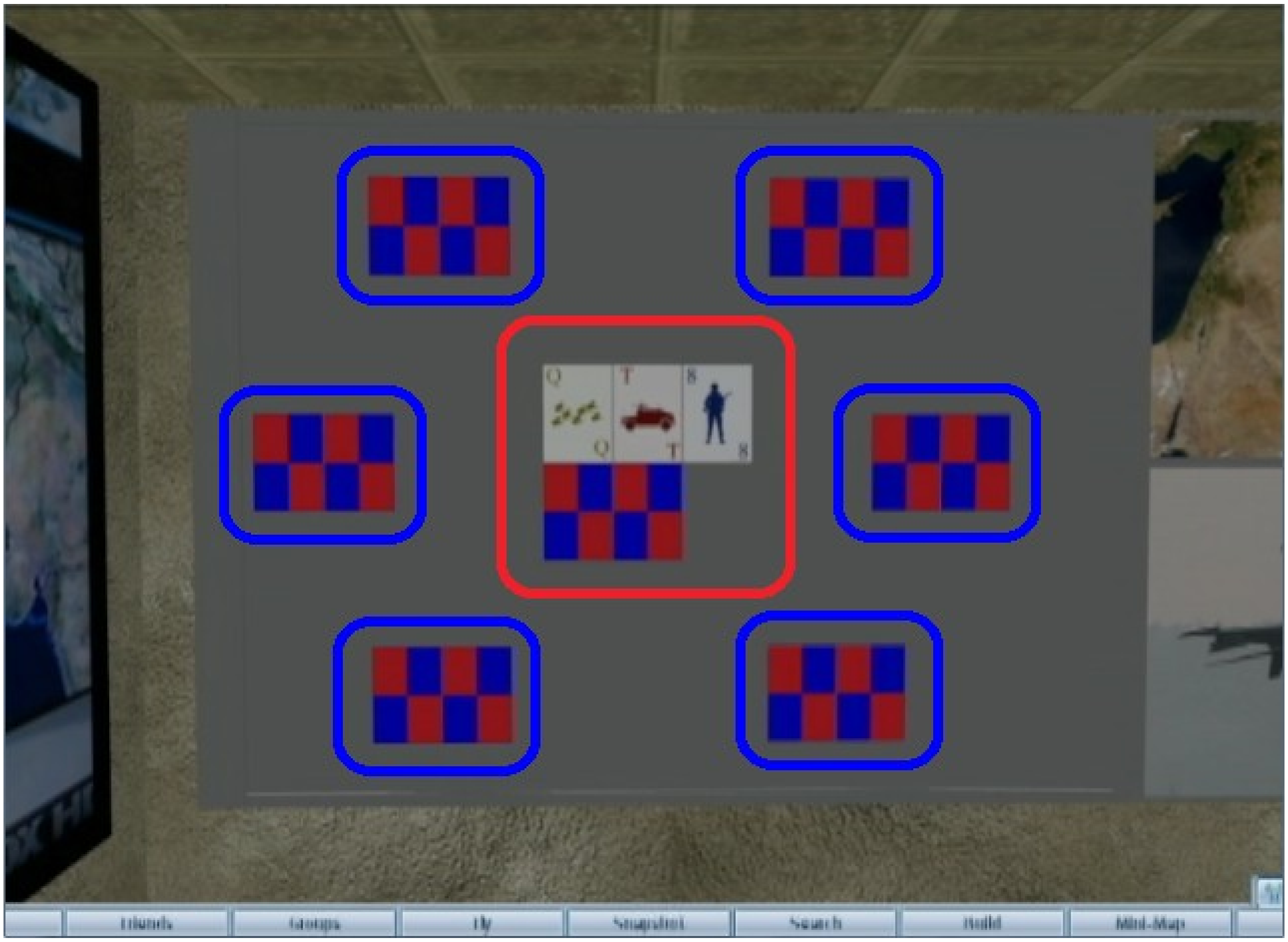
Teams discuss tactics for the current hand.

Team members interact amongst themselves, coming to a consensus on how to dedicate their resources and when to abandon a given situation. OpenSim handles all of this with its built-in communication tools. Interaction between teams is cautioned against (but not automatically forbidden), as sharing information non-openly may skew the odds of the gaming simulation.

Variable sized teams competing against each other.

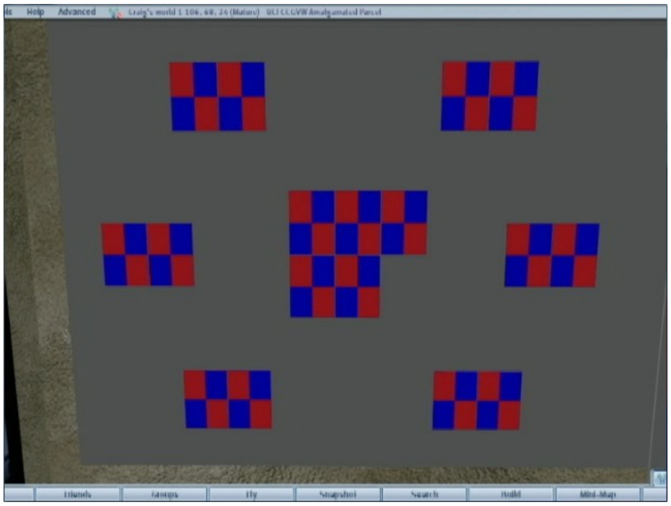
Both C2 and Texas Hold'em make use of public information, viewable by all and specific to no single party. C2 has this in the form of large information displays, either showing the most important information that must available at all times or organizational information necessary to coordinate tasks. Hold'em uses public information in the form of community cards, available to all to make the best five-card poker hand possible; these are commonly known as the flop, the turn, and the river.

In addition to public information, C2 and Hold'em both make use of private information. In C2, this can be anything from specialist data to communications to the stream of a soldier's helmet-cam. In Hold'em, it is each player's hole cards, the two cards only that player may use to make his or her five-card poker hand the best at the table.

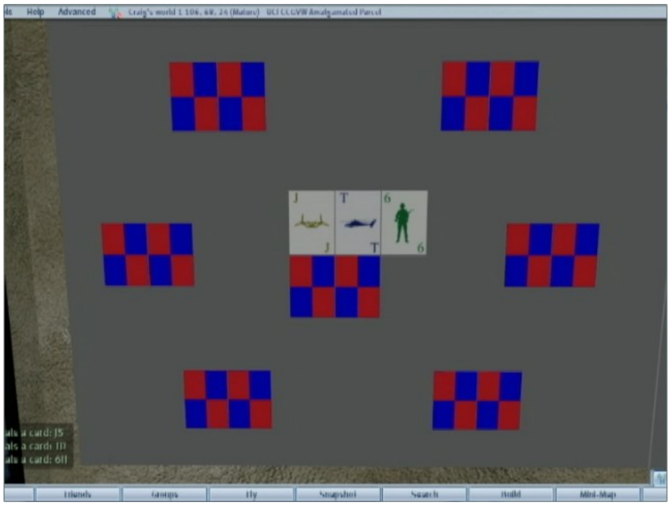


The public cards are circled with red; the pairs of private cards (face down) are circled with blue.

In addition to the sharing modes of information exchange, both games share similar resource and timing restraints. New information is revealed over time, and the information revealed this way affects each group's situation, changing who has the advantage. C2 reveals information in real time, and that information changes which resources should be committed and which should be held back. Hold'em reveals the public information cards at set intervals, but this still requires the team to re-evaluate its position, and the strategic values of its cards. The ability to re-evaluate one's standing and adapt to new information is critical in C2, just as one's ability to recognize when a new card improves or devalues an existing hand in Hold'em.

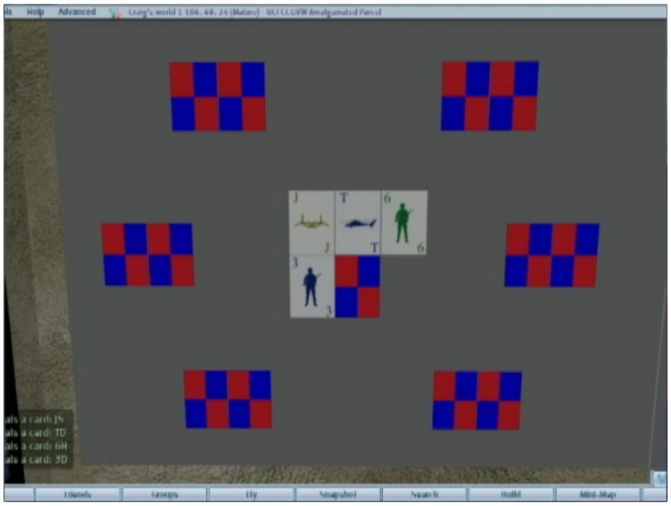


Pre-flop



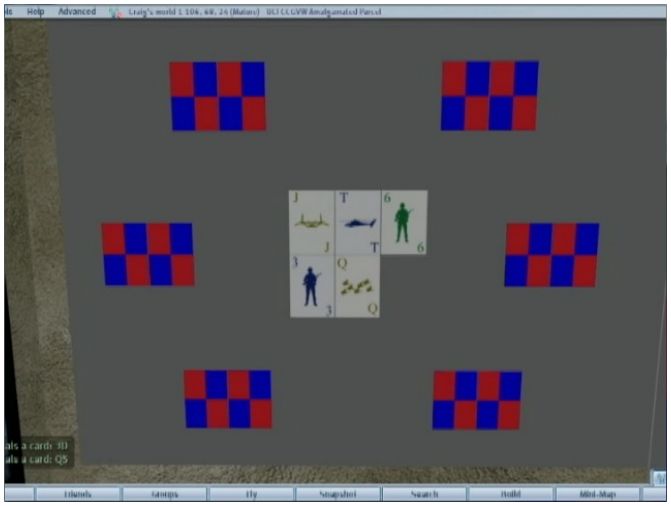
als a card: JS
als a card: TD
als a card: 6H

Flop



als a card: JS
als a card: TD
als a card: 6H
als a card: 3D

Turn



als a card: 3D
als a card: QS

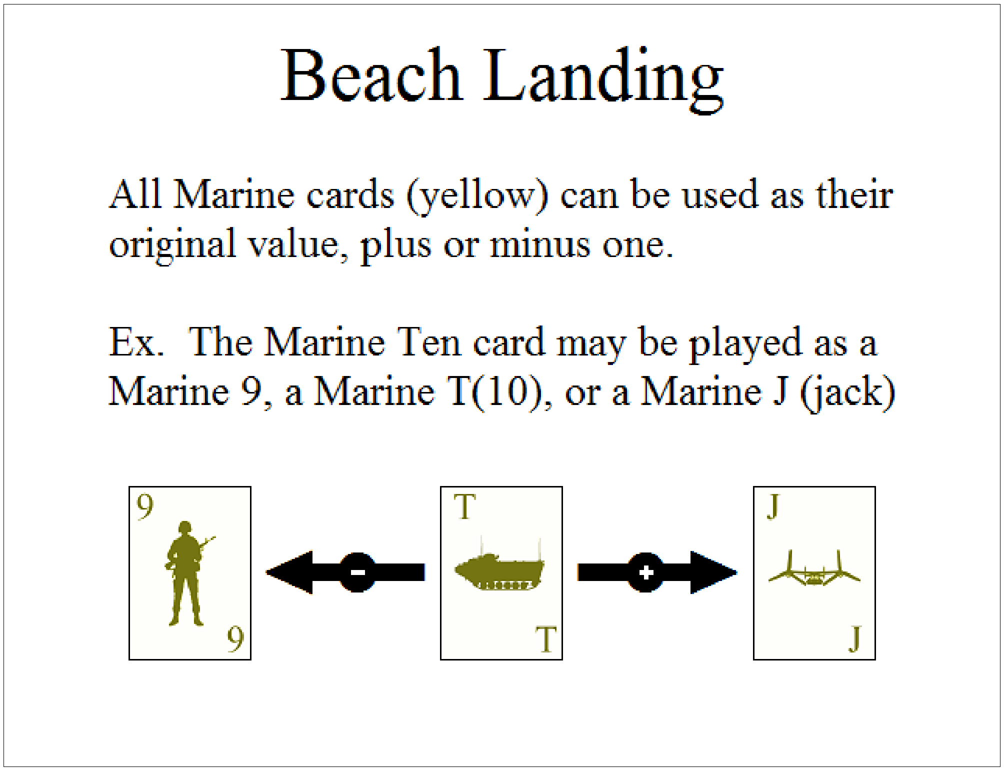
River

Features in Development for DECENT

DECENT Mission Cards

As the prototype continues to evolve, we are adding more constraints to the DECENT C2 game world, such as a deck of specialized mission cards. These cards act as additional constraints and modifications on the traditional Hold'em game, modifying the fundamental way each team's hole cards can be valued, and acting as additional ways for making the Hold'em analogy fit the environment of military C2. This allows us to increase the variability of play while still maintaining the fundamentals of Hold'em.

Some mission cards will give an advantage to certain military branches (the equivalent of traditional poker suits), while some will require or prohibit the use of specific card values. In addition to varying the DECENT's play, this also allows users with more flexible thinking skills to show their ability to adapt to different situations. Learning to value one's hole cards becomes much more difficult when the mission cards vary which cards are best. The mission cards will help to discover the most capable and flexible C2 thinkers.



Beach Landing

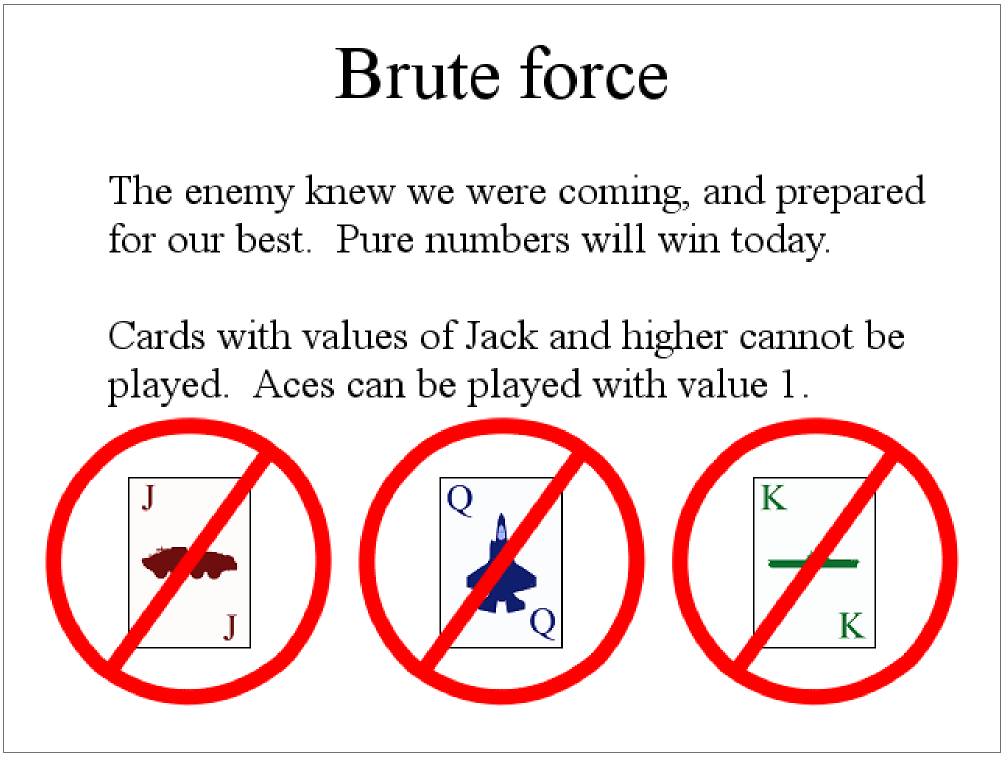
All Marine cards (yellow) can be used as their original value, plus or minus one.

Ex. The Marine Ten card may be played as a Marine 9, a Marine T(10), or a Marine J (jack)

Example mission card: Beach Landing.

In order to ensure that users who have not yet become accustomed to the game of Hold'em are not confused by this addition, DECENT can be played with or without these mission cards; the proctor may simply choose not to deal out a mission card. When played without the Mission Deck, DECENT's gameplay is exactly that of Hold'Em. Due to the static nature of the rules of Hold'em, we suggest that this only be done when teaching users the basic rules of Hold'em, and that the Mission Deck be added as soon as possible.

Because the mission cards are an additional element, not normally found in Hold'em, the structure of the gameplay must be modified when using them. Players get their cards, then a round of betting occurs. In traditional Hold'em, this would be followed by the flop (three public cards being revealed), but DECENT adds a mission step before the flop. After the first round of betting, the top card of the Mission Deck is revealed and made publicly available to all players. The mission card, itself, is placed in the public card area, below the flop and to the right of the river. To avoid any possible confusion between the playing cards and the mission card, we have made the mission cards twice the size of any other card. After the mission card is revealed, there is another round of betting, so that players have a chance to reevaluate their position, given the mission at hand. Play then continues as normal, with the flop.



Brute force

The enemy knew we were coming, and prepared for our best. Pure numbers will win today.

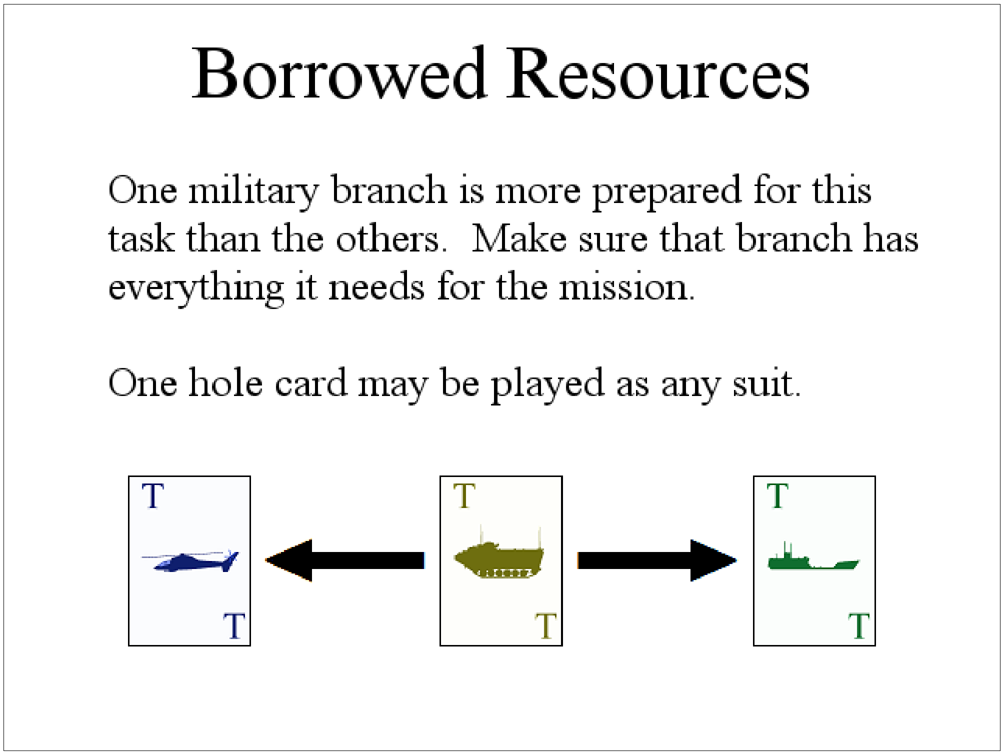
Cards with values of Jack and higher cannot be played. Aces can be played with value 1.

Example mission card: Brute Force.

In case the initial set of mission cards do not provide enough variability, or the users simply want new mission cards, additional mission cards can be made and added to the Mission Deck by an administrator (in OpenSim/SecondLife, this would be whoever owns the parcel of land). The most important thing when creating new mission cards is to make change the odds of getting any given hand without making that change too significant. For example, the Brute Force mission (above) invalidates all face cards, but does not actually cause a significant change to the odds of any hand. In effect, Brute Force just makes 10's the highest card and Aces the lowest card, while making Jacks, Queens, and Kings dead cards.

When making the visual component of a mission card, the current format is as follows:

Title in Bold
Flavor text, a description of the mission
Rules text, how the mission affects the game's rules
Image, an image which represents the rules text



Borrowed Resources

One military branch is more prepared for this task than the others. Make sure that branch has everything it needs for the mission.

One hole card may be played as any suit.

Example mission card titled Borrowed Resources, with narrative flavor text followed by rules text, and then an image of the rules text.

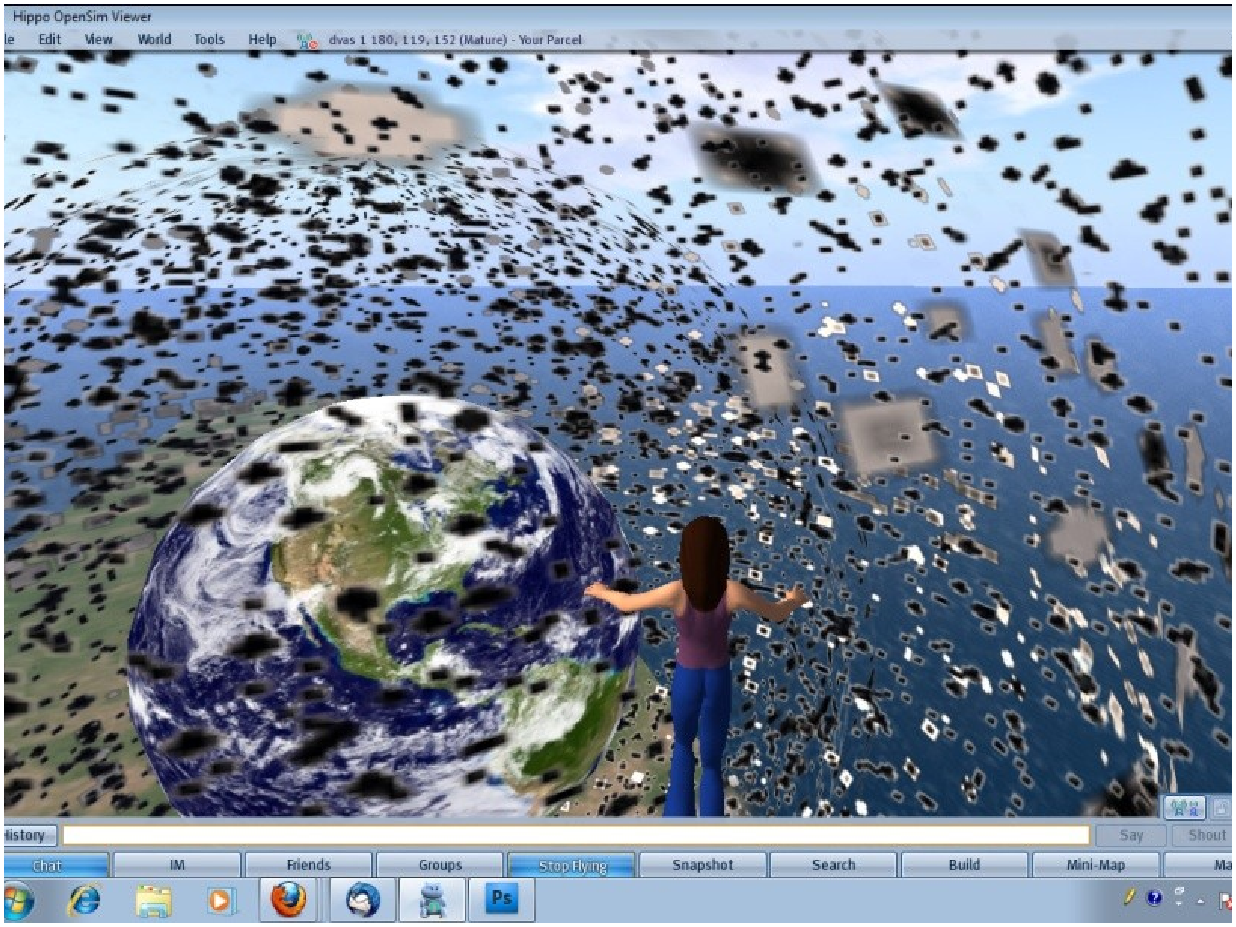
Extending DECENT C2 functionality

There are three areas of interest that we have begun to explore in order to extend the function and scope of our DECENT prototype. First, we are exploring how to more efficiently create, tailor, and personalize the appearance and behavior of in-world avatars. This extension is influenced in part by related efforts with the ELICIT environment at NPS that explore how and why to tailor software agents to emulate specific people in a C2 operation [Wynn, Ruddy, Nissen 2010]. Such capabilities allow us to model people assigned different roles or decision-making responsibilities.

RAN
H ISRAEL'S
D BARAK
LIVE
DOW 37.94
LiveLeak

Personalized models of C2 user-controlled avatars used in DECENT studies

Second, we have been extending DECENT to model and simulate space-centered and ocean-centered VWs. Our objective here has been to explore what kinds of new modes of interaction with modeled and simulated objects (or clusters of objects) may arise. For example, for a space-centered VW we model and simulate space debris fields that surround the Earth, and represent hazards/threats to the post-launch trajectory and orbital placement of new satellites, as well as potentially serving as a hiding space for other satellites or space-born threats. This capability allow DECENT users to place or "fly" their avatars into the modeled space debris field at different elevations and proximity, which allows for near-object visual inspection, its field of view, its current orbital trajectory, and more. Furthermore, these models can interoperate with the data importing and *spherecasting* (i.e., decentralized, net-centric control of shared spherical data visualizations) capabilities providing by NOAA's *Science on a Sphere* visualization systems (http://sos.noaa.gov/), used for visualizing weather and remote sensing data sets.



Space-centered VW with debris fields at different altitudes, trajectories, and sizes

Third, and in the opposite direction, we have also studied how to model selected regions of the ocean floor, so as to investigate C2 problems involving physical oceanography, underwater navigation and surveillance, as well as for associating remotely sensed data on ocean bottom (e.g., sonar mapping or acoustic tomography), sea life (fish, mammals, vegetation) and water conditions (thermoclines, haloclines, etc.). Basic technical challenges involve data acquisition and transformation from multiple sources, with different resolutions and different meta-data, all of which make creating plausibly accurate ocean floor and associated sea life modeling time-consuming. Accordingly, we have focuses our efforts to near-shore regions (e.g,. from Southern California coastline to Catalina Island) which also present challenges of physical space (~250 sq. miles, ocean depths to >1000') to data model (150x150 sq. meter virtual regions) for visualization and interaction in OpenSim "mega-regions" (interconnected regions with seamless boundaries). Our results so far though (e.g., http://www.youtube.com/watch?v=N4Kra3YGUYU) indicate challenges remain in achieving higher levels of ocean floor and sea life modeling are necessary before for engaging levels of verisimilitude and immersion in an oceanic VW are to be realized. Nonetheless, current results may be sufficient to support simple training exercises in near-shore underwater surveillance.

Extending DECENT to support the full range of modern CGVW capabilities and affordances for decentralized collaboration

Recent studies, such as those found in Bainbridge [2007, 2010] and Scacchi, *et al.* [2011], as well as emerging ventures commercializing emerging CGVW technologies, reveal a diverse a growing set of socio-technical affordances (i.e., new ways and means for net-centric, decentralized collaborative work) are both supported and being used in practices including:

- *Group presentation, communication, conferencing, and social interaction* – virtual meetings of many different kinds that incorporate a sense of place at a distance, along with relevant work media (interactive reports, documents, presentations, 3D models, etc. -- many examples in Second Life), often as an alternative to currently available solutions provided by WebEx, GotoMeeting, and Skype-based online meetings.
- *Prototyping and review* – interactive design, construction, and modification of virtual objects, composite systems, or mirror worlds as potential enterprise products or services that can be used in proposals or design presentations (e.g., see Lopes [2009], along with Encitra.com at http://www.youtube.com/watch?v=M4cYiHZqN90).
- *Training, education, rehearsal, learning* – providing VW-based simulators where people can enact simple/complex behaviors to understand how best to use/service a simulated device (e.g., see projects the Discovery Science Center and Intel [Scacchi 2010], also see examples from the *Little Big Planet* 2 game portal).
- *New product demonstration* – virtual product showroom, often with modeled/simulated interactive controls for selecting or customizing product features/attributes, such as color, appearance, accessories, etc.(see EON Reality online case studies).
- *Identity role-playing, team building, and other social processes* – often training centered VW but focusing on role-play, especially with attention to workplace diversity issues (cf. FutureWork Institute for identity role play, IBM efforts on team building).
- *Multi-media storytelling and avatar control/choreography* – creating video-audio animations (recordings) or live virtual action (live broadcast) of VWs for the purpose of illuminating narratives, telling stories through “machinima” for such purposes.
- *Mirrored worlds and memorialization* – creating virtual worlds that seek to strongly represent, primarily through visual means, some aspect, venue, or enterprise facility also found in the physical world. One reason for this may be to

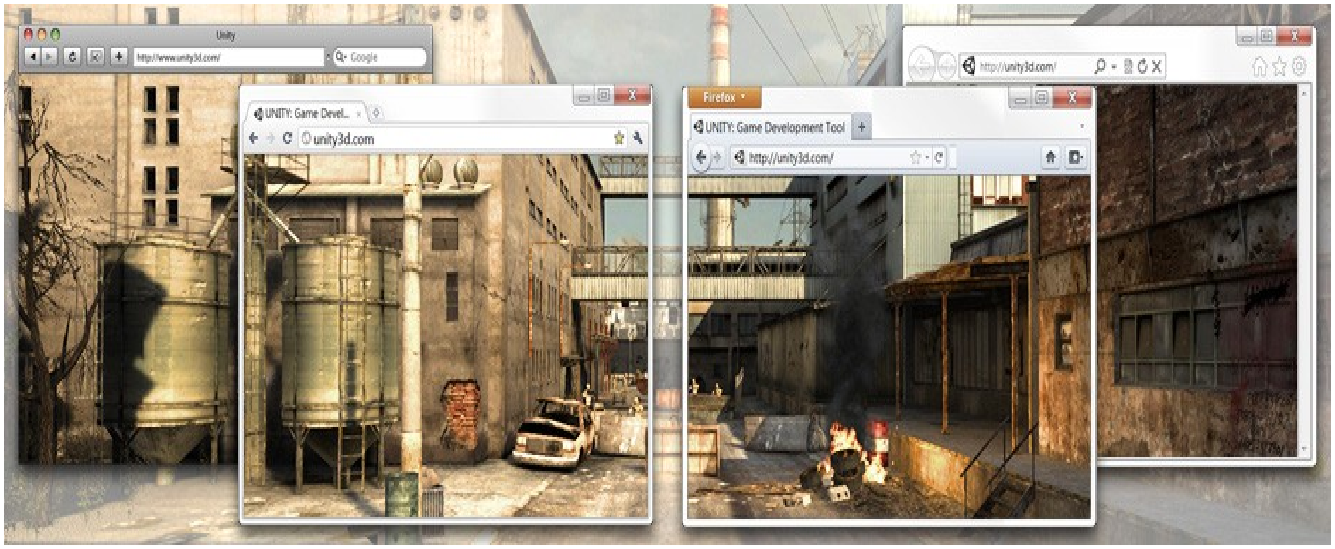
recreate or commemorate places that may no longer exist. Another reason may be to help new users more readily acclimate during their initial immigration from familiar physical worlds, to seemingly familiar virtual worlds, so as to enable follow-on activities like training or role-playing. Vws that mirror physical worlds may also include support for control devices that affect action in the corresponding physical world place, and vice-versa using augmented reality techniques.

- *Game development and/or modding* – software development kits or modding tools [Scacchi 2011] specific to a game engine (CGVW run-time environment) that streamline CGVW development using engine-specific content development tools (e.g., *Unreal Development Kit* for the Unreal 3 engine from Epic Games), rather than general-purpose programming tools or interactive development environments (e.g., Microsoft's *Visual Studio*).
- *Socio-technical process discovery* – ethnographic, virtual ethnographic, and computational (text data mining) approaches to discovering socio-technical processes emerging within CGVW work or play activities.
- *Enabling human behavior transformation* – CGVW designed to enable reflection, modification, and evolution of human behavior through repeated system-based training or usage settings, most clearly observed in CGVW for improvement of human health, ability, recovery, and self-managed care.
- *Modeling, analyzing, and developing complex intellectual property regimes accommodating multiple heterogeneous IP licenses* – understanding how developer or user-create objects or multi-object compositions within CGVW can be manipulated, exported, or imported across system boundaries, potentially for monetary or other forms of capital gain.

Affordances such as these can all support new ways and means of conducting collaborative research, development, and education in C2 domains, and thus collective represent a new engine for innovation and advancement, as well as to the creation, sharing, and enactment of new kinds of scientific knowledge.

DECENT Migration Candidate: *Unity3D*

In addition to adding functional improvements, we are currently in the process of considering another virtual environment platform to use as the permanent home of DECENT. Unity3D is a promising platform, having higher levels of graphical fidelity, mesh support, built-in HTML support, and multi-browser compatibility, though in a proprietary closed-source software format. These promising features will help migrate DECENT from a competent training prototype to a fully realized and distributable training system for use across the military. Unity3D also features support for smart phone and other mobile devices. This functionality would grant DECENT a convenience that few other training systems can boast.

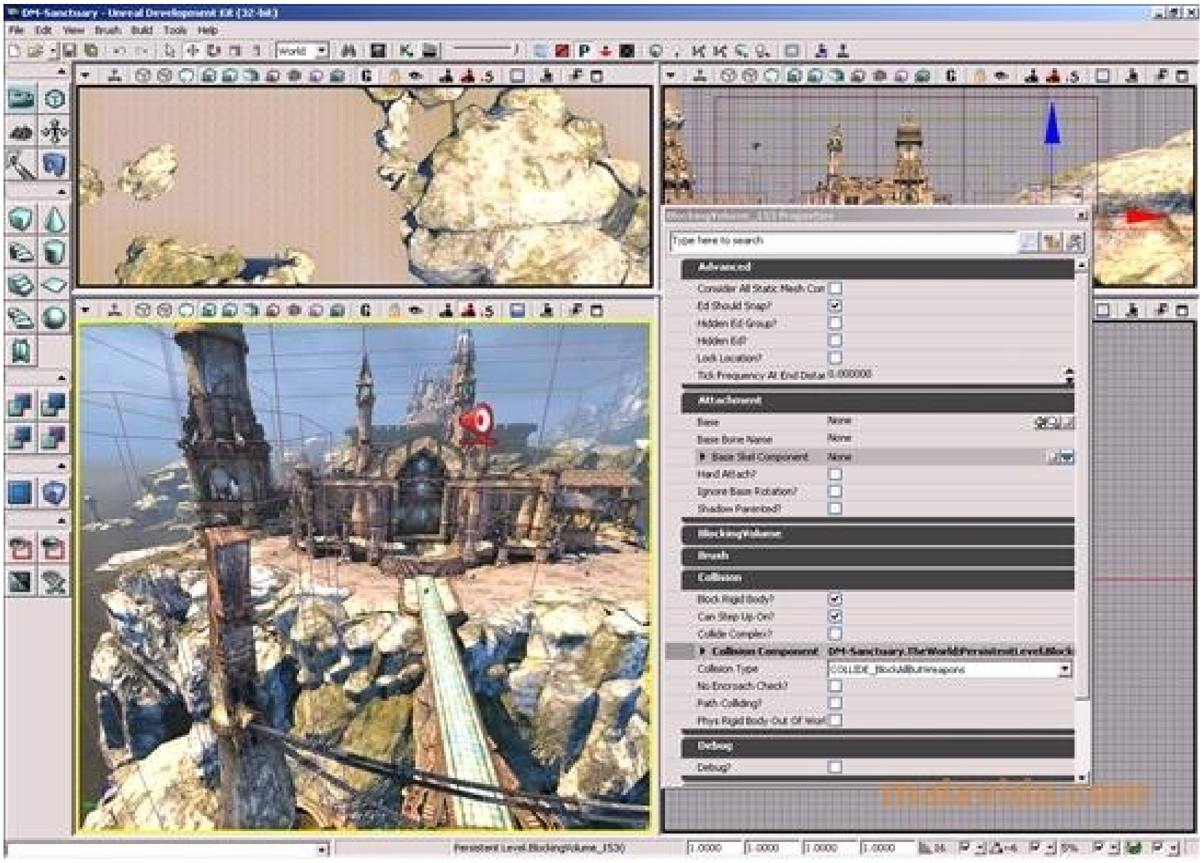


Unity 3D web clients running on different Web browsers (Apple Safari, Opera, Firefox, and Microsoft Internet Explorer).

Unity 3D Mobile client, available for iPhone and Android-based smart-phones. Also see, Back, Kimber, *et al.* [2010] for application to physical-virtual world interfacing and control.

DECENT Migration Candidate: Unreal Engine and Unreal Development Kit (UDK)

The Unreal Engine, created by Epic Games in North Carolina, offers the highest levels of graphic quality possible with commercially available software. With support for very high polygon count models, multiple layers of texture support, bump and normal mapping, and advanced lighting and rendering options, the Unreal Engine can provide support for levels of visual fidelity unparalleled by most software. While these features allow for higher levels of immersion and realism, it also requires artists with professional level talent, much like it main competitor, the CryENGINE 3 from CryTek (based in Germany). The Unreal Engine cannot have content easily added by laypeople, as OpenSim can. Assets are created in external 3d modeling packages such as Autodesk's 3DStudio Max or Maya. Last, Unreal games are typically configured to run in a network-centric, client-server architecture for multi-player game play, where clients may be PCs (Windows-based), smartphones (iPhone), or game consoles (Sony PlayStation 3, Xbox 360 Live), communicating with remote servers for exchanging in-game activity and multi-player chat/voice streams.


DM-Sanctuary - Unreal Development Kit (32-bit)
File Edit View Brush Build Tools Help
World
Type here to search
Advanced
Attachment
BlockingVolume
Brush
Collision
Debug
Persistent Level.BlockingVolume_1530

Consider All Static Mesh Con ☐
Ed Should Snap? ☑
Hidden Ed Group? ☐
Hidden Ed? ☐
Lock Location? ☐
Tick Frequency At End Distar 0.000000

Base None
Base Bone Name None
Base Skel Component None
Hard Attach? ☐
Ignore Base Rotation? ☐
Shadow Parented? ☐

Block Rigid Body? ☑
Can Step Up On? ☑
Collide Complex? ☐
Collision Component DM-Sanctuary.TheWorld:PersistentLevel.Block
Collision Type COLLIDE_BlockAllButWeapons
No Encroach Check? ☐
Path Colliding? ☐
Phys Rigid Body Out Of Worl ☐

Debug? ☐

Screenshot of Unreal Development Kit for modeling and updating an existing virtual world (lower left panel)

The graphic fidelity and mobility of Unity3D and the Unreal Engine come at the cost of interconnectivity. The departure from OpenSim would make transfers between DECENT and other OpenSim-based military training environments much more complicated, due to the loss of seamless Hypergrid movement.

Comparing OpenSim to Unity 3D and to Unreal

OpenSim most notable shortfalls compared to Unity and Unreal is a lack of support, at present, for smartphone clients, dearth of online tutorials with worked examples, and comparatively small developer community. The visual quality and physical realism of animated objects and characters in Unity and Unreal is well beyond what OpenSim can achieve, as its physics engine is still modest, compared to the advanced physics engines with hardware acceleration supported by game engines like Unreal. Similarly, OpenSim at present requires a separate, client-side browser (e.g., Hippo, Imprudence, or others compatible with Second Life), rather than an application that is designed to be plugged into an existing Web browser.

Both Unity and Unreal lack support for scalable and extensible hypergrids based on multi-processor server clusters, and both proprietary system lack the openness that open source software affords. With time and more participation in open source software development, and perhaps with external support from industry and government, the OpenSim environment may advance to near parity with their proprietary, closed source software alternatives.

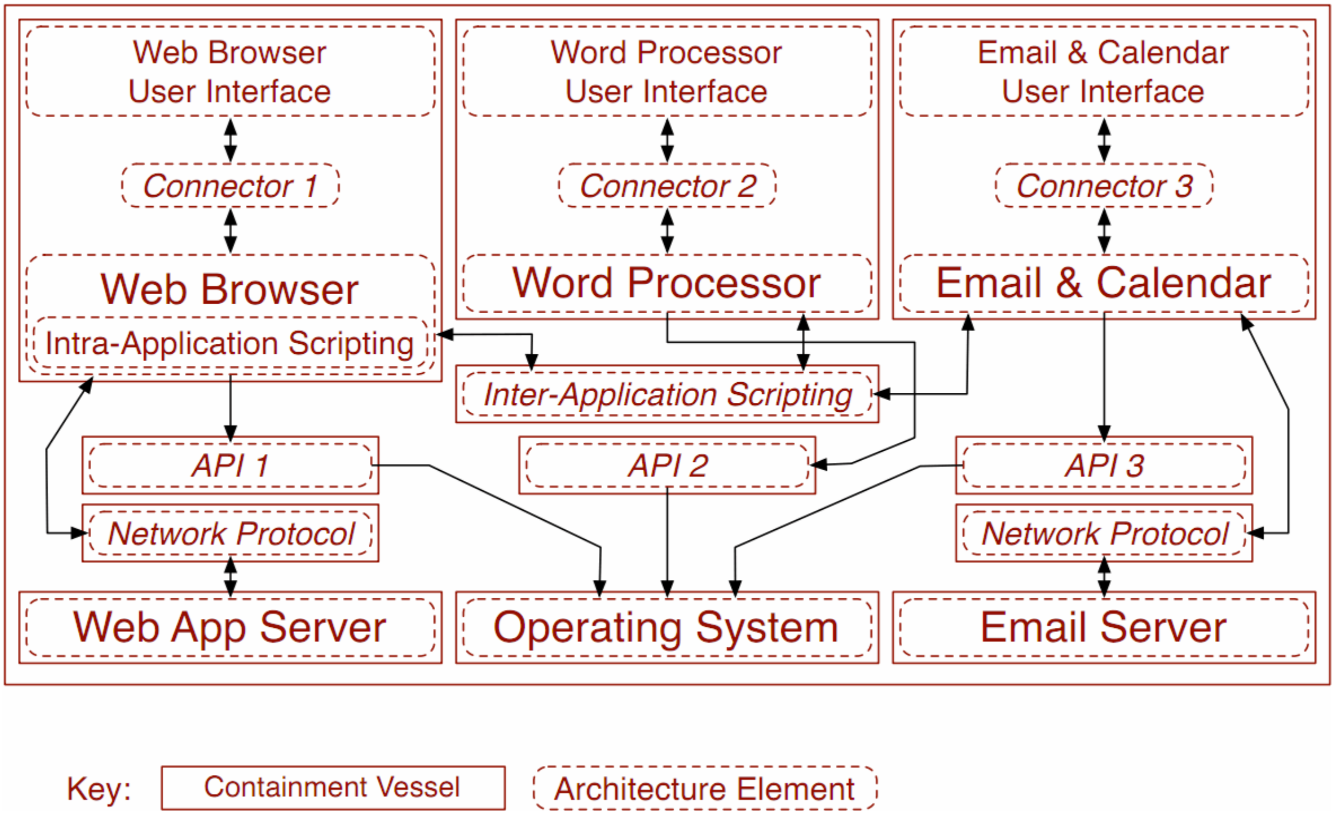
Overall, our view is that research investigations should evaluate and experiment with all leading candidate technologies, in order to note their comparative advantages and weaknesses, as this has been our choice to invest our effort with OpenSim, but to consider where, when, what, and why it may be appropriate to consider commercial alternatives to develop and deploy advanced command and control systems and virtual world environments.

Under-explored topics for DECENT

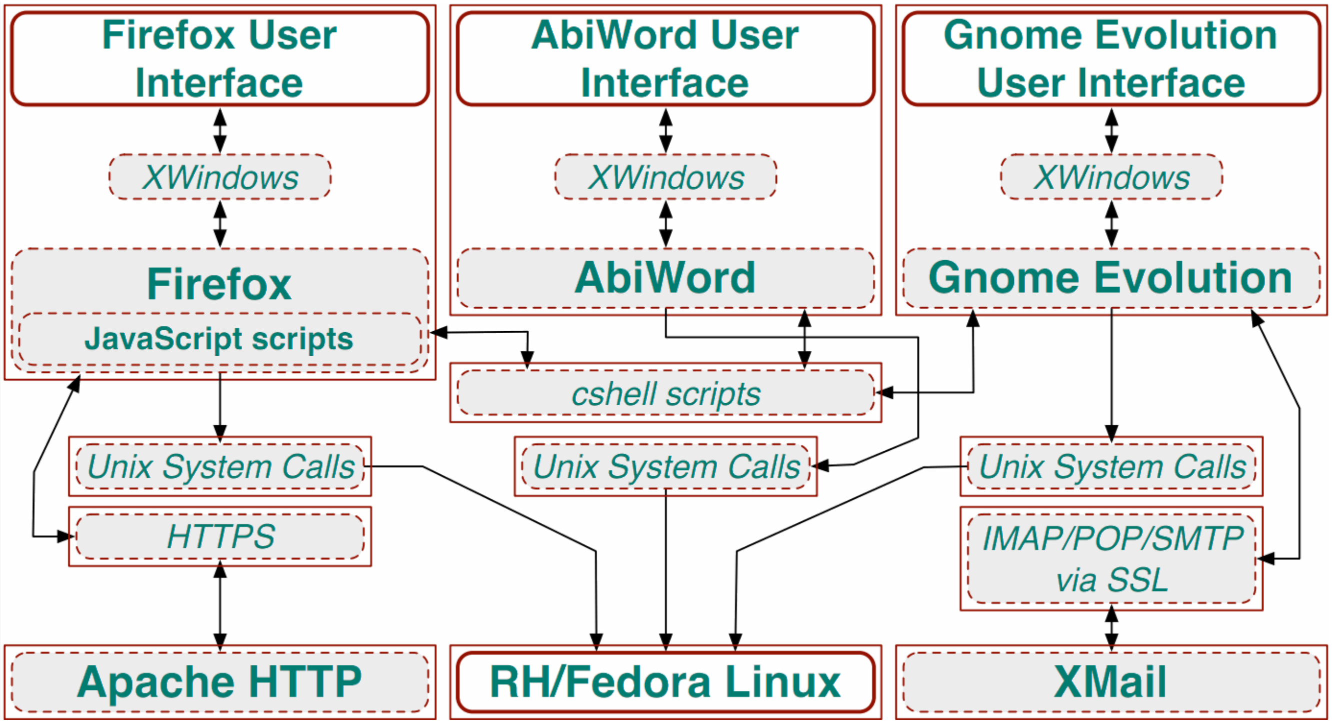
To no surprise, there are many other topics for investigation that were beyond the scope of resources available for us to explore. For example, most CGVW software technologies offer *little/no ready support for integration of external application programs or other software components*. So our efforts to model and simulate a decentralized C2 virtual world make no attempt to undertake such integration studies.

Next, the *underlying software architectures of CGVW are rarely disclosed or made open*, even when realized using open source software (OSS) components. So it is a major technical challenge to evaluate, assess, or compare at a deep technical level what architectural choices or trade-offs have been made in designing, building, and/or deploying an operational CGVW system. In simple terms, this makes comparing OSS CGVW technologies like Delta3D [see http://www.delta3d.org/], OpenSim, and any of dozens of OSS game engines accessible on the Web (e.g., via search at SourceForge.net) impractical at present. Thus, there is a basic research need to develop open architecture (OA) frameworks for specifying CGVW systems [Scacchi and Alspaugh 2008].

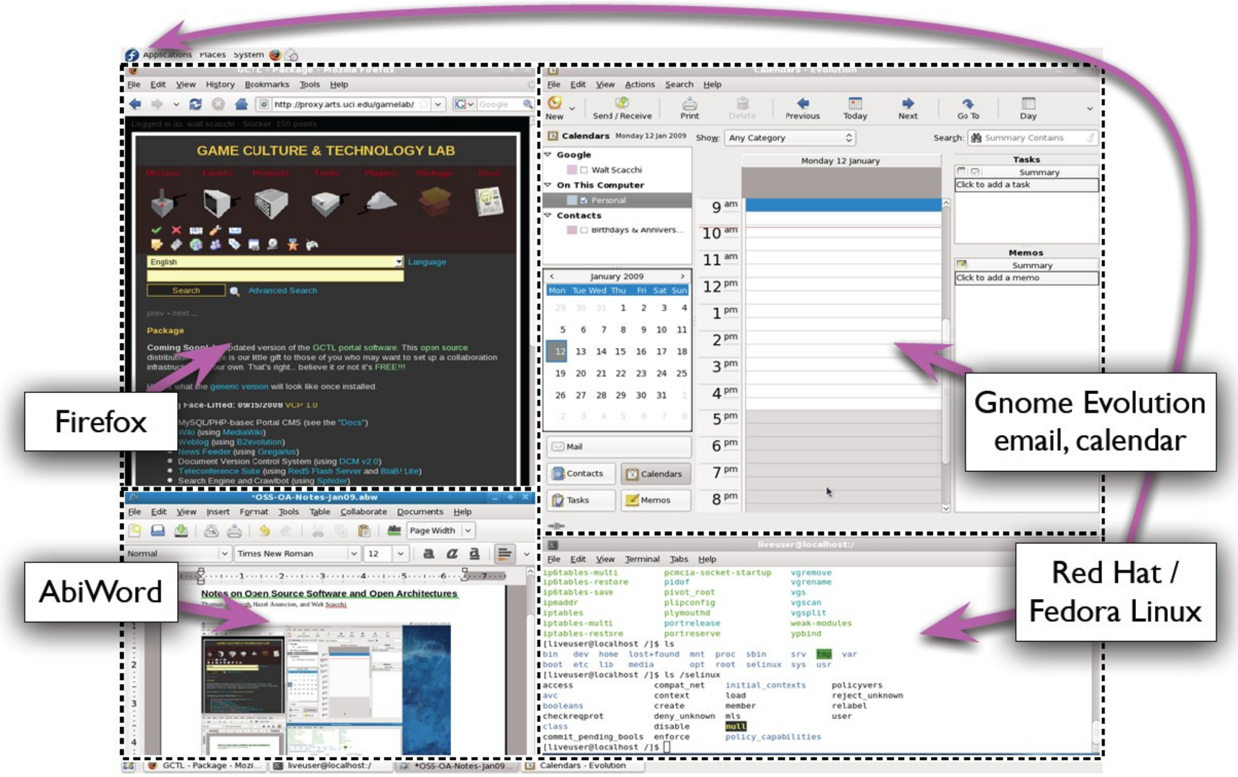
Similarly, the topic of *securing a VW for military C2 applications* is a major concern. VWs like DECENT are envisioned, developed, and extended as an open architecture system [Scacchi and Alspaugh 2008]. Recent advances in developing architectural level security schemes for designing, building, and deploying open architecture software systems are relevant and readily applicable to VWs applications, and well as Web-based system architectures, such as those for the C2RPC [2010]. For example, Scacchi and Alspaugh [2011] have developed and demonstrated a conceptual approach based on existing research technologies that can be used to specify, model, and analyze the security of an OA system with secure/contained elements, as suggested in figures below.



Secure architectural design for generic software applications common to C2 systems, from Scacchi and Alspaugh [2011].



Secure build-time architecture for generic software applications available for use in C2 systems, from Scacchi and Alspaugh [2011].



Coming Soon! ... updated version of the GCTL portal software. This open source distribution ... is our little gift to those of you who may want to set up a collaboration infrastruct... our own. That's right... believe it or not it's FREE!!!

... what the generic version will look like once installed.

... Face-Lifted: 09/15/2008 VCP 1.0

- MySQL/PHP-based Portal CMS (see the "Docs")
- Wiki (using MediaWiki)
- Weblog (using B2evolution)
- News Feeder (using Gregarius)
- Document Version Control System (using DCM v2.0)
- Teleconference Suite (using Red5 Flash Server and BlaB! Lite)
- Search Engine and Crawlbot (using Sphider)

```
ip6tables-multi        pcmcia-socket-startup   vgremove
ip6tables-restore      pidof                   vgrename
ip6tables-save         pivot_root              vgs
ipmaddr                plipconfig              vgscan
iptables               plymouthd               vgsplit
iptables-multi         portrelease             weak-modules
iptables-restore       portreserve             ypbind
[liveuser@localhost /]$ ls
bin   dev  home  lost+found  mnt  proc  sbin     srv  tmp  var
boot  etc  lib   media       opt  root  selinux  sys  usr
[liveuser@localhost /]$ ls /selinux
access                 compat_net     initial_contexts     policyvers
avc                    context        load                 reject_unknown
booleans               create         member               relabel
checkreqprot           deny_unknown   mls                  user
class                  disable        null
commit_pending_bools   enforce        policy_capabilities
[liveuser@localhost /]$ 
```

Sample run-time software application deployment for generic software applications available for use in C2 systems, from Scacchi and Alspaugh [2011].

Last, most CGVW software technologies provide very basic security mechanisms, and thus are quite amenable to remote attacks, penetration, and possible code injections. Furthermore, CGVW may allow for new modes of malware that may enable activities including avatar impersonation or remote control (e.g., who/what is controlling this avatar, and with what authorization?) and other ill-defined vulnerabilities. So CGVW technologies should only be considered for experimental purposes until more robust security capabilities are in place, tested, validated, monitored, and evolved [cf. Scacchi and Alspaugh 2011] and are not at this point of technology development ready for supporting real C2 applications, such as those in the C2RPC [Garcia 2010, Gizzi 2011]. However, they may be appropriate for experimentation with future C2 system architectures that may include Web-based and VW-based software elements, that may be accessible from smart-phones, as well as open to access, monitor, and control physical devices and sensors deployed in physical world settings, whether on land, sea, air, or space.

Conclusions and recommendations for future study

This report seeks to describe and document the results of a small-scale one year research study that investigates how virtual world concepts, techniques, and tools can be employed to support experimental/prototyping efforts for command and control applications. We reported on our efforts to investigate and prototype a VW we called DECENT as a platform for exercising and assessing the potential of a VW-based approach to decentralized C2, as well as to compare our efforts with others closely related. Overall, we found this effort gave rise to very promising results that point to additional opportunities and system extensions for new ways to consider the potential of decentralized approaches to C2 that merit further systematic investigation and experimentation.

We also found that a number of computer games also suggest system features applicable to C2, as well as distributed individual/group problem-solving, using popular card games. Again, these may be candidates for further systematic study.

We similarly identified and compared a small set of related technologies that could be compared to the efficacy of the VW technologies that we employed (OpenSim, an open source software toolkit for building, navigating, and socially interacting). OpenSim provides many interesting affordances, some which are common to most VWs. Understanding the potential for how VWs may be designed, built, deployed, and evolved seems that a significant opportunity area for further study. In addition, there is still need to determine how best to evaluate and compare the efficacy of VWs that seek to mirror physical sites or physically-located human problem solving and social interaction. There is also need to evaluate and compare the efficacy of alternative VW and computer game development technologies, whether open source software, or proprietary, closed source software.

Last, we also reported on topics that essentially have little/no published research study or results that are central to the continuing development of VW technologies for C2 applications. Principal among these are topics surrounding the security of VWs intended for experimental studies or future applications to the C2 domain. VWs are still very early in their technical development, and far from ready for deployment in actual application settings. Nonetheless, these technologies do merit ongoing study and investment as they offer new categories of affordances that can enable/support both centralized and decentralized approaches to C2 in ways that legacy C2 technologies and approaches cannot. Decentralized VW-based approaches may offer the potential to substantially reduce the cost and dramatically shorten the time to design, build, and deploy C2 systems that embrace new generations of low-cost, mobile technologies that future C2 workforces may expect, whether for use in physical or virtual/cyberspace worlds. So much remains to be studied, and time for appropriate and realistic research investments is at hand. In the near-term, such research is likely to still be considered risky, but the longer-term benefits may most quickly arise and be demonstrated through such near-to-mid term research investments. This is the future opportunity now at hand.

References

Albert, D.S. and Hayes, R.E., (2003). *Power to the Edge: Command and Control in the Information Age*, Command and Control Research Program, Washington, DC, http://www.dodccrp.org/files/Alberts_Power.pdf

Back, M., Kimber, D., Rieffel, E., Dunnigan, A., Liew, B., Gattepally, S., Foote, J., Shingu, J. and Vaughan, J. (2010). The virtual chocolate factory: mixed reality industrial collaboration and control. In *Proc. Intern. Conf. on Multimedia* (MM '10). ACM, New York, NY, USA, 1505-1506. Also see, http://www.youtube.com/watch?v=-J5qi8yquoo for video recording.

Bainbridge, W.S. (2007). *The Scientific Research Potential of Virtual Worlds*, Science, 317, 472-476, 27 July 2007.

Bainbridge, W.S. (Ed.) (2010a). *Online Worlds: Convergence of the Real and the Virtual*, Human-Computer Interaction Series, Springer-Verlag London Limited.

Bergin, R., Adams, A., Junior, R., Hudgens, B., Chinn Yee Lee, J., and Nissen, M. (2010). Command & Control in Virtual Environments: Laboratory Experimentation to Compare Virtual with Physical, *Proceedings 15th International Command & Control Research & Technology Symposium*, Paper 075, Santa Monica, CA.

C2RPC (2010). Command and Control Rapid Deployment Continuum Overview, http://www.afcea-sd.org/wp-content/uploads/2010/12/YoungAFCEA_C2RPC.pdf

Garcia, P. (2010). Maritime C2 Strategy: An Innovative Approach to System Transformation, *Proceedings 15th International Command & Control Research & Technology Symposium*, Paper 147, Santa Monica, CA.

Gizzi, N. (2011). Command and Control Rapid Prototyping Continuum (C2RPC) Transition: Bridging the Valley of Death, *Proceedings 8th Annual Acquisition Research Symposium,* Vol. 1, Naval Postgraduate School, Monterey, CA.

Hudson, K., and Nissen, M. (2010). Command & Control in Virtual Environments: Designing a Virtual Environment for Experimentation, *Proceedings 15th International Command & Control Research & Technology Symposium*, Paper 052, Santa Monica, CA.

Korolov, Maria, (2011). Second Life vs. OpenSim. *Hypergrid Business*. 28 May 2011.

http://www.hypergridbusiness.com/2011/05/second-life-vs-opensim/. Also included in Appendix 3.

Lopes, C.V., (2009). The Massification and Webification of Systems' Modeling and Simulation with Virtual Worlds, in *Proc. European Software Engineering Conference/Foundations of Software Engineering (ESEC-FSE'09)*, 63-70, Amsterdam, The Netherlands, ACM Press. http://portal.acm.org/citation.cfm?id=1595696.1595708

MOSES, (2011), *Military Open Simulator Enterprise Strategy*, http://fvwc.army.mil/moses/

MOSES Server Map, (2011). http://fvwc.army.mil/moses/server-allocation-map/

Neville, James, (2011). "Army Extends MOSES to Other Researchers". *Hypergrid Business*. 21 May 2011. http://www.hypergridbusiness.com/2011/05/army-extends-moses-to-other-researchers/

OpenSim, (2011). *The Open Simulator Project,* http://opensimulator.org/

OpenSim Hypergrid, (2011). *The OpenSim Hypergrid*, http://opensimulator.org/wiki/Hypergrid

Scacchi, W. (2010). Game-Based Virtual Worlds as Decentralized Virtual Activity Systems, in W.S. Bainbridge (ed.), *Online Worlds: Convergence of the Real and the Virtual,* Human-Computer Interaction Series, Springer-Verlag London Limited, 225-236.

Scacchi, W., (2011). Modding as a Basis for Developing Game Systems, *Proc. 1st. Workshop Games and Software Engineering* (GAS'11), 33rd Intern. Conf. Software Engineering, Waikiki, Honolulu, HI, May 2011.

Scacchi, W. and Alspaugh, T. (2008). Emerging Issues in the Acquisition of Open Source Software within the U.S. Department of Defense, *Proc. 5thAnnual Acquisition Research Symposium*, Vol. 1, 230-244, NPS-AM-08-036, Naval Postgraduate School, Monterey, CA

Scacchi, W. and Alspaugh, T. (2011). Advances in the Acquisition of Secure Systems Based on Open Architectures, *Proceedings 8th Annual Acquisition Research Symposium,* Vol. 1, Naval Postgraduate School, Monterey, CA.

Scacchi, W., et al. (2011). *The Future of Research in Computer Games and Virtual*

Worlds, NSF Workshop Report, February 2011.

Wynn, D., Ruddy, M, and Nissen, M., (2010). Command & Control In Virtual Environments: Tailoring Software Agents To Emulate Specific People, *Proceedings 15th International Command & Control Research & Technology Symposium*, Paper 019, Santa Monica, CA.

Appendix 1: The Texas Hold'em game

Hold'em is a form a poker, popularized by the movie, Rounders, and the broadcasting of the World Series of Poker. Play typically involves between two and ten players at a single table, and the flow of the game is controlled by a dealer. At the beginning of play, one player is randomly given the Dealer Chip. The dealer deals cards as if he or she was sitting in the seat of the player with the Dealer Chip. Cards are dealt starting with the player to the left of the Dealer Chip.

At the beginning of each hand, the player to the left of the dealer pays half the minimum bet; this is called the Small Blind. The next player pays the minimum bet; this is called the Big Blind. The Blinds ensure that money is always in the pot, but that every player does not have to spend money every hand.

Each player is then dealt two cards, face down; these cards are called hole cards. Hole cards are unique to each player, and may be used in conjunction with the public cards to make a five-card poker hand.

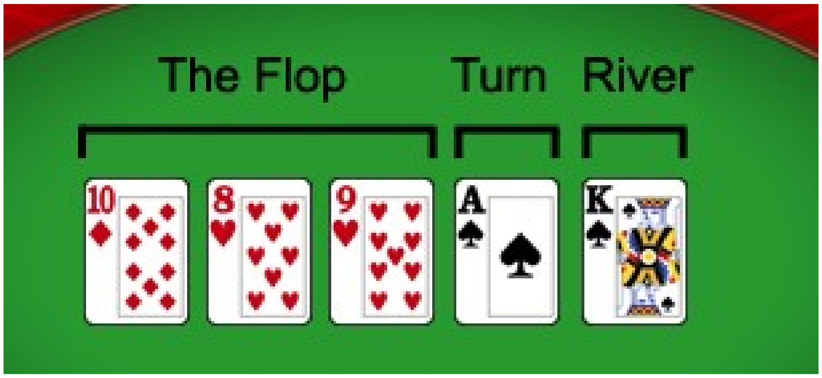
Hole cards.

After the hole cards are dealt, there is a round of betting, starting with the player to the left of the Big Blind. Players may call (add money equal to the current bet or blind), raise (add money in excess of the current bet or blind, increasing the current bet), fold (pay no money and forfeit play this hand), or check (add no money and continue playing the hand; this may only be done if the player is there is no current bet).

Once all betting is done, the dealer removes one card from the remaining deck, face down; this is called burning a card. The dealer then deals three cards, face-up, into the public card area; these three cards are called the Flop. After the flop is revealed, there is another round of betting, starting with the small blind.

After the second round of betting is done, the dealer burns another card, then deals one card, face-up, into the public card area; this card is called the Turn. After the turn, there is another round of betting, again starting with the small blind.

After the third round of betting is done, the dealer burns another card, then deals a final card, face-up, into the public card area; this card is called the River, and is the last card revealed. At this point, all public cards are known. After the river, there is another final of betting, again starting with the small blind.



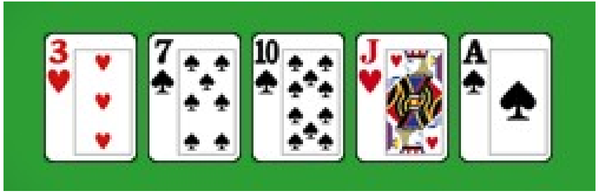
The flop, turn, and river.

Once the last round of betting is finished, each remaining player, starting with the player who made the last bet, may either reveal his/her cards. If a player wishes, he/she may forfeit the hand by not choosing to not reveal his/her cards; this is known as mucking. Once all remaining player's hands are revealed or mucked, the player with the best five-card poker hand wins the hand, and gets all the money in the pot.

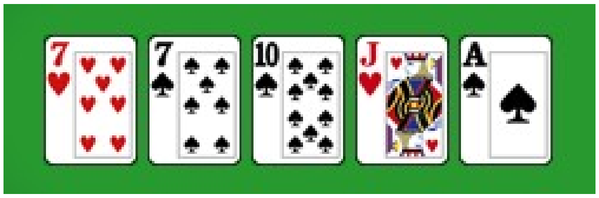
In order to ensure that no player's position has an unfair advantage, the Dealer Chip rotates after each hand, so that each player has the dealer's position in order, and each player has a turn as the small and big blinds.

Hand ranking is as follows, starting with the weakest hand possible:
High card, Pair, Two-pair, Three of a kind, Straight, Flush, Full House, Four of a kind, Straight Flush, Royal Flush

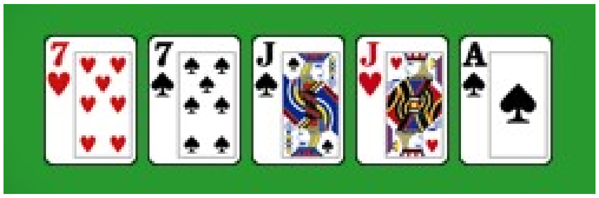
High card
No pairs, flushes, or straights.



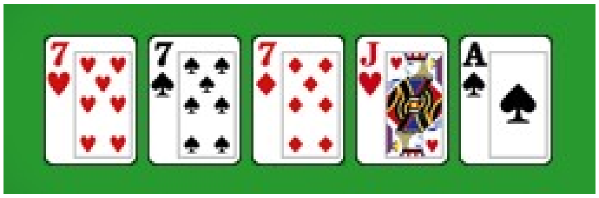
Pair
Two cards of the same value.



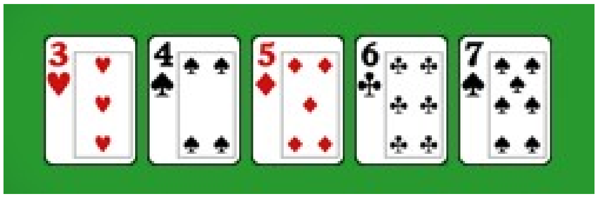
Two-pair
Two sets of two cards with the same value.



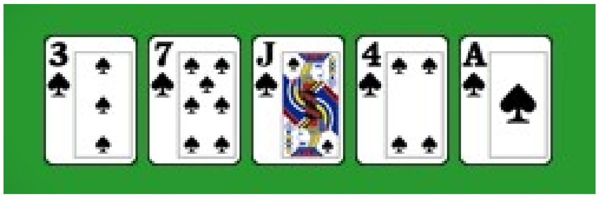
Three of a kind
Three cards of the same value.



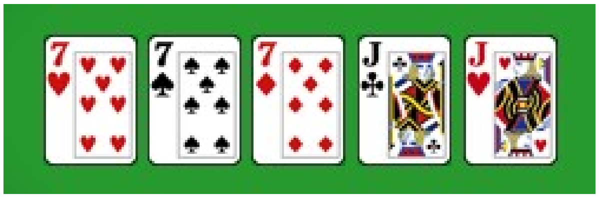
Straight
Five cards in consecutive order.



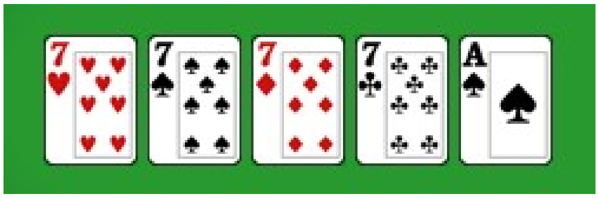
Flush
Five cards which share a suit.



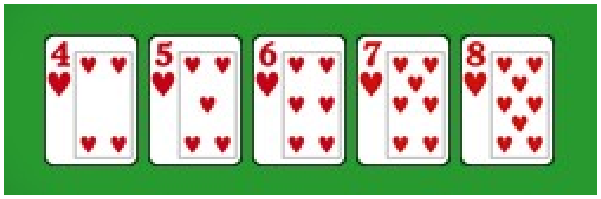
Full house
Three cards of one value and two cards of another value.



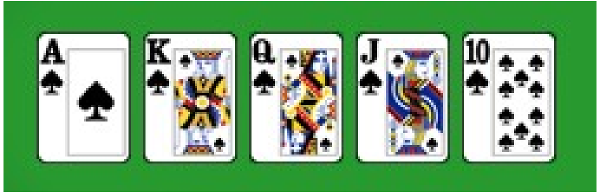
Four of a kind
Four cards with the same value.



Straight Flush
Five cards in consecutive order and which share a suit.



Royal flush
A straight flush, starting with a 10 and ending with an Ace.



Appendix 2: Presentation on Mission Resource Scheduling game

As part of our collaboration with the Intelligence Systems Division of one of our aerospace/defense industry partners, we at UCI engaged in a study of games to improve the analytic performance of generating information needs required for intelligence collection. Immersive game-based virtual worlds offer the potential to teach [Gee 2007, Scacchi 2008], simulate, and support analyst groups by providing environments for learning, training, and collaboration in complex scientific or technical work environments [Henckle and Lopes 2009, Lopes 2009, Scacchi 2010]. Prior studies of complex human performance (e.g., performing complex problem solving and effective physical dexterity during laparoscopic surgery) that were preceded or not with computer game play has shown significant performance improvements for those who played games [Rosser, et al. 2007]. In addition, other studies have shown that different games, visual interaction methods, and overall play experience further mediate human motivation and performance [Flores, *et al.* 2008]. Thus, there is need for carefully designed computer game that employs an effective information visualization and interaction scheme that can best facilitate individual and group human performance improvement of collaborative intelligence collection and related C2 environment tasks [Mark 2002].

Potential approach and concept

We systematically studied three areas of research: 1) The effectiveness of playing games to derive collection plans from information needs, 2) The degree of game based analogy to real world simulations for analysts to generate collection plans that meet information needs, 3) The effectiveness of multi-player competitive games to improve group solution generation for intelligence collection planning. Results in these three areas provided insights to the quantitative effectiveness of games for improving a key Intelligence Community enterprise priority to enhance community mission management. In addition, these problem areas need to analyze and compare both individual and small group performance.

Realizing our research study entailed a set of activities that included a review of relevant research literature, development of analytical concepts and hypothesized associations, design of an experiment, pilot testing the experiment, recruiting subjects, conducting the experiment, analyzing experiment data, and publishing the results from the experiment and overall study.

Critical technical issues

At UCI, we conducted an informal system development experiment along the following lines. The UCI team would create an immersive game-like visual environment involving a scheduling activity; the design of this virtual world needed careful thought, but it incorporated intuitive visualization and game elements. For example, Professor Lopes at UCI has been working on immersive visualizations of basic Computer Science algorithms, like bubble sort, quick sort, etc., and the results have been surprising, based on the effect that the visualization had made on understanding the intuitive information processing steps involved in these algorithms. We anticipated similar gains in insight and decision-making performance may arise for people who learn to play resource scheduling and information collection games prior to performance at work.

We could next qualitatively test the power of visualizations in this environment for enhancing the effectiveness that you mention above by comparing the individual and group performance with different experimental treatments (e.g., no visualization support, traditional text and 2D graphics, and immersive 3D game-based virtual world). However, we recognized the need to systematically measure individual performance and group performance. Further, we also need to derive performance metrics as well as analyze problem-solving and decision-making strategies, as well as negotiation strategies in the group context.

Expected performance, robustness

We think that this will provide insights to the quantitative effectiveness of games for improving a key Intelligence Community enterprise priority to enhance the management of intelligence collection and related missions. The basis of our efforts led to a decision to focus on the integration of a common, well-known poker card game, Texas Hold'em, to serve as a medium through which we could setup and study how well different users (individuals or groups) could act to realize individual or shared mission resource allocation outcomes. This effort is still in progress, though all the technical elements are in place, but the resources needed to conduct game play-based studies are needed to continue this effort, as are resources needed to further refine and extend the functionality of our virtual command and control environment to integrate other information system applications (e.g., those providing or simulating ISR kinds of data sources for fusion or knowledge management). Thus, this effort awaits provision of additional support for ongoing study and experimentation.

References

E. Flores, G. Tobon, *et al*., Improving Patient Motivation in Game Development for Motor Deficit Rehabilitation, *ACM 2008 Intern. Conf. Advances in Computer Entertainment*, 381-384. http://portal.acm.org/citation.cfm?id=1501839

J.P. Gee, (2007). *What computer games can teach us about learning and literacy, 2nd Edition*, Palgrave Macmillan, New York. http://www.amazon.com/Video-Games-Learning-Literacy-Second/dp/1403984530/ref=ntt_at_ep_dpi_2

A. Henckel and C.V. Lopes (2009). StellarSim: A plug-in architecture for scientific visualization in virtual worlds, in *Proc. 1st Intern. Conference on Facets of Virtual Environments (FAVE)*, Berlin, Germany. http://www.ics.uci.edu/~ahenckel/StellarSim.pdf

C.V. Lopes, (2009). The Massification and Webification of Systems' Modeling and Simulation with Virtual Worlds, in *Proc. European Software Engineering Conference/Foundations of Software Engineering (ESEC-FSE'09)*, 63-70, Amsterdam, The Netherlands, ACM Press. http://portal.acm.org/citation.cfm?id=1595696.1595708

G. Mark, (2002). Extreme Collaboration, *Communications of the ACM*, 45(6), 89-93. http://www.ics.uci.edu/~gmark/extreme%20collaboration.pdf

J.C. Rosser Jr, P.J. Lynch, L. Cuddihy, D.A. Gentile, J. Klonsky, and R. Merrell, (2007). The Impact of Video Games on Training Surgeons in the 21st Century, *Archives of Surgery*, 142(2), 181-186. http://archsurg.ama-assn.org/cgi/content/short/142/2/181

W. Scacchi (2010). Game-based Virtual Worlds as Decentralized Virtual Activity Systems, in W.S. Bainbridge, (Ed.), *Online Worlds: Convergence of the Real and the Virtual*, Springer, New York, 225-236. http://www.ics.uci.edu/~wscacchi/Papers/New/Scacchi-OnlineWorlds-Chapter-2009.pdf

W. Scacchi, R. Nideffer, and J. Adams (2008). A Collaborative Science Learning Games for Informal Science Education: *DinoQuest Online*, in IFIP International Federation for Information Processing, Volume 279; *New Frontiers for Entertainment Computing*; P. Ciancarini, R. Nakatsu, M. Rauterberg, M. Roccetti (Eds.); Boston: Springer, 71–82, 2008. http://www.ics.uci.edu/~wscacchi/Papers/New/Scacchi-Nideffer-Adams-2008b.pdf

Appendix 3: SecondLife versus OpenSim

Maria Korolov
http://www.hypergridbusiness.com/2011/05/second-life-vs-opensim/

28 May 2011

Second Life	**OpenSim**
If you're looking for a social group, you can find anything you want. Political activists. Role players. Educators. Artists. Designers. Sexual minorities.	If you already have a group, you can bring your group to a grid — or start your own grid.
Large user base for event promoters looking for audiences, retail merchants looking for customers, marketers looking for influence, and activists looking for publicity.	Small groups of widely dispersed users. Some merchants may find niche pockets of under-served customers on new grids.
$295 a month per region, plus $1,000 setup fee. Texture uploads cost money.	Free if you run it on your own computer or USB stick. Starting at $9.95 a month per region with no setup fee for professional hosting. Up to $100 a month for a high-performance region, or on a high-end commercial grids. Texture uploads are typically free.
Unreliable for mission-critical business meetings due to problems with voice, unexpected region crashes or restarts, login problems, lag, and other issues. But stable enough for classes, in-world talk shows, fund raisers, and social gatherings.	Some enterprise users report higher stability than Second Life, due to high-end hardware and plenty of bandwidth. Typical commercial grids have less stability, however. Open grids — where users can connect regions hosted on home computers — can have extremely poor performance and stability on those regions. Grids running older software can also have instability issues due to bugs that have since been fixed. Grids running untested, experimental versions of OpenSim can also suffer due to the appearance of new bugs that haven't yet been fixed.

Second Life	OpenSim
Lots of content available both through in-world merchants and freebie shops and through a gigantic web-based marketplace.	Content slowly becoming available from in-world merchants on commercial grids and for download from websites. As grids mature, infringing content in freebie stores on open grids is discovered and removed, and replaced with non-infringing original content. Some merchants beginning to offer DRM-free content on websites and in hypergrid-enabled shopping areas.
Some content protection and digital rights management technology available to protect content creators, but content theft is still frequently reported.	Content protection is completely up to the grid owner. Some commercial grids, including Avination and InWorldz, have the same level of content protection as Second Life. Other grids offer more freedom to users, allowing them to back up their inventories or regions. Private grids run by companies, schools, groups or individuals can put as much or as little content protection in place as they want or need.
Linden Lab owns all the content in Second Life. Users just get a license to use it. Linden Lab can remove individual items from user inventories, entire regions from the grid, or shut down any user account at any time.	Individual grid owners determine the content use policies on their grids. Some commercial grids follow Linden Lab model. Others allow their residents to have rights to their content. Meanwhile, companies, schools, groups and individuals who run their own grids have full ownership of those grids — similar to the way they have full ownership of their websites.
Linden Lab allows individual users to make backups of content in which they themselves have created every part of the object.	Individual grid owners determine backup policies. Some, like InWorldz and Avination, mirror those of Second Life. Other grids allow more backup options. Owners of private grids can backup any objects, can make backups of entire inventories of individual users, of entire regions, or of the whole grid.
All Second Life land is provided by a single vendor, Linden Lab. However, resellers and	More than 50 different vendors rent land on individual grids or as standalone regions, mini-grids, or run full grids for customers — not counting in-world developers who subdivide and resell land in individual commercial grids.

Second Life	OpenSim
middlemen may step in to subdivide or improve virtual land.	
Second Life uses the commercial Havoc physics engine.	OpenSim grid owners can take advantage of OpenSim's modular nature to use any of a number of either open source or commercial physics engines. Most OpenSim grids tend to use the default physics provided with the OpenSim software, ODE, which is inferior to that available in Second Life particularly when it comes to vehicle physics.
Second Life uses LSL, the Linden Scripting Language.	OpenSim currently supports more than 95 percent of all LSL commands, and adds a number of unique OSSL commands. Users can also write their own scripting commands and include them as an OpenSim module — or create a completely new scripting engine. InWorldz, for example, has deployed its own scripting engine, called Phlox.
Second Life uses a voice system from Vivox.	Grid owners can also choose to install modules for a couple of different free voice systems, including Freeswitch and Whisper/Mumble, or buy a commercial license. Avination, for example, has a license to Vivox, the same voice system used in Second Life.
Second Life uses the Linden dollar currency. Users can also make off-grid transactions via PayPal or PayPal Micropayments.	OpenSim grid owners can create their own in-grid currency. Many commercial grids, including Avination and InWorlds, have done this. As long as all regions on the grid are hosted by the grid owners, this is as secure as the currency system in Second Life. OpenSim grids can also install modules that use the multi-grid OMC currency from Virwox, or enable in-world payments via PayPal or PayPal Micropayments. With both OMC and PayPal, final confirmation of transactions take place on a webpage, for maximum security.
There are several third-party exchanges that trade Linden dollars for US	The OMC currency can be traded for US dollars, Euros, and Linden dollars on the Virwox exchange. It is accepted on 28 different grids, and if a particular grid goes out of business,

Second Life	OpenSim
dollars or Euros. However, officially, the currency is not actually owned by users but is licensed, and Linden Lab can terminate that license at any time without a refund.	the currency will still retain its value. Avination’s currency is also now traded on Virwox, but is unlikely to retain value if the grid closes. Other in-grid currencies can only be bought from their grid owners and may or may not be redeemable in the future.
Teleports only between regions on the Second Life grid.	Teleports between any hypergrid-enabled grids by using a hypergate, link region, or simply entering the hypergrid address in the Map dialog’s search field. Currently, hypergrid teleports are limited — it doesn’t work between grids running versions of OpenSim that are too far apart, or between regions located too far apart on the map, and does not support friends or instant messages. These limitations are expected to be addressed in future releases.