Serious Gaming Technologies Support Human Factors Investigations of Advanced Interfaces for Semi-Autonomous Vehicles

Robert Stone, Robert Guest, Eugene Ch'ng, Christopher McCririe,
Christopher Collis[1], Rama Mannur and Imran Rehmi
Human Interface Technologies Team
University of Birmingham
Electronic, Electrical & Computer Engineering Department
Edgbaston
Birmingham, B15 2TT
UK

r.j.stone@bham.ac.uk

ABSTRACT

Since the advent of highly capable uninhabited vehicles, notably in the application domains of offshore oil/gas exploration and defence, attention has increasingly focused on the development of technologies necessary to endow remote systems with complete autonomy. However, this approach has not met with widespread success. Operational experiences frequently point to the fact that the human operator still has a significant role to play in the future of uninhabited vehicles, as part of a control continuum that ranges from direct teleoperation during critical mission phases and recovery modes of control to the high-level supervision of single or multiple platforms. However, few (if any) usable guidelines and/or affordable experimental test beds exist to help ensure that human factors issues are adopted early in the design lifecycle of uninhabited systems. To help redress this situation, research under way within the University of Birmingham and the UK's Human Factors Integration Defence Technology Centre has resulted in the development of an experimental Synthetic Environments technology demonstrator test bed, codenamed Alchemy. The test bed is designed to support the generation of new human factors knowledge relating (initially) to operator display and control requirements for uninhabited vehicles, such as iSTAR UAVs (Intelligence, Surveillance, Target Acquisition & Reconnaissance Uninhabited Air Vehicles), deployed in support of homeland security operations or urban combat. The test bed has evolved from an early PC demonstrator, exploiting the Microsoft DirectX Application Programming Interface and .NET framework, to one that now exploits the power, quality and support of software tools emerging from the serious gaming community. This evolution is also helping to support the exploitation of serious games technologies in other defence applications, from close-range weapons training to military surgery.

INTRODUCTION

One of the most significant historical drivers behind technological developments in robotics and (semi-) autonomous systems (e.g. Sheridan, 1992) has been the desire to remove humans from direct contact with hazardous or safety critical environments, such as subsea, in space, within nuclear processing facilities or on

---

[1] At the time of writing, a Graduate Engineer with EADS Defence and Security Systems UK.

Stone, R.; Guest, R.; Ch'ng, E.; McCririe, C.; Collis, C.; Mannur, R.; Rehmi, I. (2006) Serious Gaming Technologies Support Human Factors Investigations of Advanced Interfaces for Semi-Autonomous Vehicles. In *Virtual Media for Military Applications* (pp. 8-1 – 8-20). Meeting Proceedings RTO-MP-HFM-136, Paper 8. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int/abstracts.asp.

the battlefield. The evolution of teleoperated systems – robot-like devices controlled primarily by a human operator – spans a good 4-5 decades. During that time, there have been all manner of attempts to improve human performance whilst working remotely, from stereoscopic television to *telepresence* systems utilising head-mounted displays and instrumented body gear, and from advanced forms of force and tactile feedback (including exoskeletal controllers) to sophisticated interfaces based on dynamic computer-generated or *virtual* imagery (e.g. Stone, 1992). In parallel, developments with autonomous vehicle and manipulator technologies have continued at a pace, although it is recognised internationally that, for the foreseeable future, semi-autonomous systems are best served by carefully blending human skill with reliable automatic subsystem support in order to minimise workload and maximise safe operation. Although technological developments in autonomous, semi-autonomous and teleoperated robot systems have accelerated over the past three decades, the majority of field-based systems – those performing *actual* remote operations in hazardous environments – still rely on the human operator to carry out the “fundamentals”, such as remote imagery interpretation, decision making, task/route planning, navigation, motion control and manipulation.

With the exception of a small number of institutions, such as the Oak Ridge National Laboratories (e.g. Draper, 1985; 1994), NASA’s Jet Propulsion Laboratory (e.g. Brooks & Bejczy, 1985) and domain-specific guidelines such as those by the first author and others (Stone *et al*., 1984; Stone, 2002; Jacobus *et al*., 1992; Graves, 1998), many of the academic and military laboratory-based R&D programmes have consistently failed to generate realistic guidelines to help develop interfaces to support the human operator in controlling remotely operated systems safely and efficiently. This comment applies as much to standard interface design issues – console sizes and layout, hand controllers, other manual input devices – as it does to more recent multimedia screen interfaces utilising alphanumeric, 2D/3D graphical, video or multimedia-based content. Furthermore, the international standards community also admits that, as far as advanced computer-based human interfaces are concerned, it is further behind delivering underpinning standards for technological developments than it would like to be.

In short, the role and needs of the human operator has been, but must no longer be ignored. This lesson was learned during the closing stages of the UK’s civilian and military advanced robotics programmes in the 1980s, not to mention those instigated by DARPA in the 1980s and 1990s (Figure 1). Specifically in the case of autonomous systems, there is a widespread view that the role of the human should neither be reduced to zero activity, nor to a level that simply instigates a series of vehicle launch/recovery procedures. Instead, the human-system interface for future autonomous and semi-autonomous systems must be capable of supporting the operators in high-level mission planning (on- and off-line; simulated or real), mission deviation planning, and (perhaps most importantly in the case of sophisticated, high-value assets) “intelligent” reversion to levels of manual control. Manual control is likely to be required in cases where the vehicle(s) is (are) unable to perform autonomously, due to faulty or damaged subsystems, incomplete or highly uncertain environmental sensor data or other “degraded autonomy” situations. In these cases, the effectiveness with which the human can take over control and retain a high level of situational awareness will depend on the display and control qualities of the user interface.

Figure 1: Historical US and UK Defence Advanced Robotics Initiatives Focusing on Autonomy: the DARPA Autonomous Land Vehicle Project (Left) and one of the UK Ministry of Defence MARDI Programme Vehicles (Mobile Advanced Robotics Defence Initiative; Right).

Despite the seemingly obvious importance of the human operator's role in uninhabited vehicle systems, there have not, to date, been any *integrated* human factors or human-centred programmes of research and development leading to the production of sensible, reasoned and meaningful advanced operator interface guidelines or standards. Where attempts have been made to develop focused standards and guidelines, the lack of relevant uninhabited platform systems human factors data has forced some authors to take short cuts. One notable example of this is Appendix B3 (Human Computer Interface) of STANAG 4586, *Standard Interface of the Unmanned Control System (UCS) for NATO UAV Interoperability*. Here, in the absence of any appropriately *packaged* knowledge relevant to UAV human interface design (and relevant knowledge certainly exists in the UAV community), the authors have used slightly modified extracts from the civilian standard, ISO 9241, *Ergonomic Requirements for Office Work With Visual Display Terminals*.

Furthermore, in cases where the relevant human factors knowledge does not exist, affordable and accessible experimental testbeds are elusive. In order to address this issue, work undertaken as part of the research programme mapped out for the UK's Human Factors Integration Defence Technology Centre (www.hfidtc.com) has been actively reviewing the Synthetic/Virtual Environment (SE/VE) arena, with the aim of recommending methodologies and processes for exploiting simulation in support of a range of human factors activities, from training to human-centred prototyping and from real-time visualisation for C4I to applications in defence medicine.

THE HUMAN FACTORS INTEGRATION DEFENCE TECHNOLOGY CENTRE

The UK Defence Technology Centre (DTC) concept was announced by the British Government in 2002, with the aim of establishing world-class centres of excellence, each conducting research into innovative science and technology with the aim of contributing to an enhanced UK defence capability, together with additional valuable "spin-out" of results into civilian sectors. One of the first DTCs to be launched by the UK's Ministry of Defence (MoD) in 2003 focuses on Human Factors Integration (HFI – Human Systems Integration, or HSI, in the US) over a broad range of military domains. The HFI DTC consists of a number of commercial and academic organisations, each bringing a set of skills and military applications experience to the table that, together, contribute to a unique consortium in the global defence community (Stone & Lane, 2004).

The scope of first phase of work of the DTC (2003 – 2006) brought together a number of research themes and generic or "enabling" technologies (Stone, 2003) into four broad Work Packages: (a) HFI and Network-Enabled Capability (NEC), (b) education and training, (c) updating MoD HFI processes and (d) HFI awareness/exploitation. The work described herein, whilst originating outside of the original scope of the HFI DTC's research plan, was, in 2004, incorporated within the broader SE overview project contained within (c), above. The DTC's Phase 2 (2006 – 2009) research programme, which commenced in April 2006, incorporates a number of work packages addressing many of the issues described in the present paper relating to affordable/accessible SEs for military applications.

Although the early work focused on the use of Uninhabited Air Vehicles (UAVs) for urban operations support, it is now addressing more general Human Factors issues of exploiting low-cost experimental VE or SE test beds to generate new processes, knowledge and guidelines (e.g. relating to perceptual-motor skills, safety issues, operator training needs and situational awareness) to assist in the future development of "high-tech" defence systems demanding significant human operator involvement.

The reasons underlying this approach are clear. Legacy processes in HFI and whole-life cycle procurement simply do not currently adapt well to the introduction of new technologies into the defence community, particularly in the pre-concept phases of procurement cycles (such as *CADMID*[2] in the UK) and during the concept, assessment and demonstration stages themselves. Similar comments can be directed at the early stages of the Synthetic Environment Development & Exploitation Process (*SEDEP* – Euclid RTP 11.13), one major aim of which is "to overcome the obstacles that prevent SEs being exploited in Europe by developing a process and an integrated set of prototype tools, which will reduce the cost and timescale of specifying, creating and utilising SEs for collective training, mission rehearsal and simulation-based acquisition".

PROJECTS *ALCHEMY* AND *TOMSAV*

Project *Alchemy* (the generic title given to the projects described herein) was originally conceived to demonstrate how low-cost SE test beds could be developed quickly and cheaply, thereby supporting the rapid development of technology demonstrations (military systems, scenarios, etc.) and, ultimately, investigations into the use of different forms of interactive display and control devices. *Alchemy* builds upon a real-time, interactive 3D (*i3D*) demonstrator originally developed by one of the authors (Collis) for a Birmingham University Undergraduate Final Year project called *TOMSAV* (Teleoperation Of Multiple Semi-Autonomous Vehicles). *TOMSAV* was an experimental VE demonstrator to investigate human operator situational awareness display requirements for the control of a fleet of semi-autonomous submersibles deployed to carry out surveillance activities around a disabled nuclear submarine (DISSUB). *TOMSAV* was developed specifically to address information display requirements during the transition from the supervisory control of up to six Underwater Uninhabited Vehicles (UUVs) to the tracking or direct manual control of individual vehicles. Once an individual vehicle had been selected for direct teleoperation, the remaining platforms could be programmed to exhibit a range of behaviours, including swarming to a remote, "stand-off" location and "flocking" – following the human-controlled lead submersible in formation. Individual UUVs were programmed to exhibit unanticipated behaviours, such as "straying" from the swarm. The system was designed with a reconfigurable interface, enabling a range of system control, status and navigation parameters to be displayed on-screen, including vehicle "health", thrust, bearing, pitch and overview map (Figure 2).

---

[2] *CADMID*: Concept – Assessment – Demonstration – Manufacture – In-Service – Disposal

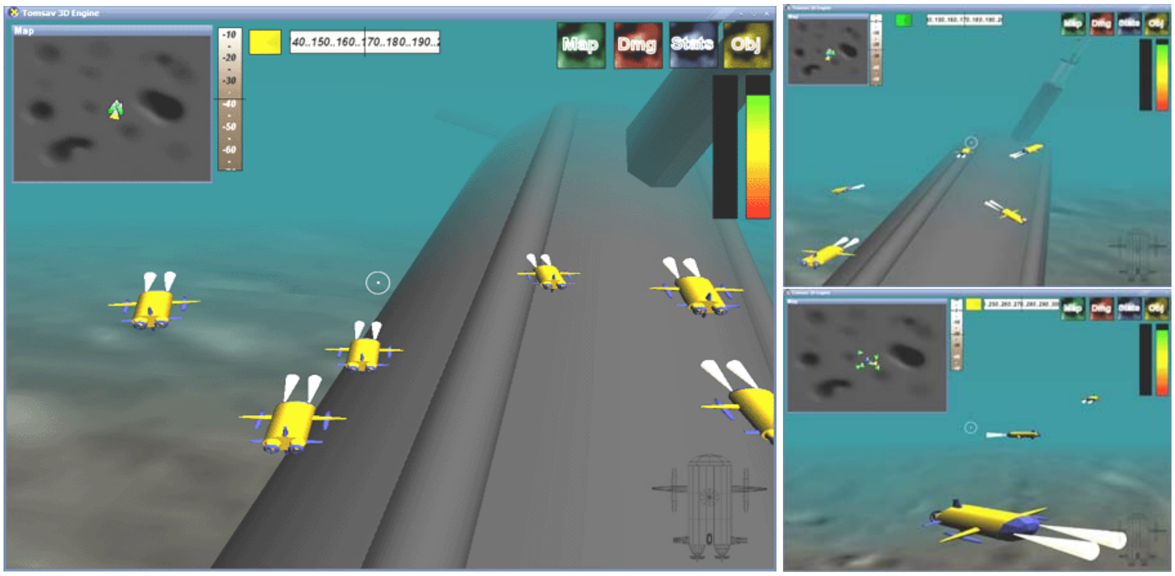


Figure 2: Screen Shots of the *TOMSAV* Experimental Demonstrator Showing Various User Interface Elements and UUV Behaviours.

The *TOMSAV* project was designed to demonstrate the feasibility of using tools made freely available with and for PC platforms to provide low-cost experimental test beds to support human factors studies of remote systems. A key aim of the project, therefore (and one which was been preserved throughout the development of *Alchemy*) was to deliver affordable and accessible (and, if possible Open Source) i3D software engine for research purposes, using desktop or laptop PC hardware. *Alchemy 1* exploited Microsoft's DirectX 9.0 API in its early developmental stages, by virtue of the fact that it enabled software developers to access specialised hardware features (e.g. graphics accelerators, joysticks and other interactive devices) without having to write hardware-specific code. "Managed" DirectX also enabled researchers to exploit the then new .NET languages (such as C#) for developing high-performance multimedia and games applications.

THE *ALCHEMY* UAV AND CONTEXT

Having successfully demonstrated the exploitation of DirectX and .NET during the *TOMSAV* Project, attention turned to extending the relevance of the research to that under way within the HFI DTC. Project *Alchemy*'s virtual UAV was modelled on one of the real-life ducted turbofan iSTAR UAVs, developed by Allied Aerospace (US). This UAV is one of a number of "Organic" Air Vehicles (OAVs) being considered in the US for "sentinel" duties – deployed by battlefield personnel to provide risk-free reconnaissance and surveillance support within urban settings (especially in close proximity to architectural structures). For the purposes of Project *Alchemy*, the actual design of the vehicle was unimportant. The choice of a semi-autonomous hovering platform was made purely to investigate the role of the human operator in deploying, controlling and retrieving a small, hovering UAV in an urban setting. An additional uninhabited ground ("marsupial") vehicle for deployment/ launch support has also been modelled (Figure 3). This system is based on the marsupial telerobot under development by the Space and Naval Warfare (SPAWAR) Systems Center (San Diego) as part of the Mobile Detection, Assessment and Response System - External (MDARS-E) Programme (Figure 4).

The iSTAR class of UAV/OAV is destined for deployment in urban settings, such as in the vicinity of buildings and confined spaces. To this end, a three-dimensional model of the University of Birmingham's

Electronic, Electrical and Computer Engineering (EECE) building was constructed (also shown in Figure 3), together with the adjacent annexes, car parks and open ground (parts of which also feature in experimental C4I/mobile computing work associated with the HFI DTC's work programme (Stanton, 2006)). The EECE building has many ideal features for conducting virtual urban iSTAR UAV operations, including prominent ledges, recesses and set-back windows, constrained passages and a roof-mounted radar dish installation. These features provided excellent targets and challenging landing sites within which the virtual vehicle could "perch" for simulated surveillance activities.

The virtual building also reproduced a covered walkway, through which it is possible to fly the UAV and then lose direct line-of-sight control for the purposes of flying to the rear of the site. This feature provided opportunities to investigate the handing-over of control to a second human operator or reverting from exocentric (third-person) to egocentric (first-person) control. In addition to the building and its immediate environs, two offices were modelled in greater detail and were "occupied" by armed virtual insurgents (also shown in Figure 5). These offices provided researchers with ample detail with which to interrogate the virtual UAV user during experiments (addressing recognition, recall, etc.) in which the vehicle is being flown in a reconnaissance mode of operation.

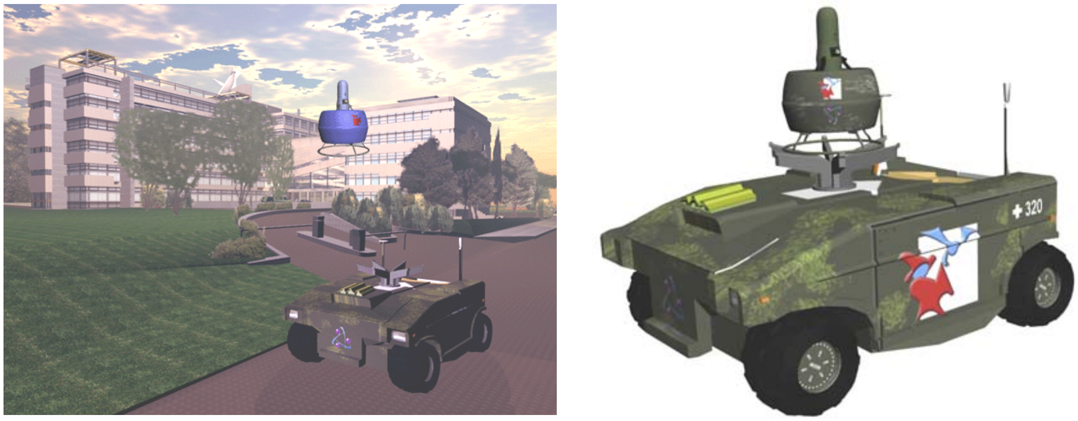


Figure 3: The *Alchemy* Virtual Marsupial Vehicle and iSTAR UAV.

Figure 4: The SPAWAR MDARS-E Marsupial Vehicle Launching an Allied Aerospace iSTAR UAV (source: www.nosc.mil/robots/resources/marsupial/marsupial.html).

Figure 5: One of the Virtual "Insurgent-Occupied" Offices (3D insurgents and AK-47 weapons sourced from www.turbosquid.com).

The 3D models of the vehicles, payload and "urban" settings were constructed using a variety of resources, including 3ds max (and the freely available version, *gmax*), converted to .X or BSP format for integration within the DirectX and .NET run-time modules/framework. Gmax is one of a number of increasingly popular "light" 3D modelling packages. The software is available for download from the Internet free of charge (www.turbosquid.com/gmax) and is supported by a large global user community, as can be seen by some of the highly detailed Microsoft *Flight Simulator* aircraft models available on the Web (e.g. at www.simviation.com). Models of the virtual insurgents, automobiles and weapons were purchased at minimal cost (again in 3ds/gmax format) from the Turbosquid site. Gmax models can be converted to other popular formats (e.g. for current-generation game engines and editors or other i3D delivery and browsing packages) using a combination of 3ds max, gmax *Tempest* (also available at the Turbosquid site) or the *MilkShape 3D* Modeller (www.swissquake.ch/chumbalum-soft/). Using these and similar Web-accessible resources, it was possible to introduce additional 3D models into the *Alchemy* demonstrator, if required.

INTERFACE OPTIONS

As emphasised earlier, the early *Alchemy* test bed was been designed to show how quickly and cost effectively an experimental HFI tool based on SEs could be produced and used to investigate key HFI and HFE issues – avoiding the need to gain direct access to scarce prototype or operational hardware, or having to conduct experiments in potentially unsafe urban settings. To this end, the *Alchemy* test bed supports investigations of:

- vehicle control (basic and advanced),
- real-time information display,
- the development and support of human skills in supervisory control and teleoperation,
- the role of wearable/head-tracked devices (Figure 6), novel interface devices and so on.

The first generation of the test bed supported a range of configurations suitable for experimentation, including "overview" (exocentric) *vs.* teleoperation (egocentric) control modes (Figure 6); UAV sensor type (e.g. conventional camera *vs.* thermal imager *vs.* night vision; camera field of view (narrow, wide, panoramic); keyboard + mouse *vs.* joystick control *vs.* tracked, head-mounted display plus joystick control; transmission time delays from/to the operator control station.

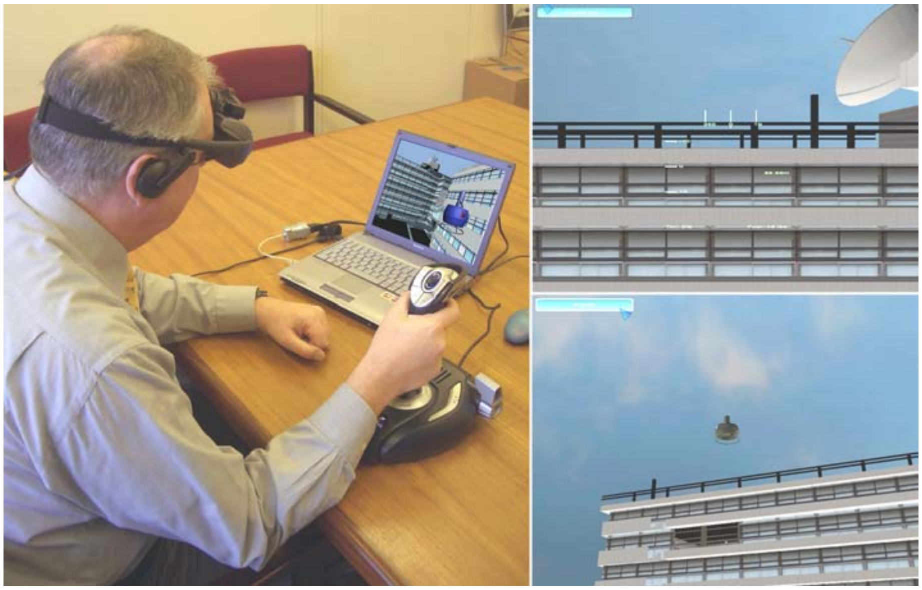


Figure 6: One Interface Variation for *Alchemy*: Head-Mounted Display, Head Tracker and Force Feedback Joystick. The Images on the Right of the Figure Show Typical Screen and HMD Views – Exocentric, or Third-Person/Overview (Lower) and Egocentric, or First Person/Teleoperation (Upper, with Simple Head-Up Flight Control Symbology).

In addition to the independent "interface technology" variables listed above, the *Alchemy* system supported a number of dependent variables, including objective recording of operator performance whilst flying the UAV, from task times to collision events. The system also allowed simple secondary task features to be programmed into a flight scenario prior to launch. These varied from the appearance of simple visual stimuli at different screen positions, to the presentation of recognition and cognitive tasks. As well as visual stimuli, the *Alchemy* programme generated simple auditory and haptic feedback cues, should the experimenter wish to use these in a given scenario (e.g. to simulate engine problems, turbulence, obstacle proximity, etc.).

Serious Gaming

At about the time the first variant of the *Alchemy* testbed was being demonstrated, it was becoming obvious that a new genre of interactive 3D content development and rendering tools was becoming available, many at very low cost or even free of charge. This development is significant in that it promises to "open up" simulation to previously non-specialist i3D user communities (including Human Factors specialists) to a much greater extent than had been the case with developments in, for example, Virtual Reality during the closing decade of the last Century. This "new" field of endeavour, referred to as *serious gaming*, focuses on the exploitation of computer games software tools such as those underpinning the "first person shooter" (FPS) or "role-playing" games currently being enjoyed by youngsters and adults alike, all around the world. These tools take the form of software development kits (SDKs), released by leading games developers shortly after the publication of a new product, such as *FarCry* or *Half-Life 2*, together with a growing number of supporting content generation packages. The tools enable games players to develop their own virtual humans ("avatars") or computer-generated forces, environments, weapons and adversaries, thereby prolonging the longevity of the game they have purchased. Over the past 12-18 months, the availability and affordability of these tools have also very rapidly generated interest from the serious applications community, including those responsible for designing training and real-time visualisation systems for defence, medicine and education, to mention but three examples.

Long before today's home gaming revolution, and during the late 1980s, when Virtual Reality was just about to break out of its NASA and Department of Defense "homes" (Stone, 1992, 2005a;b), the future potential of computer games to solve the accessibility and affordability problems of modelling and rendering tools for "serious" interactive 3D applications had already been recognised. In the 1980s, for example, basic developments in *modifiable* 3D games technologies for exploitation by communities other than those supporting home entertainment were well under way. For example, *Battlezone* – a successful 3D wire frame tank game published in 1980 for the Atari – was developed a year later into a serious game for the US Army to support training for the *Bradley* military vehicle (*The Bradley Trainer*). A major step forward in the history of interactive 3D was provided by *The Colony* – a first-person science fiction game created in 1988 by David Smith (who was also accredited with developing the first VRML Internet toolkit in 1995). *The Colony*, a combat and logical reasoning game, featured a crashed spaceship and a multi-level underground colony infested with aliens. The significance of Smith's game did not just centre on its revolutionary (for 1988) true 3D and "First-Person Shooter" (FPS) qualities. The underlying development software for *The Colony* was eventually commercialised as a 3D toolkit (*Virtus WalkThrough*) and was subsequently modified for use as a virtual scene planning tool for the 1989 20th Century Fox underwater science fiction film *The Abyss*.

However, despite these early activities, it was not until the 1990s that titles started to emerge that were to set the scene for a decade of games engine development. Such revolutionary titles as *Wolfenstein*, *Doom* (the first game to support a user editing function), *Quake*, *Heretic*, *Hexen*, *Unreal* (with its well-exploited "unrealed.exe" editor addition) and *Half Life* provided over 6 years of first-person action. The graphics of some of the early versions of these games may appear crude and simple today. But all of these games had one thing in common – an essential ingredient in their potential exploitation for serious applications. As long as the user's attention is captured and he or she is required to maintain a spatial and temporal awareness of the 3D situation in order to survive within the scenario, and as long as the simulation responds meaningfully in real time, the underlying engine can be used to develop a training simulator capable of delivering valid, reliable and believable content to highly motivated students of all ages and skills. Indeed a version of *Doom II* was used to train US Marines at the Marine Corps Modeling and Simulation Management Office ("McMismo"), Quantico Base, Virginia. Tom Clancy's *Rainbow Six*, mentioned earlier, despite being put on temporary hold following the events of September 11th, 2001, was actually modified using maps and scenarios requested by the US Army to train troops to fight terrorists in urban terrain.

Based on Epic's *Unreal* engine, *America's Army* was originally produced to allow young Americans to investigate military career opportunities (whilst reducing the cost of preparing and distributing printed information). *America's Army*, a distributable 3D game developed by the Moves Institute specifically for the US Army, had, by early 2005, developed into one of the most successful online games ever. Directed and managed by the U.S. Military Academy's Office of Economic & Manpower Analysis at West Point, *America's Army* was, in December 2003, the focus of intense cross-discipline interest at a "Serious Games Day", held at the Wilson International Center in Washington DC, the precursor event to a Serious Games Summit now staged annually.

The UK's effort in the adaptation of games engines for evaluating collective team working, procedures and communications includes *DIVE* (Dismounted Infantry Virtual Environment), developed by QinetiQ and Maverick Developments and based on the original *Half-Life* engine, recently "upgraded" to take account of latest developments offered by the *Half-Life 2* software development kit. More recently, projects sponsored in part by the HFI DTC have developed a range of serious games demonstrators, ranging from the *Alchemy* experimental testbed (described further below), to proof-of-concept part-task training simulators focusing on close-range weapons (Figure 7), ship navigation competencies, defence mental health studies and naval vessel disposal. Together with subject matter expert support from the Royal Centre for Defence Medicine (RCDM) and significant development effort by TruSim (a division of Blitz Games – a leading independent UK games developer), an *Interactive Trauma Trainer* has also been demonstrated, the ultimate aim of which is to support the acquisition of life-saving decision skills for treating battlefield casualties (Stone & Barker, 2006; Figure 7).

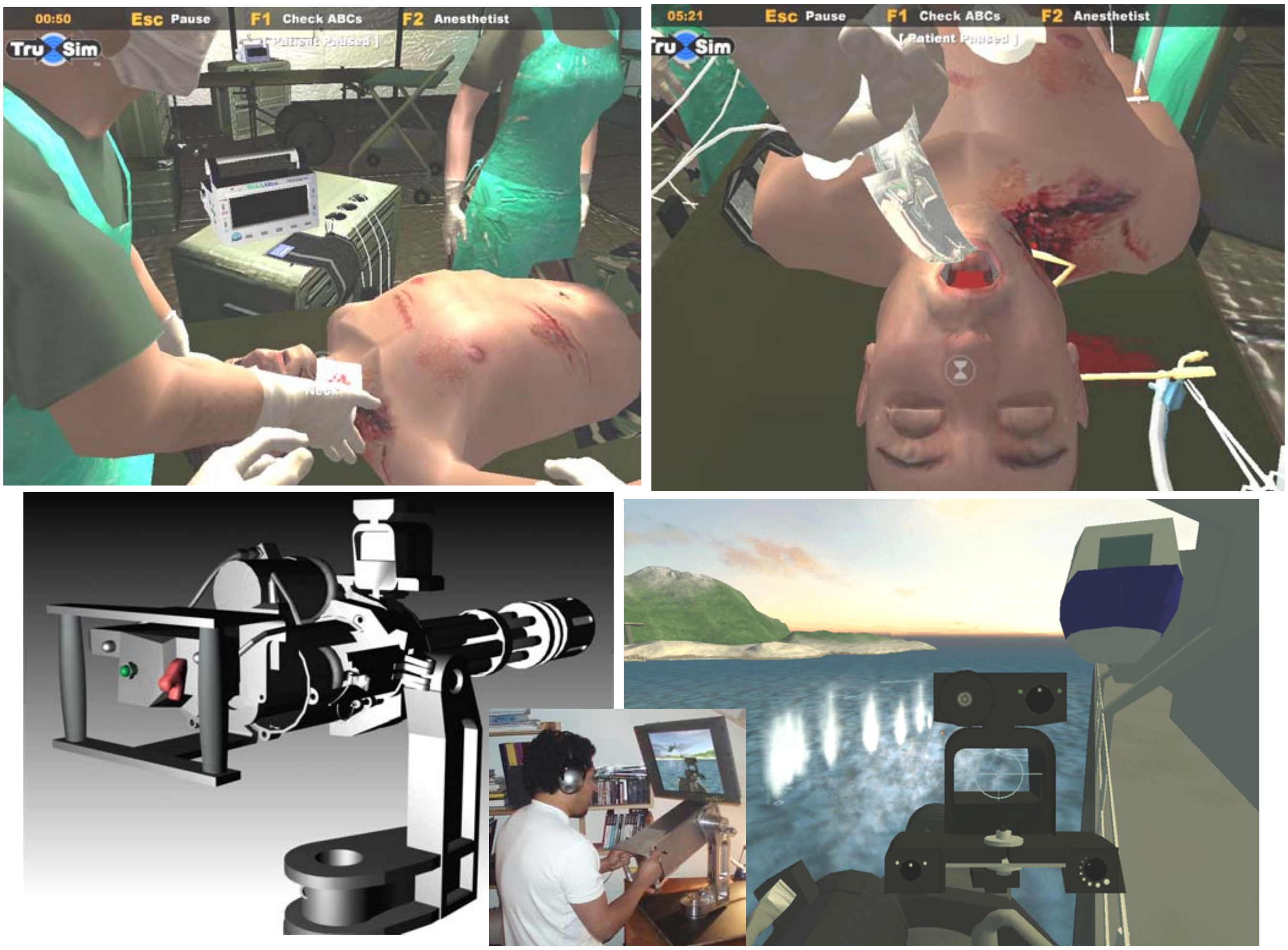
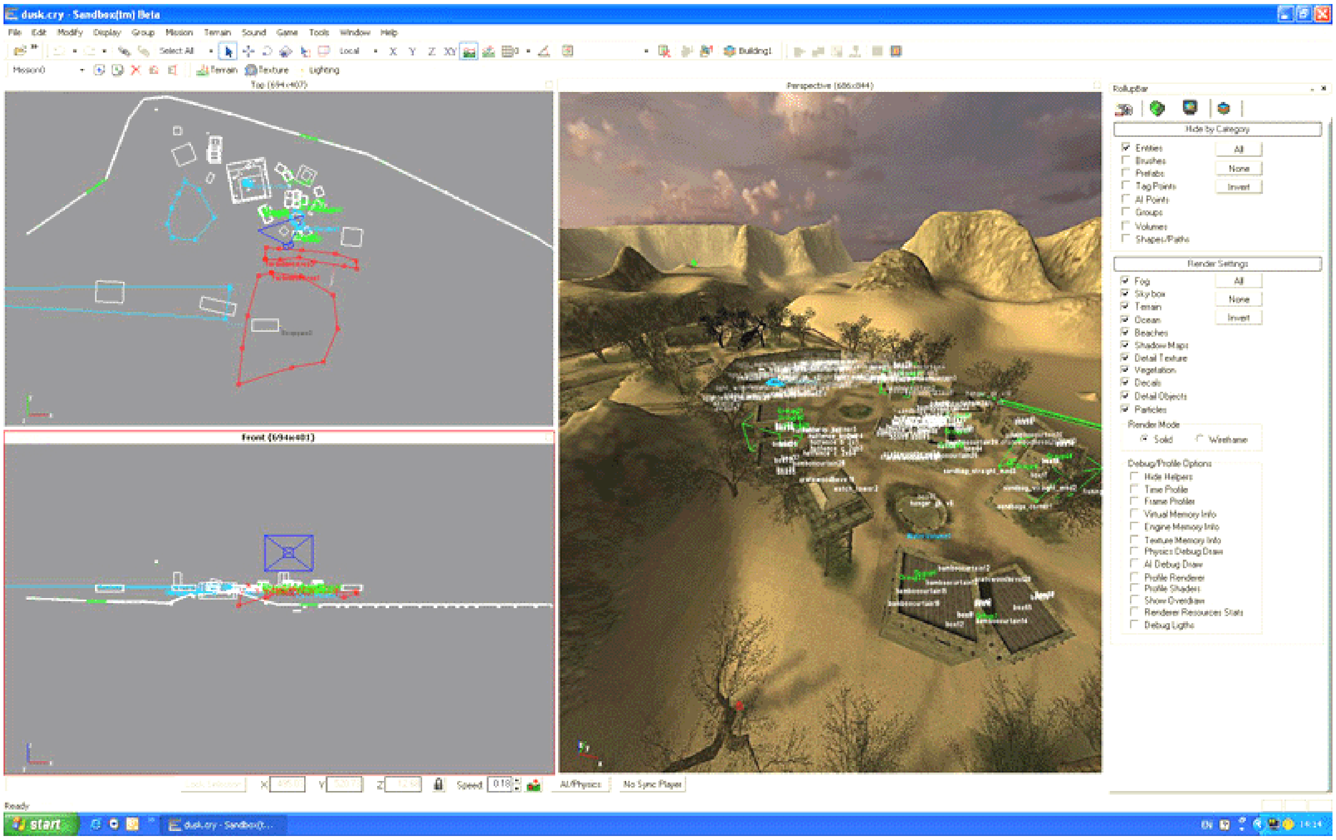


Figure 7: Screen Images of the HFI DTC's Interactive Trauma Trainer (Upper Left and Right) and the Royal Navy Minigun Part-Task Training Demonstrator (Lower Left, Middle and Right).

Alchemy 2

With the emergence of serious games "products", together with their free-of-charge (for research) support tools, it was decided to explore the possibility of exploiting one or more games engines in support of investigations being conducted both within the University of Birmingham and the HFI DTC. At the time of writing, a comprehensive review of serious gaming technologies relevant to military Human Factors experimentation and part-task training is being concluded. However, in order to assess existing engine capabilities at the time when the initial, informal investigations using the first *Alchemy* testbed were coming to a close, it was decided to concentrate on one proprietary engine in order to evaluate the pros and cons of developing rapid scenarios around the existing iSTAR UAV-marsupial vehicle concept.

The game and associated engine chosen for this purpose was *FarCry* – a popular first-person action game based on a real-time engine developed by the German company Crytek (the *CryENGINE* – now owned by the games company Ubisoft). The *CryENGINE* is one of the gaming communities' leading engines (with particular strengths in the real-time rendering of expansive external environments out to 2km) and supports most of the currently available video and computer hardware types. As well as supporting effects such as bump mapping, static lights, networking, integrated physics, Artificial Intelligence, shaders, shadowing and so on, the *CryENGINE* features its own real-time game editor called *Sandbox* (Figure 8). The *Sandbox* is one of the more intuitive editing packages available on the market and can also be downloaded free of charge from the Web. At the time of writing *Crysis*, Crytek's latest action game, is under development, with early demonstrations being shown at major gaming conferences worldwide. The quality and functionality of the software underpinning *Crysis* – the *CryENGINE 2* (effectively a new engine) – promises new collision, particle and lighting physics, highly realistic external environments and a new generation of "believable" virtual humans ("avatars"), thereby offering serious games researchers even greater capabilities than its predecessor.



RollupBar

Hide by Category
- ☑ Entities
- ☐ Brushes
- ☐ Prefabs
- ☐ Tag Points
- ☐ AI Points
- ☐ Groups
- ☐ Volumes
- ☐ Shapes/Paths

All None Invert

Render Settings
- ☑ Fog
- ☑ Sky box
- ☑ Terrain
- ☑ Ocean
- ☑ Beaches
- ☑ Shadow Maps
- ☑ Detail Texture
- ☑ Vegetation
- ☑ Decals
- ☑ Detail Objects
- ☑ Particles

All None Invert

Render Mode
● Solid ○ Wireframe

Debug/Profile Options
- ☐ Hide Helpers
- ☐ Time Profile
- ☐ Frame Profiler
- ☐ Virtual Memory Info
- ☐ Engine Memory Info
- ☐ Texture Memory Info
- ☐ Physics Debug Draw
- ☐ AI Debug Draw
- ☐ Profile Renderer
- ☐ Profile Shaders
- ☐ Show Overdraw
- ☐ Renderer Resources Stats
- ☐ Debug Ligths

Figure 8: The User Interface for the *CryEngine Sandbox* Editor.

Using some of the original 3D model assets developed in support of the first *Alchemy* iSTAR UAV, the *Alchemy 2* demonstration was actually constructed by one of the authors (Guest) in just 2 weeks, exploiting the geometric import and effects library inherent with the *Sandbox* editor. The scenario was based around a Middle-Eastern insurgent training camp scenario (built from scratch using 3ds max and images from *Google* and other Web resources), with a small collection of buildings surrounded by rough terrain and undulating sand dunes. In parts of the scenario, failed armed forces missions were indicated by burning vehicles and a downed helicopter. Having been "delivered" by the teleoperated marsupial vehicle to a location close to the training camp, the task of the virtual UAV operator was to fly a low-altitude mission around the camp, approaching from different directions and noting key features of the camp, including the presence of insurgents and military assets (Figure 9). As with the *Alchemy 1* testbed, the VE was designed to support different forms of input and display device, together with normal camera and night vision effects. In addition to those investigated before, the opportunity was also taken to investigate the pros and cons of flying a virtual UAV using a typical games console input device (the Microsoft *Xbox S* controller, shown in the central insert in Figure 9). These investigations continue at the time of writing.

Figure 9: Various Views of the *Alchemy 2* iSTAR UAV Demonstrator, Including Launch, Flight Over Insurgent Camp and Night Vision Capability (middle insert shows the Microsoft *Xbox S* controller used as one of the experimental UAV flight control options).

Alchemy 2 Extension: Remote Scenario Surveillance Using UAV-Jettisoned Sensors

One interesting scenario, discussed after the early *CryENGINE* implementation of the iSTAR UAV, was how additional technology could be provided to maximise tactical data returns from the zone of operation. It was felt that these small UAVs (which are somewhat "unstealthy", due to engine noise) could quite easily become the target for small arms fire and, if damaged, could crash without sending sufficient amounts of useful data back to the command post. In addition, certain detailed objectives might not be visible from the UAVs, due to restricted camera views, even at relatively low altitudes. One concept solution worthy of study related to the idea of jettisoning small sensor modules from the UAV, dispersing them around the operational area before retreating to a safe location or, in the case of heavy small arms damage, before the vehicle impacted with the ground. These modules would possess a self-righting capability and would deploy a miniature camera mounted on a motorised platform which could scan the immediate area, producing a 360^o panorama of "stitched-together" digital pictures. The picture file, together with GPS and/or digital compass data could be relayed back to the command post (via the UAV if still intact, or via other appropriate uplinks) and could be displayed in conjunction with a digital map to enhance the situational awareness of the operational zone before deploying armed personnel.

To investigate this, a series of undergraduate degree student projects were conducted at Birmingham University to develop proof-of-concept hardware and software. Two of these are summarised here. The first project (by one of the authors – Mannur) concentrated on developing a demonstrator "deployable" mini camera module, as might be jettisoned from a low-altitude iSTAR UAV. The prototype system consists of a PC-hosted controller which relays user commands to the camera module via encrypted RF 433 MHz technology. As can be seen in Figure 10, the camera, stepper motor scanning cradle and GPS antenna are exposed once the four "orange peel" housing segments deploy. These segments also serve to provide an upright and stable platform for the camera (and could provide a surface for solar panels). Once the camera inside is exposed, it rotates 360^o, capturing static pictures and/or video of its immediate location as well as noting the compass orientation of the captured images. This information is then relayed back to the user. Once the scan transmission has been completed, the housing segments close and the module settles into a dormant, power-saving state, awaiting further commands from the remote user.

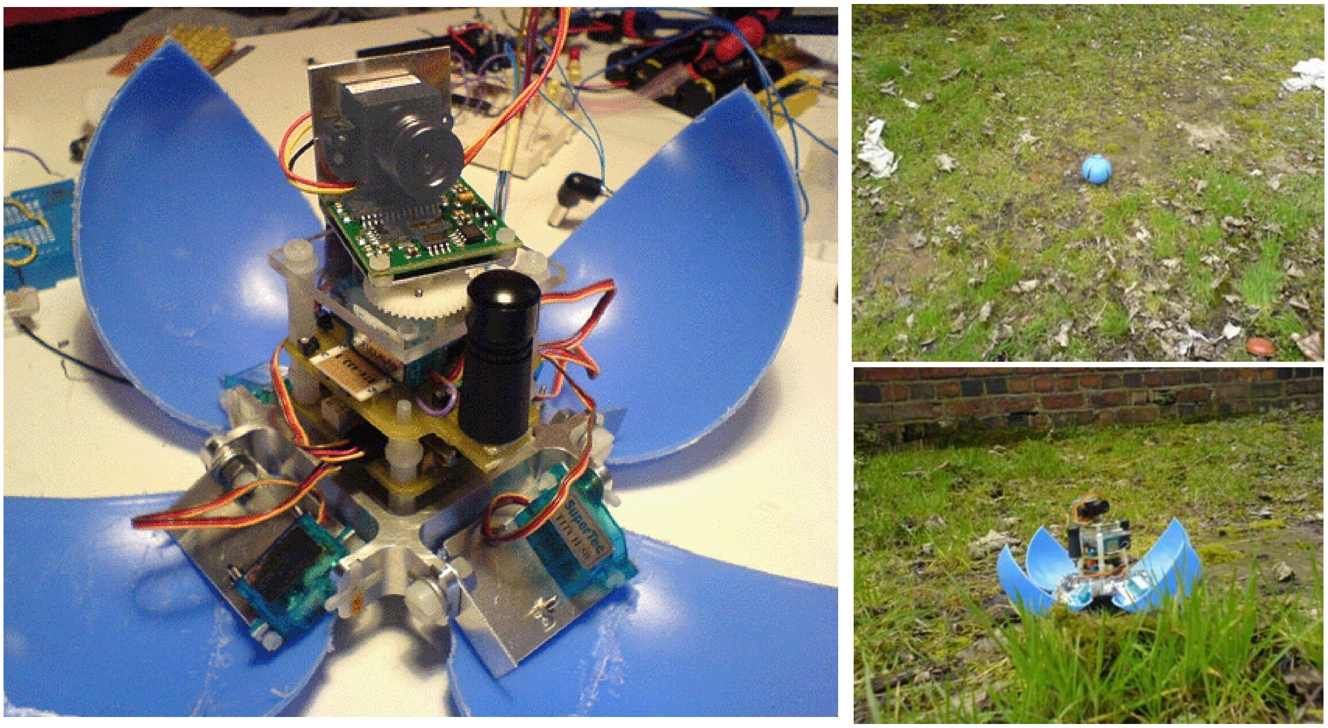
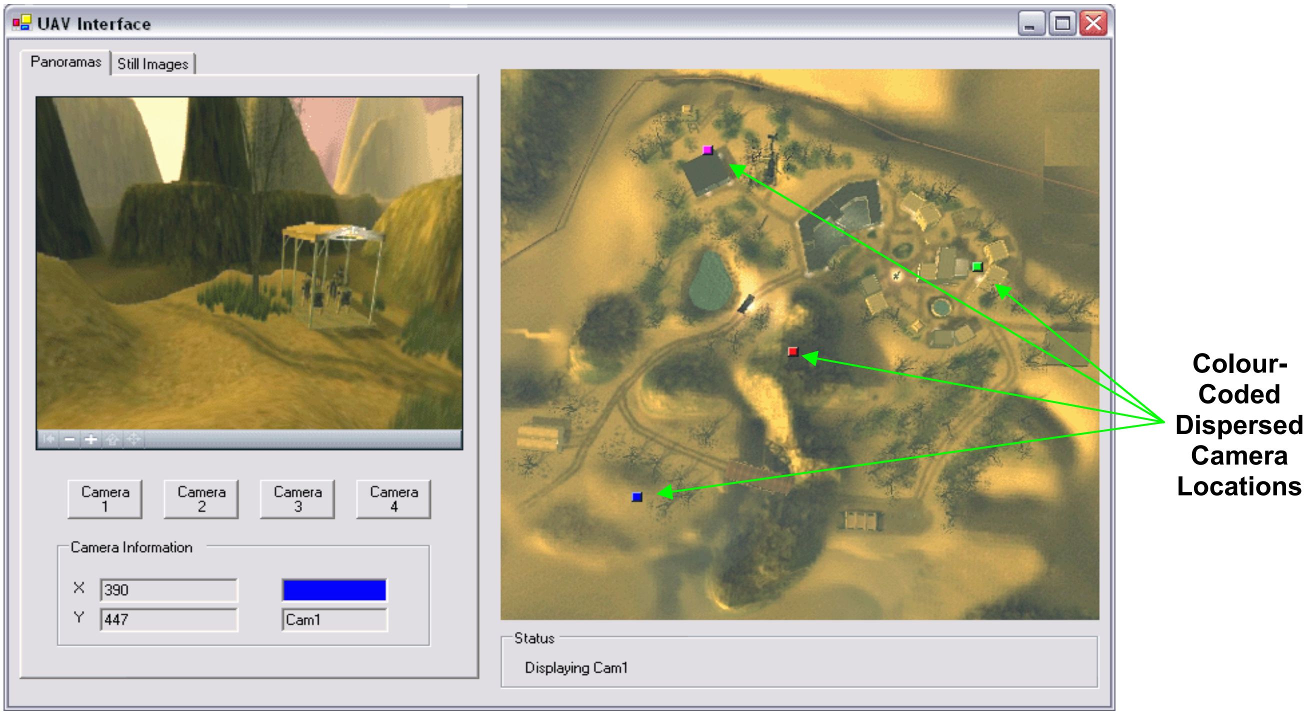


Figure 10: Self-Righting, 360^o Panoramic Camera/GPS Deployable Module Concept ("deployed" state images shown right).

The most promising implementation of the panoramic imaging user interface is shown in Figure 11. This demonstration was developed as part of a second undergraduate project by another of the authors (McCririe) using the original *CryENGINE* Middle-Eastern "training camp" Virtual Environment scenario, described earlier. The user interface was designed in Visual Basic / .NET and features two main functions – the first to trigger the processing of the raw images captured from the VE; the second to display the processed (and partly processed) information in a manageable, effective form.

The processing of the images was achieved using standard off-the-shelf photo stitching software compatible with Visual Basic Scripting Language (VBS). VBS was used to automate the process of creating a *QuickTime* panorama (.mov or .qt), which includes the initial retrieval of the images, the application of stitching and blending algorithms, file compression and export to a given location in memory. Having created the panorama, a colour-coded marker is placed on a two-dimensional map or aerial image of the VE at the location the camera was "deployed" (on the right of Figure 11; this information would be derived from GPS data returns in an operational system). Left-clicking the mouse on this marker displays the relevant panorama in a *QuickTime*-like viewing window (on the left of Figure 11), enabling the user to pan left and right and to zoom in and out, again using the simple mouse interface.

Also in Figure 11, "Camera 1" (located in the bottom left-hand sector of the aerial view) has been selected - it is just possible to discern two vehicles armed with medium-calibre machine guns positioned under a covered structure. These vehicles were not visible from the UAV, even flying at quite a low altitude.



X 390
Y 447
Cam1

Status
Displaying Cam1

Figure 11: User Interface Concept for Dispersed Camera Map and Panorama Display.

During the development of the main interface, some of the limitations of the commercial photo-stitching software became evident. For example, if the images were taken in an area with a high number of complex objects, all in close proximity with the camera, some parts of the panorama appeared slightly disorientated. This led to a second display option in which separate images, taken at 45^o intervals (i.e. north, north-east, east, etc.) were also made available and accessed in a similar way to the panorama.

A final development worthy of note here is another student project (by Rehmi), again based on the *CryENGINE* UAV implementation. However, rather than focusing on a defence application, this project addressed the possible future exploitation of uninhabited vehicle technologies in support of rescue missions following natural disasters, such as earthquakes or tsunamis. In this demonstrator, which was developed with the support of subject matter experts from the Islamic Relief aid agency, the UAV pilot has to fly the vehicle over a devastated island fishing village, noting the location of survivors, hazardous areas, UAV surveillance "perching" sites, access routes and so on, prior to the deployment of rescue services. The village has been cut off from the rescuers' base camp by a landslide which blocks the main estuary access. Figure 12 shows some of the images from this project.

Figure 12: UAV Deployment in Support of Disaster Zone Rescue Missions, Visualised Using the Crytek *CryENGINE*.

Alchemy 2 Extension: Urban Combat UAVs and *Personal Air Vehicles* (PAVs)

One of the HFDI DTC's remits is to investigate channels of exploitation and "spin-out" of knowledge and capabilities into other defence and civilian sectors. An opportunity arose to test this spin-out aspiration when the University of Birmingham was approached by an aerospace company called Kestrel, based in the UK's Midlands region. Kestrel is a small innovative company specialising in the design and development of novel aerospace vehicles, both inhabited and uninhabited. The company's full range of products can be seen at www.kestrelaerospace.com. With increasing interest in their uninhabited and personal air vehicle (PAV) systems being shown by the likes of NASA, the DoD and MoD, Kestrel wanted to be able to visualise some of their designs in a form that would ultimately support the company's marketing, investment, design and, in due course, training programmes. As many of these applications would help to test the strengths of serious gaming technologies as applied to the support of system lifecycles (such as the UK CADMID cycle mentioned earlier), Birmingham researchers acquired basic 3D models of the UAV and PAV concepts and modified their geometries so that they were more suited to visualisation in real-time VE or games engine packages. Once again the *CryENGINE* environment was chosen and, in 2 days and 10 days respectively, virtual flight models of the company's *Seeker* portable (10kg, 1m tall) Urban Combat UAV and *Kestrel* PAV respectively were implemented within the Middle Eastern scenario (Figure 13).

Figure 13: *Seeker* UAV (Left) and *Kestrel* PAV (Right) Design Concepts Visualised Using the Crytek *CryEngine*.

CONCLUSIONS

One of the biggest problems facing those in the human factors and systems engineering communities has been the absence of affordable, accessible tools supporting rapid and timely investigations into new equipment concepts, hypothetical scenarios, user interface designs, ergonomics prototypes, part-task training needs, and so on. Tools that are readily usable by more than just those with strong software or i3D competencies. Tools that do not demand complicated technologies to be instantly exploited by end users in the defence arena. The serious gaming community looks set to provide those tools – not only under "free-for-research" licensing conditions, but, increasingly, as freely distributable, open source engines, many of which are emerging from militarily-supported organisations (e.g. see www.devmaster.net).

The investigations described herein suggest that real-time simulation and serious games technologies look set provide a credible alternative to more conventional early Human Factors investigations and experimental trials (not to mention scenario generation), especially at a time when physical defence resources and personnel are stretched to the limit in support of Western coalition forces in many corners of the world. What is also becoming apparent is that the software tools and hardware resources necessary to produce the content and run-time functionality demanded by high-quality simulated environments are no longer the sole province of specialist companies. The financial shackles imposed by the "supercomputer" and "black-box" vendors of yesteryear – especially the enormous start-up costs and year-on-year maintenance charges – have been broken. As was found in the *Alchemy* Projects described above, the unique 3D models, the games engine functionality and the photorealistic detail cost virtually nothing to produce. Add a reasonably powerful laptop with an appropriate graphics card and a joystick or games console hand controller from a local high-street computer vendor and one has access to portable simulation tools that can easily compete with their defence-procured counterparts, typically costing many hundreds of thousands of dollars. Currently, the worldwide situation for serious gaming is looking very encouraging indeed, not just from a commercial perspective, but for the future of developers of all ages and backgrounds as well. The availability of i3D modelling and rendering tools at very low prices, even free from the Web, is something that those in the Virtual Reality community could only have dreamed of as it struggled to make inroads into the serious applications markets of the 1990s – and, indeed, continues to struggle today.

REFERENCES

Brooks, T.L., and Bejczy, A.K. (1985). Hand Controllers for Teleoperation; a State of the Art Technology Survey and Evaluation. Jet Propulsion Laboratory Publication 85-11. Ellson, D.C.

Draper, J.V. (1985). The Human-Machine Interface in Mobile Teleoperators. *Proceedings of the Department of Energy Workshop on Requirements for Mobile Teleoperators for Radiological Emergency Response and Recovery* (Foltman, A.J., Ed.). Dallas, TX: Argonne National Laboratory.

Draper, J.V. (1994). Teleoperator Hand Controllers: A Contextual Human Factors Assessment. Oak Ridge National Laboratory Publication ORNL/TM-12762.

Graves, A. (1998). User Interface Issues in Teleoperation. Unpublished Research Report. De Montfort University, Leicester, United Kingdom.

Jacobus, H.N., Riggs, A.J., Jacobus, C.J., and Weinstein, Y. (1992). Implementation Issues for Telerobotic Handcontrollers: Human-Robot Ergonomics. In M. Rahimi & W. Karwowski (Eds.), *Human-Robot Interaction*. London: Taylor & Francis.

Sheridan, T.B. (1992). Telerobotics, Automation and Human Supervisory Control. MIT Press.

Stanton, N. (2006). Experimental Studies in a Reconfigurable C4 Test-Bed for Network Enabled Capability. Proceedings of HFM-136 Workshop on Virtual Media for Military Applications. *In Press*.

Stone, R.J. (1992). Virtual Reality and Telepresence. *Robotica*; 10; 461-467.

Stone, R.J. (2002). Human-System Interfaces for EOD Telerobots: Introduction to Human Factors Design Guidelines. Unpublished report prepared under contract to the QinetiQ Centre for Robotics & Machine Vision (Contract No. C4004-20747).

Stone, R.J. (2003). The UK HFI Defence Technology Centre: Background, Research & Collaboration. *Proceedings of I/ITSEC 2003* (Orlando; 1-4 December).

Stone, R.J. (2005a). The Convergence of Computer Games and Virtual Reality: Opportunities for Future Synthetic Environment Training. *Proceedings of HCI 2005*; Las Vegas; 22-27 July, 2005.

Stone, R.J. (2005b). Serious Gaming - Virtual Reality's Saviour? In *Proceedings of VSMM 2005*. Ghent, Belgium, 3-7 October, 2005; pp.773-786.

Stone, R.J., & Barker, P. (2006). *Serious Gaming*: A New Generation of Virtual Simulation Technologies for Defence Medicine & Surgery. *International Review of the Armed Forces Medical Services* (In Press).

Stone, R.J., Day, P.O.., Rogers, H.G., and Kelly, C.J. (1984). ROV Workstation Design Handbook. UK Department of Energy Contract No. E/5B/CON/753/802. Unpublished British Aerospace Document; Ref. JS10014.

Stone, R.J., & Lane, K.P. (2004). The UK Human Factors Integration Defence Technology Centre: developments in command & control, training and synthetic environments. *Proceedings of IMAGE 2004*; Phoenix, Arizona; 12-16 July, 2004.