State of the art in Virtual 3D Worlds (for Training)
Introduction

**Prepared by eTS**

Virtual Reality
- Types of Virtual Reality
  - VR Hardware
  - VR Software
Virtual Worlds
- Concepts and background
- Categories of 3-D virtual worlds
- Educational potential of 3-D virtual worlds
- Working with students/trainees in 3-D environments

References and additional resources

Software for Virtual Worlds

**Prepared by UPAT**

Introduction
Representative Software
Comparison
References and additional resources

OpenSimulator

**Prepared by UPAT**

- Introduction
- Architecture
- Installation
- Regions
- Terrains
- 3D Graphics
- 3D Viewers and User Accounts
- Avatars
- Textures
- Physics Engine
- Communication
- Basic Interactivity
- Scripting (LSL)
- Non Playable Characters
- Special Scripts
- HUDs (Head-Up Displays)
Integration with LMS
References and additional resources
Introduction

This introduction is based on “Virtual Reality / Technologies and applications in education” report developed by Borislav Stoyanov from e-Training Solutions for project AVARES – Enhance attractiveness of Renewable energy training by Virtual Reality. Since VR development had significantly accelerated the author had to update and expand certain sections to keep the information useful and up-to-date as of the time of writing this introduction – March 2016.

Virtual Reality

The virtual reality concept appeared much earlier than the modern computers. In the middle of the twentieth century Morton Heilig suggested the creation of the Sensorama, a theatre experience designed to stimulate the senses of the audience - vision, sound, balance, smell, even touch (via wind) - and so draw them more effectively into the productions1.

The digitally generated world has entered our life more than three decades ago, however many of the ideas used to build it already existed for much longer. They can be found in well-established social technologies, such as books, theater, film, music. The idea of a virtual world where people negotiate space as a psychological apparatus, rather than a physical reality, has been exploited for centuries, and can be traced as back as to the ancient Greek mythology. The concept of virtuality is far wider and encompasses much more than the computer mediated communities. So the digital virtual worlds can be considered as an extension of other, older forms of virtuality.

The virtual reality (VR) simulators developed for military and aerospace purposes are the granddads of the virtual worlds. Virtual reality as a term applies to computer-simulated environments that can simulate physical presence in places in the real and in imaginary worlds. The term virtual reality itself was proposed and popularized in the 1980s by Jaron Lanier, a researcher and engineer who contributed to a number of products to the emerging VR industry.

Today virtual reality is used to describe a wide variety of applications commonly associated with immersive, highly visual, 3-D environments2. Most of the current VR environments are displayed via a computer screen or special stereoscopic displays and rely mainly on visual stimulus, however there are simulations include additional sensory information, such as sound and force feedback (tactile information).

In near future remote communication environments will become available. They will provide users with virtual presence (telepresence and telexistence) and will use either standard input devices such as a keyboard and mouse, or multimodal devices like wired gloves and omnidirectional treadmills. The

---

simulated realities in such environments can be similar to the real world and provide lifelike experience (combat training, pilot simulators), or can differ significantly from reality (VR games).

The origin of virtual worlds on personal computers can be traced back to early games such as Maze War³, which was developed in the early 1970s at NASA. In this game images of eyeballs were used to represent avatars, there were maps showing the levels, and it was played over networks. In 1986, LucasFilm Games developed Habitat⁴, mostly two-dimensional environment that included humanoid avatars, and users could access the game through online service on their computers.

When World Wide Web started to spread in the mid-1990s, many online virtual worlds started to appear and some experts predicted that these 3-D views would become the standard way to browse the web. One of the earliest – the Active Worlds⁵ platform allowed people to join for free or pay a monthly fee for premium features. However, the first virtual worlds didn’t reach the expected level of popularity mainly because the hardware and bandwidth requirements were much higher than the average at the time. The 90ies have witnessed appearance of several VR-focused products – the EyePhone by VPL research, Nintendo’s Virtual Boy, Sega VR. The main reason they failed to reach popularity was the technology limitations in computing power resulted in high costs and lack of realism. However, VR progressed in other directions - car manufacturers started using it to design cars and test user experience; medical scientists used it to study or treat wide range of medical conditions – from physical to physiological. The governments, military, NASA, developed ways to incorporate VR into training.

Fast forward several years: in September 2012 Oculus⁶ closes its Kickstarter campaign after raising 2 million dollars, in March 2014 it’s bought by Facebook. In 2015 VR become affordable for a first time: Google cardboard headsets were presented which converted any smartphone into a head-mounted display. Various magazines delivered ready cardboard headsets to their subscribers. Instructions were published online how to create such headsets at home, DIY style, requiring only a plan, cardboard and a smartphone. The big question is will the trend be sustained in 2016 and beyond. Will after the initial hype the industry manage to provide content to exploit the existing, already powerful and affordable hardware: games, education materials and ultimately movies?⁷ According to a study by Accuray Research LLP published in March 2016 the Global AR and VR Market is poised to grow at around 16.7% in the next 5 years to reach approximately $2.95 billion by 2020⁸.

³ Maze War http://en.wikipedia.org/wiki/Maze_War
⁴ Habitat: http://www.youtube.com/watch?v=VVpulhO3jyc
⁵ Active Worlds: http://www.activeworlds.com/
⁶ Oculus: https://www.oculus.com
Types of Virtual Reality

**Immersive**
Completely involves the user’s personal viewpoint in the virtual world. The user experiences immersion or the feeling of being a part of that world. Usually a HMD (Head Mounted Display) is used, a helmet or face mask that holds the visual and auditory displays. A large projection displays are used to create a virtual background such as a room.

**Non-immersive**
- **Text-based VR**: The reader of a certain text forms a mental model of a virtual world in his mind.
- **Augmented VR**: The idea of taking what is real and adding to it in some way so the user obtains more information from their environment.

**Window on world (or Desktop VR)**
Systems that use conventional computer monitor to display visual world.

**Tele-Presence**:
Variation that links the remote sensors with the senses of human operator in real world. Popular applications include teleconferencing and operating remote controlled vehicles.

**Mixed Reality**
Merging Tele-presence and Virtual Reality systems. The user can see and hear a virtual world and can operate that world using Tele-presence.

**VR Hardware**

**Image Generators**
The hardware responsible for generating images. It is one of the most time consuming tasks in a VR system.

**Manipulation & control devices**
Tracking the position of real world objects such as head or hand. Conventional control devices (mouse or joysticks) or VR specific devices like the Instrumented Glove that uses inbuilt sensors that fit on fingers for tracking.

**Stereo Vision**
Works by creating two different images of the world, one for each eye. When the images are shown in rapid speed the brain fuses these two images in a single image producing the 3D experience.

**Head Mounted Displays**
These use some sort of helmet or goggles to place small video displays in front of each eye, with special optics to focus and stretch the perceived field of view. Most HMDs use two displays and can provide stereoscopic imaging. Others use a single larger display to provide higher resolution, but without the stereoscopic vision.

**VR Software**

Software tools for VR application development include:

**3D modeling software**
3D modelling software allows developing a mathematical representation of three-dimensional surfaces of objects. There are 3 popular methods of 3D modelling: Polygonal, using points in space called vertices that form Polygonal mesh; Curve, where surfaces are defined by curves; and the most recent one - Digital Sculpting.

**2D graphics software**
In VR 2D graphics software usually plays supporting role. Vector based graphics software is used in creating flat-surface elements for the GUI like info signs and visuals. Bitmap (raster) based graphics software is the more important one since it is used to produce textures to be used to “cover” 3D models and environment elements.

**VR simulation software**
It is used for very diverse applications on different software and hardware platforms. The simulation software is used to program how varied objects behave and set the rules that the virtual world follows. Its applications include military, transportation industry, medical, emergency services, police, education.

**Digital sound editing software**
It allows generating, converting and editing audio data. There are 2 main types – Destructive, which edit the data itself and Real-Time, which doesn’t edit the data itself but applies edits and processing over it to take effect while reproducing.

---

Virtual Worlds

“Virtual World” term is used to describe digital spaces that can be explored from within, where users can navigate through, interact with objects, other users and AI bots. Users can exchange information via text, audio, still images, animation and video. Usually the user’s presence takes is facilitated by an “avatar” - a digital 3D object that is used to represent the user. This representation is chosen by the user who may choose if his virtual identity has any real-world resemblance\textsuperscript{12}.

The currently popular virtual worlds are three-dimensional (3-D) computer rendered environments which can be accessed over a network, usually via Internet, populated by users in form of avatars who interact with the simulated environment and other users. These virtual worlds had moved beyond gaming and chat environments and transforming into powerful communication and education tools. The sensory immersion and the way of communication with other users make them a feasible alternative approach to tasks as distance learning and training, world-wide communication and collaboration. The number of private and public virtual world users grows steadily, from 300 million users world-wide in 2008 to forecasted 1 billion users in 2017\textsuperscript{13}, populating existing and new virtual worlds that are constantly being developed.

According to Gartner Research\textsuperscript{14} statement dated 2008: "Public virtual worlds, which are suffering from disillusionment after their peak of hype in 2007, will in the long term represent an important media channel to support and build broader communities of interest\textsuperscript{15}.”

Forrester Research\textsuperscript{16} predicted: “Within five years, the 3-D Internet will be as important for work as the Web is today. Information and knowledge management professionals should begin to investigate and experiment with virtual worlds\textsuperscript{17}.”

Virtual worlds are becoming a major technology for teaching, learning, research and collaboration. Virtual worlds constitute a growing online space for collaborative play, learning, edutainment and work.

Concepts and background

Virtual worlds rely on developing the centuries old human desire to get free from the boundaries of the real world. In 3D cyberspace users can interact with the virtual environment in a more life-like manner which can lead to development of new forms of human-machine interaction (HMI). Interaction with a computer


\textsuperscript{14} http://www.gartner.com


\textsuperscript{16} http://www.forrester.com

\textsuperscript{17} Getting Real Work Done In Virtual Worlds, Forrester Research, January 2008, http://www.forrester.com/Getting+Real+Work+Done+In+Virtual+Worlds/fulltext//E-RES43450?objectid=RES43450
by using a keyboard and mouse which became prevalent in the last 30 years is unnatural to humans and forces people to adapt to the existing technology.

Ideally a virtual environment would allow users to fully immerse in a highly convincing world which they can explore by means of all senses and interact naturally using new forms of communication and understanding.

Real life in virtual life

A virtual world can offer freedom from real life constraints and besides mimicking the real world experience allow the users to:

- Move though by walking, running, flying to explore the open spaces and three-dimensional objects (e.g. buildings, etc.)
- Interact with objects and perform virtual operations (open doors, access computer terminals, watch movies, etc.)
- Interact and communicate with other world residents (e.g. write to, talk, play, attend events, etc.)
- Take part in activities (e.g. training sessions, presentations, seminars, etc.)

All these and many more activities taking place in a 3-D environment replicate real-life experiences but without real-life constraints and with some degree of anonymity.

The software behind virtual worlds

The primary programming language for designing VR worlds is the Virtual Reality Modeling Language. VRML is a text scripting language for describing specific to 3-D objects properties such as lighting, texture and camera angle.

The virtual worlds precursors are computer programs initially known as Multi-User Dungeons (MUDs). The first MUDs appearing in late 1970s were text-based, words were used to describe interactions between the users and the virtual world. It is to be noted that all the interactions, environments, and communication occurred without graphics.

The current generation of 3-D virtual worlds roots can be found in the multi-user text-based games played over TelNet network in 1979. Looking back at the lots of variations and breakthroughs that took place in the last 35 years, we can summarize the history of virtual worlds in the following way:

- MUDs (Multi-User Dungeons)
- TinyMUDs
- MOOs (Multi-User Dungeons Object Orientated)
- MMORPGs (Massively Multiplayer Online Role-Playing Games)
- 3-D social virtual worlds.
Social virtual worlds such as There\textsuperscript{18}, Second Life\textsuperscript{19} and Active Worlds\textsuperscript{20} inherit similar features from MMORPGs, however they are not centered around a game. Games do exist in these virtual worlds but not the games are the central driving force but the social interactions taking place there\textsuperscript{21}.

These virtual worlds combine the power of the contemporary 3-D graphics and broadband Internet, allowing users to create new identity and contacts, alternative and sometimes supplementing the real world.

**Design of 3-D virtual worlds**

Currently there are millions of people visiting virtual worlds daily and this figure is expected to grow over the coming years. What is the main reason the users are attracted to these virtual places? How they spend their time there? How can the virtual worlds designers meet the users' needs?

Designing a virtual world is a challenging but rewarding task that opens almost endless possibilities for developers. Within the technological framework a designer possessing a creative imagination can do anything in the virtual world.

There are three critical attributes of effectively designed virtual worlds shaped by the experience of many designers, developers, researchers and users\textsuperscript{22,23}:

1. Thematic design of space
2. Promotion of presence
3. Awareness of the unique qualities of human behavior in online social environments.

**Thematic design of space**

The sense of three dimensional space and the feeling of being sharing it in real time with other people is one of the major differentiators between 3D virtual worlds and webpage-based environments. Virtual worlds designed and implemented in ways that suggest presence within a recognizable space are supposed to support more intense and timely communication among participants within those spaces. Using metaphors as a foundation helps to construct virtual spaces more efficiently.

---

Promotion of presence

A virtual space alone will not urge participants to communicate easily and effectively. It is the visualized presence that makes the critical difference between the virtual worlds and other online communication technologies.

The sense of sharing the 3D space with other users is an important factor to create a useful communication medium out of a virtual world. The sense of presence should be considered as a crucial component of virtual world design. It is a sense that tells the users if they are in the immediate vicinity of others and as it would be in real life it would be difficult to establish a successful interaction if it’s missing. No interaction among users and information will flow smoothly if it’s not present. In a 3D virtual environment the awareness of others originates from being emotionally and cognitively immersed in a shared space and the feeling of “being there.” However how “there” can be defined and described? Studies outline several aspects of presence within a virtual world: environmental, personal, and social.

The environmental presence is defined by the level of presence of participants and participants’ interaction. In this aspect the virtual worlds are similar to the other web-based environments. They all offer means of facilitating the users’ communication. Discussion boards, forums, chat logs, and blogs are tools that are shared in most types of environment. However 3D virtual worlds offer some possibilities that are not present in web-based social spaces. Mimicking the real world the 3D virtual environment participants can modify the position, shape, texture of environment individual elements and an entire island to denote presence. On the opposite, most web-based environments do not allow participants to modify the structure and appearance of the core elements.

The personal presence can be described as the awareness of being at a certain place physically. As in the environmental presence, the personal presence in virtual worlds and traditional web-based spaces is similar. A text chat in a virtual world and in Skype or Facebook illustrates how little the difference is. A really unique feature is the existence of nonverbal cues such as body position, gestures, proximity that avatars in virtual worlds easily represent.

The social presence in virtual words is registered through other participants’ verbal and nonverbal manifestation of behavior such as expressions, signs, gestures. Research has pointed out three important elements of social presence - context, communication, and interactivity – suggesting that each plays an important role in activating the various knowledge domains. For example, some authors suggest that a student’s perceptions of the presence of the instructor and the other students in an online class are related to the student’s attitude about the course and level of satisfaction with the process of learning. There are studies that suggest significant connection between virtual learning environments with high degrees of presence and all levels of learning, and that the virtual reality stimulation of the senses matched with the social presence plays a positive role in learning.

Summarizing all above the sense of not only feeling physically present, but also feeling at the same time the presence of other users is the most significant difference between 3D virtual environments and other web-based environments.
Human behavior

Unlike the traditional online environments which do not require that the learner is present in a visualized shape the virtual worlds can be accessed only using such shape called avatar. However, despite that the avatars are important prerequisite their presence alone can’t make the virtual worlds effective environments for communication. Another important factor is the real time connection, without it no effective communication can occur and this differentiates 3D virtual environments from other online technologies. Additionally, each user must act in order to establish fluid and meaningful communication. The user-to-user communication taking place in virtual worlds often mimics typical real world face-to-face interactions - body language, mimicry, gestures rendered by the avatars. The way participants behave in regard of issues of personal space and appearance, and how the acceptable behavior is modeled play a role in how effectively and efficiently communication will occur within the virtual space. In order to design effective virtual worlds it is important to understand the impact of constructs on the basis of human behavior and draw practical guidelines24.

3-D User Interface Design

Designing the user interface is a crucial task when developing any virtual environment application. Special attention is to be paid to any potential problems that might appear. Users, specifically the digital migrants, often have difficulties to orient and navigate in 3-D spaces. Adult users (and trainers) face much steeper learning curve compared with children and adolescents who use their experience gained from computer and video games.

The real physical world is still much more realistic regarding ways to perceive and understand it and contains many constrains that are not fully represented in the current generation of a virtual reality worlds. Thus significant effort is required when designing user interfaces and interaction techniques for 3-D applications. Obviously adaptation of the existing traditional interaction styles to 3-D does not provide a feasible solution to the problem. Innovative 3-D user interfaces that match ever closer real-world interactions, are constantly being developed25.

A design of 3-D user interface which is based on real-world mimicking should involve certain interaction techniques:

- Navigation - the task of navigation is the most prevalent user action in most large-scale 3-D environments, and presents challenges such as supporting spatial awareness, providing efficient and comfortable movement between distant locations, and making navigation lightweight so that users can focus on more important tasks.

• Selection and manipulation - interaction techniques for 3-D manipulation in VEs should provide means to accomplish at least one of three basic tasks: object selection, object positioning, and object rotation.

• System control - it refers to a task in which a command is applied to change either the state of the system or the mode of interaction. The issuing of a command always includes the selection of an element from a set. Thus, some similarities can be seen between system control and object selection techniques.

• 2-D interaction in 3-D environments - a common misconception of 3-D user interface design is that since the applications usually contain 3-D worlds where users can create, select, and manipulate 3-D objects, the interaction design space should only utilize 3-D interaction. In reality, 2-D interaction offers a number of distinct advantages over 3-D interaction techniques for certain tasks.

Users’ interactions in 3-D virtual worlds

An avatar is a 3-D digital representation of the user. Some virtual world platforms such as Second Life and OpenSim allow the users to customize their avatars. Users can select the appearance of their avatars, initially starting from a male or female human form. Next, the skin color, head and face from, hair type, body style and clothing can be adjusted. Additional clothing and accessories can be later on acquired in-world. Residents can always update or completely “redesign” their appearance, even choose non-human forms such as animals, plants, fiction characters or non-living objects.

Real-world physical limits do not necessarily apply to the avatars. In Second Life virtual world avatars can walk across the virtual landscape however they can also jump over very large distances, fly or just float in the air observing the goings-on below, or when in a hurry simply teleport themselves to a selected location. Users can also or detach their view from the avatar and fly around thus gaining an additional point of view, including on themselves.

Once logged inside the users can join groups and clubs. There are different types of groups - some provide services in-world, others specialize in a particular area like study subject or cause. The group membership can be displayed in the users’ profiles, an easy way to communicate to each other areas of interest and expertise.

There are users that explore every option available when configuring their avatar with enthusiasm. They continuously update their avatars, try out new outfits and to see what they are most comfortable with. Typically these users establish a relationship with their avatars, and the more they share experiences the more these relations strengthen. Sometimes the owner may go even further and starts to perceive him/her as being at one with the avatar, moving through the virtual environment as a single entity.

On the other end of the spectrum there are users that regard the avatar as just a technical requirement to use a virtual world. They often consider the avatars as an unnecessary hurdle when implementing their objectives in the virtual world. These users might get annoyed or irritated observing how other users get
involved with their avatars, often customize it and spend extra time and efforts on their digital representation, rather than focusing on a specific in-world task. 26

To summarize, people react differently to “diving” in a virtual world. There are people, who when encounter a virtual world, instantly feel at home and start to “live” there. Others are interested in the tasks and outcomes, minimizing the “virtual life” when possible. And there are people that don’t want to use a virtual world at all. These environments are not for beginners or technophobes. There cases when 3-D virtual worlds bring more and greater rewards than other “conventional” computer technologies. However these rewards to be achieved an initial extra efforts are required such as acquiring the basic skills, keeping effective communication and socializing effectively.

**Categories of 3-D virtual worlds**

The contemporary virtual worlds can be categorized as follows:

1. Role plays - multiplayer role play online games:
   - World of Warcraft, Everquest, Guild Wars
2. Social - open-ended exploratory immersive worlds:
   - Second Life, OpenSimulator, CyWorld, ActiveWorlds
3. Work-related - corporate and business 3-D spaces:
   - Project Wonderland, IBM’s Metaverse
4. Training - 3-D training simulations and serious games:
   - Forterra’s OLIVE platform
5. Mirror worlds - using geo-spatial databases and mapping services:
   - Google Earth, Planet Earth, Unype

The above categorization of virtual worlds is neither exhaustive nor definitive; it can be used as a starting point for exploring the wide range of applications that are currently available. More information on the

---

27 [http://www.worldofwarcraft.com](http://www.worldofwarcraft.com)
28 [http://everquest.station.sony.com](http://everquest.station.sony.com)
29 [http://www.guildwards.com](http://www.guildwards.com)
30 [http://www.secondlife.com](http://www.secondlife.com)
31 [http://opensimulator.org](http://opensimulator.org)
32 [http://us.cyworld.com](http://us.cyworld.com)
34 [https://lg3d-wonderland.dev.java.net](https://lg3d-wonderland.dev.java.net)
35 [http://eightbar.co.uk/2007/05/08/the-ibm-innovate-quick-internal-metaverse-project](http://eightbar.co.uk/2007/05/08/the-ibm-innovate-quick-internal-metaverse-project)
37 [http://earth.google.com](http://earth.google.com)
38 [http://www.planet-earth.org](http://www.planet-earth.org)
39 [http://www.unype.com](http://www.unype.com)
virtual worlds’ categorization and terminology can be found in *Serious Virtual Worlds, A scoping study* by Sara de Freitas\(^{40}\).

A comprehensive list of 3-D virtual worlds can be accessed at ArianeB website\(^{41}\).

Some of the application domains of 3-D virtual worlds today\(^{42}\):

- Education and training
- Social communication, networking and interaction
- Commercial (business and e-commerce)
- Medical
- Industrial Design
- Entertainment (single- and multi-player games)
- Tourism
- Media publishing, etc.

The more the technology develops, the more new application domains will appear.

**Educational potential of 3-D virtual worlds**

In the recent years the 3-D virtual worlds started to attract attention as platforms for learning. They offer new learning delivery channels through which training organizations can provide experiential or simulated learning and group activities in a shared space. A virtual world can provide a perfect multi-dimensional/sensory environment and a host of tools for informal learning, coaching, brainstorming sessions allowing real time sharing and exchange and also recording and capturing the ongoing activities. Nowadays the existing 3-D virtual worlds provide immersive learning delivery platform that can be adapted to different training scenarios\(^{43}\):

- Discovery learning by clicking on objects with associated information
- Reinforcement learning by offering a knowledge repository, tools, etc., associated with objects in 3-D
- Collaborative workspaces, such as 3-D classrooms and informal sites for discussion, encouraging school-style study and research
- Traditional instructor-led learning through a distance delivery method
- Simulated learning by modeling a process or interaction that closely resembles the real world in terms of fidelity and outcomes.

\(^{40}\) Sara de Freitas, Serious Virtual Worlds, JISC, [http://www.jisc.ac.uk/media/documents/publications/seriousvirtualworldsv1.pdf](http://www.jisc.ac.uk/media/documents/publications/seriousvirtualworldsv1.pdf)

\(^{41}\) [http://arianeb.com/more3Dworlds.htm](http://arianeb.com/more3Dworlds.htm)


\(^{43}\) 3D Learning and Virtual Worlds, Xerox white paper, 2009, [http://www.xerox.com/businessservices](http://www.xerox.com/businessservices)
3-D virtual environments possess several significant advantages over other training approaches:

- The experience can be much more engaging than a typical page-turning course
- The learner can learn by doing
- Expensive videoconferencing is not required for real-time online activity
- A user's learning experience can be designed to fit specific task needs with a flexibility and immediacy that is impossible in real life
- Exploration and discovery are encouraged
- Fantasy and imagination can be unleashed
- Virtual 3-D spaces often allow full recording of any activity, interaction, or exchange, enabling past events to be re-experienced or re-used
- Creed, skin color, look, and status within the organization do not count much in virtual spaces. Further, people with major physical handicaps appear as capable and as beautiful as anyone else, reducing discrimination

The ability to inhabit any type of body and to customize one's own look gives many people the opportunity to express themselves as they truly feel and not as society forces them to be.

Working with students/trainees in 3-D environments

There are several organizational steps to be taken before the training starts to ensure a smooth and successful training process.

- Find highly motivated teachers/trainers, explain the training objectives and ensure they possess adequate command of the virtual environment.
- Communication with the students/trainees – establish communication with the target group on a familiar social networking site they already use.
- Dissemination – publish as much information concerning the training as needed. Use varied dissemination channels.

Evaluate outcomes – prepare evaluation questionnaire, check the knowledge gained and the level of satisfaction among the trainees.

---

Virtual 3-D worlds: practical notes

There are many existing virtual world platforms and versions and they offer different functionalities. For example people that used SecondLife could expect OpenSim to behave in the same matter but this not always true.

Most virtual worlds are not web based services. Besides the setup of a centralized server the use of a virtual world environment requires installation of a client application running on PCs connected to the virtual world.

In order to install, configure and run such client application the users should possesses some degree of computer literacy and skills. The PC hardware also must meet a set of minimum requirements. The network should be configuration to enable the interaction between the client application and server through specific ports and protocols. In relation to the internet connection the available bandwidth must meet minimal requirements. The firewall at both school/organization network and at the PC level needs to be properly configured.

Ensure adequate technical support during the whole training process. At the early phase, it will be required to help during the installation and configuration of the client software and configuring the network. When the training starts and in-world activities take place the technical support will be needed for modifying the environment and using its facilities. To properly support the training different types of expertise will be required:

- knowledge of the installation and management of applications in different operating systems - Windows Mac, Linux;
- knowledge of networking for firewall configuration at the PC and network level;
- knowledge of the interactions and customizations within the particular virtual world.
References and additional resources

15. 3-D immersive virtual worlds in higher education: An Australian and New Zealand scoping study report, 2010 http://www.academia.edu/530936/3D_immersive_virtual_worlds_in_higher_education_An_Australian_and_New_Zealand_scoping_study

Web site accompanying the book "Learning in 3-D", http://www.learningin3d.info
19. Educational Uses of SecondLife (online video),
   http://www.youtube.com/watch?feature=player_embedded&v=qOFU9oUF2HA
20. Ahmad John Reeves and Shailey Minocha, Relating Pedagogical and Learning Space Designs in Second Life, The Open University, UK, 2011,
   http://www.igi-global.com/chapter/relating-pedagogical-learning-spacedesigns/52391
23. Virtual Worlds: 20+ Tools for Creating 3D Graphics and Environments,
   http://mashable.com/2008/07/10/tools-for-3d-creation/
25. 3-D virtual worlds list, http://arianeb.com/more3Dworlds.htm
28. Why virtual reality could finally mend its broken promise, Erin Carson
Software for Virtual Worlds

Introduction

A virtual world is an online community that takes the form of a computer-based simulated environment through which users can interact with one another and use and create objects. The term has become largely synonymous with interactive 3D virtual environments, where the users take the form of avatars visible to others. These avatars usually appear as textual, two-dimensional, or dimensional representations, although other forms are possible (auditory and touch sensations for example). In general, virtual worlds allow for multiple users.

The computer accesses a computer-simulated world and presents perceptual stimuli to the user, who in turn can manipulate elements of the modeled world and thus experience a degree of telepresence.

As virtual world is a fairly vague and inclusive term, the above can generally be divided along a spectrum ranging from:

- massively multiplayer online role-playing games (MMORPGs), also called virtual game worlds, the user playing a specific character is a main feature of the game.
- massively multiplayer online real-life games (MMORLGs), also called virtual social worlds, where the user can edit and alter their avatar at will, allowing them to play a more dynamic role, or multiple roles.
- Systems that have been designed for a social application include: Second Life, Active Worlds, Twinity, Kaneva, Smallworlds, Onverse.

There are various approaches for developing 3D Worlds, each one with different advantages and disadvantages.

Developing an application from scratch using a programming language and graphics libraries used to be the only option in previous decades. This option requires a vast amount of time since you'll be creating a game engine specifically for the virtual world you want and requires high level of programming skills in order to achieve something of quality.

Game development platforms are highly sophisticated software kits that allow one to create 2D and 3D games/applications using a supplied game engine with little limitation in terms of customization. They offer toolkits that can help someone carry out specific tasks but in overall it takes a lot of time for someone to learn and use it and in order to crate a 3D Virtual World with social interactions a lot of development. Two of the most important game development platforms right now are Unreal Engine 4 and Unity 3D, both offering excellent level of graphics and they are used by top game development companies, creating state of the art game experiences.
Existing MMO games are also an option in some cases, since they already offer social interactions between users. Some of them also offer capabilities to create content inside the game, but the customization options are limited and in most cases it is impossible to use scripts to design your own scenarios.

The easier option to quickly create a virtual world with social capabilities, is by using dedicated virtual world platforms. These platforms can already be online, offered as services where you connect and create your content (e.g. Secondlife, Kitely) or offered as software that you can install on your own machine as a server process, where client users will connect to. This option can definitely save a lot of time since most of the necessary Virtual World functionality is already implemented and one can focus on creating the content of the virtual world. Using scripts to create custom scenarios is usually a feature available in these platforms but customization options are limited compared to the game development platforms.

In the following section we present some of the most popular platforms, representing the different approaches described above.

**Representative Software**

**Unity 3D**

Unity is a cross-platform game engine developed by Unity Technologies and used to develop video games for PC, consoles, mobile devices and websites. Unity is notable for its ability to target games to multiple platforms. Within a project, developers have control over delivery to mobile devices, web browsers, desktops, and consoles. It is a powerful 3D engine and a user friendly development environment, Easy enough for the beginner and powerful enough for the expert.

There is a Pro edition that comes with more features and tools, however, a free version allows you to build complete games, and even publish them to the desktop and the web without paying anything. Games published under the free edition will have a small Unity watermark). Unity has been used for many popular video games published recently and is being used by both big companies and individual "indie" game developers.

As a general purpose game engine Unity can be used for almost any kind of game, including of course Multiplayer Virtual Worlds. However it requires a great amount of work to develop the required functionality of a virtual World and experience and skills with programming and working with 3D modelling, shading and other advanced concepts.
Unreal Engine 4

The Unreal Engine is another high end game engine developed by Epic Games, first introduced in the 1998 first-person shooter game Unreal. Although primarily developed for first-person shooters, it has been successfully used in a variety of other genres, including stealth, MMORPGs, and other RPGs. The code is written in C++, so the Unreal Engine features a high degree of portability and is a tool used by many game developers today. The current release is Unreal Engine 4, designed for Microsoft’s DirectX 11 and 12 (for Microsoft Windows, Xbox One, Windows RT); OpenGL (for OS X, Linux, PlayStation 4, iOS, Android, Ouya[5] and Windows XP); Vulkan (for Android); Metal (for iOS); and JavaScript/WebGL (for HTML5 Web browsers).

As of March 2, 2015, Unreal Engine 4 is available to everyone for free. There is a very active marketplace with assets, many offered for free by Epic Games itself, where one can find, download and use in his project. Unreal Engine 4 is a complete suite of game development tools made by game developers, for game developers and offers everything you need to create high quality games.

Figure 1: photorealistic scene created with Unreal Engine 4

Similarly with Unity 3D, the Unreal Engine can be used to create Virtual Worlds but requires a lot of effort and skills to use it. The quality of the graphics can be excellent but it also means that there are very high requirements in hardware to use it for development (a very expensive GPU card), and the resulting game will also need a high end computer to run.

World Of Warcraft

World of Warcraft is a Massively Multiplayer Online Roleplay (MMOR) game, where players take on the identity of characters in a narrative-rich plot, working together to overcome challenges. World of Warcraft was released in 2004 by Blizzard Entertainment and still remains the world’s most-subscribed MMORPG and holds the Guinness World Record for the most popular MMORPG by subscribers.
World of Warcraft requires a subscription fee to be paid to allow continued play, with options to pay in one-month, three-month, or six-month blocks. WoW uses Lua, a scripting language that can be used to create custom Interface modules. Unfortunately, apart from this, there is almost no room for players to create custom content.

Figure 2: players cooperating to overcome challenges in WoW

There are occasions where WoW has been used in education, but it is definitely not an option if you want to create your own content. However since it is collaborative game and players are usually forced to work together to overcome obstacles in the game, it could successfully help in language training, if a trainer oversees the students and creates scenarios for them to follow.

Minecraft

Minecraft, released in 2011, is a popular video game in which players can work together in a sandbox area to use various types of simple blocks to construct buildings in a 3-D world. While constructing creative and imaginative buildings, players can explore the land, gather resources or fight in combat. There are many occasions where Micecraft has been used as a tool for learning. Minecraft is currently being used in classrooms around the world to teach everything from STEM subjects to arts and poetry.

Unlike WoW and other MMORPs that focus on offering players a detailed, specific scenario and quests to follow in the game, Minecraft is an open world that promotes creativity, collaboration, and problem solving, with users actually forced to create their own content. In January 2016, Microsoft announced a new tool for education, called Minecraft: Education Edition or MinecraftEDU, planned to be released in 2016. The student’s characters in MinecraftEDU will be able to retain characteristics. Students will also be able to download the game at home, without having to buy their own version of the game. Finally the last
large difference is that students can take in-game photos. These photos will be stored in an online notebook with the students online notes.

![Figure 3: simple math exercises in Minecraft](image)

Minecraft is a great tool for trainers to create collaborative assignments for the students, where they need to build something based on the specifications (e.g. something they were taught in classroom). The limitations are mostly the quality of the "blockish" graphics that do not allow detailed items and the limited options to use scripts and create your own scenarios.

**Entropia Universe**

Entropia Universe is a massively multiplayer online virtual universe designed by the Swedish software company MindArk. Nearly all in-game activities require expendable resources which must be purchased from vending machines (or other players) but there is no account fee or connection charge.

**Neverwinter Nights**

Neverwinter Nights is a third-person role-playing video game developed by BioWare. Its sequel Neverwinter Nights 2 was developed by Obsidian Entertainment and published by Atari. The games are set in the fantasy world of the Forgotten Realms and the game mechanics are based on Dungeons & Dragons rules.

The game engine was designed around an Internet-based model for running a massively multiplayer online game (MMOG), which would allow end users to host game servers. More importantly the games are accompanied by powerful toolsets that allow users to create their own adventures, focusing on the scenarios and dialogues of the game. More specifically Neverwinter Nights included the so-called "Aurora toolset", a collection of tools allowing users to create their own digital adventure modules to be played either in single-player or in online multiplayer. The toolset enjoyed great popularity among the modding community, with over a thousand fan-made modules produced in it within half a year after the release. Similarly Neverwinter Nights 2 included its own toolset called the Electron. Through the Electron toolset,
Neverwinter Nights 2 gives you all the tools you need to build your own modules, campaigns, and adventures - create buildings, terrain, script encounters, write dialogues, create quests and items.

Figure 4: using the toolset to create your own adventure (source: Gamespot)

Kaneva

Kaneva, LLC is a privately owned Atlanta-based company founded in 2004 by Christopher Klaus and Greg Frame. Kaneva is a 3D Virtual World that supports 2D web browsing, social networking and shared media. Kaneva was originally founded to develop a massively multi-player online game (MMOG), but it was later decided to use the technology to develop a virtual world that combined video sharing, social networking and 3D environments. The Virtual World of Kaneva was released into beta in mid-2006.

Second Life

Second Life (Linden Lab, 2003), is probably the most popular of the social worlds, with a large active user base and the most active educational community. Secondlife is open to users over 18 years of age and offers a detailed 3D environment and avatars, voice and text communication tools. It is also a social network, with users able to be members of groups and to create and share information and objects. Second Life allows building objects, that are completely owned by their creators, and there is a very active marketplace where one can buy or sell content (clothes, accessories, homes and diverse items including educational tools). Most of Second Life itself was created by its users.
and since there is not a specific objective, creating objects and areas is one of its main activities. SL also has a vibrant economy with an enormous market for clothes, accessories, homes and diverse items including educational tools.

The environment is completely customisable for the land owner of an area. Second Life is free to use for the purpose of creating an avatar and visiting the lands created by other users, but users have the option of paying a small monthly fee to get a parcel of land where they can build their own content. More serious building projects require the purchasing of an island and the payment of a monthly rental fee, in addition to the initial terra-forming and design and building costs.

Many top universities and educational organisations have presence in Second Life, and many trainers have been using it as a tool in classes of various subject areas.

![Figure 5: Ohio University's Second Life campus](image)

Although SL is proprietary software, Linden Lab has released the source code leading to the creation of many open-source tools and viewers by the user community. The most important of these open-source platforms is OpenSimulator that allows running your own server of a Virtual World, with almost identical functionality with Second Life but no limitation in terms of land. OpenSimulator has been growing on its own and currently offers its own great community of users and the educational community surrounding it is also growing quickly. OpenSimulator, is the software we decided to use for the Virtual World that will be created in this project, so we are going to present it in detail in the next section of this report.

**Active Worlds**

Active Worlds was launched in 1997 and works much in the same way as Second Life. Although it is free to use with the limited 'tourist' account, paying a small monthly fee allows one to become a 'citizen', offering them features such as having a unique name, unrestricted access to any world, the ability to customize their avatar and to build their
own content, and access to social networking features such as voice chat, IM and file sharing (tools which are usually offered for free by default in most other social worlds). For users needing more control over their environment and more privacy, personal firewall-protected Universes are available for enterprises and educational projects. These are separate worlds from the main universe and are priced starting at $2400. A separate set of worlds and a community for educational projects is also available named Active Worlds Educational Universe and boasts over 80 participating organization. Despite this, Active Worlds is often disregarded in favor of Second Life, especially with regards to educational projects.

**Open Wonderland**

Open Wonderland (previously Project Wonderland) is a Java open-source toolkit for creating collaborative 3D virtual worlds. It has recently become a community-supported open-source project. Users can communicate with audio, share live desktop applications and documents. Open Wonderland is completely extensible; developers and graphic artists can extend its functionality to create entirely new worlds and add new features to existing worlds.

![Figure 6: Space vehicles exhibit in a Wonderland virtual museum (source: mfeldstein.com)](image)

**Open Cobalt**

Open Cobalt is a free, open source software platform for constructing, accessing, and sharing virtual worlds both on local area networks or across the Internet, without any requirement for centralized servers. One of the main features that Open Cobalt offers, in comparison with other platforms is that it does not require the use of servers to create and share virtual worlds (since it is based on a peer-to-peer synchronization architecture/messaging protocol).
## Comparison Table

<table>
<thead>
<tr>
<th></th>
<th>Free</th>
<th>Open source</th>
<th>Graphics</th>
<th>Text</th>
<th>Voice</th>
<th>User community</th>
<th>Custom content</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SecondLife</strong></td>
<td>no</td>
<td>no</td>
<td>Very Good</td>
<td>yes</td>
<td>yes</td>
<td>Very Active</td>
<td>Yes</td>
<td>Virtual World Platform</td>
</tr>
<tr>
<td><strong>OpenSim</strong></td>
<td>yes</td>
<td>yes</td>
<td>Very Good</td>
<td>yes</td>
<td>yes</td>
<td>Very Active</td>
<td>Yes</td>
<td>Virtual World Platform</td>
</tr>
<tr>
<td><strong>Active Worlds</strong></td>
<td>no</td>
<td>no</td>
<td>Good</td>
<td>yes</td>
<td>yes</td>
<td>Fairly Active</td>
<td>Yes</td>
<td>Virtual World Platform</td>
</tr>
<tr>
<td><strong>Open Cobalt</strong></td>
<td>yes</td>
<td>yes</td>
<td>Good</td>
<td>yes</td>
<td>yes</td>
<td>Not Active</td>
<td>Yes</td>
<td>Virtual World Platform</td>
</tr>
<tr>
<td><strong>Open Wonderland</strong></td>
<td>yes</td>
<td>yes</td>
<td>Very Good</td>
<td>yes</td>
<td>yes</td>
<td>Not Active</td>
<td>Yes</td>
<td>Virtual World Platform</td>
</tr>
<tr>
<td><strong>World of Warcraft</strong></td>
<td>no</td>
<td>no</td>
<td>Great</td>
<td>yes</td>
<td>yes</td>
<td>Very Active</td>
<td>No</td>
<td>MMO Game</td>
</tr>
<tr>
<td><strong>Minecraft</strong></td>
<td>no</td>
<td>no</td>
<td>Low</td>
<td>yes</td>
<td>yes</td>
<td>Very Active</td>
<td>Yes</td>
<td>MMO Game</td>
</tr>
<tr>
<td>Game</td>
<td>Multiplayer</td>
<td>Rating</td>
<td>Very Active</td>
<td>Fairly Active</td>
<td>Very Good</td>
<td>Yes</td>
<td>No</td>
<td>Game Engine</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------</td>
<td>--------</td>
<td>-------------</td>
<td>---------------</td>
<td>-----------</td>
<td>-----</td>
<td>----</td>
<td>--------------</td>
</tr>
<tr>
<td>Never Winter Nights 2</td>
<td>yes</td>
<td>no</td>
<td>Very Good</td>
<td>yes</td>
<td>yes</td>
<td>Fairly Active</td>
<td>Yes</td>
<td>Multiplayer RPG</td>
</tr>
<tr>
<td>Unity 3D</td>
<td>yes</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
<td>yes</td>
<td>Very Active</td>
<td>Yes</td>
<td>Game Engine</td>
</tr>
<tr>
<td>Unreal Engine 4</td>
<td>yes</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
<td>yes</td>
<td>Very Active</td>
<td>Yes</td>
<td>Game Engine</td>
</tr>
</tbody>
</table>

References and additional resources

1. Open Simulator Main Page, [http://opensimulator.org/wiki/Main_Page](http://opensimulator.org/wiki/Main_Page)
5. Unreal Engine [https://www.unrealengine.com/](https://www.unrealengine.com/)
6. Unity Engine [https://unity3d.com/](https://unity3d.com/)
10. Active Worlds [https://www.activeworlds.com](https://www.activeworlds.com)
OpenSimulator

Introduction

OpenSimulator is an open source 3D application server that can be used to create a Virtual Environment, accessible through a variety of clients. It can simulate virtual environments similar to Second Life, supporting the core of its protocols and functionality.

OpenSimulator has a very active community of people and groups that contributing to its development and also the creation of open content. Many universities and other educational institutes have integrated OpenSimulator as part of their courses and have successfully used it for innovative research on learning methodologies.

In the following chapters we will present and describe many of the features offered by OpenSimulator.

Architecture

OpenSimulator is a server application that runs on a machine (or multiple machines when used in grid mode) and serves client applications (3D Viewers) through HTTP messages and the Second Life protocols.

The OpenSimulator server can be used in two modes. In the standalone mode the whole simulation is run as a single process so it can only run in a single machine. This makes it much easier to configure but it is not possible to connect the hosted virtual world with other Virtual World online. In the alternative mode, called grid mode, various aspects of the simulation can be separated and run as different processes across different machines. This allows multiple Virtual Worlds running in different servers across the internet to share common users and assets data, giving users the ability to easily teleport from one to another while keeping their inventory items, thus forming a hypergrid of connected worlds. The hypergrid is effectively supporting the emergence of a Web of virtual worlds.

OpenSimulator is written in C# and is designed in a way that allows to be extended through modules. OpenSim uses a database to store almost all content of the 3D Worlds. It can be configured to use any of the popular database software (mysql, postgres e.t.c).

The software running on the server is just a command line program that communicates with the clients and sends the necessary information that will be displayed on the client’s machine. Consequently the server machine does not require a demanding 3D graphics card. Client machines on the other hand will need a machine with a good GPU since the rendering will take place in them. For the server running the simulation it is important to have a sufficient amount of memory, particularly if many users (client computers) will connect at the same time.
Installation

To Run OpenSim you need:

For Windows: .NET Framework 3.5
For Linux: At least Mono 2.4.3

Additional software:

- Freeswitch server for Voice communication (optional)
- Apache server with PHP is required for some modules (e.g. offline messages or groups) to work but is not necessary to run the simulation
- MySQL Server or other databases can be configured to be used instead of the default, already embedded in OpenSim (MariaDB).

OpenSim does not need installation. Only Extraction of the archive file. There are a number of important configurations files with parameters (such as credentials for connecting to the SQL server etc) that can be configured by the administrator. To initialize and run the simulation, the administration runs the OpenSim executable and then has access to a console displaying logging messages with the communication with clients and allowing using a set of commands. The administration can use these commands to create users and regions, import terrain height maps or folders of items in inventories and many other administrative tasks.

Regions

Regions are individual areas inside the 3D world. A virtual world can have multiple regions. Each region has a location in the word's map, and may have other regions in the four adjacent, neighbouring areas. By default the regions are independent of each other and users teleport from one region to the other (even when they move to neighbouring ones). However, a recent feature, "megaregions" allows many regions to be combined so that there is no border transition between regions for the users who are able to move from one region to the next without teleportation delays.

Terrains

The terrain is the ground surface of the regions, configured through a heightmap that indicates the distance of each individual position from the lowest plane. A specific height can be set as the sea level, so anything below will be filled with water. Separate, custom textures can be used for different height ranges, thus allowing rocky textures for higher heights (e.g. mountains), grass textures for lower heights, as desired by the creator of the 3D World. The heightmap can be manually set from an avatar (with ownership rights) inside the Virtual World using a variety of tools (flatten, raise, smooth e.t.c), but there are also automated
tools, such as L3DT that can generate a heightmap from images or other sources and then be imported in regions in the virtual world. It is even possible to have a larger height map be spread over multiple regions, thus keeping a smooth continuity between them. Finally using various tools available online, one can use actual real height data from relevant datasets and easily generate heightmaps that accurately correspond to real areas in the world (e.g. a specific island in the world).

3D Graphics

Similar to other worlds and also 3D modelling software, one can add 3D objects in a region by creating and then manipulating simple objects called "prims". Initially the user creates one of the basic prims (cube, cylinder etc) by dragging and dropping them in a position, and then can use corresponding controls to scale, move or rotate them.

Each prim has a menu with various options/tabs to modify its parameters:

- **General**: General information like the name and description of the object, the Owner and the permissions of others.
- **Object**: The location (x-y-z location) inside the world, its rotation, size and type. An object can be physical, so it follows physical laws as defined by the physics engine, if used.
- **Features**: These options allow configuring the lighting cast on the object or a Flexible Path that allows it to behave like complex, dynamic objects like cloth, hair etc.
- **Texture**: These options allow to set specific texture to each distinct plane of an object.
- **Content**: Each prim has its own inventory and so can store other objects and most importantly script files that contain code (using LSL syntax) that defines its behavior.

The parameters for modifying the shape of an object are extensive and allow creating intricate, complex objects. An important aspect is also the ability to connect multiple prims into link sets. The individual
objects will act as parts of a single object and this also facilitates the interaction between one another. For example, it is possible to link one object to another and add scripts so that it rotates around it.

Given time, one can learn to create complex, detailed objects, by joining simple prims, but there is also the option to import 3D models designed in more advanced modelling software such as Blender, Modo, SketchUp e.t.c., through the widely used Collada format. There are many communities online that offer for free existing 3D models in collada format, that can be directly imported in OpenSim.

3D Viewers and User Accounts

The Virtual World is accessible for visitors through a special type of software called a 3D Viewer. There are multiple alternative 3D Viewer options, such as Kokua, Firestorm, Imprudence and Singularity and although they have some differences in the user interface they mostly share the same functionality.

Users will have to create a user account to gain access to the 3D world and this is usually done through a web site interface where they provide a name for their avatar (First and Last name) and a password. There are different alternative user management interfaces such as Wifi, based on different technologies. If an interface is not used then the server’s administrator would have to manually create accounts for interested users that want to visit the 3D World.

To connect to a specific Virtual World running on a distant server, the user needs to know the server’s address and port on which the simulation is running (similarly to the adders of a web page) and provide the credentials for the user he had previously created as described above.

Avatars

Users entering the virtual world, control an Avatar character representing them. The avatar is a crucial aspect in order to achieve a high degree of immersion. Avatars can be customised to take the desired appearance, by modifying body parts (Shape, Skin, Hair and Eyes), clothes or using various items as attachments.
It is possible to create a custom 3d model and rig (the skeleton of the character) for the Avatar using 3D modelling software, instead of using the default shapes. For skin, hair and eyes, custom textures can be used allowing very detailed and realistic characters.

A set of different clothe types are available that can be adjusted using menu sliders but one can also create custom, complex ones, using the attachment slots. Attachment are 3D objects (single prims or link sets) that can be "attached" and then adjusted to specific slots of the character (e.g. left hand to have the avatar hold the item, head to wear a hat or helmet, chin to attach a beard). Using the "flexible path" parameters of an object, it is possible to create attachments that behave in a very realistic manner, such as hair that flow when the character moves, or following the direction of the wind.

The animation of the character (e.g walking or flying around) can also be overridden by adding custom animation files in the popular motion capture format "bvh". Many datasets of motion capture data in that format are already available online, that one can find and use directly or with minor adjustments. These may have been manually made or automatically through motion capture techniques, recording the animation of real people. It is not difficult to create a custom animation using software like QAvimator, that displays an Avatar and rig of an OpenSim character and allows setting the position of each part of the body for every time point in the timeline. Animations can be triggered inside the world manually by the user, by selecting and "playing" an animation file in his inventory, or as a result of an action (e.g touching an object) or a script.

Figure 7: using sliders to adjust the shape of an avatar

Figure 8: a custom avatar using skin textures and attachments
Controlling the character and the camera view is fairly easy and usually through the same keys used in other Virtual Worlds or popular 3D Video Games. Avatars can walk, run, crawl and even fly around, unless the region’s administrator has specifically restricted it.

Textures

Textures are 2D images that are applied on the planes of 3D objects. Creating quality textures for the 3D objects is crucial to achieve detailed, realistic worlds and for complex objects is a difficult and demanding task referred to as "UV mapping". However, when working on simpler, basic prims with limited planes it is much easier.

OpenSimulator also allows using a mask over the textures to easily modify the colour, add a glowing effect on it, or even setting a transparency value, giving it an opaque view.
Image formats with transparency capabilities such as PNG, can be used to create simple 2D objects that appear as 3D from specific angles.

Finally a recent feature is MOAP (media on a prim) that allows projecting a web site on the surface of a prim. This is a very important aspect for trainers, that allows them to project existing learning material (e.g. text or presentations) inside the Virtual World, so they do not need to create 3D objects.
Physics Engine

OpenSimulator offers a number of alternative modules that can simulate Physics behaviour for specific objects. Setting a particular object as "physical" will cause it to fall towards the earth, or respond to the collision of other objects or avatars with it.

Communication

It is possible to interact with other avatars (representing other users), through text messages, voice if a microphone is connected or even gestures and animations of the avatar. Messages can be public (displayed by all nearby avatars) or private, visible to specific only users.

Voice communication is not present by default but can be enabled by running and configuring appropriate software like FreeSWITCH. A module can also be added to allow storing offline messages that will be...
displayed to users when they log in the world (the default setting requires both users to be online to communicate with messages).

**Basic Interactivity**

Avatars can interact with various objects in the 3D world by touching (clicking) them or other events that can trigger a behaviour that has been scripted in them. For example clicking on a chair object may cause the avatar to sit on it.

Every avatar has an Inventory, which is a file directory that stores various types of files such as 3D objects/prims, sounds, animations, scripts, body parts, clothes e.t.c.

**Scripting (LSL)**

Scripts are files that describe the behaviour of an object using the LSL language developed by Linden Labs for Second Life. As mentioned earlier each prim has an inventory allowing it to contain one or more script files. Inside a script, one can define and describe different states for the object. For each state, multiple event listeners can be added that wait a particular event in order to be executed. For example a “touch” listener will be triggered when an avatar clicks on an object.

Some of the event listeners that can be used are:

- wait for an avatar to click (touch) the object
- wait for an avatar to collide with the object
- wait for an avatar to approach the object (specific radius)
- wait for specific time interval to pass (e.g every 10 seconds)
- wait for a message to be sent to a specific chat channel
- wait for the state of the simulation server to change (e.g when the server starts)
- wait for the object to be rezzed (inserted in the 3D World from a user’s inventory)
- wait for an object to be attached to an avatar

Using corresponding commands as the body of these listeners, an object can be programmed to perform various actions such as:

- move to a specific position or move specific units towards a direction
- change its size
- rotate around a specific axis or relevant to another object.
• change state (so other listeners will be used)
• change or move the texture applied in one or all of the object's planes (this can be used to simulate running water or similar effects).
• change the transparency of a texture
• make a plane of the object glow
• play a sound file (up to 10 seconds) stored in its inventory
• write a text message in a particular chat channel
• show particular message as a label over the object.
• create a dialogue menu for a specific avatar to provide an answer to a question.
• wait (sleep) a specific interval of time before the next command
• make an avatar teleport to another location inside the Virtual World (or other worlds if running is grid mode).
• give an object to a specific user (added in his inventory if accepted)
• change the speed of the avatar

As an example, a lamp object is initially in a state called “off” and waits for the event of some user clicking/touching it. When a user clicks it then it executes an action that makes the lamp turn on (emits light) and then change to another state called “on”. While on state “on” the object waits for a user to click it so it turns off and returns to state “off”.

Probably the most useful (albeit difficult to apply) feature of the scripting language is the ability of objects to send messages to specified chat channels, while being able to listen to messages in other channels. This allows having different objects across the simulation communicating with one another.

Non Playable Characters

Using a specific set of scripts, it is possible to generate and control the behaviour of NPC Characters (avatars that are not controlled by actual users).

Some of the actions available to control these characters are:

• NPC character is created at target position
• NPC walks (or runs, flies or uses any other animation) to a target position.
• NPC rotates to face towards a direction.
• NPC does some animation (wave, point, dance, attack etc).
• NPC says a message (text) in nearby chat that appears on screen.
• NPC randomly wanders around a point.
• NPC follows the user-avatar.

These actions can be used along with the scripts described earlier to program in detail the behaviour of the characters based on events triggered by real avatars or the environment.

It is therefore possible to create for example an NPC guide that senses a real user approaching him and offer to show him around the Virtual World, while talking (using sound files) or displaying text messages.
NPC characters can act as simple static bots that just give specific information, or even responsive dialogue agents that ask questions and respond to the users answers (through dialogue menus or by writing messages in specific chat channels).

**Special Scripts**

**Particles:** Using specific script commands, an object can emit particles that can be configured through a variety of parameters to simulate particular effects such as smoke, laser beams, snowflakes, rain, falling leaves e.t.c.

**Vehicle Script:** Adding and configuring a specific "vehicle" script in an object can turn it to a vehicle that can be driven around by avatars.

![Figure 15: A solar car created for project AVARES](image)

**Gun script:** This script can be configured to make a character throw specific objects (e.g. bullets or a basketball) towards a direction with adjustable speed.

**HUDs (Head-Up Displays)**

Objects can be attached on an avatar in specific slots used for HUDs. These objects will be stationary in a specific position on a user's screen, so they can be used to create custom interfaces for displaying specific messages or textures to a user and even use scripts to allow interactivity with the HUD. For example users will wear a HUD item and a small window will appear on their screen with a score updated based on their progress.

**Integration with LMS**

A very important capability of OpenSimulator and the LSL scripting language are the commands allowing it to send and receive HTTP messages, communicating with other servers or applications as well as querying and manipulating a database. This makes it possible to have other applications such as custom web sites or
even Content and Learning Management Systems fetch and display data from the virtual world or send and manipulate data inside the Virtual World.

Taking advantage of these capabilities, integrations with LMS environments have been developed, with most notable among them being SLOODLE, an integration of OpenSim/Secondlife with popular LMS platform Moodle offered as a free and open source project. SLOODLE provides a range of tools for supporting learning and teaching to the immersive virtual world; tools which are fully integrated with a tried and tested web-based learning management system used by hundreds of thousands of educators and students worldwide.

Some of the features offered by SLOODLE are:

- **Web Intercom** allows students participating in chats in Opensim using the accessible Moodle chatroom, with all discussions archived securely in the Moodle database.
- **Presenter** allows uploading slides in Moodle and having them available inside the virtual world.
- **Choice** allows capturing feedback from users in the 3D World.
- **Assignment Drop-Box** allows managing in-world assignments through the moodle interface.
- **Vendor** can be used to distribute inventory items.
- **Reg-enrol booth** allows Moodle Users to be associated with corresponding avatar characters in the world.
- **Quiz chair** allows avatars taking quizzes inside the world and managing the results by integrating the Moodle gradebook.
- **Scoreboard** can be configured to track progress of the students.
- **Rezzer** can store and recreate scenes of 3D objects and activates.
References and additional resources

1. Open Simulator Main Page,  
   http://opensimulator.org/wiki/Main_Page
2. Open Simulator Wikipedia Page,  
   https://en.wikipedia.org/wiki/OpenSimulator
3. L3DT - Large 3D Terrain Generator,  
   http://www.bundysoft.com/L3DT/
4. Second Life Portal  
   http://secondlife.com
5. Second Life Wiki  
   http://wiki.secondlife.com/wiki/Main_Page
6. LSL (Linden Scripting Language)  
7. QAvimator  
   http://www.qavimator.org/
8. FreeSWITCH  
   https://freeswitch.org/
9. SLOODLE  
   https://www.sloodle.org/