WP

WORLD OF PHYSICS

Deliverable Number	**Report IV**
Deliverable Title	**Gamification and Learning Opportunities in Virtual Worlds**
Intellectual Output Title	**Intellectual Output I: Reports on Physics Education in Schools around Europe and the state of the art in 3D Virtual Worlds**
Activity description	**Research on Gamification and Learning Opportunities in Virtual Worlds**
Authors	**UPAT**
Status (D: draft; RD: revised draft; F: final)	**F**
Date (versioning)	**30/12/2016**

Partners

University of Cyprus, Cyprus

https://www.cs.ucy.ac.cy/seit/

University of Patras, Greece

http://www.upatras.gr/en

CTE, Romania

http://www.etcenter.eu/index.php/en/

Computer Technology Institute and Press "Diophantus", Greece

http://www.cti.gr/en

ITD-CNR, Italy

https://www.cnr.it/en

NEW EDU, Slovakia

http://www.newedu.sk/

Executive Summary

This report firstly gives a quick overview of Gamification, the various Learning Types and the features offered in 3D Worlds and then presents various opportunities for learning scenarios in the Virtual World.

List of abbreviations

Abbreviation	Definition
VW	Virtual World
NPC	Non Playable Character
HUD	Heads Up Display
MOAP	Media On A Prim
Prim	Primitive Object (cube, sphere, cylinder e.t.c)

Table of contents

List of Figures

1. Gamification

1.1.   What is Gamification?

Gamification is simply the application of game-like elements and mechanics such as score, time limits and awards systems to engage and motivate people to achieve pre-determined goals such as improving performance or learning something. Most Games have some common features: There are rules, objectives and some type of reward or recognition for achieving the goal. Most games try to be fun and entertaining. Gamification tries to take the essence of games, and apply it into real world, non-game situations.

1.2.   Game Mechanisms commonly used in Gamification Strategies

Achievements: Achievements are a representation of having accomplished something. Achievements can increase the challenge of the game and motivate users even more. Achievements are often considered "locked" until you have completed the steps or tasks that are required to "unlock" them. Many gamification platforms offer Badges when completing tasks.

Appointments: The user has to log-in and/or make an action in game at a predetermined time or place to complete a task.

Blissful Productivity: Giving the user the impression that they are doing something meaningful and rewarding.

Bonuses: These are rewards that are applied for completing some special challenge or combination of challenges.

Cascading Information Theory: During the game progression the information is released in small snippets, in order for the user to retain the appropriate level of understanding.

Combos: These involve the combination of different things and usually come with additional rewards.

Community Collaboration: An entire community is motivated to work together and try to solve a riddle or a challenge.

Countdown: The user has a limited amount of time to complete a task.

Exploration: Encouraging the user to explore and discover new places / pages.

Epic Meaning: Inspiring the user that they are working to achieve something great, something bigger than themselves.

Infinite Gameplay: Some games do not have an explicit end. Many of these games have a way to dynamically generate content, tasks and rewards to keep the users engaged.

Levels: The user's level is incremented after gathering a predetermined number of points or completing specific tasks. Progressing to higher levels usually involves unlocking new features or abilities for the user.

Loss Aversion: The user is punished or losses resources if they fail to do some task or if they do some mistake or an action against some rules.

Lottery: The user completes some task or gets a reward solely by chance.

<u>Ownership:</u> The user can become attached with their character/pets or creations inside a game. This is particularly the case when the game allows a high level of customization.

<u>Points:</u> Points are a numerical value given for any action or combination of actions completed by the user.

<u>Progression:</u> The progress of the user can be displayed and measured through the process of completing itemized tasks.

<u>Quests:</u> A quest is a specific obstacle, challenge or task that the user has to complete to receive a reward. A quest can be linked with another follow-up quest, forming quest-chains.

1.3.   Tips for applying Gamification

- Create benchmarks or indicators to measure performance
- Provide meaningful rewards and recognition for accomplishment
- Keep it fun! The point of a game is to inspire, not force behaviour.
- Clearly define your goals or objectives. Focus on the goal that you want to achieve through the Gamification process (e.g. teach something), and try to identify potential distractions and eliminate them.
- Try to provide instant feedback to ensure learning and improvement.
- Most of the time quick and small rewards are most effective at driving the user's behaviour than one big albeit rare reward.
- Allowing the users to communicate and also view the progress and achievements of each other can lead to creating a strong community and engage the users even more.
- Social media offer great opportunities to apply Gamification, either on their own or as a supporting tool for other systems.

2. Learning

2.1.   Types Of Learning

The terms formal and informal learning have nothing to do with the formality of the learning, but rather with the direction of who controls the learning objectives and goals. In a formal learning environment, the training or learning department sets the goal and objectives, while informal learning means the learner sets the goal and objectives [1].

Formal education – It is organized, guided by a formal curriculum and can lead to a formally recognized credential such as a diploma or a degree, and is often guided and recognized by government. Teachers are usually trained as professionals.

Non-formal learning – This is also organized, albeit in a more loose manner and it may not be guided by a formal curriculum. This type of education may be led by a qualified teacher or by a leader with more experience. It does not result in a formal degree or diploma, but it can improve the skills and capacities of the trainees. Continuing education courses are an example for adults. Usually the learners are more engaged and interested, since they willingly participate with the aim to learn or gain skills and not just to receive a degree.

Informal learning – There is not a formal curriculum and no credits of any kind earned. The teacher can be anyone with more experience such as a family member or friend.

Two other terms worth mentioning are incidental and intentional learning, refering to the intent of the learning objectives.

An **intentional learning** environment has a self-directed purpose in that it has goals and objectives on what and/or how to learn [2].

Incidental learning occurs when the learner picks up on something else in the learning environment, such as the action of a model that causes him or her to loose focus on the learning objectives or goal and focus on an unplanned learning objective [2].

Thus, formal learning is normally always intentional. Informal learning is intentional if the learner sets an objective or goal for herself and incidental if the learning occurs haphazardly or serendipity.

Incidental learning is often dismissed by instructors, however it can be an important concept because it often has a motivating effect with the learners that lead to discovery learning. So unless other considerations prevent it, it can sometimes be worthwhile to detour from the primary objectives to take advantage of an unplanned teachable moment.

2.2.   Types of Learning Styles

One of the most common and widely-used categorizations of the various types of learning styles is Fleming's [3] VARK model (sometimes VAK) which expanded upon earlier Neuro-linguistic programming (VARK) models:

- visual learners
- auditory learners
- reading-writing preference learners
- kinesthetic learners or tactile learners

Fleming [3] claimed that visual learners have a preference for seeing (think in pictures; visual aids such as overhead slides, diagrams, handouts, etc.). Auditory learners best learn through listening (lectures, discussions, tapes, etc.). Tactile/kinesthetic learners prefer to learn via experience—moving, touching, and doing (active exploration of the world; science projects; experiments, etc.). Its use in pedagogy allows teachers to prepare classes that address each of these areas. Students can also use the model to identify their preferred learning style and maximize their educational experience by focusing on what benefits them the most.

Auditory Learners: Hear

Auditory learners would rather listen to things being explained than read about them. Reciting information out loud and having music in the background may be a common study method. Other noises may become a distraction resulting in a need for a relatively quiet place.

Visual Learners: See

Visual learners learn best by looking at graphics, watching a demonstration, or reading. For them, it's easy to look at charts and graphs, but they may have difficulty focusing while listening to an explanation.

Kinesthetic Learners: Touch

Kinesthetic learners process information best through a "hands-on" experience. Actually doing an activity can be the easiest way for them to learn. Sitting still while studying may be difficult, but writing things down makes it easier to understand.

3. 3D Worlds Characteristics Overview (Open Simulator)

3.1.   Introduction

OpenSimulator is an open source 3D application server that can be used to create a Virtual Environment, accessible through a variety of client software (3D Viewers). 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 contribute 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.

3.2.   Interacting with 3D Objects - Events and Actions

In general, objects in OpenSimulator have different states. While in a state, the object waits (or listens) for specific events to occur (e.g. someone touches or approaches the object) that trigger it, it executes some actions (e.g rotate or display a label above) or changes state. As an example a lamp 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 changes 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".

Some events that an object can wait to happen:

- User touches (clicks) an object.

- User wears or equips/attaches an object

- User takes a copy of an object

- User approaches an object.

- User collides with an object

- User selects an option in a dialogue menu or types some text in a dialogue text box

- User writes some message in a particular chat channel

- wait for an object to be attached to an avatar

- Event repeats every X seconds (timer).

Some actions that an object can execute after some event:

- Object changes position, size, color, opacity, texture, rotates etc.

- The texture assigned to the object moves/animates (e.g. running water)

- Object "says" some text in chat or plays a sound

- Label/text is displayed on top of an object.

- make a plane of the object glow

- play a sound file (up to 10 seconds) stored in its inventory

- Create a dialogue menu for a specific avatar to provide an answer to a question.

- play a sound file (up to 10 seconds) stored in its inventory

- Object stops/sleeps X seconds between two actions.

- User avatar is teleported to another position in the region or in another region.

- The walking speed of the avatar is reduced/increased.

- An object is given to the avatar (becomes available in his inventory)

Examples:

- A door is initially closed and opens (<u>action</u>: rotate) when the user clicks (<u>event</u>: touch) on it (the door can be locked so user needs to first make some action e.g type a password).
- The user is asked to select a color in a dialogue menu (<u>event</u>: option in dialogue menu) and the object is then painted with it (<u>action</u>: change color).
- An elevator moves to another floor (<u>action</u>: move to target) when the user clicks (<u>event</u>: touch) a button.
- On a presentation screen, the next slide is loaded (<u>action</u>: change texture) when user clicks Next button (<u>event</u>: touch)
- On a presentation screen, the next slide is loaded automatically every 10 seconds.
- When the user approaches an object (or area) he receives a text message.

3.3.   Messaging System

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 or even subparts of a complicated object communicating to coordinate the required behavior.

As an example, a complex device in the 3D World could consist of various button subparts and a lamp. When one of the buttons is clicked, then it sends a specific message in the channel on which the lamp sub-part is listening on. The lamp, when receiving a message in that channel, could turn on/off or change colour, depending on the message received (button clicked).

Combining this feature with the event/action mechanisms mentioned in the previous chapter, allows a skilful scripter to implement very advanced objects and behaviours in the 3D World.

3.4.   Particles

Another feature that can be helpful for various task are the particle scripts. Using scripts an object may emit small particles. The script is parameterized allowing to change the shape, texture, number of the particles, the direction they move and the duration before they disappear. Some of the effects that can be simulated with particles are: smoke, laser beam, emission of electrons, butterflies, snowflakes, rain, falling leaves.

Particles are a great way to direct students to a specific location and in many cases they are ideal for simulating some complex operations and physical phenomena, such as the operation of a laser, the emission of electrons etc.
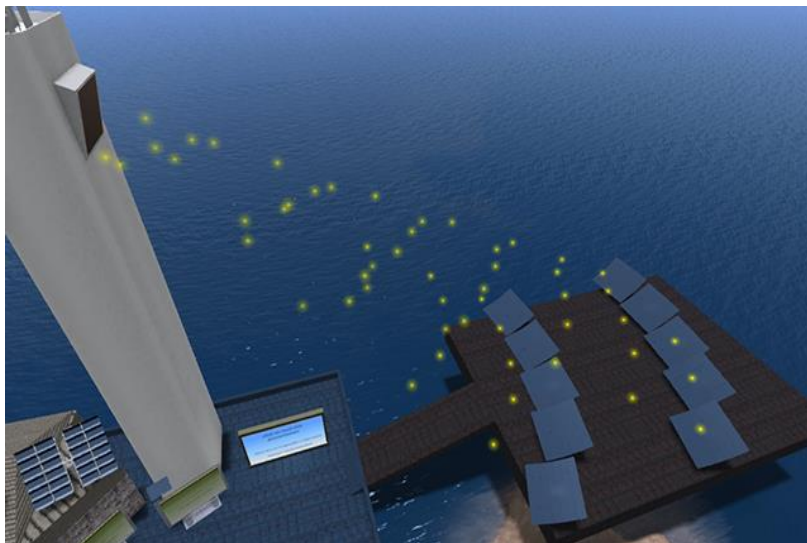

Figure 1: Particles used to visualize the direction of light

3.5.   NPC Characters

NPC characters are avatars controlled exclusively by scripts, unlike the avatars of actual users connected in the Virtual World. NPCs are crucial for self learning in

Virtual Worlds, since the teachers themselves may not be present to guide students and provide context about their objectives. NPC characters can simulate the role of an actual teacher but require a great amount of scripting to program their behaviour inside the world.

NPCs are created and controlled with the use of specific script commands that can be combined with the ideas described in the previous chapters.

Some actions that can be used to control NPC characters:

- 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 saying a message (text) in nearby chat that appears on screen.
- NPC randomly wanders around a point.
- NPC follows the user-avatar.

Some events that could trigger these actions:

NPC executes one or more of the above actions after the user...
- touches an item
- writes something to chat in a particular CHANNEL   e.g. "/5 hello"
- selects an option from a dialogue box
- writes specific text in a dialogue text box
- approaches (or collides) some object
- any other event described above.

3.6.   Heads Up Displays

In OpenSimulator, HUDs are objects that can be attached on an avatar in specific slots reserved 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.

With this technique it is possible to have custom objects, textures and text appearing on a specific part of the screen. The user receives a HUD object in his inventory. He wears it (double click for wear, or right click -> Attach Hud -> Preferred Area) and the HUD display appears on his screen. The HUD is very useful for storing data about the user's interactions in the game and providing relevant feedback.

3.7.   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.
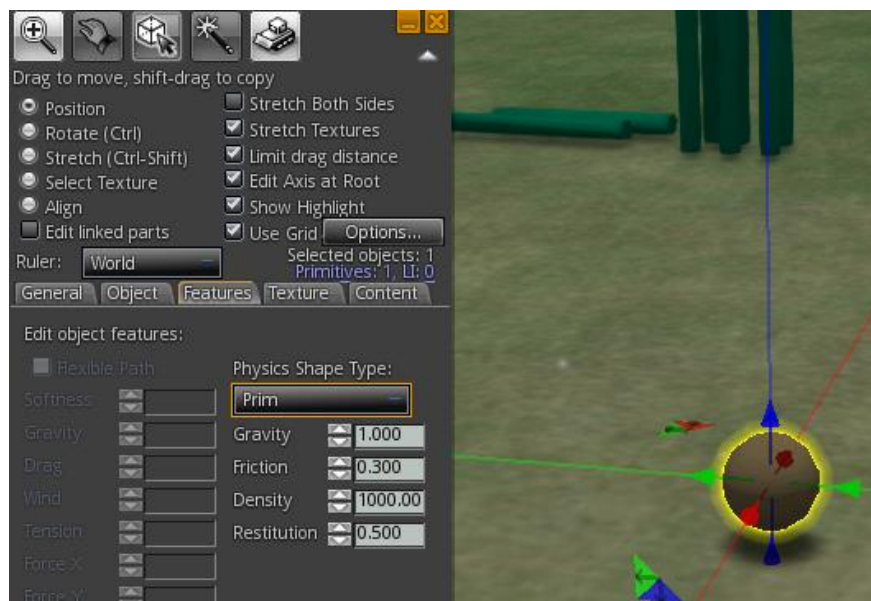
Figure 2: Making a 3D Object "physical"

3.8.   Advanced Scripts

The community of Second Life and OpenSimulator is very active and one can find very advanced, complicated scripts. Some notable examples are the "vehicle script", that can be configured and used to turn a complex object to a vehicle that an avatar can enter and control and the "gun script" that can turn an object to a gun that can be equipped by an avatar and generate custom "bullet" objects that are at a specified speed.

3.9.   Media on A Prim

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, since it can be used to project existing learning material (e.g. text or presentations) inside the Virtual World, so they do not need to create 3D objects.
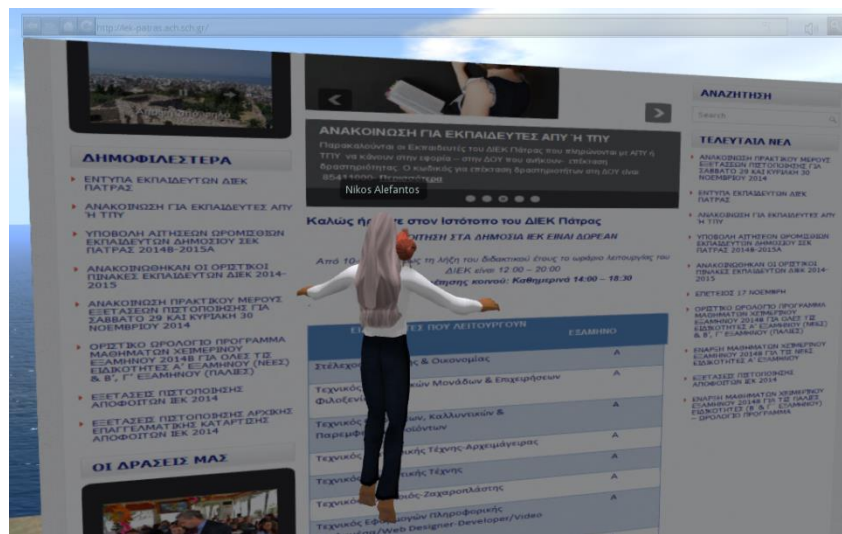
Figure 3: Projecting a website inside the 3D World

3.10.   Communication with other Users

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 in OpenSImulator 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).

4. Opportunities for Learning and Gamification

4.1.   Synchronous Learning

The easier type of learning scenarios to carry out in a Virtual World are probably those that involve the trainer to be logged in the virtual world with their avatar, along with their students. OpenSimulator allows using a microphone to communicate with voice, so it is possible to have actual learning sessions inside the Virtual World. The trainer and the students would log in the virtual world at a predetermined time and enter a virtual classroom or auditorium. The trainer can project a presentation on a 3D object's surface and control it, while using the microphone to speak to the students.

Figure 4: Student avatars in a 3D Auditorium

Alternatively if there are various 3D objects around the World that cover the learning topics of the course, the trainer can have the role of a guide. The students follow the trainer's avatar and listen to him/her explaining the theory behind the 3D objects they see.

4.2.   Asynchronous Learning

The 3D World can be used for Asynchronous Learning as well, preferably as a supporting tool. The trainer can give the theoretical background of a topic inside the classroom using traditional means and the students can visit the 3D World on their own to explore and view additional learning material or to complete a homework instructed by the trainer. Alternatively, as preparation before a lecture, the trainer asks the students to visit the 3D World on their own and then they meet in classroom and discuss on the material they observed. Obviously this use of the 3D World requires more planning for the trainer, who will need to clearly describe to the students what they should (or should not) do inside the 3D World.

The actions that the students have to do in the 3D World can vary from exploring and viewing static learning material (e.g. text presentations or simple 3D objects), to complex learning scenarios and activities (e.g. solving interactive puzzles, communicating with characters controlled by scripts).

With sufficient planning, knowledge and programming skills, a trainer could even create a 3D World that can be used on its own for self-learning purposes. This can include bots that communicate and guide the students towards the learning material, self-assessment exercises and activities, complex puzzles that test their understanding, games etc.

In the following chapters we cover various activities, learning scenarios and gamification ideas that can be implemented inside a 3D World.

4.3.   Using NPC characters

Non Playable Characters (NPCs) are a very useful feature in Virtual Worlds and an excellent way to provide feedback to users, especially in unsupervised learning.

4.3.1. NPCs as Information Providers

The easier application would be to have an NPC character stay stationary in some position and provide specific information messages (or play sounds) when a user approaches him. The messages could be the content of the course itself, or instructions that the user must follow to find or gain access to the content delivered through another medium.

Figure 5: An NPC character, using a Dialog Box to ask a question

4.3.2. NPCs as Guides

An NPC character can be used to guide the user to a specific place or a series of places (route).

Example scenario: The NPC appears when the user-avatar approaches an area or clicks some object (e.g. a closed Door). The NPC avatar waves and starts 'talking' with text messages on screen. The NPC character asks the user to follow him and starts walking a predetermined path around the region. The NPC character stops in various positions and points to something, speaking all the while with messages and after a while continues walking to another position. The NPC character stops and asks the user some question. A dialogue box with possible answers appears and the user selects one. Alternatively a text box appears and the user types his answer. Alternatively the user clicks a particular object. When the user gives the correct answer or does the necessary actions the NPC avatar continues (e.g. moves to the next position or gives a reward to the user).

An example of a simple NPC character acting as a guide:
https://www.youtube.com/watch?v=3zHfMg74AGA

4.3.3. NPCs as Quest Givers

NPC characters can be used to present the "quests" or problems that the user has to solve. Usually the NPC character would start by describing the problem and why it is important to be solved. Then depending on the difficulty and type of the game, the NCP can provide the exact steps that are required to solve the problem, or some general tips or even let the users find a solution by themselves. Optionally the NPC can mention specific rewards that the user will gain when they complete the quest.

Example scenario: The NPC remains in a position waiting. When a user passes by, the NPC approaches him and speaks to him through messages. He explains that he has lost a page of his book and asks the user to help him find it. The user/student looks around and finally discovers a page item next to a desk. When the user clicks on the page the NPC approaches him and thanks him for helping him. The page

disappears for a while (the textures' opacity is modified and a timer is set so it can be restored after some time, so another student can also find it).

4.4.   Achievement mechanisms

Monitoring the student's activities inside the virtual world and rewarding their achievements is a useful and effective way to attract their attention and motivate them to participate in the activities prepared in the virtual world, especially for unsupervised learning, since the teachers cannot be there themselves to monitor the student's actions.

A technique that can be used to implement achievement mechanisms in a Virtual World is through Head-Up Displays (HUDs). In OpenSimulator, HUDs are objects that can be attached on an avatar in specific slots reserved 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.

With this technique it is possible to have custom objects, textures and text appearing on a specific part of the screen. The user receives a HUD object in his inventory. He wears it (double click for wear, or right click -> Attach Hud -> Preferred Area) and the HUD display appears on his screen. The HUD is very useful for storing data about the user's interactions in the game and providing relevant feedback.

Some examples are presented in detail next:
- Collection of Tokens
- Collection of Puzzle Pieces
- Monitoring time spent on an area

● Using objects to solve riddles

4.4.1. Token Collection

The user wears an item that displays a small window on his screen with a score (Tokens collected and Prizes earned).
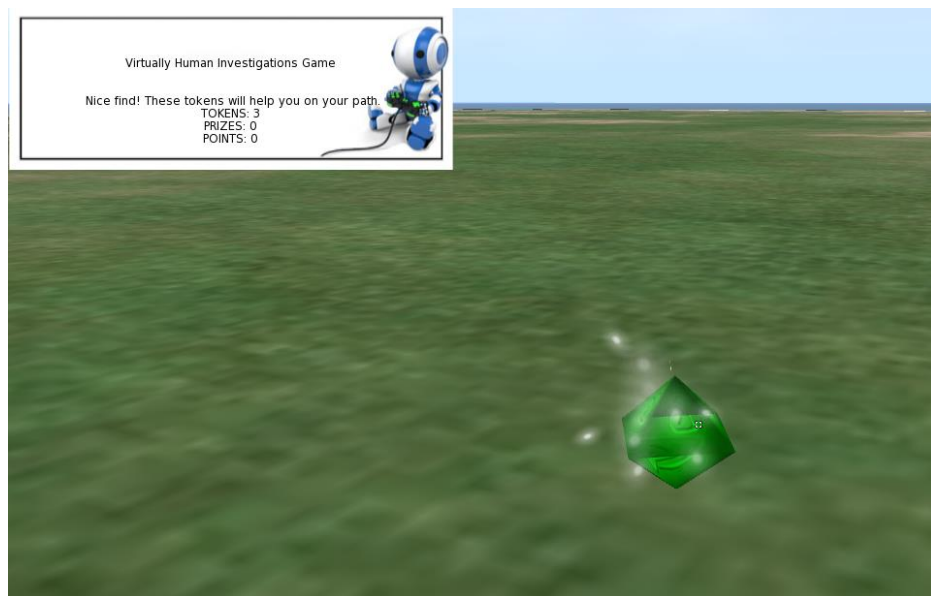

Figure 6: A HUD object displaying the number of tokens / points / prizes collected by the user

While wearing this item he gains token points when he clicks (collects) a specified item or does some specific action. Some objects may award more points than others.

Also while wearing this item he may lose token points if he triggers some traps (approaching or touching a specified object).

Some tokens may only be awarded only if a condition is met, e.g the user wears or has equipped a particular object/tool or if he has previously earned a specific reward. As an example the user clicks on a broken bottle and loses points because he was

cut. If the user has previously equipped gloves then he clicks in the broken bottle and is awarded tokens.

In general any event mentioned previously may trigger as one of the executed actions to increase or decrease the token score of the user.

4.4.2. Puzzle Pieces / Pieces Of Map

The user wears the HUD item and then proceeds to explore the world and solve some quizzes/quests. When he successfully completes a quest, he is awarded with a puzzle/map piece which appears on screen. When all quests are completed the complete Map is displayed on his screen showing the location of a hidden chamber/treasure.

Figure 7: Various states of a HUD object representing a map

4.4.3. Monitoring Time Spent in Areas

Another mechanism that can be used to monitor and reward students is adding a timer listener in the HUD item. The timer can be set to be triggered every minute and check the current coordinates of the user that wears it. If the user's coordinates are inside a specific area we are interested we increment a counter. After five subsequent triggers and consequent increments, the student is rewarded for staying in that area for 5 minutes.

In a similar manner time listeners can be used to count the time passed between two specific actions of the user. This allows offering special rewards if the user completes a task quickly or to make a task more challenging by applying a time limit in which the user should complete it.

Using this mechanism in a suitable manner could also allow trainers to extract useful information and knowledge regarding the progress of their students, their engagement, work rate, and other assessment indexes.

4.4.4. Using Objects to solve Problems

It is possible to create simple quests for students to solve. For example a candle sits on the desk and the student has been instructed that he needs to light it. Clicking on the candle has initially no effect. Nearby the user spots a matchbox. Clicking on the matchbox, the user receives a match that is automatically assigned on his right hand. By clicking on the candle again, this time the candle is lit and the match disappears from his hand. The user is rewarded a point that is displayed on his HUD item.

4.5.   Mini games

Mini games can be simple activities inside the 3D world that could either be irrelevant to the domain, and used just for entertainment purposes and to keep the students' interest, or - if planned correctly by the trainer - directly or indirectly teach concepts of the course.

Using the script ideas we presented in the previous examples one can create really engaging and sophisticated mini games. At this point we mention two ideas based on scripts that can be found online and be adapted.

Vehicles:
Vehicle scripts are very popular in OpenSim and Secondlife. They allow to turn an object like a car, a boat or anything else to a vehicle that a user can sit on and drive around the World. Driving a car inside the Virtual World is very exciting for students and could for example be the way for students to travel from one region (e.g a specific chapter of the course) to another one. A good idea would be to offer the vehicle as a reward for students that perform well, to keep the students motivated.
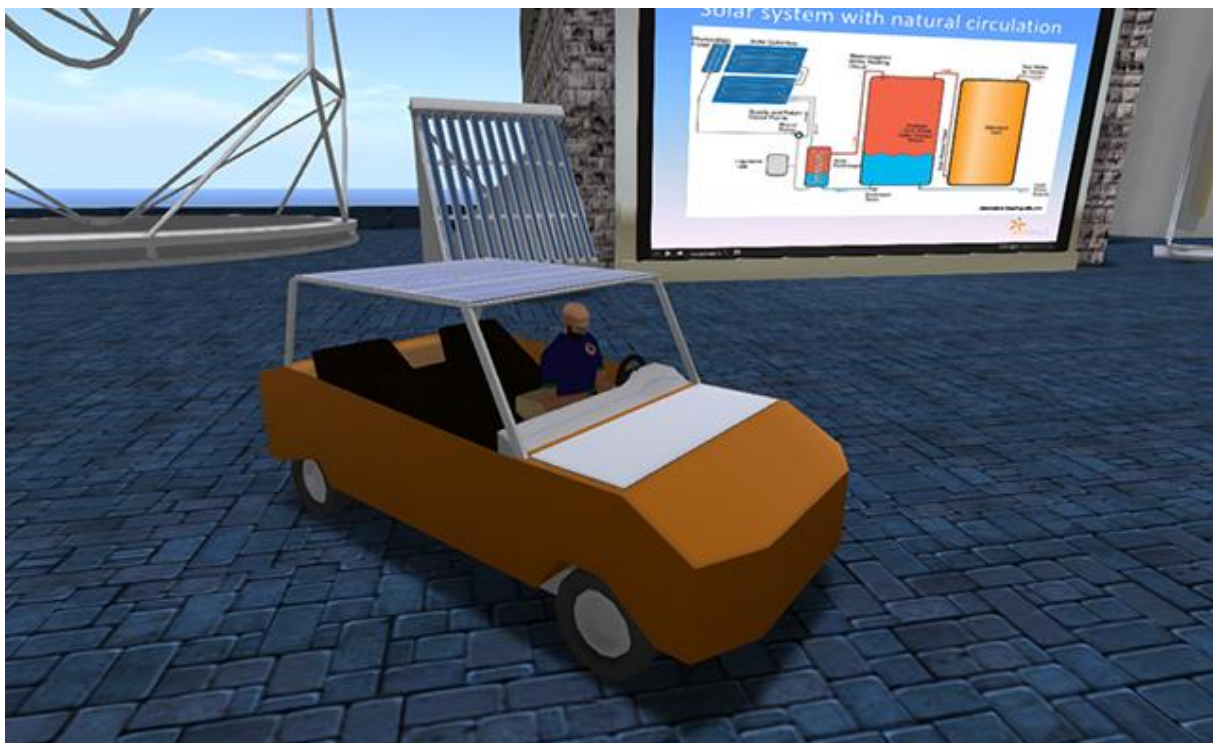
Figure 8: A solar powered car controlled by an avatar

<u>Guns:</u>

As indicated by the name, these scripts are mostly used for simulating the functionality of guns. When the user "wears" the gun attachment, a HUD is displayed on his screen with the weapon. When the user clicks, a small object is created and is thrown towards the direction the user is facing.

Figure 9: A 2D texture with transparency used on an HUD object to resemble a 3D gun

The script however can be modified at will, so it is not very difficult to use it to throw other items, like basketballs and adapt the speed and direction accordingly.

It is also possible to create Mini Games without adding any script, by taking advantage of a Physics Engine. In OpenSim an object can be set as 'physical' so gravity will affect it. The object will fall to the ground if placed in the air, slide down a slope or move when the user or another item collides with it. The user can also click and drag this item to push/pull it towards a direction.

Bowling Game:

It is relatively easy to set up a bowling game. The pins and the bowling ball are set as physicals objects. The pins are initially placed vertically so they stay up. The user drags the ball towards the direction of the placed pins. When the ball collides with the pins, the pins affected will drop.

4.6.   Self-assessment opportunities

Self-assessment opportunities are very important, especially in unsupervised learning. Virtual worlds like OpenSim offer various opportunities to collect feedback from users and respond to their answers.

If the trainer feels uncomfortable with writing custom scripts in OpenSim, there is always the option of projecting a web page with exercises on the surface of a prim (Media On a Prim).

For example, in the following figure a multiple choice quiz, created with HotPotatoes (a popular software for generating web pages with multiple choice and other types of exercises) is projected inside the 3D World on a surface of an object. The user can view and interact with the page and assess his knowledge of the domain that is taught.
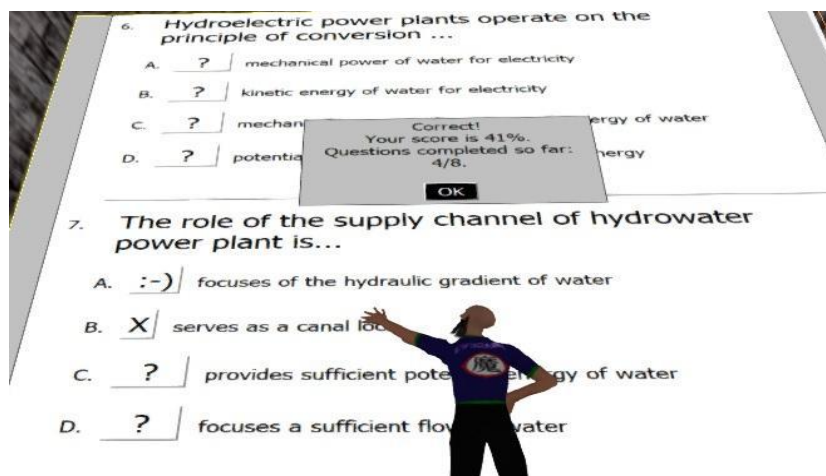

Figure 10: Multiple-choice exercises

For more confident and creative trainers, the LSL scripting language of OpenSim provides endless opportunities:

Dialogue menus are the most suitable option in most cases. When the user clicks on an object (or after any other event), a question with up to 12 answer options appears

on screen and the student is asked to click the correct one. This, will send his answer as a message in a previously specified channel. Using channel listeners we can receive his answer and provide feedback accordingly. Feedback can be any action, such as saying something, playing a sound, offering a follow up question, give a specific object as a reward, teleport to another secret location etc.

An alternative way to receive feedback is through a dialogue text-box. A question is given to the student but this time we ask them to write their answer in a provided text box. Again their answer is sent at a specific channel that we can check and respond with any actions we want.

Using these simple techniques we can create various types of assessment opportunities. It is possible for example to create a Trivia Game like the one displayed in the figure. The machine has a glass covering the face of a historical figure so it is not visible. By clicking on the question mark button, a dialogue menu with information about the figure appears and the user must answer who the person described is from a couple of possible answers. If the correct answer is provided, the glass rotates and reveals the person behind, a lamp glows with green colour and cheerful sound is played. After a couple of seconds the glass returns to the original position and another figure is loaded in the machine. This can also be combined with the HUD functionality described above, so the user is rewarded after each correct answer, collecting points displayed on his screen.

Figure 11: Trivia Machine

4.7.   Constructivism

Constructivism approaches can achieve significant results in training and are ideal for specific domains. Fortunately, it is not difficult to apply this in an OpenSim Virtual World. Actually one needs only to set a specific area of the world (or even an individual region) as a sandbox area where students are free to create their own content. The trainer will have to provide some initial instructions about creating and manipulating basic prims, then give them specific assignments (as individuals or groups) to design and finally allow students to test their knowledge while unleashing their creativity and imagination.

This strategy is a common idea in Gamification and it relies in the sense of productivity, accomplishment and ownership. Users tend to get attached with their creations and are eager to share them with their friends.

A good scenario would be to offer some general guidelines and allow the users to create their own interpretations allowing them to compete for a reward.

4.8.   Collaborative learning

Creating collaborative activities in the virtual world is possible using the different ideas presented above, but they usually require more complex planning. To ensure that users will cooperate, it is usually helpful to put specific constraints (e.g. time limit) that make specific objectives or challenges impossible to overcome unless the students cooperate and coordinate their actions in a specific pattern.

References

1. Cofer, D. (2000). Informal Workplace Learning. Practice Application Brief. NO 10. U.S. Department of Education: Clearinghouse on Adult, Career, and Vocational Education.

2. Good, T., Brophy, J. (1990). Educational Psychology: A realistic approach. New York: Holt, Rinehart, & Winston.

3. Fleming ND. The Active Learning Site_VARK
    Available at: http://www.active-learning-site.com/vark.htm. Accessed: October 10 2000.

Bibliography

1. Hoven, D. (2007). The affordances of technology for student teachers to shape their teacher education experience.

2. Dr Petrea Redmond (2011). From face-to-face teaching to online teaching: Pedagogical transitions.

3. https://en.wikipedia.org/wiki/Virtual_world

4. https://en.wikipedia.org/wiki/Experiential_learning

5. Kolb, D. A. (2014). *Experiential learning: Experience as the source of learning and development*. FT press.

6. http://work.chron.com/role-teachers-education-8807.html

7. Gratch J, Rickel J, André E, Badler N, Cassell J, Petajan E (2002) Creating Interactive Virtual Humans: Some Assembly Required. IEEE Intelligent Systems, July/August, 54-63

8. Ma, T., Xiao, X., Wee, W., Han, C. Y., & Zhou, X. (2014).A 3D Virtual Learning System for STEM Education. In Virtual, Augmented and Mixed Reality. Applications of Virtual and Augmented Reality (pp. 63-72). Springer International Publishing.

9. SLOODLE https://www.sloodle.org/

10. Dr. Sarah Elaine Eaton (2010). Formal, non-formal and informal learning: The case of literacy and language learning in Canada.

11. Vanessa Marcy (2001).- Adult Learning Styles: How the VARK© Learning Style Inventory Can Be Used to Improve Student Learning