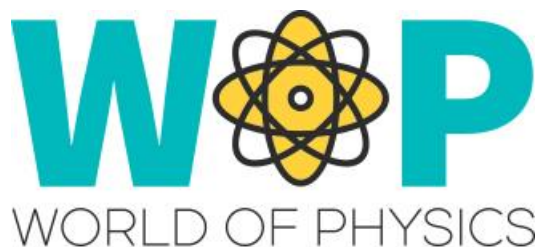




Erasmus+

Project funded by: Erasmus+ / Key Action 2 - Cooperation for innovation and the exchange of good practices, Strategic Partnerships for school education (European Commission, EACEA)



TECHNICAL GUIDE

Creating NPC Characters

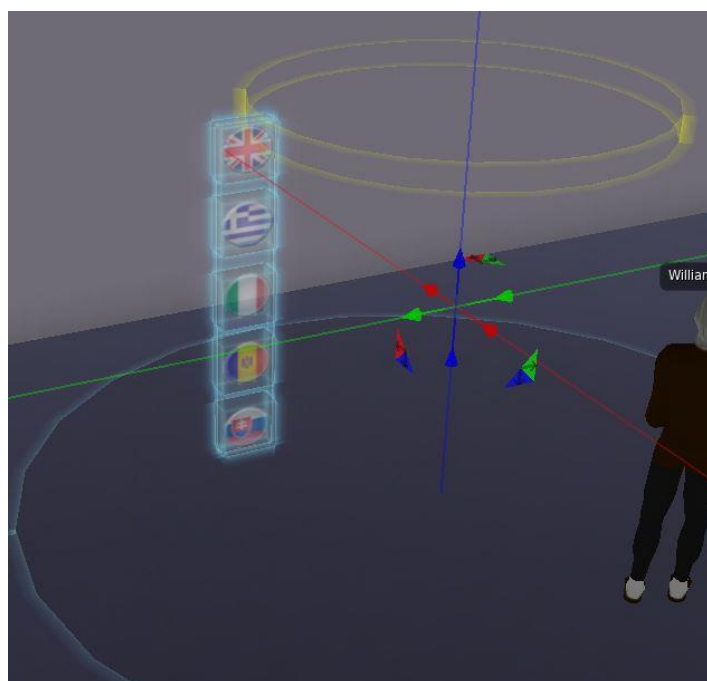
1. Introduction

NPC characters are a great way to give information to the student. They can also be used as a guide to help them navigate the areas, combined with questions to get feedback and assess them or be integrated as part of a wider learning scenario.

For the dialogue with the NPC, you can write a list of possible questions that the student will be able to ask them and what will be the answer of the NPC. For each answer you also specify what are the follow up questions the student can make.

Alternatively the NPC character can make the questions and the student will have to give the correct answer from the available ones. You can combine the dialogue of the NPC with various actions such as, walking somewhere, performing an animation, giving an object etc.

In our implementation we have buttons to change the language that the NPC will speak in. Alternatively you can have the NPC initially asking the student what language they want to use (which is more realistic).



2. Technical Details

In our approach we need buttons to be able to change the language so we use a linked set. The main NPC control script will be placed in the ROOT prim and the button scripts in the child prims to be able to change the language.

Also put in the ROOT prim the necessary files (sounds , appearance, animations etc) that you want. The appearance file is necessary to give a body to the NPC. You can use a script with the function 'osOwnerSaveAppearance' to save the appearance of an avatar as a file.

The NPC control script should generate the NPC when it loads (state_entry event) using the function: 'osNpcCreate'. It waits for avatars to approach it (sensor event) and engages them using 'llDialog' to prompt them for replies. This a similar approach as the one used in the implementation for Quizzes.

A good idea is to keep some list with the avatars that the NPC has approached in the past (known avatars) and a list with the avatars that the NPC has approached recently (you can use a timer event to regularly empty this list). You can use the first list to have different dialogue towards characters that have already met the NPC (for example the NPC will not introduce themselves every time). The second list can be used to avoid engaging the student multiple times and becoming annoying for them. A 'timer' event will clear thr list when enough time has passed to engage an avatar again.

The script can also control the NPC character by applying animations with 'osNpcPlayAnimation' or moving them with 'osNpcMoveToTarget' and can also play sounds with 'llPlaySound' or give items to the user with 'llGiveInventory'.

The scripts for the buttons just use a 'touch_start' event and use 'llMessageLinked' to send messages to the root.

You can find the scripts we use and more instructions in the "Scripts Section" here:
<http://aigroup.ceid.upatras.gr/wop-oer/scripts.html>

3. References/Links

<http://aigroup.ceid.upatras.gr/wop-oer/scripts.html>

http://wiki.secondlife.com/wiki/LSL_Portal

http://wiki.secondlife.com/wiki/Touch_start

<http://wiki.secondlife.com/wiki/LIMessageLinked>

http://wiki.secondlife.com/wiki/Link_message

<http://wiki.secondlife.com/wiki/LISetPrimMediaParams>

<http://wiki.secondlife.com/wiki/LIDialog>

<http://opensimulator.org/wiki/OSSLNPC>

http://wiki.secondlife.com/wiki/State_entry

<http://wiki.secondlife.com/wiki/Sensor>