Erasmus+
2020-1-UK01-KA201-079177

VRACE

9. NPC Characters

INTRODUCTION

- NPC characters are avatars that are controlled by scripts. They can be used to guide the users, give them information and for other applications. Through the scripts you can have the NPC characters moving around, performing animation, interacting with objects or communicating with other avatars.

APPEARANCE

- ✔ Before creating an NPC character you need to decide on their appearance. You can configure your own's avatar appearance (including clothes and attachments) and then use the LSL function **osOwnerSaveAppearance** or **osAgentSaveAppearance** , to save it as a notecard file.

```
osOwnerSaveAppearance(string notecard):key
```

APPEARANCE

- Put the notecard with the appearance in an object along with some script so it can use it. The script will contain all the code for controlling the behaviour of the NPC character.
- You can use any of the events described in previous chapters, to configure when the NPC character will act.
- There are some specific LSL commands that refer to NPC characters. We present some of them here:

GENERATION

✔ **Generating the character:** You can select the name of the avatar and the position where it will appear in the world. If the function is successful, it returns an id that you should keep in a global variable, so you can use with the functions that control this character.

```
osNpcCreate(string firstname, string lastname, vector
position, string cloneFrom):key
```

MOVING

- ✔ **Having a character moving to another position:** You specify the id of the NPC that you want to move and the target position that you want it to move towards. The NPC will walk towards that position.

```
osNpcMoveToTarget(key npc, vector target, int options):void
```

ANIMATION

- ✔ **Having a character starting and stoping an animation:** You specify the id of the NPC that you want to animate and the name of the animation to perform. After starting an animation with osNpcPlayAnimation, you can use llSleep to wait for some seconds before using osNpcStopAnimation to stop it.

```
osNpcPlayAnimation(key npc, string animation):void

osNpcStopAnimation(key npc, string animation):void
```

COMMUNICATION

- ✔ **Having a character communicating with messages:** You specify the id of the NPC that you want to send a chat message and the text of the message.

```
osNpcSay(key npc, string message):void
```