Erasmus+
2020-1-UK01-KA201-079177

10. HUD Elements, Particles, Projectiles

HUD ELEMENTS

- ✔ HUD (Head-up Display) elements are graphics that appear on specific parts of the user's screen and remain there while the user navigates the world.
- ✔ In Opensimulator you can assign a prim or linked set as a HUD element by simply finding it in your inventory and selecting to wear it (double click for wear, or right click -> Attach Hud -> Preferred Area).
- ✔ You can then edit the object's size, orientation and exact position on the screen and these settings will be saved, so next time you just need to double click it on your inventory and it will appear on that specific position.

HUD ELEMENTS

- ✔ The HUD object can have various parts and you can use scripts to implement buttons and other elements on it.
- ✔ The HUD object may be something general that remains with the user all the time, or it may be used for a specific activity only.
- ✔ It is a great way to implement your own dialogue menu for the users, instead of using the llDialogue function. This way you can adjust the exact way that the messages and the buttons will look like.
- ✔ The HUD is also very useful for storing data about the user's interaction in the game and providing relevant information.

ATTACH EVENT

- A useful event when using a HUD element is the “attach” event, that is triggered when an avatar wears the HUD object. You can use this event to store the wearer’s Id or Name.

```
attach( key id ){ ; }
```

HUD EXAMPLES

- ✔ Following are two examples, with activities that a HUD element can be used for:

HUD EXAMPLE 1

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

HUD EXAMPLE 1

- ✔ 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.

HUD EXAMPLE 2

- ✔ 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.

PROJECTILES

- ✔ You can use scripts to implement objects that dynamically generate 3D objects. This allows interesting functionality like throwing projectiles, taking advantage of the Physic's Engine.
- ✔ The corresponding LSL command is llRezObject that generates an object at a specified position with an initial velocity.

```
llRezObject( string inventory, vector pos, vector vel,
rotation rot, integer param );
```

PARTICLES

- ✔ You can use scripts to generate particles.
- ✔ These are images that are emitted from the object in a specific pattern and you can use them to produce effects like smoke, falling leaves, laser beams e.t.c.
- ✔ The LSL function for particles is “llParticleSystem”, that can be customized using a list of values.

```
llParticleSystem( list rules );
```