

# SCRIPTING

## ALGEMEEN

- Lua & LuaBind, zie Wiki Scripting pagina en Lua demo project.

## A.I.

Het voorgestelde A.I. systeem is gebaseerd op dat van Ragnarok Online (ook in Lua). Een entiteit die bestuurd wordt door de AI, is een Finite State Machine. Van elke mogelijke staat wordt vooraf de semantiek afgesproken (bv. IDLE = Hovercraft voert geen bewegingen uit, ...).

In de C++ code spreken we een aantal interfaces af die vanuit Lua aangeroepen kunnen worden. Dit zijn de basisbewegingen, en op zijn minst komen deze overeen met de controls waar een (menselijke) gebruiker toegang tot heeft (vooruit, links, rechts, ...).

```
luabind::def("turnLeft", turnLeft);
```

Om van uit het script toegang te hebben tot de variabelen in de code zijn er twee mogelijkheden: de algemene manier, zoals in Ragnarok: *GetV(V\_MOTION, MyID)* waarbij *V\_MOTION* een constante integer is, die ergens in Lua gedefinieerd is. En in de gamecode verwijst dezelfde integer *hopelijk* naar de "motion" variabele). De andere mogelijkheid is effectief getters voorzien voor elke variabele waar we van buitenaf aan willen kunnen, en deze dan allemaal binden aan het script. Ikzelf ben eerder geneigd om de eerste oplossing te kiezen ☺.

Het Lua script zelf dan bevat een main/process/decide functie, die afhankelijk van de huidige staat een andere functie oproept – die dan op zijn beurt een actie uitvoert en de staat eventueel verandert. Deze functie wordt dan om de zoveel tijd (hoe vaak is een beslissing van de AI nodig?) opgeroepen vanuit de C++ code.

## MOOD / MUZIEK

Een tweede idee is het gebruiken van Lua om de beslissingen over oa. de sfeer te scripten: als ik X dicht bij mijn voorganger ben, verander de sfeer naar "SPANNEND", en deze verandering van sfeer zorgt ervoor dat de muziek verandert. De reden van dit niet als core game/engine functionaliteit te schrijven is het mogelijk maken om verschillende triggers te maken voor de sfeer: afhankelijk van de positie in de race, na een bepaalde event, enz.