

Zoek met behulp van Google de volgende zaken op:

Een kort voorbeeld van het gebruik van een lijngrafiek in *Matplotlib*

Bestudeer dit voorbeeld regel voor regel.

Experimenteer met dit voorbeeld door kleine dingen te veranderen en te kijken wat het effect is.

Een kort voorbeeld van het gebruik van de *JSON* module die standaard bij Python zit

Bestudeer dit voorbeeld weer regel voor regel.

Experimenteer ook met dit voorbeeld door kleine dingen te veranderen en te kijken wat het effect is.

Opgaven bij de huidige les:

Laat in een programma *race* een aantal auto's van klasse *Car* het tegen elkaar opnemen met een merkegebonden constante versnelling en maximale snelheid. Na elke *deltaT* drukken ze hun merk, versnelling, snelheid en positie af, een plaatje hoeft niet. Gebruik geen timers en haal de tijd maar 1x per lusdoorgang op! Het programma stopt als de eerste auto de volledige baanlengte heeft afgelegd.

Bereken in een zelf geprogrammeerde eindeloze simulatie-lus steeds de snelheid en de positie door middel van numerieke integratie (Google op keyphrases *Python*, *current time* en *StackOverflow*).

Bouw in die lus ook een wachttijd in van *0.05 s* (Google op keywords *Python*, *sleep* en *StackOverflow*). De eindeloze lus zit in een klasse (*race*-)*Track*. Een *Track* bevat een referentie naar een aantal *Car*'s. Elke *Car* bevat een referentie naar de *Track*. Elke *Car* berekent zelf z'n snelheid en positie. De benodigde *deltaT* wordt berekend door de *Track*.

Gebruik de *JSON* module om alle tijdstippen met daarbij steeds merk, versnelling, snelheid en positie van een alle *Car*'s naar het achtergrondgeheugen (bijv. harddisk) weg te schrijven.

Maak een tweede programma, *analyse*, dat de weggeschreven gegevens leest en voor elke auto grafieken van snelheid en positie laat zien met behulp van *Matplotlib*. Test dit grondig.

Breid het programma uit zodat uit de positie door middel van numerieke differentiatie weer de snelheid wordt berekend. Laat het programma ook deze grafieken tekenen en vergelijk ze met de oorspronkelijke snelheid. Omdat integratie en differentiatie elkaars omgekeerde zijn, zouden beide snelheden bij goede benadering overeen moeten komen.

--//--