

Python voor Natuur- en Sterrenkundigen

Eindopdracht: Angry Birds

Uiterste inleverdatum: vrijdag 11 december, 17:00 uur

In de eindopdracht gaan we kogelbanen plotten van “Angry Birds” op verschillende planeten. De benodigde informatie over de vogels en planeten moeten worden ingelezen uit twee bestanden welke je kunt downloaden van de website. In deze bestanden is er per regel informatie over een vogel of planeet te vinden. De datapunten zijn gescheiden door tabs. Regels die beginnen met een # worden beschouwd als commentaarregels en moeten bij het inlezen worden overgeslagen. De gebruikte eenheden voor de datapunten vind je terug in de commentaarregel van elk bestand. Let erop dat je misschien de datapunten moet omrekenen naar een andere eenheid voor de rest van de opdracht.

We gaan ervan uit dat de vogels geen last hebben van de luchtweerstand. De kogels zijn dus in een vrije val. Per vogel is er een startsnelheid v_0 en lanceerhoek θ_0 opgegeven. De baan van een kogel of projectiel wordt beschreven door de volgende formule:

$$\vec{v}_0 = v_{0x}\hat{i} + v_{0y}\hat{j}$$

De kogel ondervindt geen versnelling in de horizontale richting. De vogels zullen worden gelanceerd vanaf een lanceerplatform dat zich op 5 meter van de oppervlakte van de planeet bevindt. De vogels worden dus dicht bij de oppervlakte van de planeet gelanceerd en ondervinden daarom in de verticale richting de zwaartekracht. De zwaartekracht is een versnelling g die we per planeet berekenen (zie hieronder). Op basis hiervan kunnen we de formule vereenvoudigen tot:

$$\begin{aligned}x - x_0 &= (v_0 \cos \theta_0)t \\y - y_0 &= (v_0 \sin \theta_0)t - \frac{1}{2}gt^2\end{aligned}$$

(x_0, y_0) is de startpositie van de kogel (de locatie van het lanceerplatform). Voor een gegeven tijdstip t kan met deze formules een locatie (x, y) worden berekend.

Per planeet worden de massa van de planeet en de radius (de afstand van het middelpunt van de planeet tot de oppervlakte) gegeven. Met deze gegevens kan de valversnelling g worden berekend, welke geldt voor objecten vlakbij de oppervlakte van de planeet:

$$g = \frac{GM}{r^2}$$

En we hebben hiervoor de gravitatieconstante nodig:

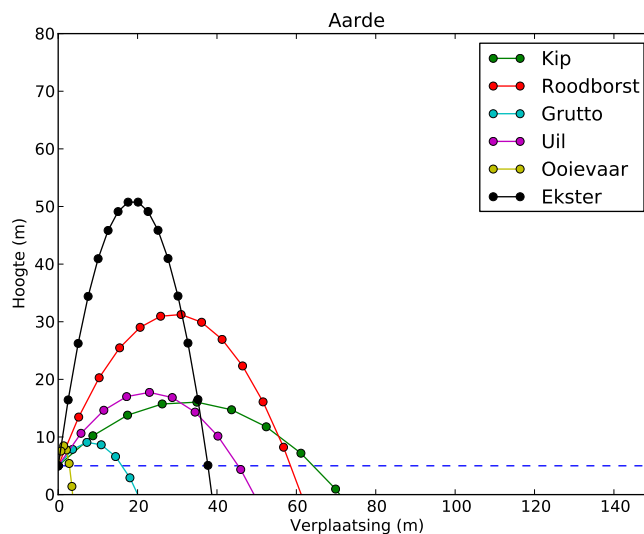
$$G = 6.67384 \times 10^{-11} \text{N} \cdot \text{m}^2/\text{kg}^2$$

Maak een module `natuurkunde.py` met alle constanten en functies voor de natuurkundige formules die je gebruikt. Importeer deze module vervolgens in je hoofdprogramma.

Nu moet er per planeet een plot worden gemaakt van de kogelbanen van alle vogels. Zoals is gezegd worden de vogels gelanceerd vanaf een lanceerplatform op 5 meter hoogte. De lanceersnelheid en lanceerhoek lees je uit het databestand. Gebruik een tijdspanne van 0 tot en met 20 seconden en genereer 50 tijdstappen (Hint: `np.linspace`). Bereken vervolgens de locatievectoren $((x, y)$ locaties) voor alle vogels op alle tijdstippen. Zodra dat is gedaan, kun je alle banen

plotten. Markeer elk datapunt met een teken (bijvoorbeeld een stip) en zorg voor verbindende lijnen tussen de stippen. Vang de “handles” op zodat je later een legenda kunt maken.

Plot ook een **stippellijn** $y = 5$. Geef de plot een titel (bijvoorbeeld de naam van de planeet) en label de assen. Als bereik mag je voor de x-as $[0, 150]$ gebruiken en voor de y-as $[0, 80]$. De liefhebber mag ook aan de hand van de resultaten geschikte bereiken voor de assen berekenen! Maak ook een legenda zodat het duidelijk is welke baan bij welke vogel hoort. Sla de plot op als PDF bestand en geef het bijvoorbeeld de naam van de planeet (Aarde.pdf). Een plot voor een planeet ziet er bijvoorbeeld zo uit:



Als laatste: gebruik de resultaat-array om voor elke vogel de maximaal bereikte hoogte (y) te bepalen. Let goed op langs welke as van de NumPy array je `np.amax` moet toepassen. Gebruik bijvoorbeeld een array met deze waarden als returnwaarde van je plot-functie.

Nadat je alle plots hebt gegenereerd heb je dus per planeet een array met de maximaal bereikte hoogtes per vogel. (Of misschien heb je deze allemaal in 1 array gestopt, natuurlijk ook prima!). Schrijf een nette tabel met deze gegevens naar het bestand `maxima.txt`. Bijvoorbeeld:

Maximaal bereikte hoogtes (m)						
	Kip	Roodborst	Grutto	Uil	Ooievaar	Ekster
Venus	17.3	34.0	9.37	19.0	8.77	55.7
Aarde	16.0	31.2	9.06	17.7	8.47	50.8
...						

Opmerkingen

- Je mag ervan uitgaan dat het bestand met vogels altijd 6 vogels bevat. Je kan dus van te voren NumPy arrays maken waarin je bij het lezen van het bestand direct de lanceersnelheden en lanceerhoeken kunt opslaan.
- Voor het inlezen van de verschillende files en het maken van de plot kun je natuurlijk nette, aparte functies maken.
- Let op: `np.cos` verwacht radialen en geen graden. Om graden om te zetten naar radialen kan je gebruik maken van `np.deg2rad`.

- Voor de liefhebber: door slim gebruik te maken van NumPy arrays kan je de locaties voor alle vogels voor 1 tijdstip in 1 keer berekenen. Je past dan dezelfde formule eenmaal toe op een array van 6 elementen. (Of 6,2 elementen waarbij je met bijvoorbeeld `[:,0]` alleen het x-coördinaat selecteert). Scheelt je weer het schrijven van een loop!
- In plaats van `.format(lijst[0], lijst[1], lijst[2], lijst[3])` kun je ook schrijven `.format(*lijst)`.

Inleveren

Uiterste inleverdatum: vrijdag 11 december, 17:00 uur

Hoe inleveren?

1. Digitaal de Python code inleveren, dit zijn twee bestanden dus een `angrybirds.py` en `natuurkunde.py`. Zorg ervoor dat de namen van de makers in het commentaar in de Python code genoemd worden! In het eventuele `.pyc` bestand zijn we niet geïnteresseerd. Stuur de code in een e-mail naar `pm@liacs.leidenuniv.nl`.
2. Deponeer ook de papieren versie van het verslag (inclusief de Python code) in de speciaal hiervoor bestemde doos "Python-opdrachten Programmeermethoden" in de postkamer van informatica, kamer 156 van het Snellius-gebouw.
3. Het verslag (uiteeraard weer in LaTeX, zie de eerdere opgaven) moet het volgende bevatten: een beschrijving van het programma (waarin het probleem kort maar duidelijk wordt uitgelegd), de plots die je hebt gegenereerd met je programma en omschrijf iets dat je in de plots opviel, een tabel met de gewerkte uren (uitgesplitst per week en per persoon). Voeg je Python code in door gebruik te maken van LaTeX listings. Voor de plots in PDF formaat kun je gebruik maken van de `graphicx`-package, zoals jullie gewend zijn met gewone afbeeldingen.

Het programma moet werken op een Linux machine met Python 2.7 waarop ook NumPy en matplotlib zijn geïnstalleerd.

Normering: Commentaar en verslag: 3 punten; modulariteit (functies, module): 3 punten; werking: 4 punten. Het cijfer voor deze opdracht telt voor 70% mee in het cijfer voor het Python-deel.