

Python Advanced - Labo8 - fietsdeelsysteem "Velo"

Inleiding

Je bouwt een object-georiënteerd simulatiemodel voor het fietsdeelsysteem "Velo" te Antwerpen. Je vindt een JSON-dataset met startinformatie op deze locatie:

<https://portaal-stadantwerpen.opendata.arcgis.com/datasets/637e9ec3779846faa94709b8509afd4d/explore>

Verdere uitleg: https://nl.wikipedia.org/wiki/Velo_Antwerpen

Officiële website: <https://www.velo-antwerpen.be/>

De opdracht

- Je bouwt een realistisch model bestaande uit objecten van minstens de volgende elementen: (fiets-)stations, slots (ook plaatsen genoemd), fietsen, gebruikers, fietstransporteurs, loggen van gebruik.
- Stations bevatten een aantal plaatsen of slots, fietsen kunnen gekoppeld zijn aan een slot enz...
- Fietstransporteurs zijn de vrachtwagens die rondrijden door de stad en in bulk fietsen verdelen van volle stations naar legere stations. Je modelleert ze als een bijzondere soort gebruikers.
- Je logt alle verplaatsingen en houdt ze bij in objecten die je via compositie bindt aan de juiste objecten.
- Je maakt gebruik van objectgeoriënteerd programmeren en bouwt met andere woorden het systeem op uit klassen en objecten. Je doet dit om je programma beter te structureren, met het oog op de overzichtelijkheid, onderhoudbaarheid en herbruikbaarheid van componenten. Je structureert ook door gepast op te delen in bestanden/modules en kiest gepaste namen voor alle entiteiten.
- Bij de eerste start van je toepassing creëer je aan de hand van de json-dataset een "reële" situatie (er zijn bvb ca 4200 fietsen, 309 stations en +55000 gebruikers). Je vult deze set aan met eigen data, je genereert bvb random namen en familienamen voor gebruikers.
- Je zorgt ervoor dat de data wordt opgeslagen en bijgehouden in één of meerdere bestanden.
- Start je de toepassing opnieuw dan geef je de keuze om opnieuw te beginnen of om verder te gaan vanaf de bestaande situatie.
- De interface is de terminal. Je zorgt er voor dat er relatief gebruiksvriendelijk met de toepassing kan worden omgesprongen door gebruik van een menu.
- Acties die manueel moeten kunnen geïnitieerd worden zijn het ontlenen van (een) fiets(en), het terug inchecken van (een) fiets(en) voor zowel gebruikers als transporteurs.
- Een extra actie die je moet kunnen initiëren vanuit de interface is het aanmaken van html-output. Je genereert een verzameling van webpagina's waarin je de toestand kan opvragen per station of per gebruiker en per fiets. Het betreft hier een statische website die je genereert in de folder `_site`. Je zorgt hier voor een zekere gebruikersvriendelijkheid via css.
- Je zorgt ook voor een simulatiemodus waarbij je de tijd kan versnellen en het registreren van verplaatsingen doorheen de stad kan automatiseren. De simulatiemodus kan worden gestart vanuit de terminal-interface, maar ook door de applicatie te starten met "-s" argument.

- Je mag uiteraard extra functionaliteit toevoegen bovenop het gevraagde en gebruik maken van Python libraries.

Hoe en wat dien je in?

- Je dient een zip-bestand in met als naam
2324_python_advanced_AchternaamVoornaam_opdracht8.zip
- Je werkt met een virtual environment (venv) en lijst gebruikte libraries op in een requirements.txt bestand.
- Je stuurt de venv folder zelf niet mee!
- Je maakt ook een folder reflectie aan met daarin een markdown (.md) bestand waarin je een reflectieverslag maakt over je toepassing. Je beschrijft - in markdown - de werking, de features en de keuzes die je hebt gemaakt. Je beschrijft tot slot ook kort de eventuele moeilijkheden die je hebt ervaren, en eventuele problemen die je niet hebt kunnen oplossen.