

Miniprojecten Programming

versie 1.4

datum: 12-10-2015

auteur: Rory Sie

versie	auteur	datum	beschrijving
0.9	Rory Sie	16-08-2015	Draftversie
1.0	Rory Sie	18-09-2015	Eindversie: wijze van oplevering is veranderd
1.1	Rory Sie	7-10-2015	De ontbrekende figuur in Appendix B is toegevoegd
1.2	Rory Sie	7-10-2015	link naar RDW API is vervangen
1.3	Rory Sie	12-10-2015	Vragen omtrent het Miniproject zijn verwerkt in Appendix B.
1.4	Rory Sie	15-10-2015	dead links vervangen en beschrijving Pkpass aangepast

Inleiding

Het Miniproject heeft als doel nog een keer de tijdens de colleges geleerde vaardigheden in praktijk te brengen. Uiteindelijk gaat het er niet alleen om dat je kunt uitleggen hoe iets geprogrammeerd wordt, maar dat je het ook in de praktijk kunt brengen.

Het Miniproject is er in twee varianten: A en B. Variant A is het normale Miniproject en dit vindt plaats in lesweek 7, 8, of 9 van blok A. Je krijgt een volle studieweek voor het Miniproject, wat neerkomt op zo'n 42 uur per teamlid. Je kunt 100 punten halen voor deze variant en daarmee maximaal een tien halen.

Variant B betreft de lange variant van het Miniproject die niet alleen meer tijd in beslag neemt (vanaf lesweek 3), maar ook uitdagender is en meer zelfstandigheid vereist. Je kunt voor deze variant maximaal 120 punten halen. In theorie kun je dus maximaal een twaalf halen, in de praktijk kunnen we je alleen een tien geven, maar deze variant geeft je wel meer kans op een hoog cijfer. **Let op:** Als je variant B kiest, dan moet je het eindresultaat in dezelfde week opleveren als de rest van de klas.

Opzet Miniproject

Planning

Maandag (of eerste bijeenkomst volgens rooster)	Ochtend: kick-off, tekenen teamcontract. Middag: project, begeleiding volgens rooster
Dinsdag	Project, begeleiding volgens rooster
Woensdag	Project, begeleiding volgens rooster 12.00 uur: inleveren Voortgangsverslag
Donderdag	Project, begeleiding volgens rooster
Vrijdag	17.00 uur: inleveren Applicatiecode, Screencast en Eindverslag.

Inleveren

De op te leveren deliverables zijn voor ieder Miniproject:

1. Voortgangsverslag:
 - **Lever het Voortgangsverslag in via Sharepoint uiterlijk woensdag 12.00 uur 's middags**
 - Dit is een eerste versie van het Eindverslag, waarbij je een update geeft over hoe ver je bent met het bouwen van de applicatie. Laat zien dat je gestructureerd te werk bent gegaan, dus:
 - neem PSDs of workflows op in je voortgangsverslag en
 - laat zien dat je een planning en taakverdeling hebt gemaakt.
2. De applicatie:
 - **Lever altijd een werkende versie van je applicatie in via Sharepoint, uiterlijk vrijdagmiddag 17.00 uur.** De docent moet namelijk genoeg tijd hebben om de applicatie te bekijken.
 - Zet alle code in een .zip-bestand met de bestandsnaam <klas>_team<teamnummer>_<Miniprojectnummer>.zip, bijvoorbeeld V1A_team2_A3.zip . Als je als team gebruik hebt gemaakt van een Git-repository, dan kun je deze met de docent delen door een email te sturen met onderwerp "Programming Klas <klas> team <teamnummer> Miniproject <Miniprojectnummer>"
 - Iedere applicatie is voorzien van een Graphical User Interface (GUI)
3. Een beschrijving van hoe je applicatie opgestart moet worden.
 - Zet dit in een **README.txt**-bestand in dezelfde map als je applicatie. Als je als team gebruik hebt gemaakt van Git(Hub), dan mag je de README.md in de repository zetten.
4. Het Eindverslag:
 - **Lever het Eindverslag in via Sharepoint uiterlijk vrijdagmiddag 17.00 uur.** De docent moet namelijk genoeg tijd hebben om het verslag te lezen. Een template voor het verslag kun je vinden op https://drive.google.com/open?id=0B6XotC2QRuK_TDh5YWxRUFBGQzg
 - Het verslag bevat de volgende onderdelen:
 - Zelfreflectie: Houd iedere dag bij welke taken je hebt uitgevoerd, en hoe lang je daarmee bezig bent geweest. Als je liever de tijd bijhoudt terwijl je bezig bent, dan kun je (als team) gebruik maken van een tool als toggl.com of <https://trackingtime.co> .
 - Peer-feedback:

- Geef voor ieder van je teamgenoten *tops* en *tips* aan. Schrijf dus op wat ze al goed doen, en wat ze kunnen verbeteren.
 - Geef op een schaal van 1-10 aan wat per teamlid de bijdrage aan het project is geweest. 1 staat voor een hele kleine bijdrage, 10 staat voor een hele grote bijdrage.
 - Applicatieworkflow: Een beschrijving van de workflow van de applicatie. Dit mag in één of meerdere PSD's, of in een ander soort diagram (bijv. *activity diagram*)
5. Een Screencast:
- **Lever de screencast in via Sharepoint uiterlijk vrijdagmiddag 17.00 uur.** De applicatie zal niet worden gepresenteerd voor de klas. In plaats daarvan vragen we jullie als team een *screencast* (video) te maken van de correcte werking van de applicatie. Een screencast is het opnemen van je computerscherm, waarbij je die werking van de applicatie laat zien.
 - Screencast maken: Op Windows kun je gebruik maken van software als CamStudio (<http://camstudio.org>) of Screencast-O-Matic (<http://www.screencast-o-matic.com>). Op de Mac kun je gebruik maken van Quicktime Player of gebruik webgebaseerde systemen als Screenr (<https://www.screenr.com>).
 - Dit mag met of zonder gesproken uitleg in de video, maar het kan handig zijn om de werking uit te leggen. Tip: Houd rekening met de tijd die het neemt om een dergelijke screencast te maken. Ga uit van minstens een halve dag werk.

Beoordeling

Het Miniproject telt voor 50% mee voor je eindcijfer en je moet minimaal een 5,5 halen. Ieder project wordt beoordeeld aan de hand van de leerdoelen zoals die zijn beschreven in de studentenhandleiding. De beoordelingscriteria zijn te vinden in de **rubric** van het Miniproject: https://drive.google.com/open?id=0B6XotC2QRuK_ZzhVUjhjdVIMDQ
 Lees deze rubric aandachtig door, want hier word je als team op beoordeeld!

Tips voor samenwerking

- **Maak duidelijke afspraken.** Zorg dat je het teamcontract ([Appendix A](#)) uitwerkt en ondertekent.
- **Houd taken bij** en wie voor welke taak verantwoordelijk is. Maak gebruik van gratis tools als Wunderlist of Trello (web èn mobiel) om de taken bij te houden
- **Houd tijd bij**, want hiermee kun je 1) aantonen wat je bijdrage is geweest, en 2) kan de docent zien waar de knelpunten hebben gelegen in het project. Gebruik hiervoor gratis tools als toggl.com of <https://trackingtime.co>
- **Overleg vaak (meerdere keren per dag)**, zodat je snel kunt inspringen op problemen die anderen hebben met code, PSDs maken, etc. Zorg wel dat iedereen aan het woord komt, want niet iedereen kan even snel reageren en zijn/haar mening uiten.
- **Gebruik een messaging app.** Telegram (web èn mobiel) wordt in de opleiding veel gebruikt, maar je kunt ook via Facebook of Whatsapp communiceren. Dit zorgt ervoor dat je snel even iets aan de ander kunt vragen.
- **Maak backups!** Niets is zo vervelend als hard werken en er dan opeens achter komen dat je applicatie/computer crasht en je geen backup hebt gemaakt. Via Git/Github gaat dit veelal automatisch, maar je kunt ook via Google drive of dropbox werken. De voorkeur heeft echter Git/Github, want dit komt later in de studie terug.

Problemen? Kijk in [Appendix B](#) voor extra tips.

Uitleiding

Voor nu rest ons alleen nog jullie succes te wensen. Kies een mooi project uit, en ga lekker aan de slag. Als je fouten maakt, dan is dat geen probleem. Je wordt geen goede programmeur als je geen fouten maakt, maar wel als je opstaat en weer doorgaat nadat je een fout maakt.

Succes namens het ontwikkelteam:

- Esther van der Stappen, docent Hogeschool Utrecht
- Arno Kamphuis, docent Hogeschool Utrecht
- Rienk van der Ploeg, docent Hogeschool Utrecht
- Silas Herlaar, student Hogeschool Utrecht
- Maikel Martens, Experius
- Edwin Bos, Alten PTS
- Rory Sie, docent Hogeschool Utrecht

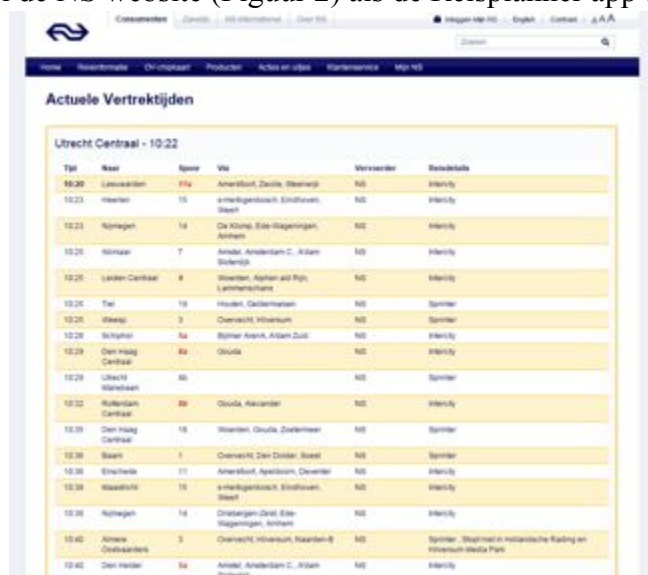
A. Normale Miniprojecten

A1. Actuele vertrektijden op de NS Kaartverkoopautomaat

Beschrijving

Om de Servicemedewerkers op de stations te ontlasten bestaat de wens om de reiziger op de NS Kaartverkoopautomaat de mogelijkheid te geven de actuele vertrekinformatie op te vragen. Dezelfde functie bestaat reeds op de NS website en in de Reisplanner app. Een grote team reizigers benadert op de stations nog steeds de Servicemedewerkers met vragen over vertrektijden, vertragingen en vertrekspreken. Door dezelfde functie ook onder te brengen in de NS Kaartautomaat is het de verwachting dat de druk op Servicemedewerkers afneemt.

Deze functie is op zowel de NS website (Figuur 2) als de Reisplanner app (Figuur 3) beschikbaar.



Tijl	Naar	Spoor	Van	Verreider	Reisduur
10:30	Leeuwarden	17A	Amsterdam, Dordrecht, Steenwijk	NS	Intercity
10:33	Heerlen	15	Amsterdam, Dordrecht, Steenwijk	NS	Intercity
10:33	Nijmegen	14	De Klomp, Ede, Wageningen, Arnhem	NS	Intercity
10:35	Rotterdam	7	Amstel, Amsterdam C., Almere, Schiedamschen	NS	Intercity
10:35	Utrecht Centraal	8	Wageningen, Arnhem, Almere, Schiedamschen	NS	Intercity
10:35	Tiel	18	Wageningen, Arnhem, Almere, Schiedamschen	NS	Sprinter
10:35	Utrecht	3	Amsterdam, Dordrecht, Steenwijk	NS	Sprinter
10:38	Wageningen	14	Amsterdam, Dordrecht, Steenwijk	NS	Intercity
10:38	Utrecht Centraal	8A	Utrecht	NS	Intercity
10:38	Utrecht	8B	Utrecht	NS	Sprinter
10:38	Rotterdam Centraal	8B	Utrecht, Rotterdam	NS	Intercity
10:38	Den Haag Centraal	18	Wageningen, Utrecht, Dordrecht, Steenwijk	NS	Sprinter
10:38	Almere	7	Amsterdam, Dordrecht, Steenwijk	NS	Sprinter
10:38	Utrecht	11	Amsterdam, Dordrecht, Steenwijk	NS	Intercity
10:38	Wageningen	15	Amsterdam, Dordrecht, Steenwijk	NS	Intercity
10:38	Nijmegen	14	Wageningen, Dordrecht, Ede, Wageningen, Arnhem	NS	Intercity
10:40	Almere	3	Amsterdam, Dordrecht, Steenwijk	NS	Sprinter
10:40	Den Haag	8A	Amstel, Amsterdam C., Almere, Schiedamschen	NS	Intercity

Figuur 2. Actuele vertrektijden op NS website.

Utrecht Centraal			
10:32	Rotterdam Centraal	8 ^B	
+3 min	Intercity		
	Via Gouda, Alexander		
10:35	Den Haag Centraal	18	
	Sprinter		
	Via Woerden, Gouda, Zoetermeer		
10:36	Baarn	1	
	Sprinter		
	Via Overvecht, Den Dolder, Soest		
10:36	Enschede	11	
	Intercity		
	Via Amersfoort, Apeldoorn, Deventer		
10:38	Maastricht	15	
	Intercity		
	Via s-Hertogenbosch, Eindhoven, Weert		
10:38	Nijmegen	14	
	Intercity		
	Via Driebergen-Zeist, Ede-Wageningen, Arnhem		
10:40	Almere Oostvaarders	3	
	Sprinter		
	Via Overvecht, Hilversum, Naarden-B		
	Stopt niet in Hollandsche Rading en Hilversum Media Park		

Figuur 3. Actuele vertrektijden in de Reisplanner app.

Opdracht

Het is de wens van de NS om op het startscherm van de automaat een “ingang” te creëren naar de nieuwe functie waarmee de klant actuele reisinformatie kan opvragen. Hierbij moet in ieder geval de reisinformatie getoond kunnen worden van het station waarop de kaartverkoopautomaat zich bevindt. Als uitbreiding hierop moet het ook mogelijk zijn om voor een ander Nederlands station de informatie op te vragen.

Het huidige startscherm van de automaat heeft de volgende layout (Figuur 4):



Figuur 4. Startscreen van de NS Kaartverkoopautomaat.

Voor het ophalen van de lijst met stations en het ophalen van de actuele vertrekinformatie is een NS API beschikbaar. Meer informatie over deze API is te lezen op de NS website (www.ns.nl/api).

Bepaal zelf welke informatie die de API biedt nuttig is voor de reiziger en hoe dit het beste gepresenteerd kan worden op het scherm. Houd hierbij in gedachten dat alle typen reizigers hiermee overweg moeten kunnen. Leesbaarheid (lettergrootte, font, kleurstelling) is hierbij erg belangrijk. Veel succes met de opdracht.

- Johan van Es en Paul Sollie, namens de NS

A2. Super-Wonder-Captain

Beschrijving

“Super-Wonder-Captain” is een spel waarbij je een superheld moet raden met zo min mogelijk hints. De hints geef je als game developer door 1) verbinding te maken met de Marvel API, 2) een willekeurige superheld te kiezen, en 3) informatie op te halen over deze superheld om hints te geven. Hoe meer hints je krijgt, des te minder punten je scoort. Raad de superheld met een beperkt aantal hints, zoals:

- Deze superheld(in) zat dezelfde verhaallijn als superheld X
- Deze superheld(in) zat in dezelfde comic als superheld X
- Deze superheld(in) heeft als superkracht Y

Marvel API

Je kunt met behulp van de Marvel API (<http://developer.marvel.com>) informatie opvragen over superhelden, hun verhaallijnen, comics en hun bedenker. Je moet hiervoor, net als bij de NS API, eerst een api key aanvragen. Dat is niet zo moeilijk: je maakt een account aan, bevestigt je emailadres en je krijgt je api key te zien.

Let op: Je krijgt twee api keys: public en private. Je hebt beide nodig om een api call te maken:

Authentication for Server-Side Applications

Server-side applications must pass two parameters in addition to the apiKey parameter:

- **ts** - a timestamp (or other long string which can change on a request-by-request basis)
- **hash** - a md5 digest of the ts parameter, your private key and your public key (e.g. md5(ts+privateKey+publicKey))

For example, a user with a public key of "1234" and a private key of "abcd" could construct a valid call as follows:

`http://gateway.marvel.com/v1/comics?ts=1&apikey=1234&hash=ffd275c5130566a2916217b101f26150` (the hash value is the md5 digest of 1abcd1234)

Figuur 5. Authenticatie voor de Marvel API.

De term **md5** staat voor een encryptiemethode. Zoek op internet hoe je in Python een zogenaamde md5 hash maakt van een string. Een goede website om mee te beginnen is:

<http://www.pythoncentral.io/hashing-strings-with-python/>

De term **ts** staat voor een timestamp: het aantal seconden sinds 1 januari 1970. Ook dit kun je in Python relatief eenvoudig berekenen (twee regels code): zoek dit op op internet.

Om de hash-value te berekenen die je moet gebruiken in je api call, doorloop je deze stappen:

1. Concateneer de strings van de timestamp, de public key en de private key
2. Geef de resulterende string als parameter in de encryptiemethode md5. Sla dit op als variabele hash. De hash is voor iedere call anders, omdat je de timestamp gebruikt die elke seconde verandert
3. Stuur nu de inhoud van variabele hash mee in de URL van de api call (zie hierboven). De URL is overigens voor iedere soort informatie anders. In figuur 4 zie je de URL voor informatie over comics. Voor superheroes is deze URL weer anders.

Als je een voorbeeld wilt van hoe de Marvel API werkt en wat voor soort JSON-response je terugkrijgt, dan kun je dat bekijken op de volgende URL: <http://developer.marvel.com/docs>

Opdracht

Je moet Super-Wonder-Captain kunnen spelen in een TKinter GUI. Je moet de naam van de speler van tevoren kunnen invullen. Daarnaast moet het aantal nog te behalen punten te zien zijn in het scherm.

Puntentelling

Je kunt 25 punten verdienen met een antwoord dat in één keer goed is. Voor de volgende acties krijg je punten aftrek:

- Fout antwoord: 1 punt
- Hint vragen: 3 punten

De speler moet een nieuwe hint kunnen opvragen. Zorg dus dat je een knop maakt waarmee de speler een nieuwe hint kan opvragen. Zorg dat het opvragen van een hint eerst werkt voor één soort hint, bijvoorbeeld de beschrijving ('description'). Hierna kun je meerdere soorten hints opvragen, zoals de comics, of de series. Begin dus eerst klein en zorg dat het werkt, en bouw de applicatie daarna uit met meerdere hints.

Als de superheld is geraden, sla dan de score op. Houd de score bij in een *highscore-database* of bestand. Je moet de highscore van die dag en de *all time highscore* kunnen tonen in een TKinter GUI.

Tip: Superheld anonimiseren

Het is gebruikelijk dat de naam van de superheld in de informatie staat die in de Marvel API zit. Zorg dus dat je de naam van de superheld wegfiltert, want anders wordt het wel erg makkelijk raden. Let wel op dat er soms afkortingen worden gebruikt in de naam van de superheld (bijv. ‘A.I.M.’), terwijl deze naam in de beschrijving anders wordt geschreven (bijv. ‘AIM’). Het kan echter ook zo zijn dat het ‘description’-veld leeg is gelaten. Dan kun je deze hint en mogelijk ook de superheld niet gebruiken.

Ook staat soms de echte naam van de superheld tussen haakjes in de naam (bijv. ‘Abomination (Emil Blonsky)’), en komt deze echte naam (‘Emil Blonsky’) ook terug in de beschrijving. Bekijk zelf of je de echte naam bekend wilt maken of niet.

Tip: Keuze maken uit tien superhelden

Het kan door de grote hoeveelheid (onbekende) superhelden in de database van Marvel best moeilijk worden om de superheld te raden. Je kunt er daarom voor kiezen om de speler tien keuzes voor te leggen van mogelijke superhelden, waarvan één de juiste is, en negen superhelden willekeurig uit de API zijn opgehaald.

Extra opdracht

Spelers kunnen natuurlijk op Internet zoeken naar de hints en op deze manier vinden over welke superheld het gaat. Houd daarom de tijd bij tijdens het raden en zorg voor puntenaftrek bij elke 10 seconden.

Zorg vervolgens dat je de high scores op twee manieren kunt weergeven:

- snelst geraden superheld (per dag/week)
- hoogste puntenaantal (per dag/week)

A3. Thuisbioscoop

Beschrijving

Groot-schermbezitters (aanbieders) kunnen thuis een film draaien voor een groep bezoekers. De films kunnen alleen op één locatie tegelijk worden vertoond. Daarnaast is de film en tijd afhankelijk van de tv-programmering. Deze is te vinden via de Films op TV API van Filmtotaal (<http://www.filmtotaal.nl/api/>). De bezoekers kopen een kaartje voor een film. Zodra iemand een kaartje heeft gekocht, krijgt deze een unieke toegangscode. Deze toegangscode wordt bij het bezoek door de aanbieder (de groot-schermbezitter) ingevoerd en gecheckt in het systeem.

Opdracht

Zorg dat je in een TKinter GUI een overzicht laat zien van films en aanbieders. De namen van de aanbieders mag je zelf verzinnen (bijv. de leden van je team), maar er is maar één aanbieder per film. Een aanbieder kan wel meerdere films vertonen.

Bezoekers kunnen hun naam en mailadres invoeren en vervolgens een film selecteren. Print vervolgens een ticket uit met een unieke code. Deze code moet je namelijk per bezoeker en per film opslaan in de database. Dit opslaan in de database kan via een simpel tekstbestand, of in een .csv-bestand (door komma's gescheiden bestand). .csv-bestanden kun je inlezen in Microsoft Excel,

en dat kan handig zijn als je met grotere bestanden werkt. Voor meer informatie over hoe je met .csv-bestanden werkt in Python, kijk dan op <https://docs.python.org/3/library/csv.html>

De aanbieder moet alle gasten voor de dag kunnen zien in een TKinter GUI. Zorg dat alle bezoekers eerst op begintijd van de film gesorteerd worden, en daarna op (achter)naam.

Let op: de aanbieders mogen elkaars gasten niet zien, dus zorg ervoor dat de aanbieders alleen hun eigen gasten mogen zien. Werk bijvoorbeeld met een aanmeldvenster waarbij je om een (veilig) wachtwoord vraagt.

Extra opdracht 1

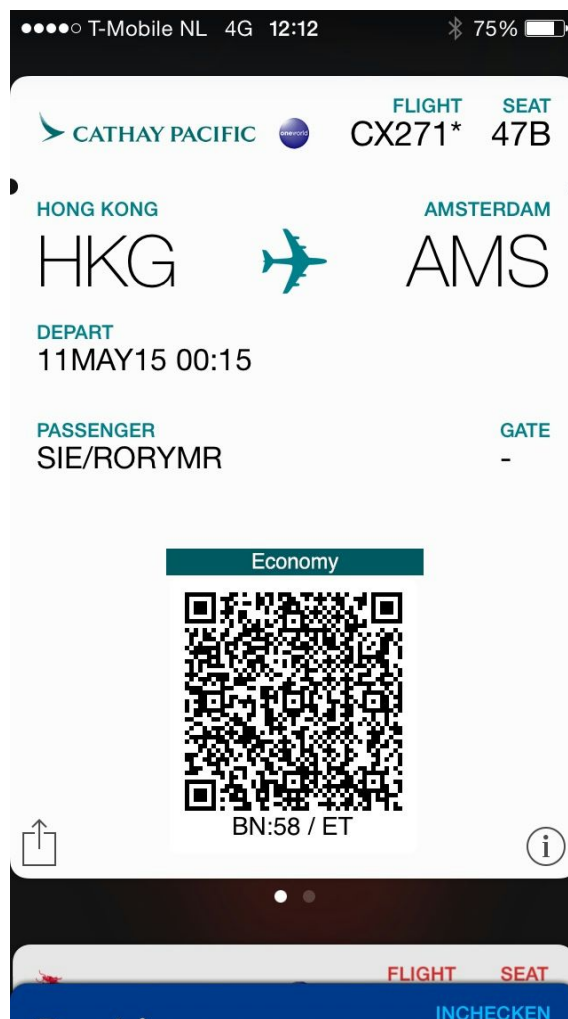
Om het checken van de kaartjes makkelijker te maken, kunnen kaartjes voorzien worden van een qr-code. Je zou ervoor kunnen kiezen deze module ervoor te gebruiken: `qr` (<https://pypi.python.org/pypi/qr>). Je kunt hiermee qr-codes aanmaken van de unieke code die je eerder hebt gegenereerd.

Tip

Als je een lokale server draait op je computer en je wilt een QR-code scannen, dan is het een stuk makkelijker als je dit niet doet via je telefoon (omdat je dan je computer/server open moet stellen), maar via je browser op je computer. Hiervoor bestaan online QR-scanners, zoals: <https://www.the-qr-code-generator.com/scan>

Extra opdracht 2

Zorg dat de bestanden ingeladen kunnen worden in iOS Passbook of Android PassWallet. Dit ziet eruit als Figuur 6:



Figuur 6. PKPass voorbeeld.

Hiervoor moet je .pkpass-bestanden aanmaken. Kijk op de website van Apple hoe je dit voor elkaar kunt krijgen:

https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/PassKit_PG/index.html#//apple_ref/doc/uid/TP40012195

Je hebt een zogeheten Team ID en Pass Type ID nodig om een pkpass bestand aan te maken. Dit kun je alleen gebruiken als je een Apple Developer account hebt. Heb je nog geen Apple Developer account, vraag dan aan je docent of hij/zij voor jou een aanvraag wilt indienen. De docent stuurt vervolgens een aanvraag naar de vaktrekker Rory Sie, voorzien van jouw voornaam, achternaam en email-adres. De vaktrekker zal jou vervolgens uitnodigen voor het Apple Developer Program.

Het Team ID is: 6HY62VM8NZ

Het Pass Type ID is:

name: HUFNTPROGRAMMING1516

ID: pass.nl.hu.fnt.programming.1516

Links:

- Online QR-scanner: <https://www.the-qrcode-generator.com/scan>
- Apple .pkpass-bestanden aanmaken: https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/PassKit_PG/index.html#//apple_ref/doc/uid/TP40012195
- Python en .csv-bestanden: <https://docs.python.org/3/library/csv.html>

A4. Inchecken bij de NS

Beschrijving

Wanneer een passagier met de trein reist, dan is deze verplicht om in- en uit te checken. De NS is op dit moment bezig om dit proces te herzien. Ze willen namelijk dat het binnenkort ook mogelijk moet zijn om in een digitale ‘wallet’ je ov-chipkaart op te slaan. Je slaat dan als het ware een QR-code-versie van je ov-chipkaart in je telefoon op. Op iOS gebeurt dat met de Passbook app, op Android kun je daar bijvoorbeeld PassWallet voor gebruiken. Een QR-code ziet er als volgt uit:



Figuur 7. Qr-code.

Deze QR-code kun je met een QR-reader of QR-code scanner uitlezen. Deze kun je voor je mobiele telefoon gratis downloaden uit Google Play of de App Store. Scan de QR code, en de app vertaalt de QR-code voor je naar de oorspronkelijke tekst. Je kunt de code ook online scannen via <https://www.the-qrcode-generator.com/scan> .

Je kunt een unieke QR-code maken voor iedere reis op basis van het ov-chipkaartnummer. Zorg dat ov-chipkaartnummers worden opgeslagen in de database, en vertaal deze naar de QR-code. Daarna kun je de QR-code scannen om het ov-chipkaartnummer terug te krijgen en op te zoeken in de database.

Opdracht

Maak een applicatie ‘Reis database’ (inclusief TKinter GUI) waarmee je verschillende gegevens van de ov-chipkaart en de reis kunt opgeven en opslaan in de database. Iedere reis krijgt hierbij een unieke ID ‘reisID’. De volgende gegevens worden opgeslagen:

- reisID,
- naam,
- ov-chipkaartnummer,
- beginstation,
- eindstation

Vervolgens wordt er op basis van de reis ID een QR-code gegenereerd, zodat de gebruiker deze QR-code kan uitprinten/opslaan op zijn/haar telefoon. Iedere reis heeft dus een unieke reis ID waarmee je deze weer terug kunt vinden in de database.

Door de QR-code scanner te gebruiken wordt de qr-code weer gedecodeerd (extra: je mag ook een QR scanner bouwen). Je krijgt dan de oorspronkelijke reis ID weer terug, zodat je de gegevens over de reiziger en de reis weer terug kunt vinden.

Maak een applicatie ‘Incheckzuil’ (voorzien van TKinter GUI) waarbij je het ov-chipkaartnummer kunt invoeren om in de database gegevens terug te vinden over de reiziger. Je mag in plaats hiervan ook een REST-server maken die een reis ID kan ontvangen en vervolgens de gegevens van de klant zoekt in de database (zie Extra).

Zorg als laatste dat je voor de NS-marketingafdeling een (automatisch) overzicht genereert van:

- het aantal reizen per ov-chipkaart,
- de populairste bestemmingen, en
- de populairste vertrekstations.

Dit hoeft niet per sé in een TKinter GUI, dus dit mag ook in een tekstbestand. Zorg wel dat je de gegevens op naam sorteert.

Extra opdracht 1: Automatisch opzoeken ov-chipkaartnummer

Als je de online QR-scanner gebruiken, dan kun je automatisch verbinding maken met een zogeheten REST-server die op je computer draait. Je hebt hier mogelijk al kennis mee gemaakt tijdens de workshop Python op de matchingsdag: Bij opdracht B2 moest je het bestand bkr-api.py uitvoeren. Hiermee startte je de server van het BKR op. Vervolgens moest je de postcode en het huisnummer van de klant in een request naar de BKR server sturen. Kijk hier nog eens naar de bestanden van de matchingsdag, waaronder de opdrachtbeschrijving <http://goo.gl/5FSEb0>.

Hetzelfde kun je ook doen met de QR-code scanner. Als je de BKR-server zo aanpast dat deze een reis ID kan ontvangen in plaats van een postcode en huisnummer, dan kun je het reis ID opzoeken in de database. Let wel op dat je qr-code dan verwijst naar het juiste serveradres, bijvoorbeeld <http://127.0.0.1:5000/lookup/<reis ID>>. Als je er niet uitkomt, dan kun je meer informatie opzoeken op de site van de onderliggende module Flask of de tutorial (zie links).

Tip

Als je een lokale server draait op je computer en je wilt een QR-code scannen, dan is het een stuk makkelijker als je dit niet doet via je telefoon (omdat je dan je computer/server open moet stellen), maar via je browser op je computer. Hiervoor bestaan online QR-scanners, zoals:

<https://www.the-qr-code-generator.com/scan>

Extra opdracht 2: qr-code opslaan in iOS Wallet / Android PassWallet

Zorg dat de bestanden ingeladen kunnen worden in iOS Passbook of Android PassWallet. Hiervoor moet je .pkpass bestanden aanmaken. Kijk op de website van Apple hoe je dit voor elkaar kunt krijgen:

https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/PassKit_PG/index.html#//apple_ref/doc/uid/TP40012195

Je hebt een zogeheten Team ID en Pass Type ID nodig om een pkpass bestand aan te maken. Dit kun je alleen gebruiken als je een Apple Developer account hebt. Heb je nog geen Apple Developer account, vraag dan aan je docent of hij/zij voor jou een aanvraag wilt indienen. De docent stuurt vervolgens een aanvraag naar de vaktrekker Rory Sie, voorzien van jouw voornaam, achternaam en email-adres. De vaktrekker zal jou vervolgens uitnodigen voor het Apple Developer Program.

Het Team ID is: 6HY62VM8NZ

Het Pass Type ID is:

name: HUFNTPROGRAMMING1516

ID: pass.nl.hu.fnt.programming.1516

Links:

- Online QR-scanner: <https://www.the-qrcode-generator.com/scan>
- Flask REST-server: <http://flask.pocoo.org/docs/0.10/>
- Flask tutorial:
<http://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world>
- Apple .pkpass-bestanden aanmaken:
https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/PassKit_PG/Chapters/YourFirst.html

AX. Vrije opdracht (Teamproject)

Dit betreft een vrije opdracht, die moet voldoen aan de volgende eisen:

1. Data wordt opgeslagen in een database.
 - a. Dit is minimaal een tekstbestand, maar kan ook een CSV-bestand of een sqllite database zijn.
 - b. Optie: Sla gegevens op een veilige manier op, dat wil zeggen dat bijvoorbeeld persoonlijke gegevens worden gecodeerd
2. Input:
 - a. Er wordt input van de gebruiker gevraagd, in de vorm van tekst, een qr-code of iets vergelijkbaars.
 - b. Er wordt gewerkt met complexe data, dat wil zeggen dat er meerdere soorten informatie worden gecombineerd. Het eenvoudigste is om dit via een online API te doen, omdat je die data dan kunt combineren met input van de gebruiker. Maar je kunt er ook voor kiezen om zelf verschillende soorten data te combineren, bijvoorbeeld door zelf data te verzamelen van gebruikers, van het web ('web scraping'), of juist meerdere APIs te combineren (bijvoorbeeld NS vertragingen en klachten via Twitter).
3. Output:
 - a. De applicatie is voorzien van een TKinter GUI waarin informatie op een gebruiksvriendelijke wijze wordt gepresenteerd
 - b. Optie: Je kunt ervoor kiezen om een visualisatie van de informatie te maken, of bepaalde informatie te exporteren naar een qr-code.

Beoordeling

In de beoordeling houdt de docent rekening met de complexiteit van de opdracht. Hoe complexer de opdracht, hoe meer punten je kunt halen. Vraag daarom van tevoren aan de docent of de opdracht die je zelf hebt bedacht groot genoeg is, en of je daarmee akkoord gaat. Uiteindelijk wordt iedereen beoordeeld op de criteria die in de rubric staan.

Naast dat een applicatie complexer van aard is als deze meer acties moet verrichten (complexere workflow), kan er bijvoorbeeld hoger gescoord worden als er gebruik is gemaakt van een .csv bestand (zie opdracht A3), of een sqllite database om data op te slaan (zoek op Google voor de werking van sqllite). Een TKinter GUI is complexer van aard als er inputgegevens worden gevraagd, en op basis hiervan actie wordt ondernomen, in plaats van alleen een visualisatie van data.

B. Lange Miniprojecten

B1. Parkeergaragebeheer

Beschrijving

De parkeergarage aan de Kruisstraat wil graag dat auto's bij aankomst/vertrek worden herkend door middel van hun kenteken. Doe dit door een foto van een auto te maken en het kenteken uit te lezen door middel van een bestaande Pythonmodule. Sla het kenteken op in de database, zodat je bij vertrek kunt meten hoe lang de auto in de parkeergarage heeft gestaan.

RDW API

Haal vervolgens met het kenteken de gegevens op bij het RDW (<https://overheid.io/documentatie/voertuiggegevens>), want deze garage staat in het Centrum van Utrecht. Sinds 2015 is het verboden om met auto's die als hoofdbrandstof diesel en een datum van eerste afgifte van vóór 2001 hebben het centrum van Utrecht in te rijden.

Deze informatie moet worden opgeslagen in de centrale database van het parkeergaragebeheer, maar dat moet wel veilig gebeuren. Zorg dus dat de informatie gecodeerd wordt.

Links

- Google tesseract-ocr: <https://code.google.com/p/tesseract-ocr/>
- pytesseract Python tesseract implementatie: <https://pypi.python.org/pypi/pytesseract>
- openalpr: <https://github.com/openalpr/openalpr/tree/master/src/bindings/python>
- Image Text Recognition: <http://francescopochetti.com/text-recognition-natural-scenes/>
- AES Encryptie van gegevens:
<http://pythonprogramming.net/encryption-and-decryption-in-python-code-example-with-explanation/>

Extra opdracht

Zorg ervoor dat de klant zijn of haar factuurgegevens kan invoeren. De gegevens moeten gekoppeld worden aan het kenteken van de auto, zodat bij het parkeren automatisch gefactureerd kan worden.

B2. Wachtdienst NS

Beschrijving

Bij de onderhoudsdienst van de kaartautomaat werkt een aantal monteurs. Deze monteurs installeren nieuwe software op de automaten, maar verhelpen ook problemen met de hardware en software van de kaartautomaten. In de avond draaien ze wachtdienst, want het kan zomaar zijn dat een kaartautomaat het begeeft.

Opdracht

De NS wil graag dat er een zo efficiënt mogelijk systeem komt voor de monteurs. Ten eerste moet je bijhouden welke kaartautomaat kapot is. Hiervoor moet je de unieke naam/code van de

kaartautomaat weten, en de locatie. De locatie bestaat uit het station, maar nog beter: de geolocatie, dat wil zeggen de breedtegraad (latitude) en lengtegraad (longitude). Deze kun je via de NS API opvragen per station.

Ten tweede moet je weten of monteurs beschikbaar zijn of niet. Niet alle monteurs zijn altijd beschikbaar, anders zou iedere monteur dag en nacht werken. Houd dus bij wanneer een monteur beschikbaar is. Houd ook de contactgegevens van de monteur bij (adres, telefoon, lengte- en breedtegraad).

Ten derde moet er efficiënt gezocht worden naar monteurs. Zorg dus dat je eerst begint bij de monteur die het dichtst bij de locatie van de kaartautomaat woont. Daarna ga je voor de monteur die daarna het dichtst bij de locatie van de kaartautomaat woont, enzovoort. De afstand kun je bepalen door gebruik te maken van de Google Distance Matrix API (<https://developers.google.com/maps/documentation/distancematrix/intro>).

Nu komt het deel waarbij je contact gaat leggen met de monteur. Je kunt via de Telegram API (<https://core.telegram.org>) berichten sturen. Je kunt in plaats van de telefoonnummers van de monteurs de telefoonnummers van het team gebruiken. Stuur een Telegrambericht naar de dichtstbijzijnde beschikbare monteur. Wacht een bepaalde tijd, en ga dan over op de volgende monteur, enzovoort.

Links

- Google Distance Matrix API: <https://developers.google.com/maps/documentation/distancematrix/intro>
- NS API: <http://www.ns.nl/api/home>
- Telegram API: <https://core.telegram.org>

BX. Vrije opdracht

Het betreft hier een vrije opdracht, die moet voldoen aan de volgende eisen:

1. Data wordt opgeslagen in een database.
 - a. Dit is minimaal een CSV-bestand of een sqlite database.
 - b. Sla gegevens op een veilige manier op, dat wil zeggen dat bijvoorbeeld persoonlijke gegevens worden gecodeerd
2. Input:
 - a. Er wordt input van de gebruiker gevraagd, in de vorm van 1) tekst, en 2) een qr-code of iets vergelijkbaars (afbeelding).
 - b. Er wordt gewerkt met complexe data, dat wil zeggen dat er meerdere soorten informatie worden gecombineerd. Het eenvoudigste is om dit via een online API te doen, omdat je die data dan kunt combineren met input van de gebruiker. Maar je kunt er ook voor kiezen om zelf verschillende soorten data te combineren, bijvoorbeeld door zelf data te verzamelen van gebruikers, van het web ('web scraping'), of juist meerdere APIs te combineren (bijvoorbeeld NS vertragingen en klachten via Twitter).
3. Output:
 - a. De applicatie is voorzien van een TKinter GUI waarin informatie op een gebruiksvriendelijke wijze wordt gepresenteerd
 - b. Maak een visualisatie van de informatie, of exporteer bepaalde informatie naar een qr-code, .pkpass of iets vergelijkbaars.

Beoordeling

In de beoordeling houdt de docent rekening met de complexiteit van de opdracht. Hoe complexer de opdracht, hoe meer punten je kunt halen. Vraag daarom van tevoren aan de docent of de opdracht die je zelf hebt bedacht groot genoeg is, en of je daarmee akkoord gaat. Uiteindelijk wordt iedereen beoordeeld op de criteria die in de rubric staan.

Naast dat een applicatie complexer van aard is als deze meer acties moet verrichten (complexere workflow), kan er bijvoorbeeld hoger gescoord worden als er gebruik is gemaakt van een .csv bestand (zie opdracht A3), of een sqlite database om data op te slaan (zoek op Google voor de werking van sqlite). Een TKinter GUI is complexer van aard als er inputgegevens worden gevraagd, en op basis hiervan actie wordt ondernomen, in plaats van alleen een visualisatie van data.

C. Team herkansingsprojecten

Om aan de herkansing mee te doen, moet je je hebben ingeschreven in Osiris. Hierna zal de docent contact opnemen via e-mail zodra de herkansingsperiode begint. Je kunt vervolgens aangeven of je als team of als individu het Miniproject wilt herkansen.

Het is niet de bedoeling dat bij de keuze voor een herkansingsopdracht een keuze gemaakt kan worden uit de A- of B-Miniprojecten.

C1. iNStagram

De NS wil een nieuwe pilot beginnen waarbij ze op de stations op grote schermen recente Instagram-foto's van die locatie weergeven. Het idee is als volgt: Je staat op station Utrecht Centraal. Station Utrecht Centraal heeft een bepaalde geolocatie, weergegeven door een lengtegraad (longitude) en breedtegraad (latitude). Met deze gegevens kun je via de Instagram API alle foto's of videos op een locatie opzoeken.

Opdracht

Zorg dat deze service voor ieder willekeurig station werkt. Vraag daarom via een TKinter GUI input van de gebruiker in de vorm van een stationsnaam. Haal vervolgens via de NS API alle NS-stations op, en zoek of het gegeven station er tussen staat.

Haal de gegevens *lat* en *long* op uit de response van de NS API en kijk vervolgens via de Instagram API of er media zijn gepost in de buurt van die locatie (dit staat automatisch op een straal van 1 km). Haal de URLs van de media op Instagram op en geef ze weer in de TKinter GUI, zodat je de links kunt kopiëren en plakken in je webbrowser.

Maak als laatste per type medium (foto, video) een overzicht van hoeveel er zijn gepost en wanneer.

Extra opdracht

Zorg dat zodra je op een link klikt, je de URL opent in een nieuw browservenster. Of nog beter: Je laat de media rechtstreeks zien in de TKinter GUI.

Benodigheden

1. Module requests om verbinding te maken met de API's
2. In het geval van de NS moet je een XML response kunnen uitlezen, bij Instagram is de response in JSON-formaat
3. Access token van Instagram. Hiervoor moet je een Instagram-account hebben, ingelogd zijn, en vervolgens een access token opvragen via <http://instagram.pixelunion.net>

Links

- Instagram API media/search: <https://instagram.com/developer/endpoints/media/>
- Instagram access token generator: <http://instagram.pixelunion.net>

CX. Vrije opdracht (Teamproject)

Let op: Dit is een vrije opdracht. Kies hier alleen voor als je heel erg zeker weet dat het gaat lukken. Over het algemeen leveren vrijere opdrachten ook meer problemen op. Aan de andere kant kun je gemotiveerder zijn om de oplossing te vinden. Denk dus goed na. Leg je idee voor de vrije opdracht sowieso voor aan de docent.

Deze vrije opdracht moet voldoen aan de volgende eisen:

1. Data wordt opgeslagen in een database.
 - a. Dit is minimaal een tekstbestand, maar kan ook een CSV-bestand of een sqlite database zijn.
 - b. Optie: Sla gegevens op een veilige manier op, dat wil zeggen dat bijvoorbeeld persoonlijke gegevens worden gecodeerd
2. Input:
 - a. Er wordt input van de gebruiker gevraagd, in de vorm van tekst, een qr-code of iets vergelijkbaars.
 - b. Er wordt gewerkt met complexe data, dat wil zeggen dat er meerdere soorten informatie worden gecombineerd. Het eenvoudigste is om dit via een online API te doen, omdat je die data dan kunt combineren met input van de gebruiker. Maar je kunt er ook voor kiezen om zelf verschillende soorten data te combineren, bijvoorbeeld door zelf data te verzamelen van gebruikers, van het web ('web scraping'), of juist meerdere APIs te combineren (bijvoorbeeld NS vertragingen en klachten via Twitter).
3. Output:
 - a. De applicatie is voorzien van een TKinter GUI waarin informatie op een gebruiksvriendelijke wijze wordt gepresenteerd
 - b. Optie: Je kunt ervoor kiezen om een visualisatie van de informatie te maken, of bepaalde informatie te exporteren naar een qr-code.

Beoordeling

In de beoordeling houdt de docent rekening met de complexiteit van de opdracht. Hoe complexer de opdracht, hoe meer punten je kunt halen. Vraag daarom van tevoren aan de docent of de opdracht die je zelf hebt bedacht groot genoeg is, en of je daarmee akkoord gaat. Uiteindelijk wordt iedereen beoordeeld op de criteria die in de rubric staan.

Naast dat een applicatie complexer van aard is als deze meer acties moet verrichten (complexere workflow), kan er bijvoorbeeld hoger gescoord worden als er gebruik is gemaakt van een .csv bestand (zie opdracht A3), of een sqlite database om data op te slaan (zoek op Google voor de werking van sqlite). Een TKinter GUI is complexer van aard als er inputgegevens worden gevraagd, en op basis hiervan actie wordt ondernomen, in plaats van alleen een visualisatie van data.

D. Individuele herkansingsprojecten

D1. Nim-7

Beschrijving

Bij het spel Nim-7 krijg je aan het begin van het spel 7 munten. Twee spelers mogen om de beurt één of twee munten pakken. Degene die de laatste munt pakt, heeft gewonnen. Kijk voor een beschrijving op [https://nl.wikipedia.org/wiki/Nim_\(spel\)](https://nl.wikipedia.org/wiki/Nim_(spel))

Opdracht

Zorg dat je het spel Nim-7 in een TKinter GUI kan spelen. Maak een TKinter GUI met de volgende onderdelen:

- 2 knoppen:
 - “pak 1 steen”
 - “pak 2 stenen”
- een label dat aangeeft: “speler 1 aan de beurt” of “speler 2 aan de beurt”
- een label dat aangeeft hoeveel munten er nog op de stapel liggen

Vereisten:

- Start met een willekeurige speler.
- Het spel eindigt als iemand de laatste munt heeft gepakt.

Tip

Probeer het spel eerst eens “offline” te spelen. Verzamel 7 munten, stenen, pennen, paperclips, of jelly beans. Vraag met een klasgenoot of deze met jou het spel Nim-7 wil spelen.

Extra opdracht

Kopieer je bestanden en probeer een tweede versie van Nim te maken waarbij je tegen de computer speelt. Kun je een optimale strategie bedenken waarbij de computer altijd wint?

Appendix A. Teamcontract

Werk samen met je team deze afspraken uit. Voeg zo nodig nog extra afspraken toe. Zorg dat ieder teamlid het teamcontract ondertekent. Lever een kopie van dit contract in bij je docent.

Team:

Teamleden:

- 1)
- 2)
- 3)
- 4)
- 5)

Vergaderingen

Plan aan het begin van het project de vergaderingen. Op deze manier kan iedereen in zijn/haar agenda rekening houden met de vergaderingen.

Dag	Tijd	Locatie

Als een teamlid niet deel kan nemen aan de vergadering, dan laat deze dat ... uur van tevoren weten. *

Als dit niet gebeurt, dan resulteert dit in een waarschuwing (zie consequenties).

Communicatie

We hebben de voorkeur voor communicatie via email / Telegram / telefoon / Skype / Facebook / Google+ / ... *

We zullen reageren binnen ... uur *

Als dit niet gebeurt, dan resulteert dit in een waarschuwing (zie consequenties).

Beslissingen

We nemen alleen beslissingen als iedereen / de meerderheid het hiermee eens is.*

Als er (grote) dingen worden gedaan waar geen beslissing over is genomen, dan resulteert dit in een waarschuwing (zie consequenties).

Verantwoordelijkheid

Leden van het team zullen de overeengekomen taken uitvoeren zoals ze zijn afgesproken tijdens de vergadering. Als er onverwachte problemen zijn, dan worden deze tijdig met het team gecommuniceerd.

Als dit niet gebeurt, dan resulteert dit in een waarschuwing (zie consequenties).

Consequenties

Ieder teamlid gaat akkoord met de bovenstaande afspraken. Als dit niet gebeurt, dan resulteert dit in een waarschuwing.

Bij drie waarschuwingen wordt het desbetreffende teamlid tijdelijk uit het team verwijderd. Het verwijderde teamlid zal binnen ... * uur communiceren met het team hoe hij/zij denkt de situatie anders aan te pakken. De rest van het team moet hiermee akkoord gaan, voordat het teamlid weer wordt toegelaten.

Bij vier waarschuwingen wordt het teamlid permanent uit het team verwijderd.

Ondertekening

Ik ga akkoord met de bovenstaande afspraken en ik ben mij ervan bewust dat als ik me hier herhaaldelijk niet aan houd, dit consequenties kan hebben.

Teamlid	Handtekening
1	
2	
3	
4	
5	

* doorhalen wat niet van toepassing is / invullen

Appendix B. Omgaan met tegenslagen

Inhoudelijke problemen

Ik kom er niet uit, of ik krijg een foutmelding.

Probeer eerst eens te kijken wat er gebeurt in je code. Je kunt print-statements gebruiken, maar nog beter: breakpoints. Dan kun je zien wat er op welk moment gebeurt, en welke waarden variabelen krijgen.

Als je een foutmelding krijgt, of iets werkt niet zoals het moet, bekijk dan de figuur hieronder, want het kan zijn dat jouw probleem (inclusief oplossing) erbij staat.



Figuur 8. Veelvoorkomende problemen.

Kom je er nog steeds niet uit? Kijk vervolgens op Google / StackOverflow of je een oplossing kunt vinden. Als je problemen hebt om een bestand uit te lezen, zoek dan bijvoorbeeld naar “read file Python”. Neem altijd het woord Python op in je zoektermen, want anders krijg je zoekresultaten over bijvoorbeeld Java, PHP en C++.

Nog steeds geen oplossing? Vraag aan klasgenoten of ze tegen een vergelijkbaar probleem aan zijn gelopen. Probeer niet klakkeloos hun oplossing over te nemen, maar probeer er zelf uit te komen.

Als je zelf op de oplossing komt, dan blijft deze beter hangen dan wanneer je die van iemand anders kopieert.

Nog steeeds geen oplossing? Vraag dan aan de student-assistent of de docent of deze je kan helpen. Je krijgt waarschijnlijk niet de oplossing aangereikt, maar je komt wel weer verder.

Het lukt niet om verbinding te maken met de API.

Dit kan meerdere oorzaken hebben. Krijg je wel verbinding, en krijg je dus een response? Dan zit daar een status bij. Kijk of je daar iets mee kunt als je op Google zoekt. Als de status van de response OK is, dan ligt het waarschijnlijk aan het uitlezen van de responsedata (JSON, CSV of XML). Probeer in PyCharm door de code te lopen met breakpoints, om te kijken waar het probleem zit.

Krijg je geen verbinding met de API? Check dan of je wel een API key (eventueel inclusief encoding) hebt meegestuurd, en dat je internetverbinding goed werkt.

Groepsproblemen

Eén van de teamleden doet zijn/haar werk niet, en lift mee op de rest.

Zorg dat individuen verantwoordelijk zijn voor hun eigen taken. Maak afspraken over wie welke taak op zich neemt, en zorg dat iedereen daarmee akkoord gaat. Als iemand zich dan niet aan de afspraak houdt, dan kun je diegene daarop aanspreken.

Als je als team vaak genoeg afspreekt (minstens elke dag), dan zorg je er ook voor dat je die persoon kunt aanspreken op zijn/haar gedrag. Als iemand zich niet houdt aan de afspraken en niet komt opdagen bij de vergaderingen, dan leg je dit vast in het verslag.

Een teamlid weigert het teamcontract te tekenen.

Samenwerken, taken verdelen en plannen behoort tot de leerdoelen van deze cursus. Indien een teamlid hier niet mee akkoord gaat, dan weigert deze de cursus volledig te volgen. Dit teamlid mag zich melden bij de docent, die het teamlid (indien nodig) verwijst naar de propedeusecoördinator Hans van der Meer (hans.vandermeer@hu.nl).

Het team heeft bepaald dat iemand een waarschuwing krijgt, wat nu?

Communiceer als team de waarschuwing met het betreffende teamlid. Documenteer de waarschuwing door deze ook via email te versturen.

Een teamlid reageert niet binnen de afgesproken tijd na drie waarschuwingen te hebben ontvangen.

Het teamlid wordt automatisch uit de groep verwijderd. Het resterende team neemt contact op met de docent om de drie waarschuwingen te laten zien. De docent zal het teamlid vervolgens officieel op de hoogte stellen.

Een teamlid stelt een oplossing voor na drie waarschuwingen te hebben ontvangen, maar het team gaat niet akkoord.

Probeer er nogmaals uit te komen. Als dit niet lukt, neem dan contact op met de docent om toestemming te vragen om het teamlid uit het team te verwijderen.

Ik ben uit het team gezet, wat nu?

Het is praktisch onhaalbaar om de opdracht individueel af te ronden. Je wordt daarom verwezen naar de volgende mogelijkheid om het Miniproject te doen (herkansing).

Eén van de teamleden overheerst de discussie of beslist alles/ teveel.

Zorg dat iedereen aan het woord komt. Het is handig om daarbij telkens 'het stokje' (bijv. een pen) over te dragen aan ieder teamlid totdat iedereen aan het woord is geweest of zijn/haar mening heeft gegeven. Probeer iedereen naar zijn/haar mening te vragen, dat voorkomt dat mensen met een gefrustreerd gevoel achterblijven.

Er is niks mis mee wanneer één iemand de leiding neemt, maar het is niet de bedoeling dat deze persoon beslissingen neemt zonder daarbij de mening van de anderen te hebben geraadpleegd. Zorg daarom dat je van tevoren afsprekt dat bij grote beslissingen ofwel 1) iedereen akkoord moet gaan, of 2) de meerderheid akkoord moet gaan. Als iemand niet akkoord is, vraag dan waarom, want als je weet waar het verzet vandaan komt, dan kun je aan een oplossing werken.

Een teamlid komt niet of te laat opdagen.

Spreek het teamlid hierop aan op een constructieve manier. Vraag waarom deze niet is op komen dagen, terwijl er toch van tevoren afspraken zijn gemaakt.

Heeft het teamlid zich verslapen? Vraag wat de reden is. Er kunnen namelijk ook andere redenen dan luiheid meespelen: Er zijn ICT'ers die graag 's nachts werken/studeren, omdat het dan rustiger is, maar dat gaat wel ten koste van hun slaap. Kijk of je iets later in de ochtend kunt afspreken, als de voortgang van het project dat toelaat.

Heeft iemand andere verplichtingen, zoals een bijbaan/zorg voor familieleden?

Kijk of je het teamlid op een andere manier kunt betrekken bij vergaderingen, door op een ander moment af te spreken. Of houd elkaar op de hoogte van gemaakte afspraken/ taken die worden gedaan.