Algoritme voor BDICT

Computational Thinking (CT) vragen doen een beroep op de volgende denkvaardigheden:

* Decompositie: deel het probleem op in kleinere stukken/stappen
* Patroonherkenning: herken patronen in gegevens en data
* Abstractie: herleid vraagstelling tot de kern van het probleem
* Algoritme: bedenk een logische volgorde van stappen om tot het antwoord te komen

We moeten een elektronische leeromgeving (ELO) maken waar docenten vragen kunnen uploaden en waar studenten deze vragen kunnen beantwoorden en ervan leren.

Voor elke vraag moet opgegeven worden voor hoeveel punten iedere denkvaardigheid wordt aangesproken. Dit wordt gebruikt om aan te geven op welke denkvaardigheid de vraag het meeste betrekking heeft, maar ook op welk niveau deze denkvaardigheden aangesproken worden.

Het maximaal aantal gescoorde punten per denkvaardigheid wordt opgeslagen als de highscore van de leerling. Op basis van deze score wordt bepaald welke vragen herhalend zijn en welke verdiepend.

De student kan dus opgeven of hij herhalende vragen wil doen of verdiepende. Bij herhalende vragen kan hij geen nieuwe highscore halen; de moeilijkheid van alle 4 de denkvaardigheden ligt onder zijn highscore. Als hij ‘verdiepen’ aanklikt krijgt hij vragen te zien waarmee hij wél een nieuwe highscore kan halen. Ook kan de leerling naast herhalen of verdiepen aangeven of hij een specifieke denkvaardigheid wil herhalen of verdiepen of dat het hem niet uitmaakt op welke denkvaardigheid de vraag het meeste betrekking heeft. Het filteren en sorteren van de vragen gebeurt op basis van de input op de volgende manieren:

|  |  |  |
| --- | --- | --- |
| Input | Filteren | Sorteren |
| Herhalen op alle vaardigheden | Zoek alle vragen waarbij geen nieuwe highscore gescoord kan worden | Sorteer deze op basis van het totaal aantal punten voor de vraag (dit sorteert op algehele moeilijkheid) |
| Herhalen op specifieke vaardigheid | Zoek alle vragen waarbij geen nieuwe highscore gescoord kan worden | Sorteer deze op basis van het aantal punten voor de specifieke vaardigheid |
| Verdiepen op alle vaardigheden | Zoek alle vragen waarbij ergens een nieuwe highscore gevestigd kan worden | Bereken van elke vraag hoeveel alle highscores in totaal opgehoogd kunnen worden en sorteer deze van klein naar groot (kleinere totale ophoging eerst) |
| Verdiepen op specifieke vaardigheid | Zoek alle vragen waarbij een nieuwe highscore gevestigd kan worden in de aangegeven categorie én waarbij de hoogste ophoging (mede) in de aangegeven categorie ligt (anders hoort de vraag bij het verdiepen in een andere categorie) | Sorteer deze vragen op basis van de kleinste ophoging voor de desbetreffende categorie |

De gefilterde en gesorteerde vragen worden op een rijtje gezet voor de gebruiker en de moeilijkheid wordt in absolute punten getoond.

Als het programma geïntegreerd wordt in de ELO kan ook het percentage berekend en aangegeven worden. Hiermee kan je snel zien voor hoeveel procent iedere vraag betrekking heeft op iedere denkvaardigheid. Ook kan de invoer dan gedaan worden in procenten. Hiermee (en met één datapunt in punten) wordt berekend voor hoeveel punten de percentages staan.

Handleiding

Het programma leest een CSV bestand (Opgaven.csv) in en schrijft het resultaat weg naar Result.csv en naar de terminal. Het programma (Main.py) moet je runnen met Python vanuit dezelfde map als waar deze bestanden in staan. Er is ook nog een Opgaven.xlsx bestand. Deze kun je gebruiken om nieuwe scores voor elke vraag in te stellen of te laten genereren door Excel. Momenteel genereert Excel in dit bestand een random waarde tussen 0 en 100 voor elke score bij elke vraag. Excel rekent ook automatisch de totaalscore voor elke vraag voor je uit. Deze totaalscore wordt door het programma ook gebruikt. Als je dus de scores voor vragen in het Opgaven.csv bestand bewerkt moet je ook handmatig de totaalscore erin aanpassen, anders werkt het programma niet correct meer.

In Opgaven.csv kun je zonder problemen je highscore aanpassen. Doe dit als je geen nieuwe waardes door Excel wilt laten genereren maar juist de huidige waarden wilt bewaren met een nieuwe highscore. De highscores staan nu op 75 omdat je dan makkelijk kunt narekenen dat het programma correct werkt. Pas de highscores (in Opgaven.csv) aan naar de waarden voor een bepaalde vraag als je wilt weten hoe het programma zou werken als/nadat je deze vraag correct beantwoord zou hebben.

Je kunt ook onderaan het Opgaven.xlsx bestand vragen toevoegen als je wilt. Dit bestand moet je na het bewerken opslaan als CSV bestand. Kies de bovenste optie voor CSV in het opslaan als menu van Excel. Dit bestand moet Opgaven.csv heten. Belangrijk is dat de waarden met ; worden gescheiden en dat de vragen en scores in dezelfde kolommen/rijen zijn blijven staan.

Testrapport

Het programma is niet modulair geschreven dus leent zich niet voor unit testing en integratie testing. Ik heb uitvoerig systeemtesten uitgevoerd om na te gaan dat het programma correct werkt bij alle mogelijke verschillende inputs. Het programma werkt correct bij alle mogelijke verschillende inputs. De juiste vragen worden gefilterd en deze worden (zoals beschreven in het algoritme) op de juiste manier gesorteerd, ongeacht wat je als input geeft qua highscore en wat je wilt dat het programma doet.