

# Tafels leren

*Je gaat opnieuw een Python programma testen.*

## Tafels leren

Met het Python-programma tafels leren .

Use cases

- Als een student wil ik mijn tafels oefenen binnen een bepaalde tijdslimiet, zodat ik mijn snelheid en nauwkeurigheid bij het vermenigvuldigen kan verbeteren."

*Het programma heeft een tijdslimiet. Bij elke nieuwe som wordt de tijdslimiet getoond.*

- "Als een student wil ik in staat zijn om de moeilijkheidsgraad van mijn oefeningen te verhogen, zodat ik kan blijven uitdagen en verbeteren van mijn rekenvaardigheden."

*Na elke drie goede antwoorden wordt het niveau verhoogd en worden de vragen moeilijker.*

- "Als een student wil ik via een game-achtig programma worden uitgedaagd, zodat ik ga proberen een betere score dan te halen en zodoende motivatie krijg om meer te oefenen. "

*Het 'spel' telt punten en geeft aan op welk niveau je bent uitgekomen. De punten kunnen worden vergeleken met anderen.*

- "Als een student wil ik dat dezelfde vraag niet twee keer achter elkaar wordt gesteld, zodat ik een breder scala aan vragen kan ervaren en mijn vermogen om verschillende vermenigvuldigingsproblemen op te lossen kan verbeteren."

*Het programma zou niet twee maal dezelfde vraag achter elkaar moeten stellen.*

## Opstellen testplan

Maka een testplan zoals je dat hebt geoefend.

Maak de volgende hoofdstukken:

1. **Inleiding**,
2. **Use cases**.
3. **Scenario's** per use case.
4. **Stappenplan** per scenario.
5. Voer je stappenplan uit.
6. Beschrijf je **bevindingen** per scenario.
7. Beschrijf de algehele **conclusie**.

## Inleveren

Volledig testplan in pdf.

## Code

```
import time
import random

# De tijd wanneer het spel start
start_time = time.time()

# Tijdslimiet voor het spel in seconden
time_limit = 20

# Teller voor het aantal goede antwoorden
correct_count = 0

# Bepaal het initiële bereik voor de getallen
number_range = 5

# Bewaar de laatste vraag
last_question = None

while True:
    # Bereken hoeveel tijd er al verstreken is
    elapsed_time = time.time() - start_time
```

```
# Als de tijdslimiet is bereikt, stop dan met het spel
if elapsed_time > time_limit:
    print("Tijd is op!")
    print(f"Je hebt {correct_count} vragen goed beantwoord.")
    print(f"Je eindigde op moeilijkheidsgraad {number_range - 4}.")
    break

# Bereken de resterende tijd en print deze
remaining_time = time_limit - elapsed_time
print(f"Resterende tijd: {remaining_time:.2f} seconden")

# Verhoog het bereik elke keer dat de gebruiker vijf vragen correct beantwoordt
if correct_count % 3 == 0 and correct_count > 0:
    number_range += 1
    print(f"Moeilijkheidsgraad verhoogd! Je zit nu op moeilijkheidsgraad {number_range - 4}.")

# Genereer een nieuwe vraag
num1, num2 = random.randint(1, number_range), random.randint(1, number_range)

# Controleer of de nieuwe vraag hetzelfde is als de vorige vraag
# Als dat het geval is, genereer dan een nieuwe vraag
while (num1, num2) == last_question:
    num1, num2 = random.randint(1, number_range), random.randint(1, number_range)

# Bewaar de huidige vraag als de laatste vraag
last_question = (num1, num2)

correct_answer = num1 * num2

# Stel de vraag aan de gebruiker
user_answer = int(input(f"Wat is {num1} * {num2}? "))

# Controleer het antwoord
if user_answer == correct_answer:
    print("Correct!")
    correct_count += 1
else:
    print("Fout! Het juiste antwoord was: ", correct_answer)
```

---

Revision #3

Created 14 July 2023 10:15:52 by Max

Updated 14 July 2023 10:29:56 by Max