

Proefpracticum Methodiek van de Informatica

masterMind

14 februari 2019

1 Praktische Richtlijnen

De bedoeling in dit proefpracticum is een **werkend** programma te implementeren waarin de tot nu toe behandelde leerstof i.v.m. functies, variabelen, controle- en lusstructuren praktisch zal toegepast worden.

Voor dit proefpracticum moet je enkel je broncode indienen. Aanwijzingen om de broncode in te dienen vind je weldra op Toledo.

Indien je beslist mee te doen aan dit proefpracticum (niet verplicht, maar wel absoluut aan te raden), moet je code ingediend worden ten laatste **zondag 17 maart 2019 om 13u00**.

1.1 Groepswork

Het proefpracticum mag in groepjes van 2 of individueel¹ gemaakt worden.

Het is voldoende om éénmaal je code in te sturen per groepje via Toledo.

Lees ook zeker de gedragscode na op Toledo. Op plagiaat staan zware sancties.

1.2 Bevraging

Dit proefpracticum wordt klassikaal besproken in de week van 18 maart. **Breng een afgeprinte versie van je code (MastermindGiven en MastermindAsked) mee naar de feedbacksessie.**

1.3 Geschatte inspanning

De inspanning voor dit practicum wordt **per persoon** als volgt geschat.

Doornemen opgave	1u
Uitdenken van de oplossing	5u
Implementatie	9u
Totaal	15u

¹Het examenpracticum zal individueel gemaakt worden.

1.4 Vragen en tips

Als er vragen zijn over dit practicum of de opgave, stel ze dan op het gepaste forum op Toledo. Zo zien ook je collega's je vraag en het bijhorend antwoord. Mogelijk wordt er nog aanvullende informatie toegevoegd op Toledo. Hou dit dus zeker ook in de gaten.

2 Achtergrondinformatie

Brainteaser spelletjes hebben al vaker hersencellen op hol gejaagd tijdens hun zoektocht naar de beste, de snelste, of de kortste oplossing voor een gegeven probleem. Mastermind is een leuk voorbeeld van zo'n denkspelletje. Het werd rond 1970 uitgevonden door de Israëliër Mordecai Meirowitz. Hoewel aanvankelijk alle grote spelletjesfabrikanten zijn idee maar niets vonden, hield hij toch hardnekkig vol, tot hij uiteindelijk terecht kwam bij een klein Engels bedrijfje dat bereid was het spelletje op de markt te brengen. Het kende onmiddellijk een wereldwijd succes. Men vond het zo goed dat het in 1973 de Game of the Year Award kreeg, en tal van andere awards volgden in de jaren daarop. Tot op heden werden er meer dan 55 miljoen exemplaren van het Mastermind spelletje verkocht in meer dan 80 landen.

Hoe wordt dit spel gespeeld? De spelregels zijn heel eenvoudig. Het doel is een geheime code te raden die bestaat uit een combinatie van een aantal gekleurde schijfjes. Je mag een beperkt aantal keer raden, en na elke gok krijg je informatie terug over de kwaliteit van je antwoord. Deze informatie bestaat uit zwarte en witte schijfjes. Een zwart schijfje betekent dat een schijfje van je antwoord de juiste kleur heeft en dat zijn positie overeenkomt met de positie van het schijfje in de geheime combinatie. Een wit schijfje betekent dat een schijfje uit je gok de goede kleur heeft, maar dat zijn positie niet overeenstemt met de positie van die kleur in de geheime combinatie.

Misschien kan een klein voorbeeldje deze spelregels verduidelijken. Stel dat je geheime code bestaat uit de volgende reeks van schijfjes: rood, geel, blauw, rood (voor de speler is deze combinatie uiteraard niet gekend). Je doet een eerste gok: groen, rood, blauw, geel, en als antwoord krijg je: zwart, wit, wit. Nu weet je dat je drie goede kleuren hebt gegokt (als speler weet je niet welke), waarvan er eentje zich op de juiste positie bevindt. De juiste kleuren zijn uiteraard rood, blauw en geel, waarbij het de blauwe is die zich op de juiste positie bevindt. Na enkele gokjes kan de speler de informatie uit al zijn vorige gokken gebruiken om zo vlug mogelijk de geheime code te ontcijferen. Als aanvulling geven we jullie nog enkele interessante links mee naar volgende site die een aardige historische beschrijving van het spelletje en interessante oplossingsstrategieën geeft: <http://www.tnelson.demon.co.uk/mastermind/>

3 Opgave

Praktisch wordt er van jullie gevraagd Mastermind te implementeren waarbij men tegen de computer kan spelen. De computer kiest een willekeurige combinatie uit, en de speler moet deze combinatie raden. Een combinatie bestaat uit vier posities. De speler kan kiezen uit 6 kleuren. Er worden 10 kansen gegeven om de juiste combinatie te vinden. Het spel eindigt van zodra de juiste combinatie gevonden wordt, of als de 10 kansen verspeeld werden. In het eerste geval wordt de speler gefeliciteerd, en in het laatste geval wordt aan de speler bekend gemaakt wat de gezochte combinatie was.

Let op: Je code moet compatibel zijn met Python 3.

Om mastermind grafisch voor te stellen, mag je gebruik maken van de Tkinter package waarvan de documentatie beschikbaar is op <http://docs.python.org/2/library/tkinter.html>. Aangezien deze code vrij complex is, krijg je deze grotendeels van ons (in *MastermindGiven.py*).

3.1 Checklist

Hieronder volgt een beknopt stappenplan voor het practicum. De uitgebreide beschrijving volgt in de volgende secties. De door ons voorziene tests zijn opgesteld in volgorde van het stappenplan, dus het is aangeraden om deze volgorde strikt te respecteren.

1. implementeer en test *get_nb_black_white_matches*
2. implementeer en test *create_combination* *
3. implementeer *create_empty_circles* en test het resultaat via de GUI *
4. implementeer en test *any_color_in_combination*
5. implementeer en test *all_colors_in_combination*
6. implementeer en test *is_sublist_of*

* nadat je stappen 2 en 3 hebt geïmplementeerd, kan je de GUI gebruiken

Optioneel:

1. Voeg een knop “*sublist*” toe en test deze via de GUI
2. Voeg een knop “*hint*” toe en test deze via de GUI

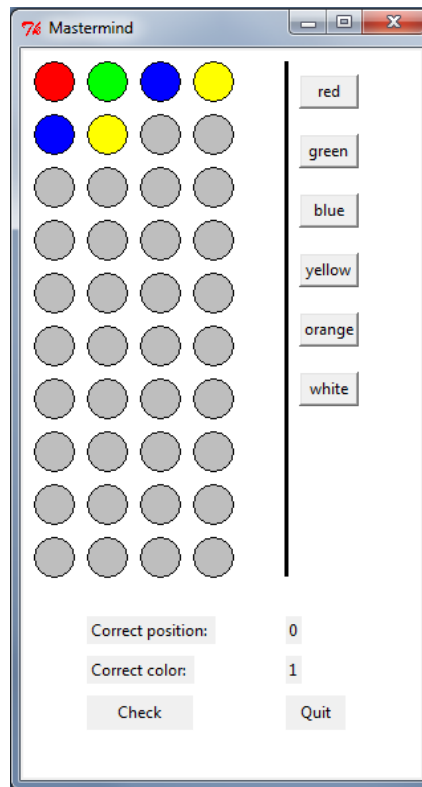
3.2 Basis functionaliteit

Het programma zal bij elke controle van een gok moeten nagaan hoeveel juiste posities er gevonden zijn (zwarte schijfjes), en hoeveel juiste kleuren op foute posities (witte schijfjes). Deze functie, *get_nb_black_white_matches(given,guess)*, moet je uitwerken. De functie moet een tuple retourneren dat bestaat uit het aantal juiste posities (zwarte schijfjes) op de eerste plaats en het aantal juiste kleuren op een foute positie op de tweede.

Een aanzet (de functie-hoofding) is reeds gegeven in *MastermindAsked.py*. Je mag uiteraard deze functie nog uitbreiden met extra hulp-functies.

Ook voorzie je een functie *create_combination(nb_elements)* die een willekeurige combinatie kleuren genereert bestaande uit *nb_elements* elementen.

Je krijgt van ons een aantal testen, maar om de werking van je code volledig na te gaan, verwachten we dat je voor elke functie die je definieert ook zelf een reeks **testen** schrijft.



Figuur 1: Mastermind GUI - grafische user interface voor mastermind

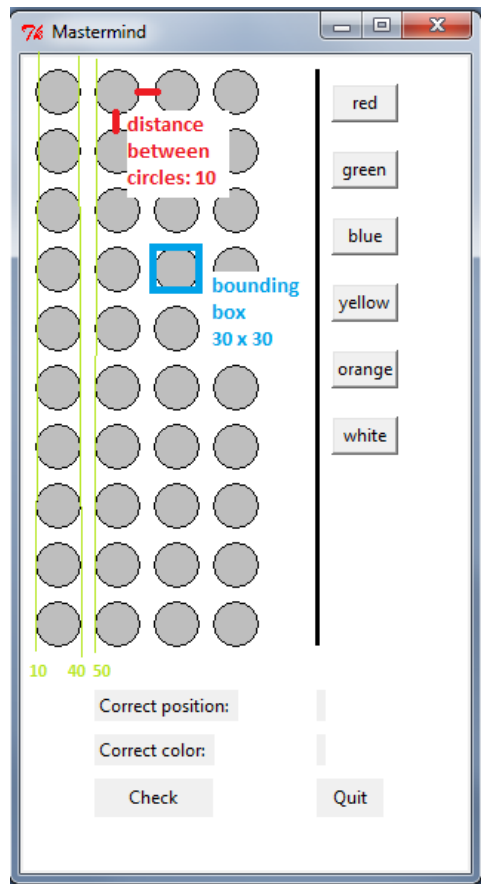
3.3 Grafische user interface (GUI)

De voornaamste functie die instaat voor de grafische user interface (GUI), is *draw_board* (in *MastermindGiven.py*). Deze functie krijg je integraal. Deze functie zal, zoals geïllustreerd in Figuur 1 een scherm tonen met een aantal cirkels die in de loop van het spel de gokken van de speler zullen tonen. Rechts op het scherm worden de verschillende kleuren getoond die de speler kan kiezen. De speler kan een kleur kiezen, door op de bijhorende knop te klikken. Onderaan wordt na elke gok weergegeven hoeveel juiste posities er waren en hoeveel juiste kleuren op foute posities. Een gok wordt gecontroleerd wanneer de speler op de check-knop klikt. Het spel kan beëindigd worden door op de quit-knop te klikken.

In *MastermindAsked* moet je enkel nog zorgen voor de functie *create_empty_circles* die, bij het opstarten van het spel, de lege cirkels tekent en opslaat. Zoals weergegeven in Figuur 2, zal je 10 rijen moeten tekenen die telkens bestaan uit 4 cirkels. Een cirkel kan getekend worden door de functie *create_oval(x1, y1, x2, y2)* op te roepen. De parameters stellen de coördinaten van de linkerbovenhoek en rechteronderhoek van de omhullende rechthoek voor. Je mag veronderstellen dat de zijden van deze omhullende rechthoek steeds 30pixels zijn. Tussen alle cirkels, en tussen elke cirkel en de rand van het scherm zijn er steeds 10 lege pixels. Merk op dat de (0,0)-coördinaat zich bovenaan links in het scherm bevindt en dat de waarden van de y-as verhogen indien er naar beneden wordt gegaan.

Wanneer je bijvoorbeeld de tweede cirkel wil tekenen voor de derde gok, kan dit met volgende code:

```
canvas.create_oval(50, 90, 80, 120)
```



Figuur 2: Teken van lege cirkels in de mastermind GUI

Tijdens het spel worden deze cirkels opgevuld met de geraden kleuren. Om dit mogelijk te maken, worden de cirkels opgeslagen. Je functie retourneert daarom een geneste lijst (matrix): `[[circle1_guess1, circle2_guess1, ...], [circle1_guess2, ...], ... [circle4_guess10]]`. Zo kan je de tweede cirkel van de derde gok bijvoorbeeld opslaan met volgende code:

```
ovals[2][1] = canvas.create_oval(50, 90, 80, 120)
```

Ook kan je, in plaats van de verzameling cirkels te beschouwen als een matrix, deze bekijken als een verzameling lijsten (1 lijst per gokbeurt).

```
ovals[2].append(...)
```

3.4 Extra functies

Als speler is het interessant om een hulplijn te kunnen invoeren tijdens het spel. Zo wil een speler mogelijk weten of een bepaalde kleur of verzameling kleuren voorkomt in de oplossing. Implementeer daarom onderstaande functies. **Merk op, je hoeft deze enkel in *Mastermind_asked* te implementeren. Een uitbreiding wordt behandeld in sectie 4. Daar zullen we enkele functies oproepen via de grafische interface.**

- *any_color_in_combination(colors, given)*: Bevat de verzameling colors minstens 1 kleur die ook voorkomt in de geheime combinatie?
- *all_colors_in_combination(colors, given)*: Behoren alle kleuren in de verzameling colors tot de geheime combinatie?
- *is_sublist_of(sublist, given)*: Behoort de sublist (in volgorde) tot de geheime combinatie?

4 GUI Uitbreidingen

Wanneer je nog tijd over hebt, kan je ook volgende functionaliteit implementeren. Deze uitbreiding mag enkel uitgewerkt worden wanneer bovenstaande functionaliteit reeds volledig geïmplementeerd is.

Merk op: de variabelen die je in MastermindGiven definieert buiten een methode, zijn aanroepbaar binnen elke functie in het programma. Wanneer je deze variabelen echter wil wijzigen, moet je, in de functie waarin verandering zal gebeuren, deze variabelen declareren als “global”. Zoals bijvoorbeeld gebeurde in de draw_board functie:

```
def draw_board(root):
    ...
    # indicate which variables of the program will be altered
    # in this function
    global canvas, ovals, correct_position_label, correct_color_label
    # alter the global canvas variable
    canvas = ...
```

4.1 Uitbreiding 1: is_sublist_of functie in GUI

Voeg een knop “sublist” toe die voor de huidige ingegeven combinatie (dit hoeft geen volledige combinatie te zijn, maar kan bestaan uit 1, 2, 3, of 4 vakjes) meldt indien deze een sublist is van de geheime combinatie. Je kan hiervoor de functie *is_sublist_of* gebruiken die je reeds hebt geschreven. Concreet voeg je dus een functie (bv. *is_sublist*) toe aan *MastermindGiven*, en zorg je dat deze gekoppeld is aan de sublist-knop die je ook moet voorzien. Zorg achteraf ook dat de huidige rij (waarin nu de kleuren van je “sublist” staan weergegeven) in je gebruikersinterface terug wordt leeggemaakt. Toon het resultaat in een bericht (‘is sublist’ of ‘is geen sublist’). Als de gebruiker een combinatie van 0 elementen probeert te controleren, toon je een foutboodschap.

TIP: onderzoek in de code van mastermindGiven hoe de “check” en “quit” knop toegevoegd worden en gekoppeld worden aan hun functie. In deze code vind je ook hoe je berichten kan tonen aan de gebruiker.

4.2 Uitbreiding 2: “Hint” functionaliteit

Als tweede uitbreiding implementeer je de hint functionaliteit. Concreet moet je (in *MastermindGiven.py*) ten eerste de GUI uitbreiden met een knop “Hint”. De plaats van de “Hint”

knop mag je zelf bepalen. Als men op de “Hint” knop klikt, moet er een functie opgeroepen worden die een boodschap toont aan de gebruiker waarin als tip voor een willekeurige positie de juiste kleur wordt onthuld (bv. “Color at position 3 is yellow”). Let op, men mag slechts 1 hint per spel krijgen. Indien er dus meerdere keren op “Hint” geklikt wordt, moet er een gepaste foutmelding op scherm komen.

Tenslotte kijk je na of je code ook zou werken indien bepaalde constanten een andere waarde zouden krijgen.

Heel veel succes!

Het MI-team