

Informatica — Universiteit Leiden

Toptoets — woensdag 4 juni 2008

We gaan vanmiddag *programmeren*. Dat doen we niet op een gewone computer met Windows, maar met *Linux*. We gebruiken de *programmeertaal* C++. We beginnen met het maken van een lopende grafische klok. Daarna gaan we werken aan een programma dat met *Kunstmatige Intelligentie* (AI) te maken heeft: het programmeren van het spel Vier-op-een-rij.

Hoe programmeer je een grafische klok in Linux?

We gebruiken Linux. Klik linksonder op het groene beest, en open een *terminal* window. Hierin kun je Linux-commando's typen. Tik letterlijk het volgende commando in:

```
cp -R /home/kosters/Klok .
```

Let hierbij op de punt aan het eind. Druk op de Enter-toets na het commando. Hiermee kopieer je een complete map (oftewel directory), *Klok* geheten, met inhoud en al naar je “eigen” *home-directory* — zoals dat heet. Daar kun je vervolgens zelf met de map aan de slag. Ga naar je nieuwe directory *Klok* met

```
cd Klok
```

en kijk wat er in zit met

```
ls -l
```

Als het goed is, zie je vijf files (bestanden): *Xsowl.cc*, *Xsowl.h*, *Xsowldef.h*, *klok.cc* en *makefile*. We gaan straks het C++-programma *klok.cc* zelf aanpassen met

```
nedit klok.cc &
```

Let op het *&* (“ampersand”) symbool! Met dit Linux-commando open je in een nieuw window een *editor* waarmee je een bestand kunt bewerken, in dit geval *klok.cc*. Dit bestand bevat C++-“code” voor een klok. Voer de volgende drie stappen uit om te bekijken wat er gebeurt als de Linux-computer deze C++-code uitvoert:

- klik op het witte terminal venster, en geef Enter;
- type daar het commando `make` (zonder `&!`) in;
- type —als `make` klaar is— het commando `./klok &` in; het mag ook zonder `&` (wat zou het verschil zijn?); mag het op jouw computer ook zonder `./?`

Deze zelfde stappen moet je doen telkens als je iets gewijzigd hebt in `klok.cc`; denk aan het tussentijds saven. Het proces dat je hier steeds uitvoert noemen we wel *compileren*.

Deze klok is nog niet erg boeiend. Je kunt er nu wat tijd eraan besteden om hem aantrekkelijker te maken — bijvoorbeeld een secondewijzer maken en deze laten lopen.

Je kunt de volgende dingen proberen:

- Vervang de tekst “Gezien hoe laat het is?” in de C++-code door een andere tekst. De kleuren en lettertypes in de editor zijn behulpzaam bij het “begrijpen” van de C++-code. Bewaar de veranderingen door “Save” te kiezen in het “File” menu. Voer opnieuw het `make` commando uit zoals hierboven beschreven en kijk wat de klok nu doet.
- In de code staat twee keer een regel zoals deze:

```
tekenwijzer (venster,xcentrumklok,ycentrumklok,
             langewijzer,min * 6,black);
```

Vervang de kleur `black` eens door een andere kleur (`red`, `green`, `yellow`, ...) en kijk wat het programma nu doet. Wat gebeurt er als je de kleur `white` gebruikt?

- Maak een secondewijzer, bijvoorbeeld een groene. Wijzig voor dit punt en het volgende voornamelijk op de plekken die gemarkeerd zijn met `//TODObegin//` en `//TODOeind//`. (Alles op een regel na `//` is overigens commentaar in C++.)
- Laat de secondewijzer nu netjes lopen: haal de “oude” steeds weg.

Het spel Vier-op-een-rij

We gaan nu het spel *Vier-op-een-rij* programmeren — en wel met Kunstmatige Intelligentie.

Ga eerst terug naar je eigen home-directory met

```
cd
```

Kopieer vervolgens de juiste files met

```
cp -R /home/kosters/Spel .
```

en werk verder in de directory `Spel`, met `cd Spel`. Speel eerst maar eens met het spel (compileren met dezelfde stappen als bij de klok, aanroepen met `./spel &`). Je kunt voor iedere zet kiezen uit ingebouwde strategieën of zelf een zet in een kolom doen — door op een van de cijfers te klikken met de muis.

Er zijn drie strategieën voorgeprogrammeerd:

random doet een willekeurige zet

recursie rekent 8 zetten diep door, en doet dan de “beste” zet; er wordt van *recursie* gebruik gemaakt

slim kijkt met behulp van een “evaluatie-functie” één zet vooruit

Je kunt de verschillende strategieën zelf tegen elkaar laten spelen. Wij gaan die derde strategie proberen te snappen en zullen die ook verbeteren.

Zoek in de C++-code (met `ncedit spel.cc &`) weer naar `//TODObegin//` en `//TODOeind//`, en wijzig daar naar believen. Probeer eens:

- Speel wat met het spel, wijzig de kleuren, etcetera.
- Zoek de “evaluatie-functie” als op in `spel.cc`. Verander wat aan de voorkomende getallen 5, 50 en 1000. Wat gebeurt er als je sommige 0 maakt? Wat zouden ze betekenen?
- Laat “slim” eens tegen “slim” spelen. Wat valt je op? Los dit probleem op. Tip: een goede zet voor de tegenstander is misschien een goede zet voor jezelf.

dr. Walter A. Kosters, Informatica, Universiteit Leiden

<http://www.liacs.nl/home/kosters/>