

## PRE-Classes Informatica, 24.1.2017

### “Laat de klok lopen en speel een spel”

Een informaticus moet zeker kunnen programmeren. In het eerste semester van de studie krijgen studenten daarom het vak *Programmeermethoden*. Ze leren hier programmeren in de programmeertaal C++. Er wordt daarbij ook gebruik gemaakt van computers die andere besturingssystemen draaien dan Microsoft's Windows: de zogeheten *UNIX-systemen*. Je maakt nu kennis met C++ en met *Linux*, een gratis UNIX-variant voor PC's. We zullen ook aandacht besteden aan het werken in een grafische omgeving.

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 UNIX?

Log eerst in op een computer in **zaal 302**. Zorg ervoor dat je een Linux-inlogscherf ziet; mocht het een Windows-inlogscherf zijn, start de computer dan opnieuw op. Geef de “user name” (staat op het briefje, bijvoorbeeld `proefs551`) en het “password” (staat ook op het briefje, bijvoorbeeld `hyWw1o1%`; let op eventuele hoofdletters, het verschil tussen het cijfer 1 en de letter l, tussen het cijfer 0 en de hoofdletter O, etcetera), en druk steeds op de Enter-toets. Na enige tijd krijg je een scherm met iconen en dergelijke. Zoek links, bijvoorbeeld bij “Dash Home”, naar “terminal”, en open een *terminal*. Hierin kun je UNIX-commando's typen. Tik eerst `cd`, gevolgd door Enter; je zit nu in je home-directory (zie straks).

Maak een nieuwe map = directory met

```
mkdir Klok
```

Druk op de Enter-toets na het commando. Nu kun je zelf met de inhoud van de map aan de slag. Let er overigens op dat in de UNIX-wereld veel met slashes (/) gewerkt wordt, terwijl het in de Windows-wereld vaak backslashes (\) zijn. Open vervolgens een browser (Firefox of Chrome) en ga naar

[www.liacs.leidenuniv.nl/~kosterswa/proef/](http://www.liacs.leidenuniv.nl/~kosterswa/proef/)

Zet de vijf files (met rechter muisknop erop klikken, etcetera) `Xsowl.cc`, `Xsowl.h`, `Xsowldef.h`, `klok.cc` en `makefile` in de map Klok.

Ga in een terminal 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: `Xsowl.cc`, `Xsowl.h`, `Xsowldef.h`, `klok.cc` en `makefile`. We gaan nu het C++-programma `klok.cc` zelf aanpassen met

```
gedit klok.cc &
```

Let op het `&` (“ampersand”) symbool! Met dit UNIX-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 computer deze C++-code uitvoert:

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

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. Als je te zijner tijd klaar bent, kun je “uitloggen”: klik op “Logout” bij de knop rechtsboven. 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 achter `//` is overigens commentaar in C++.)
- Laat de secondewijzer nu netjes lopen: haal de “oude” steeds weg. Hoe?
- Probeer de klok mooier te maken. Zo teken je met

```
venster.DrawLine (0,0,100,100);
```

een lijn vanaf het punt met coördinaten (0,0) naar het punt met coördinaten (100,100). Wellicht uurstreepjes zetten?

- Bij het tekenen van een wijzer wordt gebruik gemaakt van de *variabelen* `xcentrumklok` en `ycentrumklok`. Deze verwijzen naar de coördinaten van het centrum van de klok op het scherm. Met het commando

```
xcentrumklok = xcentrumklok + 10;
```

wordt dit centrum 10 beeldpunten naar rechts opgeschoven. Voeg deze regel direct na de regel met `sec = pointer->tm_sec;` toe en kijk wat het programma nu doet. Kun je de klok misschien rondjes laten lopen? Hint: ook

```
xcentrumklok = (int)( 200 + 100*sin (sec*M.PI/30) );
```

is een geldige “expressie”!

## 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
```

Maak vervolgens een map `Spel` en zet daar de betreffende vijf files in (waaronder `spel.cc`) van

[www.liacs.leidenuniv.nl/~kosterswa/proef/](http://www.liacs.leidenuniv.nl/~kosterswa/proef/)

Werk verder in de directory `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 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.

## Slotopmerkingen

Wil je dit zelf thuis doen? Haal de gratis Linux-omgeving *Ubuntu* op van de website [www.ubuntu.com](http://www.ubuntu.com) en gebruik de files van het practicum. Leuke links:

[www.liacs.leidenuniv.nl/~kosterswa/pm/](http://www.liacs.leidenuniv.nl/~kosterswa/pm/) (over C++)  
[www.liacs.leidenuniv.nl/~kosterswa/proest.pdf](http://www.liacs.leidenuniv.nl/~kosterswa/proest.pdf) (over AI)

dr. Walter A. Kosters, Informatica, Universiteit Leiden  
met dank aan dr. Siegfried Nijssen