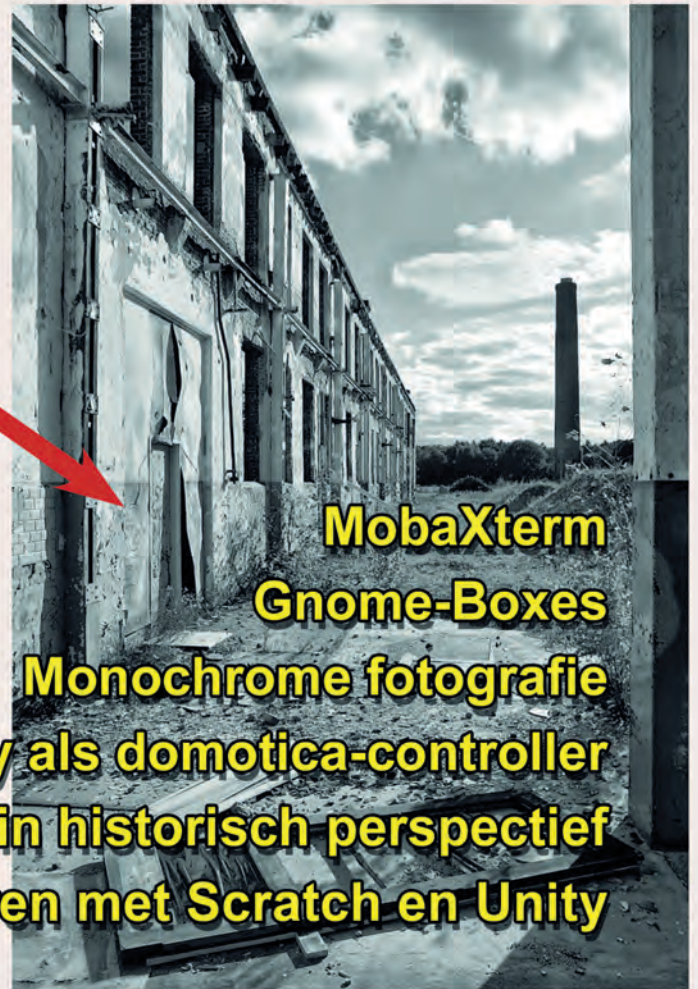


SoftwareBus



hcc!CompUsers
systemen en applicaties

2022 **1**

Officieel orgaan van **hcc!CompUsers** interessegroep

Inhoud

- 1 Voorpagina
- 2 Bij de voorplaat
- 2 Voorwoord
 - Isja Nederbragt
- 3 Afbeeldingen verkleinen
 - Joep Bär
- 4 Colofon
- 5 BASIC in historisch perspectief
 - Hans Lunsing
- 8 Gnome Boxes
 - Ton Valkenburgh
- 12 Homey als domotica controller
 - Gjalt Zwaagstra
- 15 Scratch (15)
 - René Suiker
- 19 MobaXterm
 - André Reinink
- 23 Monochrome fotografie
 - Wessel Sijl
- 27 Firefox isoleren in een Linux cgroup
 - Johan Swenker
- 29 Games maken (3)
 - René Suiker

Bij de voorplaat

Wintermaanden... korte dagen... lange avonden. Bij uitstek de tijd om de foto- of videocollectie eens flink onder handen te nemen.

Nou dan val je met je neus in de boter. Wessel Sijl verblijdde ons de vorige uitgaven al met diverse handige tips en programma's.

Zo ook nu weer, met een prima uitleg over de omzetting van kleur naar monochrome (zwart/wit).

Geïnteresseerd? Lees het artikel op pag. 23.



Isja Nederbragt



voorzitter
a.i. CompUsers

Voorwoord

Ik wens iedereen een goed 2022.

Januari is goed op weg. Het blijft alweer langer licht. De (gele) winter akoniet steekt in de tuin zijn kop weer boven de grond en tijdens mijn dagelijkse (corona)wandeling kwam ik (gele) winterjasmijn in bloei tegen. De (witte) sneeuwklodjes kondigen zich ook al weer aan.

En CompUsers functioneert sinds 1 januari 2022 praktisch gezien als intressegroep van HCC. Niet alles is al geregeld, maar dat komt wel. We doen nu 'alsof'.

De Kerstdagen en Oud en Nieuw zijn zo'n periode om terug te kijken. Zo kwam ik weer terecht bij mijn vakantie in IJsland. Het kon net, tussen alle coronagolven. Het gaat niet om corona, eigenlijk ook niet om IJsland, maar om de auto waarin we reden. Ik had het al ergens gelezen, maar nu zag ik het echt: ruitenwissers die zelf bepalen of ze water (en sneeuw) wegvegen. Je hoeft niet meer zelf te denken en doen. Eigenlijk ken ik het al langer, want mijn fiets bepaalt ook zelf of het licht of donker is. Ik hoef er niets aan te doen. Nou ja, aan het voorlicht. Het achterlicht vraagt nog eigen initiatief en een volle batterij.

En zo dacht ik weer terug aan mijn allereerste auto. Niet de geleende 2CV. Daarin moest je nog dubbel klutsen. Maar de Volkswagen Polo, de uitgekleepte versie. Uitgekleept betekent dat er een lampje gaat branden als de benzine bijna op is. Uitgekleept betekent ook dat de ruitenwischer niet automatisch naar de nulstand gaat als je hem uitzet. Dus moest je opletten wanneer je hem uitzette. In de praktijk bleek dat ik dat snel leerde en zonder nadenken deed. Hoewel mijn auto het nu zelf regelt doe ik het nog steeds zo, als reflex, als automatisme. Ik kan het niet meer afleren.

Zo denk ik ook terug aan de destijds nieuwe magnetron van een oud familielid. Er was er een met twee draaiknoppen: instellen van vermogen en instellen van tijd. Makkelijk om te leren. En nu, mijn nieuwe combi oven

(oven en magnetron in één) heeft drukknoppen, touchscreen, draaiknoppen en een heel dikke gebruiksaanwijzing die honderden keren waarschuwt dat de oven en de schalen erin heet worden.

Maakt de techniek het ons nu echt makkelijk? Apparaten zonder aan- en uitknop met heel veel standen. Je wasmachine programmeer je via voorinstellingen als 'jeans', of 'sokken'. Je moet je telefoon gebruiken om je te levisie aan en uit te zetten. Betalen met geld lukt bijna nergens meer. Alles met wachtwoorden die je nooit allemaal kunt onthouden en al die keuzes van programma's. Zonder DigiD kom je nergens meer.

CompUsers gaat ook veranderen. We zijn met ingang van 1 januari 2022 begonnen ons als interessegroep te gedragen. Dat is wennen. We zullen iets meer van te voren plannen, budget voor materiaal en software is, meer dan we gewend zijn, gekoppeld aan activiteiten. En misschien wordt het een ietsje soberder.

Gelukkelijk zal het in de praktijk allemaal wel meevallen, tenminste als er voldoende bestuursleden zijn. We gaan vol goede moed het nieuwe jaar in, met zes SoftwareBussen en hopelijk fysieke bijeenkomsten. Het hele bestuur hoopt dat we elkaar spoedig weer ontmoeten.

Je wilt ook wel eens iets schrijven in de SoftwareBus? Dat kan. Graag zelfs! Wil je liever redigeren? Dat kan ook!

Neem contact op met de redactie via het contactformulier op de website: www.compusers.nl

Sluitingsdatum volgend nummer (2022 2): 24 maart 2022

● Afbeeldingen verkleinen ●

Joep Bär

Een afbeelding kan op drie manieren worden verkleind. De eerste mogelijkheid is meestal het aanpassen van de fysieke afmetingen, waardoor het aantal pixels in de breedte en de hoogte wordt verminderd.

Andere opties

Ook het verlagen van de resolutie, gemeten in dots per inch (dpi), kan een oplossing zijn: voor drukwerk is 200 of 300 dpi meestal voldoende; op een website volstaat 72 dpi. De afbeelding kan ook in een compacter formaat worden opgeslagen, wat soms tot kwaliteitsverlies leidt: wijzig de extensie in JPG (voor foto's) respectievelijk (voor logo's) in GIF of PNG, via de optie 'Opslaan als'.

Om een afbeelding te verkleinen zijn er vele programma's beschikbaar, zowel gratis als tegen betaling. Ieder heeft zo zijn eigen voorkeur voor een specifiek programma, afhankelijk van het aantal functies en de gewenste mogelijkheden. Als je tevreden bent met je eigen programma, gebruik dit dan. Je hoeft niet zo nodig een nieuw programma te leren. Voor degene die geen programma heeft, of niet geheel tevreden is met het huidige, volgt hier een beschrijving voor het verkleinen van een afbeelding met het gratis Windows programma IrfanView: <https://www.irfanview.net/> (vergeet niet het NL taalbestand te downloaden en te installeren). Een beschrijving om het programma op je computer te installeren vind je hier. Degenen die een ander besturingssysteem gebruiken, hebben daarop al zo'n programma. En anders vind je alternatieven op internet.

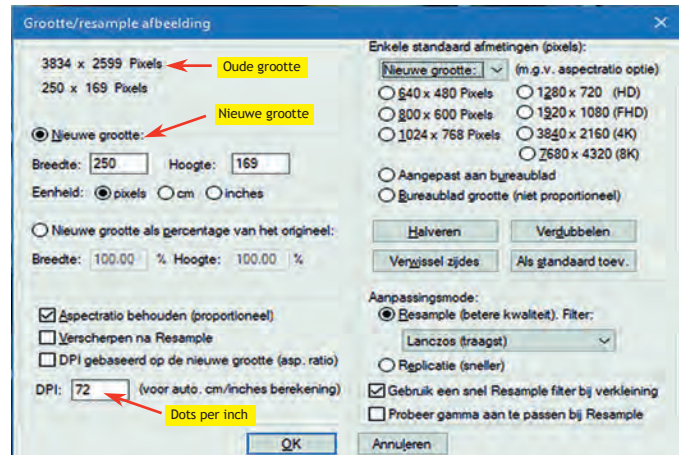
Al gebruik je een ander programma, je kunt de beschrijving gebruiken omdat de meeste programma's wel een vergelijkbaar stappenplan kennen. Sommigen denken dat met verkleinen van een afbeelding het slepen met een van de hoeken wordt bedoeld. Dit leidt slechts tot het verkleind weergeven van de afbeelding, maar de afbeelding blijft intern nog steeds even groot; het aantal bytes van de afbeelding verandert daardoor niet. Een nadeel van deze methode is dat de kwaliteit van de foto in een browser in zo'n situatie vaak (veel) slechter is dan wanneer de afbeelding eerst met een programma wordt verkleind naar het gewenste formaat. In dit artikel wordt met verkleinen bedoeld: het fysiek verkleinen van de afbeelding waardoor deze tevens minder ruimte inneemt op de harde schijf.

Een andere reden om een foto te verkleinen tot de grootste breedte en/of hoogte die op de website wordt gebruikt (bij het tonen van de afbeelding) is de tijd die het kost om de afbeelding te downloaden naar apparaten die van een langzamere verbinding gebruik maken (zoals telefoons en tablets). Als de foto 3 MB groot is kost dat vele malen meer tijd dan een foto van 100 kB. De kans dat de bezoeker de website verlaat wegens een lange laadtijd van de pagina is dan erg groot.

Verkleinen met IrfanView

Start het programma via *Bestand* » *Openen* of druk op de letter o (of O). Kies in het venster dat zich nu opent de juiste afbeelding en kies *Openen*. Deze komt op het scherm en links onderaan staan de huidige afmetingen van de afbeelding als breedte x hoogte (pixels) en resolutie (in dpi).

Ga nu naar *Afbeelding* » *Formaat wijzigen* (of *Ctrl+R*). In het nieuwe scherm kies je óf de gewenste breedte óf de hoogte en verander de waarde ervan. De andere afmeting past zich automatisch aan, zodat de verhouding gehandhaafd blijft.



Grootte aanpassen

Je kunt dit uit- en aanzetten door middel van een vinkje bij *Breedte/hoogte verhouding behouden*. Je kunt ook in het rechtervenster kiezen voor een van de standaardwaarden. Het aanpassen van de resolutie kan links onderaan achter 'DPI'. Voor een website is 72 dpi echt voldoende. Klik op OK als je akkoord bent met de nieuwe instellingen.

Sla de afbeelding nu op via *Bestand* > *Opslaan als* of druk op de letter s (of S). In het nieuwe scherm kies je in welke map en onder welke naam je het bestand opslaat. Denk eraan het origineel niet te overschrijven met de oorspronkelijke naam! Kies ook bewust voor het type bestand: gebruik dezelfde extensie als het origineel of wijzig het in een van de vele extensies waaruit je kunt kiezen. Op websites worden meestal JPG, PNG en GIF gebruikt, en ook wel WebP. De nieuwe afbeelding is nu klaar.

ATTENTIE: sla de nieuwe afbeelding op onder een andere naam of in een andere map. Je bent niet de eerste die later tot de conclusie komt dat toch een ander (groter) formaat gewenst is. Een afbeelding vergroten resulteert meestal in veel kwaliteitsverlies!

Veel afbeeldingen tegelijk verkleinen

IrfanView heeft de mogelijkheid om een groot aantal afbeeldingen tegelijk te verkleinen naar hetzelfde formaat. Kies *Bestand* » *Groepsconversie/ hernoeming*.

Selecteer de afbeeldingen die je wilt bewerken door ze aan de linkerzijde toe te voegen (klik op individuele afbeeldingen of selecteer ze en klik op *Toevoegen*). Als je de afbeeldingen hebt geselecteerd, kies dan de *Doelmap* (midden rechts). Klik dan rechtsonder op *Geavanceerd*. Selecteer *Formaat wijzigen* en maak je keuze uit *Beide zijden instellen op*, *Lange zijde instellen op* respectievelijk *Korte zijde instellen op*. Klik daarna op OK en het conversiefeest begint.

Belangrijk om te weten

Op een website kan eenzelfde afbeelding, qua formaat, op meerdere groottes worden weergegeven door de afmetingen in te stellen. Een kleine afbeelding kan groter zijn, of juist kleiner, dan het originele formaat.

Voor de onderstaande voorbeelden is in dit geval dezelfde foto gebruikt: de browser waarmee de bezoeker je website bezoekt zorgt dan zelf voor aanpassing van het formaat. Voor een website is het belangrijk dat een afbeelding zo klein mogelijk is (binnen de gewenste afmeting op het scherm). Hoe kleiner de afbeelding is in kilobytes (kB), des te sneller wordt de afbeelding geladen op de computer van de bezoeker van de website, waardoor tevens de webpagina het snelst wordt geladen. Indien het laden van de pagina te lang duurt, heb je de kans dat iemand je website links laat liggen 'want het duurt me te lang voordat die op mijn beeldscherm staat'. Op de verpakking van een SD kaart vond ik de volgende tabel. Deze geeft een goede aanduiding van de grootte van bestanden van diverse camera's.

Camera type	Bestandsgrootte
2 megapixel	900 kB
3 megapixel	1,2 MB
4 megapixel	2,0 MB
5 megapixel	2,5 MB
6 megapixel	3,2 MB
8 megapixel	3,5 MB

Om je een idee te geven van de bestandsgrootte van een afbeelding volgen hieronder enkele voorbeelden. Mijn digitale fotocamera levert opnames met 6.2 megapixels ofwel een formaat van 2848 x 2144 pixels bij 72 dpi, met een bestandsgrootte van 600 tot 3.500 kB.

Na verkleining tot 640 x 480 is de bestandsgrootte nog maar 40 tot 150 kB, en verkleining naar 320 x 240 levert bestanden op van 10 tot 60 kB. Opmerking: de onderstaande maten zijn in breedte x hoogte in pixels, resolutie: 72 dpi.

Voorbeelden:



80 x 60 x 72 = 4 kB



320 x 240 x 72 = 38 kB



640 x 480 x 24 = 66 kB

Colofon

De SoftwareBus is het verenigingsblad van CompUsers; het verschijnt zes keer per jaar. Uitgever: ProgrammaTheek BV.

Artikelen

De SoftwareBus bevat voornamelijk bijdragen van leden. Daar naast werkt CompUsers samen met andere computerbladen. Auteurs van de SoftwareBus geven impliciet toestemming om hun artikelen door te plaatsen in deze bladen. Uiteraard gebeurt dit met vermelding van auteur en bron, en eventuele vergoedingen hiervoor komen ten goede aan de auteur(s). Hebt u als auteur bezwaar tegen doorplaatsing, dan respecteren wij dat.

Abonnementen

U kunt een abonnement nemen via het aanmeldingsformulier op de website <https://www.compusers.nl/node/537>. Een abonnement wordt aangegaan voor één jaar, tenzij expliciet een andere termijn is overeengekomen. Na afloop wordt het abonnement stilzwijgend verlengd voor onbepaalde tijd. Het jaarabonnement kost € 27,00, maar leden van HCC krijgen € 9,00 korting. Nieuwe abonnementen zijn alleen mogelijk met automatische incasso. In dat geval wordt € 2,50 korting gegeven. De verzendkosten zijn nihil binnen Nederland, € 6,00 binnen de EU buiten Nederland en € 12,00 buiten de EU. Wijzigingen van tarieven worden ten minste twee nummers voor de ingangsdatum in de SoftwareBus gemeld.

Opzeggen abonnement

U kunt uw abonnement op ieder gewenst moment opzeggen. Daarbij geldt in het eerste jaar een opzegtermijn van drie maanden en, na de stilzwijgende verlenging, van één maand. U kunt een langere termijn aangeven. Opzeggen kan niet via de HCC. Gebruik om het abonnement op te zeggen het formulier op de website: <https://www.compusers.nl/node/628>. Het abonnement stopt niet automatisch bij beëindiging van het lidmaatschap van CompUsers of van HCC.

Na opzegging wordt het abonnementsgeld herrekend op basis van het aantal verzonden nummers. Hierbij worden de ledenkorting en de verzendkosten wel berekend, maar niet de korting voor automatische incasso.

Betalingen

U kunt de abonnementskosten uitsluitend via automatische incasso voldoen. Indien u langer abonnee bent en nog niet via automatische incasso betaalt, geldt een betalingstermijn van één maand na verzending van de factuur/acceptgirokaart. Bij niet tijdige betaling volgt een aanmaning en wordt de toezending van de SoftwareBus opgeschort tot de betaling binnen is. Voor een aanmaning wordt € 2,50 in rekening gebracht. Niet verzonden nummers geven geen recht op restitutie in geld of anderszins. ING bankrekeningnr. IBAN: NL13 INGB 0000 206202; BIC: INGBNL2A t.n.v. ProgrammaTheek BV.

Bestellen van losse nummers of jaargangen

Recente jaargangen (zes nummers) van de SoftwareBus zijn beschikbaar en worden op bestelling toegezonden. Kosten incl. verzending: € 21,00 voor CompUsersleden en donateurs en € 27,00 voor anderen. Losse nummers kosten € 5,15. Aanvragen: met het bestelformulier op de website <https://www.compusers.nl/node/539>. Daar staat ook hoe de kosten moeten worden voldaan.

Adreswijziging doorgeven

Gebruik hiervoor het adreswijzigingsformulier op de website: <https://www.compusers.nl/node/653>. HCC leden: ga naar <https://www.hcc.nl/contact> en volg de aanwijzingen om ook HCC uw nieuwe adres te melden.

Abonnementenadministratie

Voor contact om andere redenen: e mail naar: abonnementen@compusers.nl

Redactie

René Suiker, hoofdredacteur; Rob de Waal Malefijt, eindredacteur; Ger Stok, grafisch coördinator; Isja Nederbragt, redactielid.

Opmaak:

DTP team CompUsers: Harry van Mosseveld, Ger Stok, Rob de Waal Malefijt.

Druk: Senefelder Misset, Doetinchem

Basis Ontwerp opmaak: Okker Reclame, Veenendaal
Basis Ontwerp omslag: FIR&E, Wageningen

● BASIC in historisch perspectief ●

Hans Lunsing

Inleiding

Het artikel van Gordon Rahman over BASIC in SoftwareBus 2021 5 voerde mij weer terug naar de jaren '90 van de vorige eeuw toen ik actief was binnen de Basic gebruikersgroep. Ik werkte vooral met QBasic/QuickBasic, en later ook met PowerBasic. En dat alles in DOS, het toenmalige 16 bits besturingssysteem dat ook nog ten grondslag lag aan het toenmalige Windows (t/m Windows ME). De komst van Windows XP voor de consumentenmarkt luidde het einde van DOS in.

Ik heb enkele libraries met procedures voor zowel QBasic/QuickBasic als PowerBasic op mijn naam staan. Enkele daarvan heb ik internationaal verspreid. Ze zijn op het internet met enig zoeken nog wel te vinden. Een belangrijk deel van die procedures was ter wille van de efficiëntie in assembleertaal¹ geschreven. Ook mijn boekhoudprogramma Habo, dat in die tijd door de DOS-gg te koop werd aangeboden, was in BASIC geschreven, met ook enkele in assembleertaal geprogrammeerde onderdelen.

Het einde van DOS betekende ook het einde van mijn werk in BASIC. De visuele stijl van programmeren die door Microsofts Visual BASIC voor Windows werd geïntroduceerd sprak me niet aan. Ik ging bovendien over op het nieuwe besturingssysteem Linux. Ook al moest ik dat zelf nog grotendeels leren kennen hielp ik er ook al anderen mee. Veel tijd voor programmeren had ik toen niet meer.

Inmiddels zijn we meer dan 20 jaar verder. In de loop der tijd hebben heel wat verschillende BASIC varianten, interpreters en compilers² het licht gezien. Ze verschillen allemaal, maar de invloedrijkste was wel QuickBasic van Microsoft.



Dat werd in de praktijk de standaard waar veel BASIC varianten zich min of meer aan hielden. BASIC lijkt wat diversiteit betreft wel een beetje op Linux met al zijn distributies, die op allerlei punten verschillen, maar op essentiële punten zich toch wel aan een bepaalde standaard conformeren.

Mijn insteek was te onderzoeken wat er op het ogenblik in Linux aan BASIC ontwikkelomgevingen³ beschikbaar is. Het bleken er verschillende te zijn (één ervan, Liberty BASIC, in spe) met allemaal ook een Windows uitvoering. Het leuke was dat twee ervan, FreeBASIC en QB64, ook oude QBasic/QuickBasic programma's nog ondersteunen.



QB64.org
BASIC for the modern era

Bij deze speurtocht naar BASIC ontwikkelomgevingen kwam de hele historie van BASIC voorbij. Het leek me voor gebruikers van BASIC interessant om van de geschiedenis van deze, zeker in het verleden, invloedrijke programmeertaal een overzicht op hoofdlijnen te geven⁴. Daar is de rest van dit artikel aan gewijd. In dit overzicht komen verscheidene BASIC implementaties aan de orde die een min of meer belangrijke rol hebben gespeeld en/of spelen. Er zijn er veel meer, maar die spelen geen rol van betekenis of zijn inmiddels verdwenen.

In de jaren '90 heb ik ook Hans Lauwerier, indertijd emeritus hoogleraar wiskunde aan de Universiteit van Amsterdam, geholpen met de implementatie van de Zephyr SVGA library voor BASIC in zijn BASIC programma's voor het genereren van fractalen. Als dank kreeg ik zijn boek 'Fractals, meetkundige figuren in eindeloze herhaling' (5^e druk, september 1992). Achterin dat boek staan ruim 40 BASIC programma's voor een heel scala aan fractalen, waaronder de Mandelbrot set en de Julia set. Ze zijn allemaal geschreven in QBasic voor DOS, zonder nog van die SVGA library gebruik te maken. Ik kon ze met FreeBASIC en QB64 weer tot leven wekken.

Vanaf het begin van de jaren '80 was DOS het belangrijkste platform voor BASIC. Daarna kwam in de loop van de jaren '90 Windows, en vanaf de eeuwwisseling is BASIC er ook in allerlei gedaantes voor Linux (en MacOS), en verdween DOS van de kaart.

Het begin

BASIC, de 'Beginners All purpose Symbolic Instruction Code' is een van de oudste programmeertalen. Die werd in de jaren 1963 en 1964 ontwikkeld aan het Dartmouth College (New Hampshire, USA) door de wiskundigen John Kemeny en Thomas Kurtz. Hun doel was om het voor niet technische studenten mogelijk te maken op begrijpelijke en eenvoudige wijze een computer te gebruiken. BASIC is beïnvloed door Fortran (Formula Translating system), een veel in wetenschappelijke kringen gebruikte programmeertaal waar Kemeny en Kurtz ook aan werkten, en lijkt daar in bepaalde opzichten dan ook op. De syntax is vereenvoudigd en verduidelijkt en er worden gewone Engelse woorden gebruikt om opdrachten aan de computer te geven. Het oorspronkelijke Dartmouth BASIC bevatte maar 14 opdrachten!

De taal was niet duidelijk gestructureerd, en vooral het be-ruchte GOTO statement kon er toe leiden dat de loop van een programma vrijwel niet meer te volgen was. Dit wordt daar om wel spaghetticode genoemd. De bekende Nederlandse in formaticus en wiskundige Edsger Dijkstra zei er dan ook van 'Het is praktisch onmogelijk studenten die aan BASIC zijn blootgesteld nog goed te leren programmeren: als potentiële programmeurs zijn ze mentaal verminkt zonder hoop op herstel.' Gelukkig is er sinds dat eerste begin veel veranderd. BASIC heeft zich in de decennia daarna ontwikkeld tot een volwassen gestructureerde programmeertaal met veel mogelijkheden. Het oorspronkelijke doel, begrijpelijkheid en eenvoud, is daarbij niet uit het oog verloren.

In de tweede helft van de jaren '70 verscheen de microcomputer voor thuisgebruik, de voorloper van de pc, op het toneel. Dat leidde tot een explosieve verspreiding van BASIC omdat in vrijwel al die microcomputers BASIC was ingebouwd. Dank zij BASIC hadden al die computers, hoe verschillend ze ook waren, toch een gemeenschappelijke programmeertaal. Helaas bouwde vrijwel elke fabrikant zijn eigen variant van BASIC in, waardoor programma's toch niet zonder meer uitwisselbaar waren. Een unieke standaard was er niet. Kemeny en Kurtz zagen dit alles met lede ogen aan. Wat was er met hun BASIC gebeurd?

In Nederland werd in 1981 een populaire bibliotheek van uitwisselbare BASIC programma's met de naam Basicode opgezet. Deze programma's werkten op veel verschillende computers met een BASIC interpreter. Basicode program

ma's werden uitgezonden over de radio, zodat ze op cassette opgenomen konden worden en vervolgens op de computer uitgevoerd. Door de komst van de pc werd Basic code in feite overbodig, waarna er in 1992 een einde aan kwam.

ANSI/ISO standaard en True BASIC

Om meer eenheid in BASIC te brengen werkte Kemeny mee aan de vorming van een ANSI standaard voor 'Minimal BASIC'. Die kwam er in 1978, waarna hij in 1984 ook een ISO standaard werd. Kemeny en Kurtz publiceerden in 1985 een boek met de titel 'Back to BASIC', waarin de onverfijndheid en inconsistentie van de BASIC's voor microcomputers aan de kaak werd gesteld. Kemeny en Kurtz startten ook een bedrijf dat rond 1985 True BASIC uitbracht, dat niet verwonderlijk aan de ANSI standaard voldoet en daar nieuwe elementen aan toevoegde. In 1987 kwam daarop ook een ANSI standaard voor 'Full BASIC', dat in 1991 een ISO standaard werd. De standaard voor 'Minimal BASIC' werd teruggetrokken.



True BASIC bestaat nog steeds voor Windows, en noemt zich 'The Original BASIC since 1964'. De nieuwste versie (6) dateert van 2013. Daarvan is een demo versie beschikbaar met beperkte mogelijkheden.

Kemeny en Kurtz slaagden niet in hun opzet. De groei van de computermarkt was voor een klein bedrijf als True BASIC niet bij te houden. True BASIC is dan ook vrijwel de enige BASIC die aan de ANSI/ISO standaard voldoet. Uiteindelijk werd Microsofts QuickBasic, dat in 1985 verscheen, een de facto standaard. Dat heeft alles te maken met de komst van de pc en Microsofts dominante positie daarin.

De komst van de pc

In de jaren '80 ontstond een explosieve groei van het aantal pc's. Microsoft speelde daar met zijn besturingssysteem DOS een dominante rol in. Omdat Microsoft bij elke versie van DOS een vorm van Microsoft BASIC leverde, aanvankelijk GW BASIC, later QBASIC, werd dat in de praktijk een standaard voor BASIC. QBASIC was een uitgekledde versie van QuickBasic, een commercieel product met IDE (Integrated Development Environment) en compiler. Naast de standaardversie van QuickBasic was er QuickBasic Extended ofwel het BASIC Professional Development System (PDS). De standaardversie werd ontwikkeld tot 1988 (versie 4.5), de professionele versie tot 1990 (versie 7.1).

QuickBasic werd in 1992 opgevolgd door Visual Basic voor DOS, met een op tekst gebaseerde gebruikersinterface die de grafische interface van Visual Basic voor Windows simuleerde. Visual Basic voor DOS was geen lang leven beschoren. Het bleef bij die ene versie. Voor Windows ontwikkelde Microsoft Visual Basic met een geheel grafisch georiënteerd ontwikkelingsysteem, dat in 1991 voor het eerst verscheen en in 1998 met versie 6 voor het laatst. Het werd in 2001 opgevolgd door Visual Basic .NET, dat nog steeds verder ontwikkeld wordt. Ter onderscheid wordt het oude Visual Basic sindsdien Visual Basic Classic genoemd.

Borland bracht in 1987 TurboBASIC uit, dat syntactisch grotendeels gelijk was aan QuickBasic en ook een eigen IDE had. De ontwikkeling stopte al in 1989, waarna Bob Zale, de oorspronkelijke ontwikkelaar, het terugkocht en de naam veranderde in PowerBASIC. PowerBASIC ontwikkelde naast de DOS compiler ook een Windows (GUI) compiler en een Windows console compiler. De jongste versies zijn van 2011. Bob Zale overleed in 2012, waarna zijn weduwe het bedrijf in 2017 verkocht aan Drake Software. Tot dusver heeft dat nog niet tot verdere ontwikkeling geleid, maar de gebruikersforums worden nog steeds goed bezocht.

Het bedrijf Shoptalk Systems van Carl Gundel lanceerde in 1992 Liberty BASIC voor Windows en aanvankelijk ook OS/2. Net als Visual Basic van Microsoft biedt het een visuele ontwikkelomgeving. Zijn syntax is ook net als die van Visual Basic en PowerBASIC gebaseerd op QuickBasic, de de facto standaard. Liberty BASIC wordt nog steeds actief doorontwikkeld. De op stapel staande versie 5 (nu in alfa) zal ook voor Linux beschikbaar komen. De Windows versie draait overigens goed onder Wine, maar maakt natuurlijk Windows in plaats van Linux programma's. Er staat een probeerversie ter download waarmee geen standalone applicaties kunnen worden gemaakt. Bovendien is er van hetzelfde bedrijf Just BASIC, dat als vereenvoudigde versie van Liberty BASIC helemaal gratis is. Daarmee kunnen wel standalone applicaties worden gemaakt.

Deze ontwikkelingen leidden er toe dat er rond de eeuwwisseling vier min of meer belangrijke volwassen BASIC ontwikkelomgevingen voor Windows beschikbaar waren:

- Microsofts Visual Basic met zijn visuele ontwikkelomgeving voor grafische applicaties.
- PowerBASIC met zijn compiler voor grafische applicaties en zijn console compiler.
- Liberty BASIC met zijn visuele ontwikkelomgeving.
- True BASIC met eigen IDE en eigen taalvariant (de originele BASIC sinds 1964) moet hier ook worden genoemd.

De opkomst van Linux

Rond de eeuwwisseling was het nieuwe besturingssysteem Linux sterk in opkomst. Dat zag je terug in het feit dat nieuwe ontwikkelomgevingen voor BASIC, die in die tijd en later het licht zagen, ook versies voor Linux gingen aanbieden. Ook MacOS (toen nog Classic, vanaf 2001 Mac OS X) werd niet vergeten. De rij werd geopend met REALbasic en PureBasic.

In 1998 verscheen versie 1.0 van REALbasic, een visuele ontwikkelomgeving voor MacOS en Java. Een jaar later kwam er ook een Windows versie en stopte de ondersteuning van Java. Een Linux versie verscheen voor het eerst in 2005. In 2013 werd REALbasic hernoemd tot Xojo, en sinds 2014 ondersteunt het ook iOS. Naast de betaalde versie is een gratis versie beschikbaar. Daarmee kunnen geen standalone applicaties worden gemaakt.



PureBasic begon zijn leven in 1998 voor AmigaOS, en in 2000 voor Windows, Linux en MacOS. Het heeft een heel eigenzinnige syntax, die nogal verschilt van de QuickBasic standaard. Ook voor PureBasic moet worden betaald, maar het biedt een gratis versie aan voor kleine programma's tot 800 regels. Daarmee kan bovendien geen Windows DLL worden gemaakt, en wordt geen toegang tot de Windows API geboden. Ook PureBasic heeft een IDE met grafische ontwerpmodule.

De komst van open source

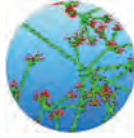
Parallel aan de opkomst van Linux verscheen ook steeds meer andere open source software op het toneel. Vrijwel alle ontwikkelomgevingen voor BASIC die na de eeuwwisseling verschenen werden uitgebracht als open source software. In 1999 had Gambas (*Gambas Almost Means Basic*) voor Linux de primeur de eerste open source ontwikkelomgeving voor de taal BASIC te zijn. Het is net als Visual Basic en daardoor geïnspireerd een visuele ontwikkelomgeving.



Gambas is nog steeds springlevend, en is sinds 2016 via Cygwin⁵ ook beschikbaar voor Windows. KBASIC, ook een visuele ontwikkelomgeving, ontstond in 2000, en kwam beschikbaar voor Linux, Windows en MacOS. Zijn naam werd later ge-

wijzigd in **Basic For Qt**. De jongste versie dateert van 2012. Zijn ontwikkeling lijkt daarna tot stilstand te zijn gekomen.

In het eerste decennium van de nieuwe eeuw verscheen vervolgens nog een viertal open source ontwikkelomgevingen, die alle vier nog actief zijn. Het zijn twee compilers, FreeBASIC en QB64, en twee interpreters, SmallBASIC en Basic 256.



FreeBASIC voor Windows, Linux en protected mode DOS verscheen in 2004. Het is een verzameling van ontwikkelgereedschappen, in het bijzonder een compiler, zonder een eigen IDE. Omdat het de bedoeling was een vrij alternatief voor QuickBasic te bieden is zijn syntax dan ook grotendeels compatibel met QuickBasic. Uiteindelijk biedt FreeBASIC veel meer dan QuickBasic, zoals meer datatypes en taalconstructies, zoals objecten. Het voorziet in een speciale 'QB' taalmodus met een hoge mate van ondersteuning voor programma's die oorspronkelijk voor QuickBasic geschreven zijn, zodat ze vrijwel zonder wijzigingen kunnen worden gecompileerd. Er zijn meerdere externe IDE's beschikbaar.

Voor Windows zijn dat met name FBIDE en FbEdit, terwijl in Linux de algemene IDE Geany ook FreeBASIC ondersteunt. Daarnaast is er voor Linux (en Windows) PoseidonFB.

QB64 verscheen voor het eerst in 2007 voor Linux, Windows en macOS. Zijn syntax is zoveel mogelijk achterwaarts compatibel met QuickBasic, maar vult de taal net zoals FreeBASIC aan met allerlei moderne elementen. Hij bestaat uit een IDE en compiler, waarbij de IDE zelfs als twee druppels water lijkt op de IDE van QuickBasic. Bijzonder is dat QB64 in BASIC is geschreven, en zich zelf compileert. QB64 compileert BASIC programma's niet direct naar machinecode, maar naar C code, waarna een C compiler die C code naar machinecode compileert. Gevolg is wel dat QB64 programma's merkbaar trager zijn dan bijvoorbeeld FreeBASIC programma's, en ook veel groter. Zelfs een simpel programmaatje resulteert al in een uitvoerbaar programma van zo'n 1,5 MB, waar FreeBASIC maar 30 KB vraagt.

SmallBASIC en Basic 256 zijn twee interessante open source BASIC interpreters. SmallBASIC begon zijn leven in 2004, en is beschikbaar voor verscheidene platforms, waaronder Windows, Linux en Android. De syntax heeft veel gemeen met die van QBASIC. Basic 256 zag in 2007 het licht en is beschikbaar voor Windows en Linux. Het is vooral bedoeld om met name studenten te leren programmeren.

Microsoft biedt in zijn Microsoft Store voor Windows 10 ook twee eenvoudige vormen van BASIC aan, nl. een verre nazaat van het oorspronkelijke QBASIC, en SmallBASIC, dat bedoeld is om kinderen te leren programmeren. Beide zijn interpreters. QBASIC biedt ook wat voorbeelden, waaronder een bij het oorspronkelijke QBASIC geleverd spelletje, genaamd nibbles.bas.

Noten

1. Assembleertaal is een symbolische weergave van machinetaal, dat is de taal waarin de instructies geschreven zijn die de processor van de computer begrijpt en kan uitvoeren. Een hogere programmeertaal zoals BASIC moet naar machinetaal worden omgezet om door de processor te kunnen worden uitgevoerd.
2. Een interpreter vertaalt de broncode van een programma instructie voor instructie naar machinetaal en laat elke instructie meteen uitvoeren. Een compiler vertaalt de broncode in zijn geheel naar machinetaal en maakt er dan een uitvoerbaar programma (executable) van.
3. Zie https://nl.wikipedia.org/wiki/Integrated_development_environment voor inhoud hiervan.
4. De historie van BASIC is ook wel terug te vinden in het Wikipedia artikel over BASIC (Engelstalig, in het Nederlands wat korter), maar daarin staan zoveel details dat de grote lijnen van de ontwikkeling wat zoekraken.
5. Cygwin is een runtime omgeving voor Windows waarmee met name Linux applicaties met minimale wijzigingen in hun broncode als Windows programma's kunnen draaien. Zijn website is <https://www.cygwin.com>

Tot slot

Wilt u ook in BASIC gaan programmeren dan hebt u de keus uit verschillende mogelijkheden die vrijwel allemaal zowel voor Windows als voor Linux beschikbaar zijn. Het ge makkelijkst is, denk ik, een visuele ontwikkelomgeving zoals die wordt geboden door Gambas, Liberty Basic en PureBasic.



De volgende versie van Liberty Basic (5) komt ook voor Linux beschikbaar, tot die tijd kan de Windows versie prima draaien in Wine onder Linux. Gambas is open source en dus gratis. Voor Liberty Basic (\$ 59,) en PureBasic (€ 79,) moet worden betaald, maar demo versies met beperkte mogelijkheden zijn beschikbaar. Xoj is ook een mogelijkheid maar is nogal duur: \$ 299, .



Naast deze drie is er natuurlijk Visual Basic .NET, maar alleen voor Windows. Het heeft verschillende edities, waarvan Visual Basic Express gratis is. Microsoft biedt een IDE genaamd Visual Studio voor allerlei verschillende talen, waaronder Visual Basic. De Community editie daarvan is gratis.

FreeBASIC en QB64 zijn vooral compilers voor de opdrachtregel (in een console of terminal). Wel is voor beide een IDE met editor beschikbaar die het werken ermee een stuk ge makkelijkker maken. Beide zijn krachtige talen waarmee ook grafische applicaties kunnen worden gemaakt. En oude QBASIC of QuickBasic programma's kunnen ermee gemakkelijk tot nieuw leven worden gewekt. Beide zijn open source software en daarom gratis.

```
MS-DOS Prompt - QBASIC
File Edit View Search Run Debug Options Help
DECLARE SUB FischerInstall ( )
DECLARE SUB Motor (MotorNo%, Direction%)
DECLARE SUB OutputMethod (Port%, AnalogSet%, OutDelay%)
DECLARE SUB FischerSet (Port%, AnalogSet%, OutDelay%)
DECLARE FUNCTION SetScaling (E1toE16%)
DECLARE FUNCTION InputHex% ( )
DECLARE FUNCTION AnalogIn% (KnoxFY%)
DECLARE FUNCTION InternalPCFunc% (KnoxFY%)
...
QBASIC interface drive routines
IBM Personal Computer
Copyright (C) Procon Technology 1995
Version 2.4
CONST E1 = 1, E2 = 2, E3 = 3, E4 = 4
CONST E5 = 5, E10 = 10, E11 = 11, E12 = 12
CONST E13 = 13, E14 = 14, E15 = 15, E16 = 16
CONST MOF = 0, EV = 1
CONST MOF = 0, MCCH = 1, MCH = 2, MON = 3
DIM SHARED Fischer%(0 TO 511)
DEF SEG = VARSEG(Fischer%(0))
DEF SEG = Fischer.bld, VARPTR(Fischer%(0))
FischerInstall
  HandOpen 30
  Arndt 39
  ArndtLeft 150
  ArndtRight 75
  ArndtDown 10
  HandClose 20
  ArndtUp 30
  ArndtLeft 150
  ArndtDown 10
  HandOpen 30
  Arndt 39
```

Websites

<https://nl.wikipedia.org/wiki/BASIC>: Informatie over BASIC, met veel aandacht voor de geschiedenis
https://en.wikipedia.org/wiki/List_of_BASIC_dialects: Uitgebreide lijst van BASIC dialecten
<https://www.nicholson.com/rhn/basic/basic.info.html#2> BASIC Programming Language Resources, met o.m. historische documentatie, en lijst van Basic interpreters en compilers
<https://www.truebasic.com/>, https://en.wikipedia.org/wiki/True_BASIC
<https://libertybasic.com/> https://nl.wikipedia.org/wiki/Liberty_BASIC
<https://visualstudio.microsoft.com/downloads/>
https://nl.wikipedia.org/wiki/Microsoft_Visual_Studio IDE voor o.m. Visual Basic
<https://docs.microsoft.com/nl-nl/dotnet/visual-basic/> https://nl.wikipedia.org/wiki/Visual_Basic_.NET
<http://gambas.sourceforge.net/>, <https://nl.wikipedia.org/wiki/Gambas>
<https://www.purebasic.com/> <https://nl.wikipedia.org/wiki/PureBasic>
<https://www.freebasic.net/> <https://nl.wikipedia.org/wiki/FreeBASIC>
<https://www.qb64.org/>, <https://en.wikipedia.org/wiki/QB64>
<https://www.xojo.com/>, <https://en.wikipedia.org/wiki/Xojo>
<http://smallbasic.github.io/>, <https://en.wikipedia.org/wiki/SmallBASIC>
<https://basic256.org/>, https://en.wikipedia.org/wiki/Basic_256
<https://www.geany.org/>: IDE voor FreeBASIC
<https://bitbucket.org/KuanHsu/poseidonfb/>: IDE voor FreeBASIC
<https://sourceforge.net/projects/fbedit/>: Windows IDE voor FreeBASIC
<https://sourceforge.net/projects/fbide/>: Windows IDE voor FreeBASIC

● Gnome-Boxes ●

Ton Valkenburgh

Als je overgegaan bent naar Linux heb je soms ook nog Windows nodig. Dit kan zijn voor specifieke programma's, voor je werk, maar ook voor een vereniging waarin je actief bent. Dual-boot kan een oplossing zijn, maar soms is een virtuele machine handiger of zelfs de enige mogelijkheid.

Inleiding

Linux is een wijdverbreid gebruikt besturingssysteem. Je komt het tegen in servers, routers, televisies, Internet of Things en diverse apparaten. Op de pc en laptop is het Linux minder gelukt om door te breken. Linux gebruikers hebben toch vaak nog Windows nodig. Wine kan hier vaak een oplossing brengen, maar niet alle Windows programma's werken goed in Wine. Dual boot heeft het nadeel dat je óf Windows gebruikt óf Linux. Snel schakelen tussen die twee lukt niet. Ook heb je twee harde schijven nodig om probleemloos met dual boot en Windows te werken. Windows vindt dat hij ei genaar is van de schijf waarop hij staat en neemt na een up grade of herinstallatie de bootsector van de schijf in bezit. Daarna werkt je Linux omgeving niet meer. Een virtuele machine kan dan een oplossing bieden. Ook is het handig om een virtuele machine te gebruiken voor het uittesten van een nieuwe versie van je besturingssysteem.

Virtuele machines

Voor de desktop en laptop zijn voor thuisgebruik de meest in teressante virtuele machines: VMware Workstation Player ([link 1](#)), Virtual Box ([link 2](#)) en QEMU ([link 3](#)). Alle drie zijn gratis voor thuisgebruik en werken onder Linux en Windows. Macintosh wordt niet door VMware Workstation Player ondersteund, maar wel door de andere twee. Welke is de beste om onder Linux te gebruiken? VMware Workstation Player is een goed performende virtuele machine, maar eist wel voor de huidige versie moderne hardware. Installatie is een fluitje van een cent. Bij sommige

updates van Linux moet de gebruiker Workstation Player op nieuw compileren. Op zich niet erg lastig, maar toch niet gebruiksvriendelijk.

Virtual Box vergt bij het installeren zorgvuldigheid betrefende de modules die ook moeten worden geïnstalleerd. Daarna werkt het probleemloos. Virtual Box heeft veel instelmogelijkheden om de virtuele machine aan te passen. QEMU is een virtuele machine waarbij je de hardwareconfiguratie kunt instellen. Dat kan een pc (x86_64) zijn, maar ook arm processors, AVR microcontrollers, Power PC, RISC V, Sparc, RX microcontrollers, Xtensa controllers en zelfs System 390. Al deze mogelijkheden maken het voor de gebruiker natuurlijk wel complex. Het vraagt ook veel hardwarekennis van de gebruiker. Bij de virtuele machine als pc (i386) is er o.a. ook ondersteuning voor TPM en Secure Boot. Dat maakt QEMU uitermate geschikt om Windows 11 te testen.

Gnome Boxes

Door de vele opties is het vrij lastig om QEMU goed werkend te krijgen. Gelukkig is het installatieprobleem opgelost door Gnome boxes ([link 4](#)). Gnome Boxes is er om virtuele machines te gebruiken, maar ook om computers op afstand te bedienen. Gnome boxes is dankzij QEMU goed geïntegreerd in de Linux kernel. Dit resulteert in een performance die duidelijk uitsteekt boven de andere genoemde virtuele machines. Het komt dicht bij de performance van de desbetreffende hostmachine. Gnome Boxes zit in de distributie van je Linux versie. Ik gebruik voor de test Kubuntu 20.04. Het geëmuleerde pc type is afhankelijk van de Ubuntu versie.

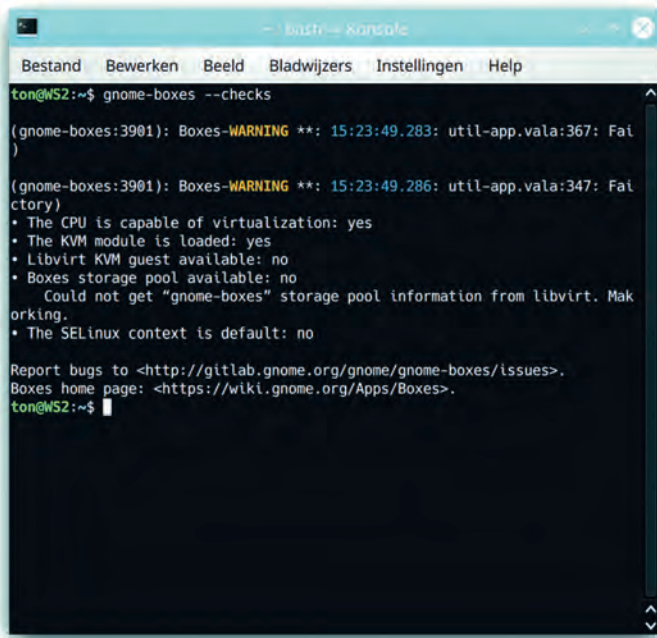
Om een virtuele machine te gebruiken is een machine met minstens 8 GB werkgeheugen nodig. Als je in de virtuele machine Windows wilt draaien, raad ik 16 GB aan. Alvorens een virtuele machine kan worden gebruikt, moet je in het BIOS van de hostmachine zorgen dat virtualisatie aan staat. Zet bij een Intel cpu *Intel Virtualization Technology* en *VT d* aan. Bij een AMD cpu zet je in het BIOS *IOMMU/AMD Vi* aan. Om te testen of de pc geschikt is en de bewuste functies aan staan, installeer je Gnome Boxes en gebruik je in de terminal het volgende commando:

```
gnome-boxes --checks
```

Met waarschijnlijk onderstaand resultaat (afbeelding 1). De pc is geschikt voor virtualisatie, maar er zijn nog wel wat dingen te regelen. Op zich werkt Gnome boxes nu, maar de waarschuwingen gaan we zoveel mogelijk wegwerken. We gaan *libvirt clients* installeren met het commando:

```
sudo apt install libvirt-clients
```

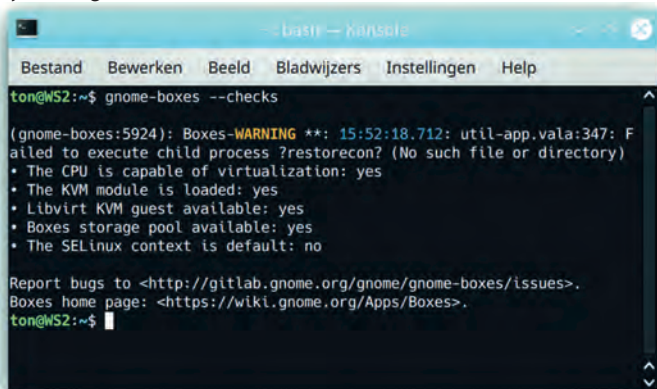
Als we na deze installatie het commando `gnome-boxes --checks` geven houden we nog een waarschuwing over (afbeelding 2). Die kunnen we negeren; deze is gerelateerd aan Security Advanced Linux (SELinux). SELinux is een Linux core uitbreiding die oorspronkelijk van de National Security Agent (NSA) afkomstig is. Deze uitgebreide, complexe veiligheidsregels zijn geschikt voor de overheid en grote bedrijven, maar niet voor de thuisgebruiker. Waar het om gaat, is dat we driemaal *yes* zien. De *Boxes storage pool* blijkt niet altijd zichtbaar te zijn, hij is er echter wel. Dit komt omdat Libvirt en Gnome Boxes allebei een andere map gebruiken om de virtuele machines op te slaan.



```
ton@WS2:~$ gnome-boxes --checks
(gnome-boxes:3901): Boxes-WARNING **: 15:23:49.283: util-app.vala:367: Failed to execute child process 'restoredcon? (No such file or directory)
(gnome-boxes:3901): Boxes-WARNING **: 15:23:49.286: util-app.vala:347: Factory
• The CPU is capable of virtualization: yes
• The KVM module is loaded: yes
• Libvirt KVM guest available: no
• Boxes storage pool available: no
  Could not get "gnome-boxes" storage pool information from libvirt. Making
orking.
• The SELinux context is default: no

Report bugs to <http://gitlab.gnome.org/gnome/gnome-boxes/issues>.
Boxes home page: <https://wiki.gnome.org/Apps/Boxes>.
ton@WS2:~$
```

Afbeelding 1: Check virtualisatie



```
ton@WS2:~$ gnome-boxes --checks
(gnome-boxes:5924): Boxes-WARNING **: 15:52:18.712: util-app.vala:347: Failed to execute child process 'restoredcon? (No such file or directory)
• The CPU is capable of virtualization: yes
• The KVM module is loaded: yes
• Libvirt KVM guest available: yes
• Boxes storage pool available: yes
• The SELinux context is default: no

Report bugs to <http://gitlab.gnome.org/gnome/gnome-boxes/issues>.
Boxes home page: <https://wiki.gnome.org/Apps/Boxes>.
ton@WS2:~$
```

Afbeelding 2: Check virtualisatie met libvirt client

De basis is gereed en we kunnen nu bijvoorbeeld Windows 10 gaan installeren. Zorg dat je een image bestand van Windows 10 (ISO) beschikbaar hebt. Dat kun je van de Microsoft web site ophalen. Start Gnome boxes door te klikken op *Toepas singen > Systeem > Boxes*. Klik links boven op het plus teken (+). Selecteer *Create Virtual Machine*. In het pop upvenster kun je kiezen om je Windows image te installeren. We kiezen voor Windows 10. Door de nieuwe eisen is Windows 11 wat in gewikkelder en de installatie daarvan komt later aan de orde.



Afbeelding 3: Create a Virtual Machine

Klik op *Operating System Image File*, selecteer het gewenste Windows bestand en open het. Je krijgt eerst de gelegenheid de virtuele machine aan te passen door op *Customize* te klikken. Afhankelijk van wat je straks met Windows wilt doen, kies je tussen de 4 en 8 GB werkgeheugen. Als schijfruimte kies je voor 128 GB. Dat is de maximale grootte waarnaar de schijf kan groeien. Klik daarna op *Create*. Volg nu de standaard installatieprocedure van Windows. Kies voor *Ik heb geen productcode*; die kun je later alsnog invullen. Je hebt dertig dagen om een en ander uit te testen. Uiteindelijk is er ook voor virtuele machines een Windows productcode nodig. Als je wilt schakelen tussen het gaststelsel en het hostsysteem druk je de linker Ctrl+Alt toetsen in. De cursor wisselt dan tussen beide systemen.

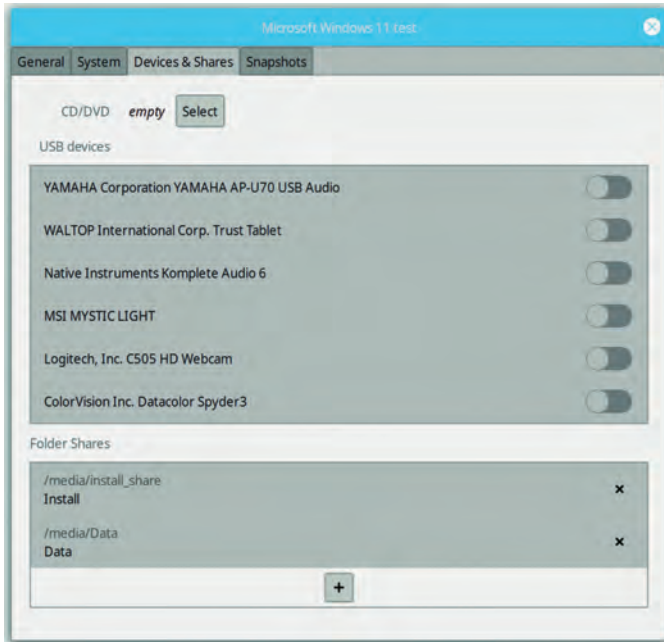
Bij een niet geactiveerde Windows zijn instellingen beperkt te wijzigen. Dat geldt ook voor de schermresolutie. Dat kunnen we oplossen met het installeren van een programma. Op de website van Spice space ([link 5](#)) halen we de *Windows guest tools* (spice guest tools) op. Na installatie en herstarten is de beperking in de schermresolutie vervallen. De resolutie past zich automatisch aan naar de venstergrootte. De muis kan zich nu soepel bewegen van gast naar host en omgekeerd. Verder kun je nu teksten en afbeeldingen kopiëren en plakken tussen host en gast. Om bestanden te kunnen delen halen we de *Spice WebDAV daemon* vanaf dezelfde website op en installeren deze. Na installatie kunnen we een netwerk station aanmaken met toegang tot mappen van de host. Ga in Gnome Boxes naar de bovenste balk rechts, klik op de drie puntjes en selecteer *Properties*. In het verschenen venster kies je *Devices & Shares*. Hier kun je kiezen welke USB apparaten je wilt gebruiken, en door op de plus te klikken kun je mappen van de host toevoegen.

Afbeelding 4 toont de door mij gekoppelde NAS share: *install_share* en de hostmap *Data*. Hierna moet je de virtuele machine herstarten.

Door middel van functies onder de andere tabs kun je diverse aanpassingen maken en snapshots maken. Het delen van bestanden met de Spice WebDAV daemon gaat prima met kleine bestanden. Wil je grote bestanden delen dan zul je een wijziging in het Windows register moeten aanbrengen. Open het register en ga naar:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WebClientParameters. Dubbelklik op *FileSizeLimitInBytes*. Vul daar de maximumwaarde van 4 GB hexadecimaal in als ffffffff (dus 8x f). Eventueel maak je de sleutel *WebClientParameters* aan en daarin het DWORD (32 bits) waarde *FileSizeLimitInBytes* met achtmaal f.

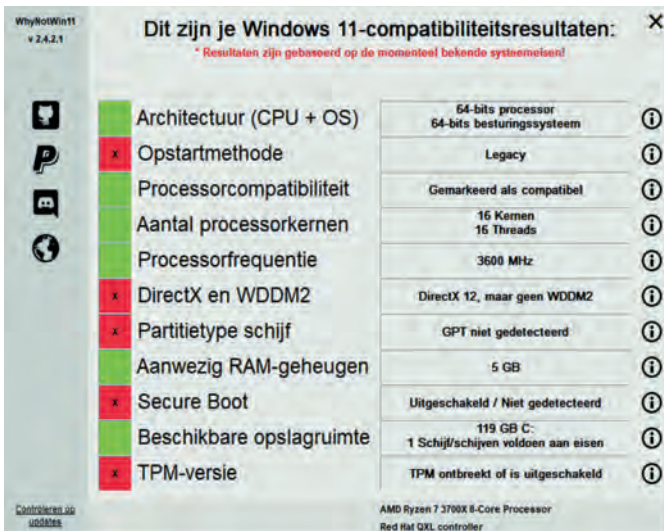
Afbeelding 4: Devices & Shares



Gnome Boxes is klaar om te gebruiken.

Als we een upgrade naar Windows 11 willen doen, moeten we eerst kijken of de virtuele machine daarvoor geschikt is. Gebruik daarvoor *Why Not Windows 11* (link 6).

Als we dat programma starten krijgen we het volgende te leestellende resultaat (Afbeelding 5).



Afbeelding 5: Windows 11 test

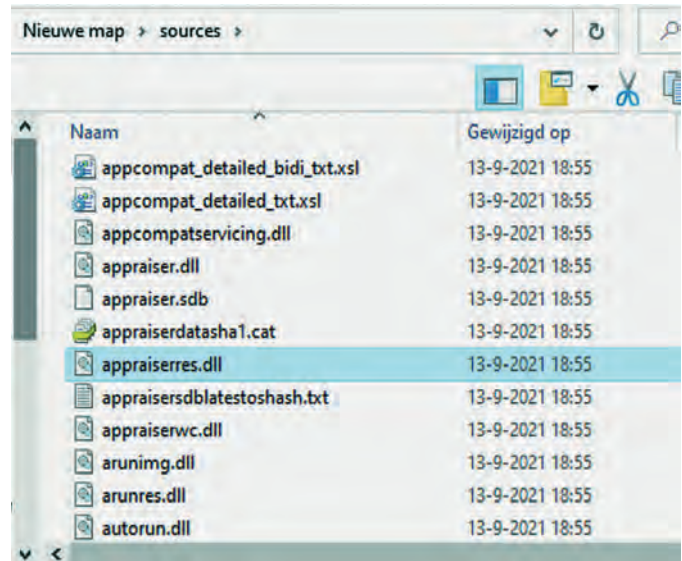
Hoewel de virtuele machine niet geschikt is voor Windows 11, is het met behulp van een bypass mogelijk een upgrade naar Windows 11 te doen. Een Windows upgrade duurt altijd nogal lang. Dit is eigenlijk alleen interessant als je Windows 10 in Gnome Boxes al langer gebruikt. In alle andere gevallen raad ik aan Windows 11 gelijk te installeren.

Upgrade van Windows 10 naar 11

Bij het starten van een upgrade wordt er eerst getest of de pc geschikt is voor Windows 11. Die test moeten we voor

komen. Als de test voor compatibiliteit met Windows 11 negatief is, wordt de installatie gestopt. Om dit te voorkomen moet daarom een programma van de installatieschijf van Windows 11 worden verwijderd.

We downloaden een Windows 11 ISO bestand in onze Windows 10 machine. De inhoud van het ISO bestand slaan we op in een map. Open met de verkenner het ISO bestand en kopieer de inhoud naar een tijdelijke map. Na het verwijderen van *appraiserres.dll* in de submap *sources* (Afbeelding 6), dubbelklikken we om de upgrade te starten in de oorspronkelijke map op *setup.exe*.



Afbeelding 6: Verwijder de file *appraiserres.dll*

In het venster dat nu verschijnt, klikken we op *Wijzigen hoe updates worden gedownload met Setup*. In het volgende venster kiezen we voor *Niet nu* en daarna *Volgende*. Ga daarna akkoord met de licentievoorzwaarden. In het volgende venster kiezen we voor *Installeren*. Je zit nu in de standaard procedure voor de Windows installatie. Als je Windows 10 gebruikte zonder Microsoft account zal er ook nu niet om worden gevraagd.

Microsoft behoudt zich het recht voor om systemen die op deze manier zijn geïnstalleerd in de toekomst niet van updates te voorzien. Of dit gebeurt, is afwachten. Je moet er dus wel alert op zijn.

Installatie van Windows 11

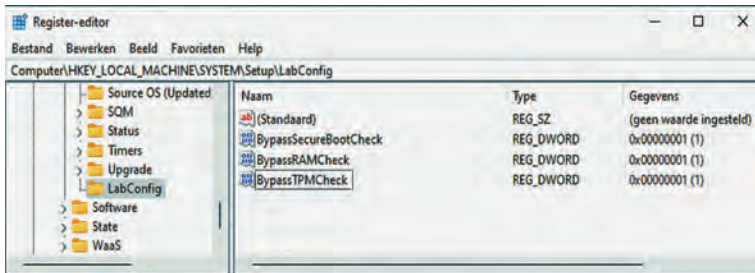
De virtuele machine mist de *Trusted Platform Module (TPM)* en *Secure Boot*. Voor Windows 11 zijn dit echter vereisten. Gelukkig is er een bypass die we kunnen gebruiken (link 7). Niet is gegarandeerd dat er op termijn updates voor de betreffende Windows zullen worden geïnstalleerd. Ik denk dat dit in de praktijk wel zal meevallen, want Microsoft heeft geen enkel belang om die updates te stoppen. Het is op zich al absurd om het voor goede machines lastig te maken om Windows 11 te installeren. We worden geacht zuinig te zijn op onze planeet! Raadpleeg eventueel ook link 8.

We starten net zoals voor Windows 10. Zorg dat je een image bestand van Windows 11 (ISO) beschikbaar hebt. Start Gnome boxes door te klikken op *Toepassingen > Systeem > Boxes*. Klik links boven op het plusteken (+).

Selecteer *Create Virtual Machine*. In het pop upvenster kun je ervoor kiezen je Windows image te installeren. We kiezen voor Windows 11. We zorgen dat *Express Install* is uitgeschakeld en kiezen *Next*. We passen de machine weer aan naar 8 GB werkgeheugen en 128 GB schijfruimte. Klik hierna op *Create*.

Bij het scherm met **Nu installeren** klikken we op **Nu uw computer herstellen**. In het volgende venster kiezen we **Problemen oplossen**. In het verschenen venster kiezen we **Opdrachtprompt**. Tik **regedit** in het nieuwe venster (cmd.exe) en geef **Enter**. We gaan nu het register aanpassen.

Ga naar **HKEY_LOCAL_MACHINE\SYSTEM\Setup**. Klik met de rechter muisknop op **Setup**. Selecteer **Nieuw > Sleutel**. Geef deze sleutel de naam **LabConfig**. Klik met de rechter muisknop op **LabConfig**. Selecteer **Nieuw > DWORD (32 bits)-waarde**. Geef het de naam **BypassSecureBootCheck**. Dubbelklik hierop en geef het de waarde **1**. We maken op deze wijze ook aan **BypassRAMCheck**, **BypassTPMCheck** en geven het ook de waarde **1** (Afbeelding 7).



Afbeelding 7: Aanpassingen in het register

Sluit **regedit** af, tik in **setup.exe** en geef **Enter**. De installatieprocedure start nu op. Verder volg je de procedure zoals bij Windows 10. Als je wilt testen zonder Microsoft account verbreek je de netwerkverbinding, zodra het venster **Uw Microsoft account toevoegen** verschijnt. Daarna klik je om terug te gaan op de pijl links in het venster. Na enige tijd even geduld komt er een venster om je lokale account in te vullen. Vervolg de installatie.

Virtuele machine aanpassen

Helaas is de instelling van de standaard pc in Gnome Boxes nogal eenvoudig: geen **UEFI**, geen **TPM** en geen **Secure Boot**. Dit is eigenlijk niet meer van deze tijd. We hebben gezien dat er nogal wat ontbreekt (zie afbeelding 5). Het is echter mogelijk om een virtuele machine aan te maken die meer moderne functies heeft.

QEMU biedt de mogelijkheid om **UEFI**, **Secure Boot** en **TPM** te activeren. Een deel van de aanpassingen kan met de virtuele machinebeheerder (*virt manager*) worden aangemaakt. TPM wordt helaas nog niet bij Kubuntu 20.04 in de virtuele machine ondersteund. In de Linux versies SUSE en Fedora is dit wel het geval. Voor Debian 11 is er een niet officieel pakket voor

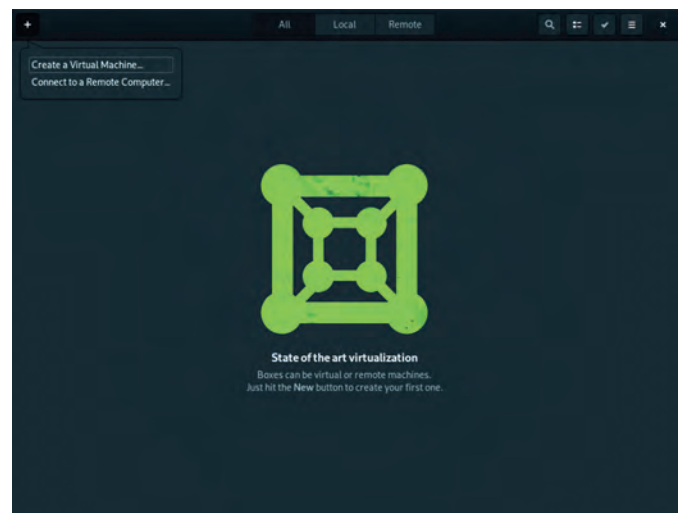
TPM. De vraag is of WDDM2.X wordt ondersteund. Als dat niet zo blijkt te zijn, is het nog steeds niet mogelijk Windows 11 zonder bypass te installeren. We zullen dit dus moeten onderzoeken.

Het netwerk zit standaard achter een virtuele router met Network Address Translation (NAT). Gast en host zitten op een ander netwerk. Wil je beide op hetzelfde netwerk hebben, dan moet je een Bridge gebruiken. Daarvoor moet je de virtuele machine en configuratiebestanden aanpassen. Alle hierboven genoemde aanpassingen aan de virtuele machine vallen buiten het bestek van dit artikel. In dit artikel wil ik laten zien hoe eenvoudig een virtuele machine met Gnome Boxes is te installeren. Grote aanpassingen in de virtuele machine zal ik in een ander meer in de diepte gaand artikel bespreken.

Conclusie

Gnome Boxes maakt het mogelijk om in Linux snel een virtuele machine op te zetten met bijvoorbeeld Windows. Het is dan mogelijk met kopiëren en plakken teksten en afbeeldingen uit te wisselen. Mappen van de hostmachine en USB apparaten kunnen eenvoudig worden gekoppeld. De aange maakte virtuele machine benadert de snelheid van de hostmachine.

De besproken methodes voor het installeren van Windows 11 op virtuele machines die niet compatibel zijn met Windows 11, zijn ook prima te gebruiken op een fysieke machine.



Links

1. <https://www.vmware.com/products/workstation player.html>
2. <https://www.virtualbox.org/>
3. <https://www.qemu.org/>
4. https://help.gnome.org/users/gnome_boxes/stable/
5. <https://www.spice space.org/download.html>
6. <https://github.com/rcmaehl/WhyNotWin11>
7. https://www.pctips.info/tips/windows tips/windows 11 installeren niet ondersteunde pc/#Optioneel_Windows_11_installatie_USB_maken_voor_niet_ondersteunde_PC
8. https://www.reddit.com/r/gnome/comments/q1wy49/install_windows_11_in_gnome_boxes/

● Homey als domotica-controller ●

Gjalt Zwaagstra

Iedereen kent inmiddels de Homey als domotica-spil in de huishouding.

Wat is Athom Homey en wat kun je ermee?

De Athom Homey is een geavanceerde smarthome hub die al allerlei slimme apparaten van verschillende merken met elkaar laat samenwerken. Dat betekent dat je met Homey enorm veel producten vanuit één app aanstuurt. In dit artikel vertellen we je meer over Homey en wat jij ermee kunt. Athom is opgericht met de visie om een beter huis voor iedereen te maken. Het bedrijf is een plek voor creatieve denkers en bouwers, die leven voor het samen oplossen van de moeilijkste problemen.

Alles wat zij doen, doen ze om jouw huis te verbeteren — en dat van henzelf. ‘Als je producten wilt maken die mensen écht willen, dan ga je bij ons thuis voelen’, zeggen ze. Meer weten over Homey? Check <https://homey.app>



Homey Controller. Prijs 399 euro (eenmalige aanschaf)



Homey Bridge. Prijs 69 euro (eenmalige aanschaf) internet verbinding vereist. 5 apparaten kunnen gratis aangesloten worden. Zijn meer aansluitingen gewenst, dan verloopt dat via een cloudienst. Er zijn verschillende abonnementsvormen beschikbaar. De abonnementskosten starten bij 3 euro per maand.

Wat kan de Homey?

Populaire apps

#1 Fibaro Your home is your castle	#6 IKEA Trådfri Creëer de juiste sfeer voor e...
#2 Philips Hue Transformeer hoe je licht er...	#7 Sonos A Better Way to Listen
#3 Google Chromecast Een slimme manier om jou...	#8 Spotify Listening is everything
#4 KlikAanKlikUit Laat Homey KlikAanKlikUit ...	#9 HomeyScript Script your Home
#5 Aqara & Xiaomi ZigBee Een ander soort oplossing v...	#10 Aeotec Intelligent automation.

Apparaten

Alle Homey apps in de Apparaten categorie

Fibaro Your home is your castle	Somfy RTS Stuur je Somfy zonnering automatisch en op e...	Bosch-Siemens Home Connect Connect Your Household
KlikAanKlikUit Laat Homey KlikAanKlikUit modules aansturen.	NEO Coolcam NEO Coolcam Z-Wave device ondersteuning	LED-Ring collectie Personaliseer Homey's LED-Ringdisplay
Aeotec Intelligent automation	Nexa Adds support for Texas Instruments devices.	Ring Ring ondersteuning voor Homey
HomeyKit HomeKit ondersteuning voor Homey	Harmony Hub Homey en de Harmony Hub zijn samen sterker!	Shelly Biedt Shelly apparaten via Homey
Xiaomi Mi Home Biedt Xiaomi Mi Home apparaten via Homey	Qubino Maak van elk huis een slim huis met Qubino	Tellus Voeg ondersteuning toe voor 433MHz Tellus...

Tv kijken met de Homey

Video

Alle Homey apps in de Video categorie

Google Chromecast Een slimme manier om jouw content te stream...	LG TV Bestuur je LG TV met Infrarood.
Samsung SmartTV Control your Samsung SmartTV	ONVIF Camera Ondersteuning voor ONVIF-compatibele camer...
LG WebOS Control your LG WebOS TV	Ziggo Connect your Mediabox Next with Homey
Samsung TV Bestuur je Samsung TV met Infrarood	Philips TV IR This app enables Homey to listen to signals fro...
Philips TV Bestuur je Philips TV's via JointSpace, met ond...	Sony BRAVIA Android TV Adds support for Sony BRAVIA Android TVs.

Flows

<https://www.youtube.com/watch?v=KipjWglIFB4>

Met 'flows' kan je op een simpele manier je automatisering regelen.

Je kunt:

- Mappen toevoegen, hernoemen, verwijderen en opnieuw rangschikken.
- Flows hernoemen, verwijderen, herschikken en inschakelen/uitschakelen
- Om een *OF* te krijgen: sleep een tweede of een latere conditiekaart iets naar beneden.
- Om een *Else* te krijgen: idem, beweeg een tweede of een latere actiekaart iets naar beneden.
- Klik op een [*niet beschikbare*] kaart om de code info te zien
- Er is geen aangepaste sortering/ordening van mappen of flows!

<https://support.athom.com/hc/nl/articles/360009667354>

Met dit kaartjessysteem is het voor iedereen mogelijk op een simpele manier zijn/haar huis superslim te maken. Eigenlijk kunnen we alles waar een stekker aan zit nu gaan automatiseren en zelfs ook zaken die je in je huisinstallatie aanbrengt.



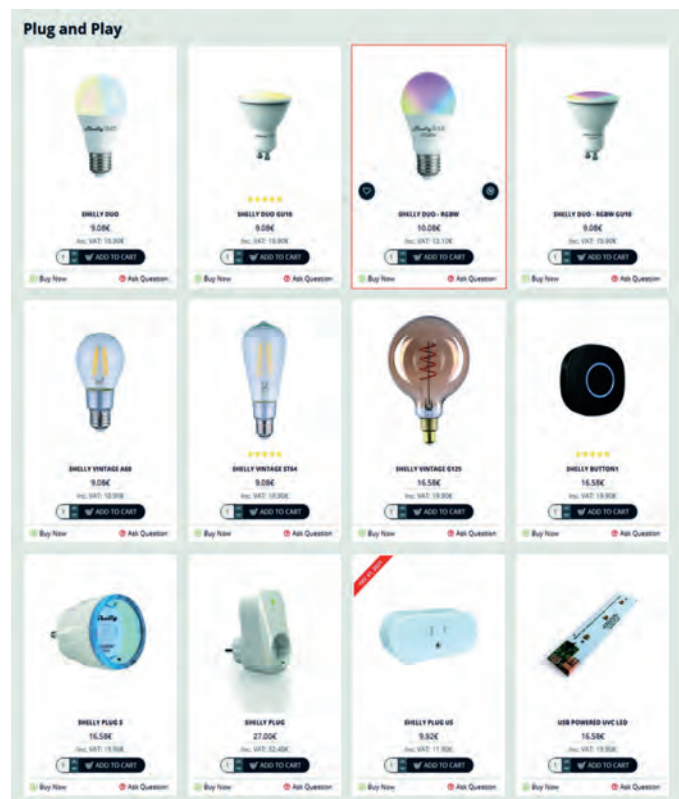
Relais voor huisinstallatie (wifi)

Met het shelly systeem wordt je hele huis slim en dat tegen beperkte kosten.

De losse onderdelen zijn bij de domoticaleveranciers en rechtstreeks bij de fabriek voor iedereen bestelbaar en je werkt direct met de shelly app met de Homey.

Als je verstand hebt van MQTT berichten zijn ze rechtstreeks aanstuurbaar en ook via de webinterface die elk device heeft.

Deze devices zijn met TASMOTA software via de wifi te patchen voor de liefhebber van Tasmota devices.



KlikAanKlikUit (KAKU) werkt met Homey

Ook apparatuur van KAKU werkt met de Homey, dus het systeem met de afstandsbediening werkt nu ook 'smart'. Men heeft nu drie systemen op de markt.

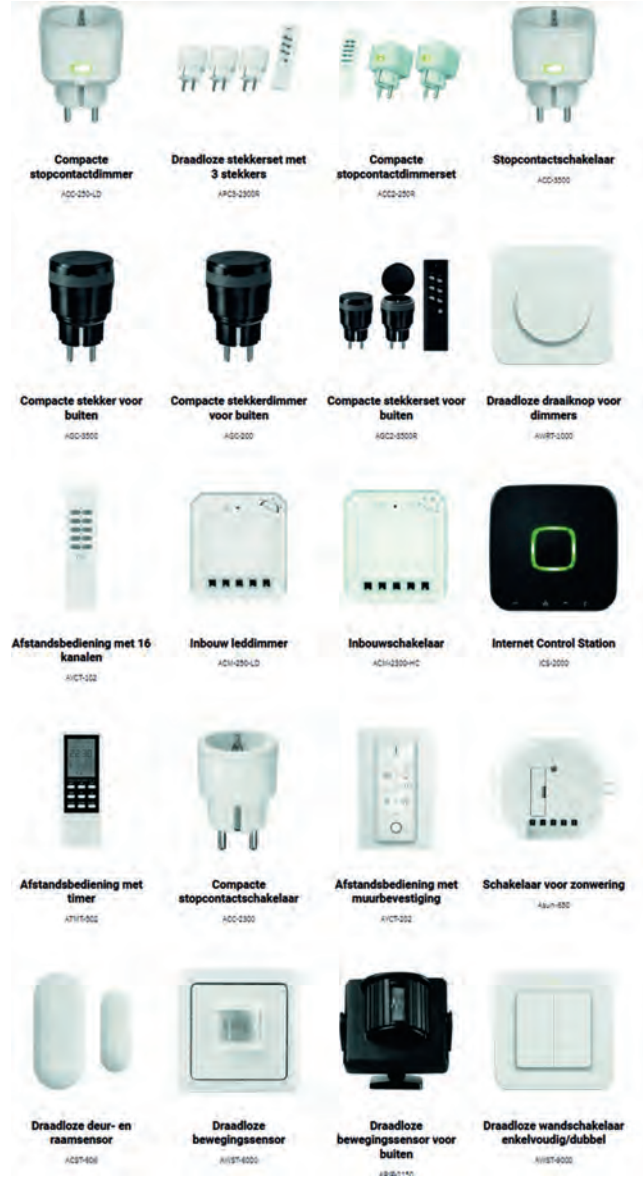
KlikAanKlikUit®



Beveiligingsproducten. Plus Line (868 MHz) speciaal voor Alarminstallaties.



Premium producten. Premium Line (Zigbee) Voor de gebruiker die alles beveiligd wil gebruiken (prijzig).



Start line (433MHz)
Voor de beginner die niet aan beveiliging doet. (goedkoop).

De keuze is reuze
Alles uitbesteden *loxtone* of *knx*.
Prijs vanaf 10.000 Euro (geen internet nodig).

Zelf bouwen
Home assistant of *Domoticz* op een Raspberry Pi (geen internet nodig).
Zelfdoen: prijs vanaf 100 euro, maar heel veel kennis nodig.
Laten doen kan honderden euro's kosten.

Homey
Beveiligd in de cloud, werking lokaal zonder internet (met flows).
Cloud Bridge: instapkosten laag, abonnement nodig, in ternetverbinding noodzakelijk.
Tuya cloud devices: prijs laag, smartphone en internetverbinding noodzakelijk. Nog geen abonnement nodig, wel voor op slag data (Action, Lidl, Hema).
Ikea, Philips: eigen implementatie, vrij prijzig, maar werkt ook met andere systemen.
AliExpress: vele mogelijkheden voor zeer kleine prijsjes.

Conclusie

Met de Homey is veel mogelijk, met een eenvoudige, gebruiksvriendelijke interface.



Scratch (15)

René Suiker

Ervan uitgaande dat we de ogen en vingers nog hebben, gaan we ook dit jaar weer op een laag pitje verder met Scratch. Niet omdat het moet, maar vooral omdat het gewoon heel leuk is. Het lage pitje komt vooral voort uit het feit dat ik bijna alles al behandeld heb, maar er is natuurlijk altijd wel iets te leren. En zo kwam het dat ik zowaar weer een idee kreeg, los van het uitwerken van het huiswerk.

Vaste lezers hebben intussen door dat ik heel enthousiast ben over Scratch, want het oogt zo simpel, maar je kunt er zo veel mee. Ik zag onlangs ook een voorbeeld van iemand die een 'bricks' spel maakte met Scratch en dat was heel een voudig. Bricks is een klassieker, je moet met een batje een balletje op een wand richten en elke keer als het balletje de wand raakt breekt er een steentje. En de bedoeling is dat je alle stenen wegspeelt.

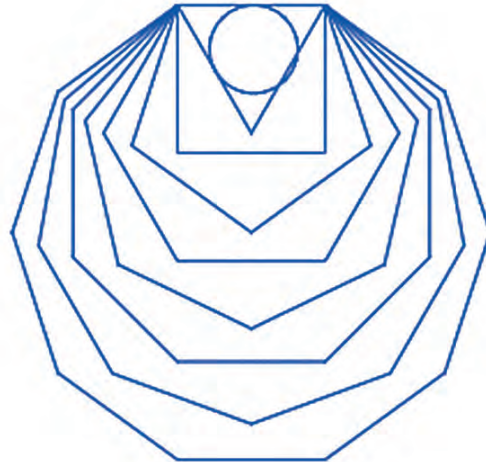
Uiteraard kun je het zo ingewikkeld maken als je wilt, maar hij nam de kijker mee in het proces, van heel eenvoudig tot erg gelikt. Voor wie de Engelse taal machtig is, deze meneer, Al Sweigart, heeft een boek over Scratch online geplaatst en dat kun je dus gratis lezen en dat behandelt eigenlijk alles van Scratch: <https://inventwithscratch.com>. De lessen die hij online zet volgen dat boek, dus als je het Bricks spel zelf wilt maken kan ik je die site van harte aanbevelen. Maar ik wilde dus iets anders.

Zoals jullie weten ben ik ook met een ander product bezig, met de intentie daar ook een reeks verhalen over te schrijven en dat gaat over Unity. Unity is een zogenaamde 'Game engine', zie elders in dit nummer. En nu wordt Scratch af en toe als speelgoed gezien, terwijl Unity dus professioneel is. En ik moet toegeven, Unity bevat heel veel krachtige elementen die het de spelontwikkelaar makkelijker moet maken een spel te ontwikkelen, maar ik wil eigenlijk eens kijken of ik met Scratch ongeveer hetzelfde kan bereiken als met Unity. Uiteraard zal ik een keer tegen de beperkingen van Scratch aanlopen, maar ik kan alleen maar zeggen (en straks aantonen): **je moet de mogelijkheden van Scratch niet onderschatten.**

Op vrijdagavonden volg ik een reeks van bijeenkomsten van de IG Programmeren, waarbij zij met Unity bezig zijn. Ik leer daar veel van en ze zijn onlangs weer begonnen om zich op beginners te richten, waartoe ik mijzelf ook reken. Zelfs op het gebied van Scratch kom ik net kijken, maar van Unity weet ik nog veel minder. Marco Kurvers, die de leiding heeft over deze bijeenkomsten, wilde mij wel bijstaan in de reeks van artikelen over Unity en ons idee was, om te beginnen met de mogelijkheden van Unity zonder te programmeren. En hierbij het ik het idee, zie ook het andere artikel, om een deeltjesversneller te bouwen. Nu is dat in het echte leven iets wat je niet zomaar in elkaar zet, maar ik had ooit eens een film gezien, over een goedge dat stuiterde zonder dat er energie verloren ging, maar waarbij juist energie werd gewonnen. Elke keer dat het de grond raakte, ging het harder omhoog dan dat het viel. Nu was dat een film en daarin is al les mogelijk en ik denk ook niet dat deze stofjes bestaan, maar het idee is wel leuk en eenvoudig te simuleren. In Unity gingen we met dat concept aan de slag en ik kreeg het idee om hetzelfde concept in Scratch ook uit te proberen.

Huiswerk

Maar eerst blikken we nog even terug op het huiswerk. Het was het huiswerk van aflevering 14, maar dat greep weer terug op de veelhoek die we in aflevering 13 beschreven. Althans, de veelhoek, het waren er een aantal, in elkaar. Het resultaat moest er ongeveer zo uitzien:



Figuur 1 Resultaat 'veelhoeken'

Ik had de code laten zien en er vielen een paar dingen op waar we niet heel gelukkig mee waren. Zie trouwens SoftwareBus 5 van 2021 voor de volledige uitwerking; voor abonnees is deze ook online volledig beschikbaar. Ik liet de uitwerking zien via een paar plaatjes:

1. De hoofdloop
2. De initialisatie
3. De veelhoekfunctie
4. De cirkel

En dan kijken we nu terug naar de vragen:

Opgave 14.1:

- a. In welk blok zat de fout:
 - a. In de hoofdloop
 - b. In de initialisatie
 - c. In de definitie van de veelhoek
 - d. In de definitie van de cirkel
- b. Gegeven dat het programma eerst de cirkel niet geheel tekende, maar ook veel trager draaide, heb je enig idee wat de fout geweest kan zijn?

Opgave 14.2:

- a. De opdracht was om zo compact mogelijk te programmeren. Zie je nog een manier om het programma, met behoud van functionaliteit, compacter te maken?
- b. Zie je een manier om het programma efficiënter te laten lopen, dus eventueel met meer code, maar snellere uitvoering?

Opgave 14.3:

- a. Pas het programma zo aan dat elke veelhoek een andere kleur krijgt
- b. Pas het programma zo aan dat de cirkel rond het midden wordt getekend en dan de veelhoeken er omheen.

Uitwerking

Opgave 14.1:

De fout zat uiteraard in de veelhoekfunctie. Dat was ook de grootste kanshebber, want daar vond de complexere logica plaats. En met programmeren zitten fouten vaak in een klein hoekje, maar meestal wel rondom de complexere zaken. We hadden te maken met een variabele 'draaien' en een variabele 'hoeken'. De variabele 'hoeken' bevatte een waarde tussen 3 en 10. De variabele 'draaien' gaf aan hoeveel graden

per hoek gedraaid moest worden. En de herhaalfunctie moest dus per hoek doorlopen worden, maar omdat hij werd aange roepen met 'draaien' in plaats van 'hoeken' werd hij dus heel veel keer doorlopen, namelijk bijvoorbeeld 120 keer als het een driehoek betrof. Dus de driehoek werd wel getekend, alleen niet in drie streepjes, maar in 120 streepjes. Omdat ze over elkaar heen vielen, viel dat niet op. De totale (aangepaste) code ziet er nu als volgt uit:

We beginnen maar met de cirkel: die zetten we rond het midden. Dus we draaien de volgorde binnen de hoofd lus even om, eerst de cirkel en dan pas de veelhoeken. Het kan hel pen om eerst een assenkruis te tekenen, zodat je een beeld hebt waar je moet zijn. Daarna is het óf wat gevorderde wis kunde, óf empirisch vaststellen waar elk figuur moet beginnen, zodat je met de veelhoek rondom de cirkel uitkomt. Als je een algoritme kunt vaststellen, dan kom je vast een eind.



Figuur 2 Totale code voor de veelhoek

Onder (1) zie je de initialisatie. Behoorlijk recht toe recht aan. Onder (2) de gecorrigeerde versie van de functie 'Maak x hoek'. Onder (3) zie je de vernieuwde functie 'Maak cirkel' en bij (4) zie je de hoofd lus.

Al met al is de code al redelijk compact en worden de veel hoeken redelijk vlot getekend. De cirkel doet er iets langer over, maar dat komt omdat hier de code in de veelhoek 60 keer doorlopen wordt. Je ziet die cirkel dus stapje voor stapje opgebouwd worden. Overigens, als je de code in de turbo mo de laat uitvoeren, dan staat het eindfiguur er ogenblikkelijk.

Opgave 14.2:

Dit was een opgave voor de cracks, want zelf zie ik niet zo gauw veel verbetermogelijkheden. Maar ik ben dan ook geen professioneel programmeur. Misschien dat een van de lezers vanuit de IG Programmeren of IG Kunstmatige Intelligentie hier nog mogelijkheden ziet? Maar het kan ook net zo goed iemand anders zijn die ineens het licht ziet. Ik ben be nieuwd.

Opgave 14.3:

Deel a is natuurlijk heel eenvoudig. Er is onder de groep 'Pen' (donkergroen) een functie beschikbaar 'verander pen kleur met een waarde'. Als je dit toepast binnen de hoofd lus bereik je snel bijna het gewenste effect (zie figuur 3 hier naast)

Toch doet dit net niet helemaal wat we beogen, al komt het aardig in de buurt. Dat brengt ons dus bij het nieuwe huis werk:

Opgave 15.1:

- Waarom is dit het niet helemaal?
- Wat kun je daaraan doen?

Opgave 14.3.b was iets complexer; niet zozeer om het helemaal in beweging te zetten, want daar is de huidige code geschikt voor, maar je moet de coördinaten even goed uitpluizen.

Ik laat het nu even aan de lezer over, want ik wil iets nieuws gaan behandelen. Eventueel kan ik volgende keer nog wel iets dieper op deze materie ingaan, mocht dat aan de orde zijn.

Wel is het misschien handig om tijdens het ontwikkelen niet gelijk alle veelhoeken in één keer te tekenen, maar eerst bij voorbeeld een driehoek. Kijk eens goed waar die moet komen en hoe zich dat tot het assenkruis en de cirkel verhoudt. Als je dat voor de driehoek hebt vastgesteld, lukt het met het vierkant ook wel en zo ga je verder. De grootte van het object hangt op grond van de logica in de functie wel af van het aantal hoeken. Dus moet je ook je startpositie hiervan af laten hangen.



Figuur 3 Opgave 14.3.a >

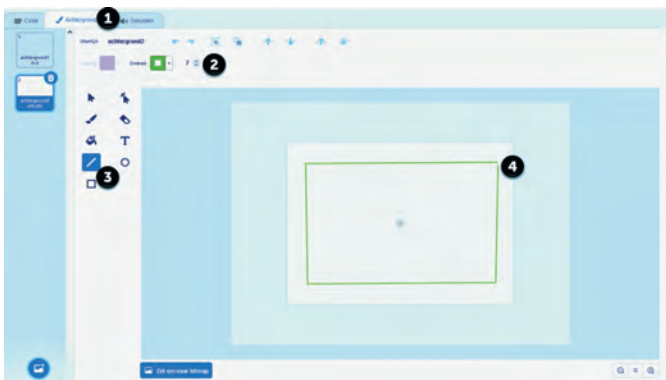
Opgave 15.2:

- Probeer de hiervoor genoemde opgave 14.3.b nu zelf, met de gegeven aanwijzingen
- Welke handige tips kun je delen om te slagen?

En toen iets nieuws

Zoals ik hiervoor al aangaf wil ik een rudimentaire deeltjes versneller bouwen. Niet zozeer omdat ik er een toepassing voor heb, maar als oud marineofficier heb ik iets met explosies en de wetten van de natuurkunde en het is natuurlijk leuk om een simulatie te maken die daarmee kan spotten. Nogmaals, een echte deeltjesversneller is natuurlijk heel complex en het voert veel te ver om zelf maar het minieme beetje dat ik er wel van snap uit de doeken te doen, laat staan dit proces helemaal te beschrijven. Maar het idee ontstond toen we met Unity bezig waren. Unity is dus geen programmeertaal, zoals Scratch wel is, maar is een zogenaamde engine, die dus het programmeerwerk voor een deel overneemt. En om een goed spel te maken, moet je dus programmeren én kun je Unity inzetten om een deel van je werk te vergemakkelijken. Dit werkt nog niet met Scratch, want dan moet je C# gaan leren. Dat gaan we in de verhalen over Unity te zijner tijd wel wat uitwerken, maar voorlopig wilden we even binnen Unity blijven en daar wat mee stoeien, voordat we feitelijk gingen programmeren. En binnen Unity heb je wat fysische wetten ingebouwd waardoor je je er als programmeur niet al te druk over hoeft te maken. Dat iets valt komt door de zwaartekracht en die kun je in Unity instellen, waardoor objecten daar dus ook vallen, totdat ze op de grond liggen. En als die grond nogal schuin is, kunnen objecten er ook vanaf rollen. En wat heeft dit nu allemaal met Scratch te maken? Niets, helemaal niets. Behalve dat ik wilde zien in hoeverre het experiment dat we met Unity gingen doen ook met Scratch mogelijk is. En zo begon dus een tweetal artikelen die op elkaar ingrijpen. Want in dit Scratch artikel kijk ik met een schuin oog ook naar Unity en in het Unity artikel verwijst ik ook naar Scratch.

Mijn conclusie is op voorhand, maar dat gaan we dus uitproberen, dat Unity krachtiger is, maar dat Scratch makkelijker is en dat je met beide tools iets boeiends kunt maken. Het idee is om een soort box te maken waarbinnen een bal letje heen en weer stuitert, maar het moet steeds harder gaan stuiten. In die box zit een gat waar die bal dan een keer met hoge snelheid doorheen schiet. In ons doolhofspel, enige tijd geleden, hebben we gezien hoe je iets kon laten bewegen en kon versnellen, dus die techniek kunnen we weer toepassen. Toen ging het met keyboardbesturing, nu moeten we het automatisch laten gebeuren. We kunnen dit zo mooi en realistisch mogelijk maken met invloeden van zwaartekracht e.d., maar laten we eerst maar eenvoudig beginnen. Het balletje valt naar beneden, stuitert omhoog tegen het plafond en stuitert dan weer terug, telkens bij elke stuit een beetje snelheid winnend. Het is de bedoeling dat hij niet alleen maar recht stuitert, maar een beetje rond, zodat hij ook een keer bij de uitgang komt. We beginnen maar even met de achtergrond:



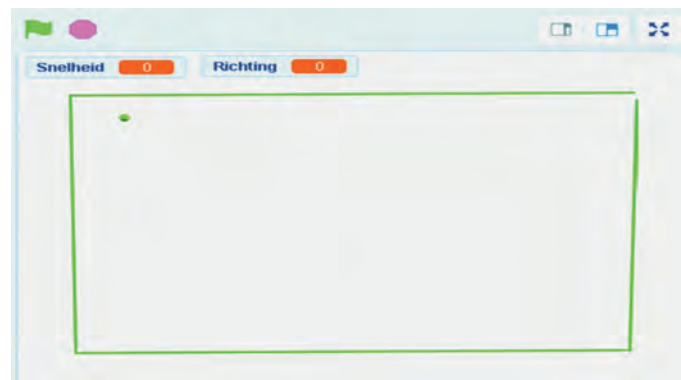
Figuur 4 De achtergrond

We maken de achtergrond door rechtsonder op achtergrond te klikken en dan een nieuwe aan te maken. Eenmaal geselecteerd zie je bij (1) dat het tabblad achtergronden geselecteerd is. Met behulp van 'Omtrek' bij (2) kun je de lijnkleur en -dikte bepalen. Ik maak gebruik van een kleine sprite (de tennisbal) en die laat ik in een relatief grote box stuiten. De lijnen van de box zijn ook dik. Het is aan te bevelen de lijnen allemaal dezelfde kleur te geven, omdat we straks gaan kijken of die kleur geraakt wordt. Bij (3) zie je het gereedschap om lijnen te tekenen. We zouden eronder ook een figuur als een rechthoek kunnen tekenen, maar dan moeten we er later een gat in maken als uitgang. Ik heb dus gewoon vier aaneengesloten lijnen gemaakt. Dan heb ik rechtsboven bij (4) een opening overgehouden, waar het balletje net doorheen zou kunnen. We gooien dus de kat even weg als sprite en kiezen de tennisbal. Ik neem aan dat bekend is hoe dat moet. De tennisbal wordt gewoon meegeleverd bij Scratch, daar hoeft je dus niet veel moeite voor te doen. De code in deze simulatie wordt dus opgehangen aan de sprite tennisbal. Om te beginnen moeten we het stuiten goed ondervangen. Ik heb daarvoor het volgende blokje ergens in de code nodig:



Figuur 5 Stuiten

We hebben een blok nodig uit de (oranje) categorie 'besturen' voor een 'als dan' constructie. Wat we ons afvragen uit de (lichtblauwe) categorie 'waarnemen' is of we een specifieke kleur aanraken. Als je op de kleur klikt kun je een kleur selecteren. Deze kleur selecteren we met het pipetje, we nemen de kleur van de rand van de achtergrond. Als deze waarde optreedt, dan keren we de bal om. Hier zie je een rood blok, dat wil zeggen, een eigen blok. De code voor 'keer om' moeten we nog schrijven. Om te kunnen omkeren moeten we een variabele snelheid en richting hebben. Scratch kent niet zoiets als een automatische beweging op basis van snelheid en richting, we kunnen echter wel de sprite een aantal stappen in een bepaalde richting laten zetten. Deze gaan we dus aanmaken onder de (bruine) categorie 'Variabelen'. Deze kunnen we aanmaken voor deze sprite alleen of voor alle sprites. Dat maakt hier niet uit, omdat we maar één sprite hebben. We maken dus een variabele 'Snelheid', want 'Richting' is binnen Scratch wel al bekend (onder de categorie 'Beweging'). Deze laten we ook zichtbaar zijn op het speelveld. Standaard komen die onder elkaar, maar als je de box redelijk groot hebt, is het misschien logischer om ze naast elkaar te zetten. Het speelveld ziet er nu aldus uit:



Figuur 6 Speelveld

Vervolgens maken we ook nog een eigen blok 'Initialisatie'. Hierin zetten we de startwaarden, dan hoeven we die niet steeds in het codeblok van de hoofd lus op te zoeken als we hieraan willen sleutelen.

De hoofd lus wordt dan wel heel simpel:

- Initialiseren
- Eeuwige lus:
 - Ga met snelheid 'snelheid' in de gekozen richting 'richting'.
 - Als je een rand raakt, keer om.
- Dit gaat door tot je op de rode vlag drukt. Dan kun je de snelheid en richting uitlezen van je scherm en met andere startwaarden kijken wat er gebeurt.

De grote uitdaging zit hem in de juiste code voor het omke ren. Tijdens proefdraaien ging de bal telkens door de zij kanten van de rechthoek. Het komt blijkbaar zeer nauw, wat overigens ook het geval was bij Unity. Daar zijn we nog steeds aan het uitzoeken waarom de bal af en toe door de wand heen schiet, zeker bij hogere snelheden.

Dat komt waarschijnlijk doordat de programmatuur zich af vraagt of je iets raakt aan het eind van een beweging. Als de beweging heel snel is, is het object al door de grens heen voordat die opgemerkt wordt. Bij Scratch, waar je dit zelf moet programmeren, zie je dit gebeuren. Bij Unity, waar de engine zorgt draagt voor de mechanica, zie je dat niet direct. Verder had ik bedacht dat, bij het stuiteren, de 'richting van inval is de richting van uitval' uitgevoerd kon worden door: Nieuwe richting = $180 - \text{oude richting}$. Dit gaat echter alleen op voor het stuiteren op een horizontaal vlak. Voor stuiteren op een verticaal vlak geldt een andere formule. Voor een stuit tegen een verticale wand geldt dat de hoek met 1 ver menigvuldigd wordt. Verder is het ook leuk om niet exact te stuiteren, dus om de baan iets minder voorspelbaar te ma ken. We willen dus de berekende hoek een klein beetje laten variëren. De totale code ziet er nu als volgt uit:

Uiteindelijk is het allemaal dus toch ingewikkelder geworden dan ik me in eerste instantie had voorgesteld, maar het past nog steeds redelijk makkelijk op één scherm. De hoofd lus onder (1) zegt in feite:

1. Initialisatie - verder beschreven bij (2)

2. Eeuwige lus:

- a. Neem 'snelheid' stappen
- b. Kijk of je een horizontale wand raakt
 - i. Indien ja, stuit horizontaal
- c. Kijk of je een verticale wand raakt
 - i. Indien ja, stuit verticaal

Voor alle zekerheid heb ik geen achtergrond getekend, maar een box, waardoor ik volledige controle had over de posities en de opening. Je ziet in de hoofd lus ook gelijk het manco van de aanpak. Als je 'snelheid' stappen neemt, dan kun je de wand al gepasseerd zijn voordat je je afvraagt of je een wand raakt. Nu treedt dat probleem op met een snelheid van rond de 30. Als ik de wanden wat dikker maak, zou het probleem dus pas bij hogere snelheden optreden. Op zich is dit natuurlijk mechanisch wel correct: op enig moment is een wand niet in staat een bewegend onderdeel vast te houden, afhankelijk van de energie die erin zit.

Onder (2) zie je de initialisatie. We maken het scherm schoon, we tekenen de wand, zie hieronder bij (3) en we gaan naar onze startpositie en stellen de richting en snelheid in. Ook hier zouden we iets met een random waarde kunnen doen, als mogelijke uitbreiding.

Onder (3) zie je de wand getekend worden. We hebben een an dere kleur genomen voor horizontale en verticale wanden. Dit voorkomt andere, meer ingewikkelde, logica om te kijken hoe we de richting moeten veranderen. We kunnen het niet alleen van de richting afleiden, want je kunt met elke bewegingsrichting zowel tegen een wand als een vloer of een plafond stuiteren.

Bij (4) zie je het effect van een horizontale stuitering. We richten ons dan op $180 - \text{huidige richting}$ en hebben deze ge modificeerd met een getal tussen 2 en $+2$. Deze formule zie je bij (6). De snelheid hebben we met 1 verhoogd, om de ver snelling te laten optreden: daar was het om begonnen. Daarna hebben nog een blokje 'herhaal tot' ingevoegd, om te voor komen dat bij het stuiteren het blokje binnen de wand blijft stuiteren. In het plaatje is die afvraag nog met de verkeerde kleur gebeurd; in mijn project heb ik dat aangepast.

Bij (5) zie je in feite dezelfde code als bij (4), alleen is de formule om de nieuwe richting te berekenen iets anders. Zoals hierboven vermeld: de nieuwe richting is de huidige richting keer 1 . Het programma is nog op veel manieren te ver fraaien, maar in essentie is dit de versneller die me voor ogen stond. Alleen, het is mooier te maken, de box oogt wat ruw, er is geen geluid en de bal gaat een keer door de muur. Als we de muur te dik maken, blijft er weinig ruimte over om te stuite ren, maar kunnen we hogere snelheden halen. Zou er een andere manier zijn om tijdens het be wegen te achterhalen of de bal door de muur schiet? Het lijkt er ook op dat de bal door de wand schiet in de hoek waar de twee wanden samenkomen. Zou daar een oplossing voor kunnen zijn, zodat we dit gedrag kunnen voorkomen? Hoe zou het komen?

Opgave 15.2:

- a. Hoe kun je voorkomen, dat de bal door de wand gaat?
- b. Zit het probleem inderdaad in de hoeken? Zo ja, wat kun je daaraan doen? Zo nee, wat is dan het grote probleem?

Ten slotte nog dit. Ik wilde in het begin de achter grond gewoon maken met lijntjes in plaats van een rechthoek. Achteraf realiseerde ik me dat je best wel met een rechthoek kunt werken en dan met het gummetje een gaatje kunt maken daar waar de bal eruit moet kunnen. Ofwel, het is niet nodig om in de code de box te maken, maar het maakt het natuurlijk wel mogelijk om elke keer de achter grond een beetje te veranderen. In die zin is het dus wel een nuttige aanvulling.

< Figuur 7 Totale code van de versneller



● MobaXterm ●

André Reinink



pSSH(t) – Ken je MobaXterm al? Nee?
Misschien is het dan tijd om ‘Silly’ Putty te dumpen!

Enkele jaren geleden was ik voor het eerst aan het stoeien met een mini computer op basis van Linux. Het gebruik van een SSH verbinding is dan nagenoeg onvermijdelijk. Vanuit Windows is ‘Putty’¹ een veel gebruikte oplossing. Maar ik ontdekte een heel fraai alternatief.

Oud speelgoed

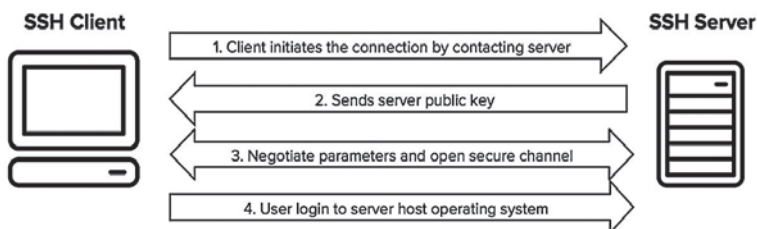
Silliputty was lang geleden een bekend stukje speelgoed. Het plastische spul kon je bijvoorbeeld op een krant drukken en daarna het op een blanco papier overbrengen. Vandaar de kwinkslag in de teaser.

Nieuw speelgoed

‘Putty is een gratis en open source terminal emulator, console en network filetransfer applicatie’. Het ondersteunt o.a. SSH. De naam Putty heeft geen officiële betekenis. Misschien heb je Putty al eens gebruikt om toegang te krijgen tot je NAS omdat je via FTP of HTTP(S) geen toegang kreeg. We staan er niet zo vaak bij stil, gebruiken het misschien niet zo vaak, maar als je het nodig hebt weet je de mogelijkheid om met Putty een SSH verbinding op te zetten wel te waarderen. Ik in ieder geval wel toen mijn NAS een probleem had.

SSH²

Secure Shell (SSH) is een versleuteld netwerk protocol om veilig te kunnen netwerken op een niet veilig netwerk. Dat kan in principe op bijna elke type netwerk. Onderstaand een grafische voorstelling van zo’n verbinding.

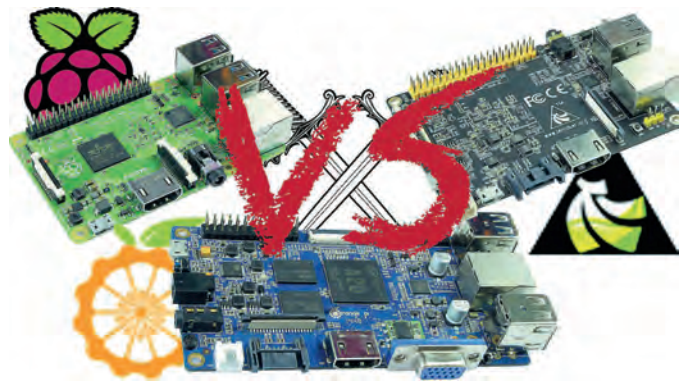


SSH wordt vaak genoemd bij gebruik in een Linux omgeving, maar gebruik met Microsoft Windows is ook goed mogelijk. SSH is ooit ontworpen als vervanger voor Telnet. Windows 10 gebruikt ‘OpenSSH’, en voor Windows 11 zal er ook wel een ingebakken oplossing zijn. De standaard communicatiepoort voor SSH is 22.

De aanleiding

Een paar maanden geleden pakte ik een oud projectje weer op. Ik had naast een oude Raspberry Pi 2 en 3 nog twee sma ken thuis liggen: een Banana Pi en een Orange Pi. Voor deze laatste twee Chinese vruchten is er een heel mooi alternatief voor de Raspberry Desktop: Armbian. Armbian is een samentrekking van ARM³ en Raspbian. ARM slaat op de processorarchitectuur. Raspbian is de oude naam van het

Raspberry Pi OS. Een tweetal grote voordelen van ARM: (relatief) goedkoop en energiezuinig.



Armbian⁴, aanvullende info

Armbian ‘Linux for ARM development boards’ heeft een relatief grote gemeenschap. Op Armbian.com vind je voor heel veel Single Board Computers (SBC) een Armbian image. Ik tel op de website een kleine 90 images.



De slogan van Armbian is: ‘Perfect for Server, Excellent for Desktop and Simple for Building’.

Analoog aan de Raspberry download je het image, schrijft het weg naar een SD kaart en start je de SBC op. In een Windows omgeving gebruik ik Rufus⁵ voor het schrijven van een image naar een SD kaart.



Banana Pi en Orange Pi

De Banana Pi en Orange Pi die ik bezit initialiseer je dus met behulp van de SD kaart. Daarna zijn er verschillen.

De Banana Pi is in staat om het besturingssysteem te ver plaatsen naar een andere schijf dan de SD kaart. Na een herstart wordt de SD kaart alleen nog gebruikt als boot medium om de hardware aan te geven waar het besturings systeem staat.

De Orange Pi die ik bezit daarentegen heeft een eigen eMMC geheugen dat gebruikt kan worden om alle benodigde data (bootinformatie en besturingssysteem) te bevatten. Na configuratie kun je de SD kaart verwijderen. Deze is niet meer nodig. Deze aanpassingen zijn allemaal te configureren via het Armbian image zelf. Na opstart en inloggen start je de configuratie op met *armbian config*. Verderop in dit artikel kom ik er nog op terug.

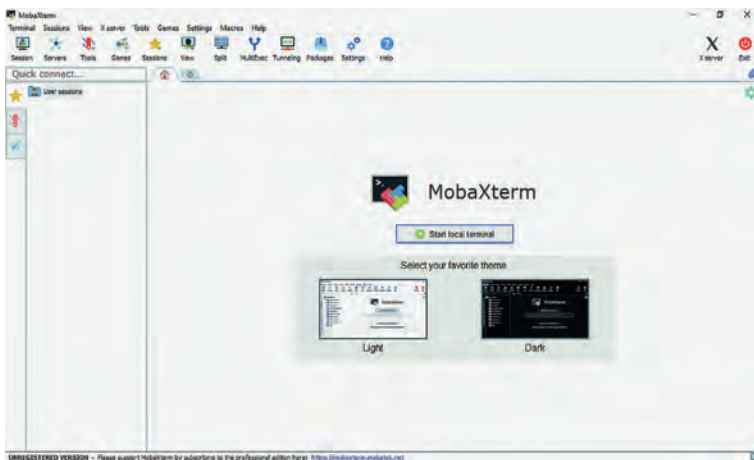
MobaXterm⁶

Tijdens ‘trial and error’ werkzaamheden liep ik tegen het programma MobaXterm aan. Het programma wordt door de makers omschreven als ‘Enhanced terminal for Windows with X11 server, tabbed SSH client, network tools and much more’. Pas later begon het me ineens te dagen dat ik de naam ‘Moba’ al eens eerder gelezen had. De makers had ik kunnen kennen van ‘MobaLiveCD’. Een executable waarmee je rechtstreeks een iso bestand van een live cd kunt starten. Helaas alleen voor 32 bits systemen. Daarnaast is er ook nog ‘MobaPhoto’. Beide programma’s lijken enigszins gedateerd of minder onderhouden. Mijns inziens niet de moeite waard.

Ik download MobaXterm Home Edition 21.4 van 7 oktober 2021. Mijn voorkeur gaat uit naar de portable edition: downloaden, uitpakken en draaien maar.

De standaard software is gratis, maar heeft een aantal beperkingen. Zoals bij heel veel andere software zijn deze beperkingen niet ‘storend’. Wil je deze beperkingen niet, dan moet je er € 49 voor neertellen.

Tijdens het downloaden zie ik dat er ook een bak vol met plug ins⁷ beschikbaar is. Meer over de beperkingen en de plug ins verderop in dit artikel.



Hier boven zie je het openingsscherm van MobaXterm. Sinds kort kun je kiezen voor een licht of een donker thema. Ik heb gekozen voor een licht thema. Er zijn voor de liefhebbers ook nog ‘skins’ beschikbaar. Ik ben geen *liefhebber*...

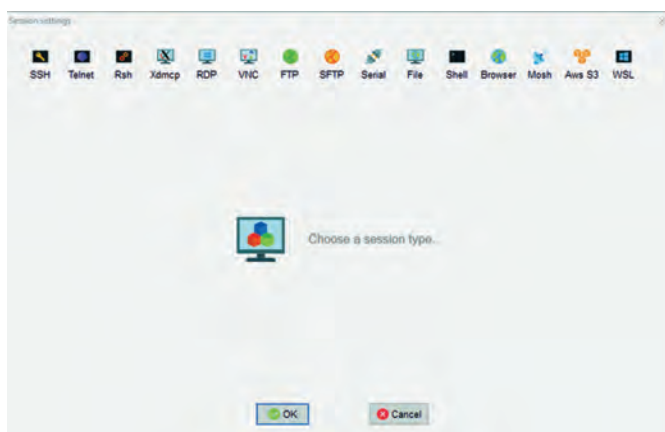
Het scherm is traditioneel opgebouwd.

Links bovenaan een menubalk. Ik kom daar nog uitgebreider op terug.

Rechts bovenaan de bekende Windows knoppen van *minimaliseren*, *maximaliseren* en *sluiten*.

Daaronder aanvullende knoppen voor een koppeling naar de Xserver en een *Exit* knop voor de openstaande sessie.

Geheel links in de kantlijn zie je de contouren ontstaan voor een bestandsbeheerder.



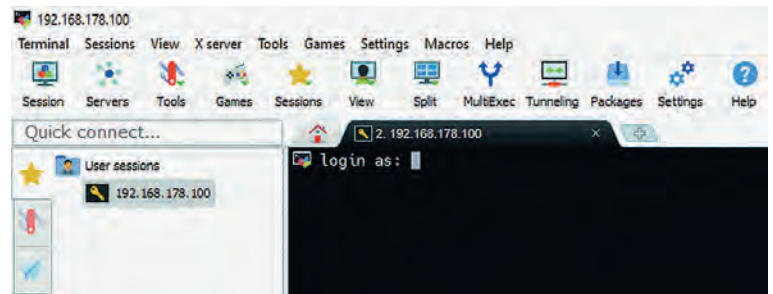
Vervolgens kun je uit een scala van verbindingsopties kiezen (Zie de afbeelding onderaan de vorige kolom).

Ik vind het een van de meest krachtige en bijzondere eigenschappen van het programma. Waarschijnlijk zul je de meeste verbindingen kennen, maar niet vanzelfsprekend ook willen gebruiken in MobaXterm.

Ik zou me goed kunnen voorstellen dat sommige lezers opties als **FTP**, **SFTP**, **Shell**, **Browser** of **WSL** (Windows Subsystem for Linux) wel eens zouden willen proberen. Ik zou willen zeggen: probeer het vooral gewoon eens uit.

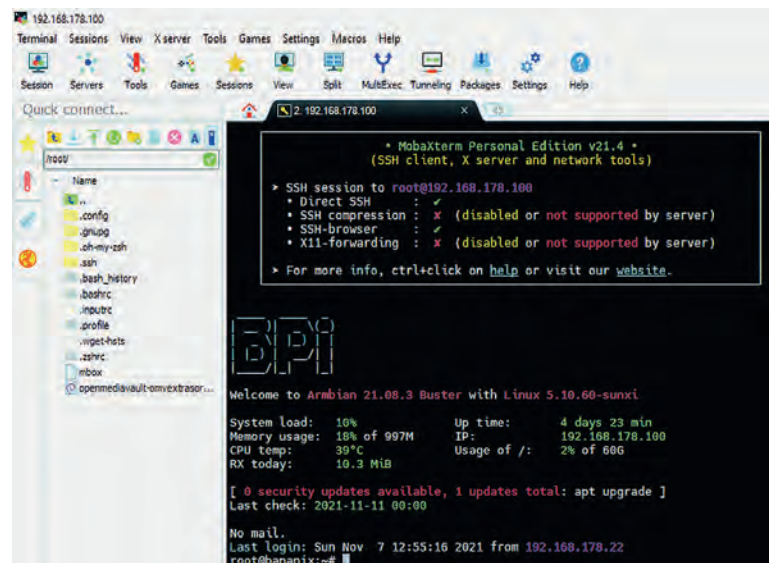
Voor dit artikel wil ik het e.e.a. beschrijven en laten zien op basis van een SSH verbinding.

Ik kies daarom dus nu voor een SSH verbinding. Vervolgens krijg je een scherm te zien waarbij je je gebruikersnaam kunt invullen (afbeelding niet getoond).



Links zie je de *Quick connect* mogelijkheid. Ik ga er even van uit dat, als je thuis een netwerkapparaat in gebruik hebt, zo als een server of een NAS, je dan weet wat het interne IP adres is. Voor dit artikel gebruik ik het IP adres van mijn Banana Pi. Vanzelfsprekend kun je ook andere interne adressen of externe adressen gebruiken. Ik heb 192.168.178.100 in de voorgaande stap ingevuld en kom in het terminalscherf van de Banana Pi.

Ik heb mijn ‘credentials’ correct ingegeven en krijg dan het volgende scherm te zien:



Bestandsbeheer

Een van de eerste opties die opvallen is dat je nu een combinatie ziet van enerzijds het terminalscherf en anderzijds een bestandsbeheerder.

De doorgewinterde Linux gebruiker kan de commando’s wel dromen om van de een naar de andere map te springen. Maar met een bestandsbeheerder wordt het wel een stuk comfortabeler als je net iets minder ervaring hebt met Linux. Best wel vreemd, als je beseft hoeveel apparatuur in onze elektronische wereld voorzien is van Linux.

Als je bijvoorbeeld `smb.conf` wilt aanpassen in `/etc/samba/` is dat heel snel geregeld. Met de navigatieknoppen ben je vliegensvlug op de juiste plek.

Als je dan dubbelklikt op `smb.conf` opent zich de interne MobaTextEditor. Naar keuze kan er ook een externe editor geopend worden.

Wil je de rechten aanpassen van het bestand? Met de rechter muis toets is dan een peulenschilletje. Hetzelfde geldt natuurlijk ook voor een map.

Split terminal mode

Als je meerdere terminals hebt geopend kun je ook meerdere terminals tegelijkertijd tonen. Dus naast *single* ook *2 terminals vertical*, *2 terminals horizontal* en *4 terminals mode*. Met een beetje beeldscherm is het best wel prettig werken met meerdere terminals.

Netwerkscanner

In het voorbeeld op de vorige pagina heb ik een IP adres gebruikt dat ik al kende. Stel dat je niet weet wat de IP adressen zijn die in je netwerk aanwezig zijn, hoe fijn is het dan dat je die met de ingebouwde netwerkscanner kunt vinden?

IP Address	Name	SSH	RDP	VNC	FTP	Telnet	Rlogin	HTTP	Other ports
192.168.178.1	friz box	X	X	X	X	X	X	X	Deep scan
192.168.178.32	volmo.friz box	X	X	X	X	X	X	X	Deep scan
192.168.178.40	sunshine13.friz box	X	X	X	X	X	X	X	Deep scan
192.168.178.55		X	X	X	X	X	X	X	Deep scan
192.168.178.100	Bananix.friz box	X	X	X	X	X	X	X	Deep scan
192.168.178.101	Fritzbox-2.friz box	X	X	X	X	X	X	X	Deep scan
192.168.178.200	orangeix	X	X	X	X	X	X	X	Deep scan
192.168.178.63		X	X	X	X	X	X	X	Deep scan

Onder het *Toolsmenu* vind je de netwerkscanner. Opstarten en enkele seconden later weet je wat er in je netwerk aanwezig is. Je ziet de Banana Pi op 192.168.178.100 en de Orange Pi zit op 192.168.178.200. Ook duidelijk te zien is dat beide SSH, FTP en HTTP ondersteunen. Natuurlijk kun je ook inloggen op je router om het een en ander op te zoeken. Maar meestal is dat een stuk omslachtiger.

Poortscanner

Misschien ben je ook nieuwsgierig naar openstaande poorten? Ook daar heeft MobaXterm aan gedacht. In hetzelfde *Tools menu* vind je ook een poortscanner. In no time heeft deze optie van adres 192.168.178.100 de bekende poorten gescand:

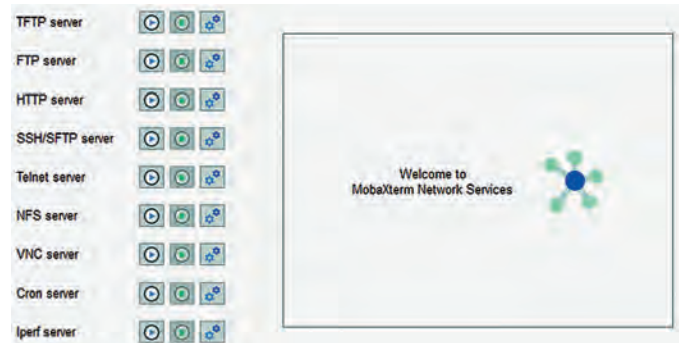
IP Address	Scan only known ports	Scan ports range
192.168.178.100	<input checked="" type="radio"/>	1 -> 65535

- Port #21 (ftp): listening
- Port #22 (ssh): listening
- Port #80 (http): listening
- Port #111 (sunrpc): listening
- Port #139 (netbios-ssn): listening
- Port #445 (microsoft-ds): listening
- Port #2049 (nfs): listening

Je kunt ook alle 65535 poorten scannen, maar dat duurt wel even iets langer. Een selectie van poorten scannen kan ook.

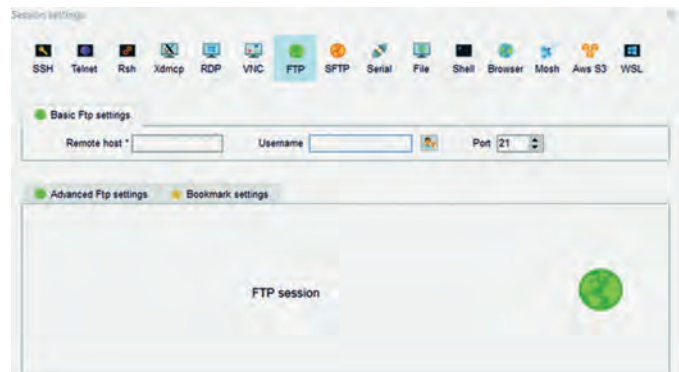
Servers

MobaXterm heeft ook een stukje gereedschap voor servers geïntegreerd. Je kunt bijvoorbeeld een FTP server, een VNC server of een HTTP server starten en stoppen. Misschien dat je het niet dagelijks nodig hebt, maar wel prettig dat het beschikbaar is.



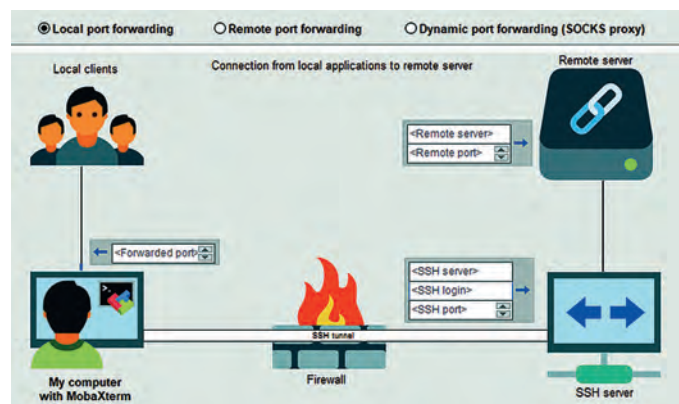
FTP en SFTP

Ik heb in het hoofdmenu als eerste een SSH verbinding gestart t.b.v. dit artikel. Misschien dat je niet van plan was SSH te gebruiken of het niet nodig hebt, maar ook een FTP en SFTP verbinding is standaard aan boord. In het hoofdmenu, Sessions, kies je dan voor FTP en SFTP. Gebruikersnaam en wachtwoord invullen en je bent 'ready to go'.



SSH tunneling

Het team van MobaXterm is er in geslaagd om de drie bekende soorten SSH tunneling (lokaal, remote, dynamisch) via een grafische interface aan te bieden. Via een SSH tunnel kun je data versturen van het ene netwerk naar het andere netwerk. Dat kan een privé netwerk zijn, maar ook een bedrijfsnetwerk.



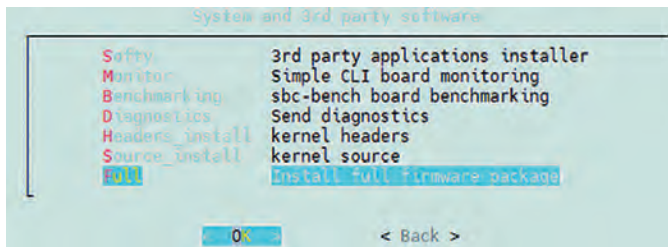
Wellicht niet voor iedere lezer weggelegd, maar het leek me goed om de mogelijkheid te benoemen.

Plug ins

In het begin van mijn artikel schreef ik dat MobaXterm ook plug ins aanbiedt. Een kleine greep uit de lange lijst: *CygUtils*, *Curl*, *DnsUtils*, *E2fsProgs*, *Emacs*, *Exif*, *FVWM2*, *MPlayer*, *Midnight Commander*, *Node.js*, *OpenSSL* of *Zip*. De eerstgenoemde, de bekende CygUtils, worden standaard meegeleverd.

Desktop

Maar hoe zit het dan met de desktop? Toen ik enkele jaren geleden voor het eerst aan de slag ging met een SBC, gebruikte ik een image dat standaard een desktop omgeving aan boord had. Ik had enerzijds een netwerkverbinding via RJ45 en anderzijds een beeldschermverbinding via HDMI. Wist ik veel dat dit allemaal niet nodig was. Armbian heeft dat heel fraai opgelost. Standaard is er geen desktop te schikbaar. Dit is wat men noemt *headless*. Via *Armbian config* kun je van de basisconfiguratie er een volledige configuratie (met desktop) van maken.



Voor wat hoort wat, dus het kost je meer schijfruimte en meer geheugen - en processorgebruik. Na een flink aantal probeersessies concludeer ik dat ik een desktop absoluut niet mis. Maar als je van plan bent om een SBC als desktopvervanger in te zetten, dan heb je de desktop natuurlijk wel nodig. De slogan van *Armbian Perfect for Server, Excellent for Desktop and Simple for Building* sluit als een bus.

Nog meer opties

Voor de echte diehards zijn er nog veel meer opties beschikbaar. Als je veel gebruik maakt van MobaXterm kun je ook macro's gebruiken. Ik heb deze mogelijkheid niet uitgetoond, maar ik stel me zo voor dat je repeterende werkzaamheden kunt vastleggen en kunt laten uitvoeren. Misschien een interessante optie voor een systeembeheerder?

Is er niets negatiefs te melden over MobaXterm?

Negatief is misschien een verkeerde typering, maar sommige opties zijn niet te configureren zoals jij dat misschien wilt. Bijvoorbeeld: de split terminal mode werkt net iets anders dan ik zou willen.

Niet storend, maar misschien ongewenst zijn de spelletjes. Althans voor mij dan. Misschien dat de ervaren systeembeheerder af en toe behoefte heeft aan een spelletje *Citris* (leuke woordspeling), *Mines*, *Soliterm* of *Sudoku*. De *Tux con solesaver* is natuurlijk wel een heel aardig onderdeel van het *Games menu*.

Misschien toch nog een minpuntje: wellicht dat de handleiding, ingebakken in de software, een beetje karig is.

Tot slot

Onderzoekt alles en behoudt het goede is een heel bekend spreekwoord. Ik heb aan de hand van een kleine computer praktijkvoorbeelden gegeven wat MobaXterm allemaal kan. Daar hoeft je thuis natuurlijk geen Chinese elektronica voor te gebruiken. Ook op een Raspberry Pi kun je MobaXterm loslaten. Maar een fan van de Chinese Pi met Armbian blijf ik wel.

MobaXterm is een prachtig stukje software met krachtige eigenschappen. Probeer het eens uit en wellicht dat het een plaatsje krijgt in je digitale gereedschapskist.



Links:

1. <https://en.wikipedia.org/wiki/PuTTY>
<https://bit.ly/2YVEODQ>
2. https://en.wikipedia.org/wiki/Secure_Shell
<https://bit.ly/3oFmt6Y>
3. https://en.wikipedia.org/wiki/ARM_architecture
<https://bit.ly/3qLlt3K>
4. <https://www.armbian.com/>
5. <https://rufus.ie/>
6. <https://mobaxterm.mobatek.net/>
7. <https://mobaxterm.mobatek.net/plugins.html>
<https://bit.ly/3CKltsw>



● Monochrome fotografie ●

Wessel Sijl

Wat is de beste weg naar een foto in zwart-wit?

In eerdere edities van de SoftwareBus heb ik uitleg gegeven over fotografie in zwart wit. Of beter gezegd, hoe je een digitaal fotobestand kunt overzetten naar zwart wit. Ik krijg nog vaak vragen over de beste methodes voor het verkrijgen van een pakkende zwart wit foto. Het is een onderwerp dat leeft, ook in deze moderne tijd met zijn vele digitale (foto) tools.

Zwart wit, of beter gezegd monochroom (één kleur ook zwart is een kleur), is nooit weggeweest. Sommige foto's lenen zich daarvoor heel goed, soms (veel) beter dan wanneer ze in kleur zouden worden getoond. In de reclame, in tijdschriften, tv of internet, wordt veel geleund op zwart wit. Dit zal er zeker aan hebben bijgedragen dat zwart wit een blijvertje is, en zich kan verheugen op een grote schare aan liefhebbers. Maar de hoofdvraag voor velen is hoe je het beste een digitaal fotobestand omzet naar zwart wit. Er zijn vele wegen naar Rome, en graag maak ik de lezer deelgenoot van de methodes die ik graag hanteer, en idem de dingen die ik liever vermijd.



Vele wegen van kleur naar zwart wit

Op locatie

Een digitale camera maakt uitsluitend foto's in kleur op basis van drie (soms vier) primaire kleuren. De beeldgevoelige beeldchip (CMOS of CCD) is op basis daarvan ontworpen. Een zwart wit foto is dus onderhuids een kleurenfoto. Ik ken slechts één camera die echt uitsluitend foto's in zwart wit vastlegt, en dat is de Leica M10 Monochrome; een zeer dure camera (meer dan 8000 euro), met een speciaal ontworpen beeldchip en dito firmware, voor de ware liefhebber. Ik ga verder niet in op deze bijzondere camera, behalve dat zijn prestaties adembenemend goed zijn.

(Als ik ooit nog eens de loterij win... staat deze hoog genoemd op mijn verlanglijstje).



De Leica M10 Monochrome

*Fantastische camera voor fotografie in zwart wit.
Voor velen financieel een brug te ver*

Alle andere digitale camera's kennen een instelling voor zwart wit, vaak in combinatie met filterinstellingen voor specifieke effecten door verschuivingen in de grijswaarden (vergelijkbaar met de gekleurde filters geel, oranje, rood, groen en blauw veelal bekend van de analoge fotografie in zwart wit, maar nog steeds actueel in de digitale wereld). Sommige, wat meer uitgebreide camera's, hebben de mogelijkheid om achteraf in de camera foto's om te zetten naar zwart wit; een soort (beperkte) digitale doka in de camera. Ikzelf gebruik deze opties zelden of nooit. De reden is simpel: de omzetting kun je thuis veel beter doen, met de eigen computer, goede toepassingen, en een groot beeldscherm in plaats van werken op het kleine LCD van de camera. In de regel heb je met de fotoprogramma's thuis op de computer RAW converter of bewerkingsprogramma veel meer verfijnde mogelijkheden voor een geslaagde conversie. Heel soms gebruik ik een van de twee opties in de camera toch wel, hetzij direct vastleggen in zwart wit, of na de op name omzetting naar zwart wit in de camera. Dat doe ik in geval van twijfel, als ik zeker wil weten of een bepaald motief zich wel of niet leent voor presentatie in zwart wit. Als je eenmaal zeker weet dat kleur dan wel zwart wit beter is, kun je desgewenst op locatie nog zaken zoals compositie of belichting daarop aanpassen. Maar in 99 % van de gevallen doe ik de omzetting naar zwart wit het liefst thuis op de computer.

Welke manier van converteren

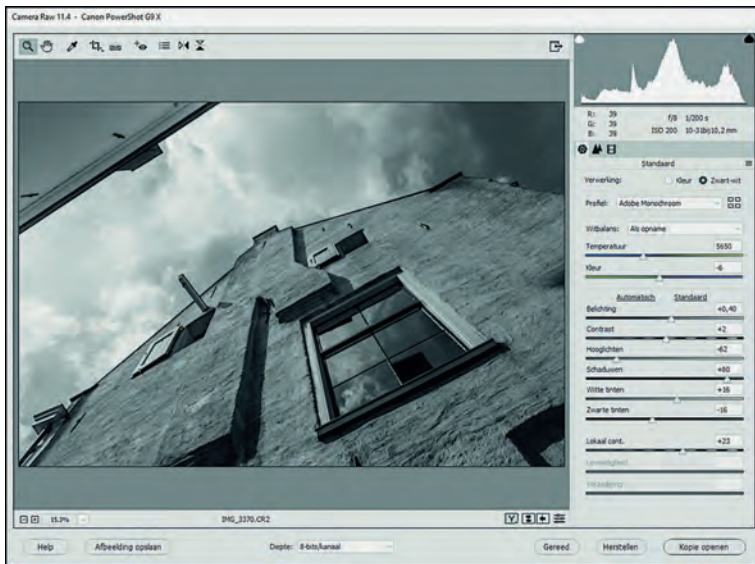
Voor conversie naar zwart wit leiden, zoals hiervoor al aangehaald, meer wegen naar Rome. Dat geldt zowel bij conversie van ruwe fotobestanden met de RAW converter (dat wil zeggen als je fotografeert in RAW iets wat ik zeer aanraad), als bij de conversie in het bewerkingsprogramma. Beide opties zal ik hieronder nader uit de doeken doen, inclusief de voor- en nadelen.

RAW conversie

De omzetting naar zwart wit met de RAW converter is een prima optie. Ik doe het zelf niet zo veel; ik prefereer om dat een fase later te doen in het fotobewerkingsprogramma.

Daarop kom ik later terug, waarom dat precies mijn voorkeur heeft.

Maar terug naar de RAW converter. Menige toepassing voor het converteren van RAW bestanden beschikt over een fraaie optie voor omzetting naar zwart wit. Het grote voordeel is dat de kleureninformatie (meestal afhankelijk van welke RAW converter je gebruikt) intact blijft. Dat behoud van kleureninformatie is erg belangrijk voor de later in te zetten filtereffecten. Bij veel RAW converters kun je de bekende filters voor zwart wit inzetten, zoals een geel of oranje filter voor een donkergrijze weergave van de blauwe lucht, zodat witte wolken er fraaier tegen afsteken. Vaak kun je ook de sterkte van die filters variëren. Verder zijn sommige RAW converters, zoals bij CameraRAW van Adobe, in staat om de structuur in de foto te verhogen; een manier om een foto meer diepte en drama te geven. En verder, heel belangrijk, is omzetting naar zwart wit via de RAW converter een goede basis voor behoud van kwaliteit met de minste kans op artefacten.



Omzetting naar zwart wit met de RAW converter
In dit voorbeeld is Adobe Camera Raw gebruikt

Echter, je mist in menig RAW converter wel de verfijningen die je kunt bereiken met een gewoon fotobewerkingprogramma. Zo kun je bijvoorbeeld een foto niet in delen converteren naar zwart wit, bijvoorbeeld de lucht met een oranje filter effect en de ondergrond met een groenfilter voor meer nuances in groene natuurtinten. Dat is de voornaamste reden waarom ik (meestal) niet de conversie doe met de RAW converter, maar een fase later in het fotobewerkingprogramma.

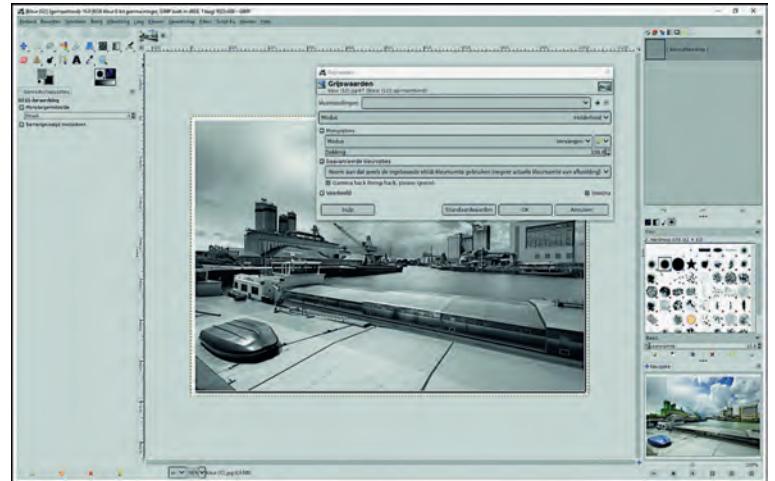
Fotobewerking

Dit is mijn favoriete plek of moment voor de omzetting van een digitaal fotobestand naar zwart wit. Dat wil zeggen als de kwaliteit het toelaat, dus met geringe kans op artefacten. Is dat risico wel aanwezig, dan doe ik de omzetting toch maar liever met de RAW converter, wat in beginsel veel beter behoud van kwaliteit garandeert. Maar goed, als de basiskwaliteit van het fotobestand in orde is (bij voorkeur op basis van een TIF bestand!) voor al of niet stevige bewerkingen, dan doe ik de omzetting naar zwart wit in het fotobewerkingprogramma (in mijn geval GIMP, soms Photoshop).

Wat ik NIET doe (en wat ik iedereen ook afraad) is de standaard omzetting naar grijswaarden. Vrijwel alle programma's beschikken over die optie. De reden waarom ik deze optie niet gebruik, is dat daarmee alle (onderhuidse) kleureninformatie verwijderd wordt. Het is dan alleen maar een bestand op basis van grijswaarden, en daarmee is er geen

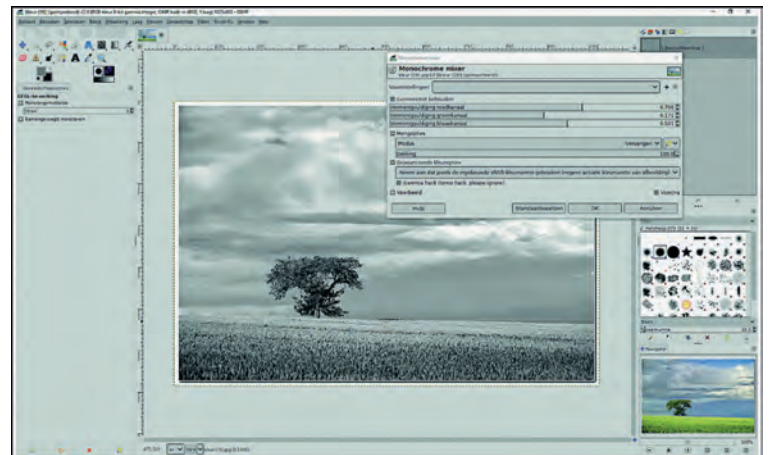
weg meer terug. Een creatieve beïnvloeding van de grijs waarden, met filters, is niet meer mogelijk. Ook andere bewerkingen zijn niet meer of slechts beperkt mogelijk. Je moet het dan doen met wat je (nog) hebt.

Er is nog een andere, min of meer vergelijkbare optie, en dat is de foto volledig desatureren, oftewel ontdoen van kleur (meestal met een schuifbalk). In feite is dat hetzelfde als de voornoemde omzetting naar grijswaarden. Ook dan is het resultaat dat de kleureninformatie voor creatieve aanpassingen van de grijs tinten volledig is verdwenen. Op zichzelf kunnen deze methoden best wel een prima resultaat opleveren, maar voor de kritische fotograaf die méér wil, of later nog andere creatieve dingen wil doen met de foto (zoals bij nader inzien de foto in duotoon presenteren), is het geen aanrader.



Omzetting naar zwart wit met 'Grijswaarden'
Dit kan goed werken, maar er zijn betere opties

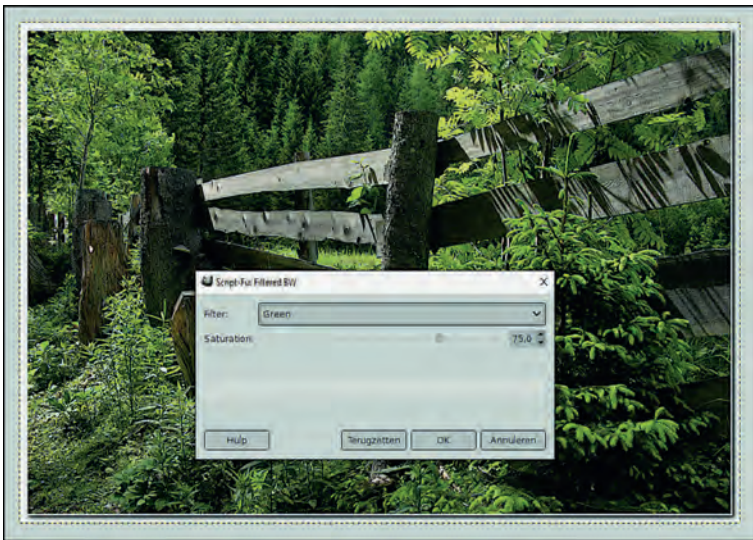
Gebruik dus bij voorkeur een optie die de (onderhuidse) kleureninformatie behoudt. De optie met de meeste flexibiliteit voor een mooie omzetting naar zwart wit is de kanaalmixer. Ook wel genoemd monochrome mixer. In bijna alle programma's die een kanaalmixer bevatten, is de werking hetzelfde: monochroom aanvinken en daarna de gewenste filtering met de kanalen (schuifbalken) rood, groen en blauw instellen. Sommige programma's hebben zelfs twee soorten kanaalmixer: een algemene en een specifiek voor monochroom (zoals bij GIMP). Het werken met de kanaalmixer vergt wel enige gewenning; velen ervaren het als lastig, alhoewel dat best meevalt. Alles kan worden gevolgd op het scherm, en als het effect niet bevalt ga je terug naar het startpunt en begin je gewoon weer opnieuw. De belangrijkste 'hobbel' die je moet nemen, is wat kennis van kleuren opdoen. Bijvoorbeeld dat geel complementair is aan blauw. Een filtering met geel zal bijvoorbeeld de blauwe lucht niet licht maar wat donkerder grijs maken, en omgekeerd zal een blauw filter de lichtgrijze lucht vrijwel wit maken.



De monochrome kanaalmixer biedt meer flexibiliteit, maar vergt wat gewenning

En verder is het gewenning door te ondervinden in welke mate het grijs door een filtering wordt beïnvloed. Heb je eenmaal wat vaardigheid opgedaan met de kanaalmixer, dan zijn de mogelijkheden eindeloos.

Als de kanaalmixer toch te moeilijk of omslachtig wordt ervaren, zijn er prima andere, veel eenvoudiger opties, die feitelijk hetzelfde doen (op basis van de niet zichtbare kanaalmixer). Dat kunnen ofwel standaardonderdelen van een fotoprogramma zijn, ofwel een plug in. De instelling gaat simpelweg met een keuze uit beschikbare filters, en vervolgens wordt de keuze bevestigd. Deze eenvoudige en snelle opties leveren in de praktijk goede kwaliteit. Sommige RAW converters hebben vergelijkbare mogelijkheden, en ook daarbij wordt (onderhuids) geleund op een kanaalmixer. Als ik een snelle conversie wil doen naar zwart wit, hanteer ik als gebruiker van GIMP graag de plug in 'Filtered BW'; snel, eenvoudig en goed. Gewenst filter kiezen, eventueel de sterkte van het filtereffect aanpassen (de standaardinstelling van 75 % is al vrij goed), en op OK klikken.



Met Filtered BW kan een snelle en goede omzetting naar zwart wit worden bereikt

De finishing touch

Zodra een goede omzetting naar zwart wit is verkregen, kunnen nog wat extra effecten worden toegevoegd om een foto meer zeggingskracht te geven. Bij zwart wit foto's komt het immers relatief sterk aan op gevoel en soms ook drama, en dan zijn aanvullende verbeteringen heel nuttig.

Kijk, om te beginnen, dus na de omzetting naar zwart wit, nog eens kritisch naar de levels en het contrast, en speel daar wat mee. Ook het spelen met de curve, of de instelling 'schaduw / hoge lichten' (o.a. GIMP en Photoshop beschikken over zo'n optie), kan dat stukje extra dimensie geven.

Vaak komt dat een zwart wit foto ten goede. Dit soort bewerkingen kun je uiteraard ook doen op bepaalde (geselecteerde) delen van de foto. Wat ik bijvoorbeeld vaak doe, is bij landschapsfoto's de lucht en de ondergrond met verschillende filtereffecten omzetten naar zwart wit. Zoals oranje filter voor de lucht en groenfilter voor de ondergrond. Maar uiteraard zijn veel meer combinaties mogelijk, afhankelijk van het soort foto en wat je er mee wil uitdrukken.

Een andere optie is het verscherpen. Normaliter ben ik daar mee wat terughoudend. Je kunt de verscherping ook over drijven. Maar bij een zwart wit foto kan het vaker en nuttiger worden ingezet in vergelijking met een kleurenfoto. Een fraaie manier van verscherpen die ik zelf graag gebruik, is de optie 'Unsharp Mask'. Vrijwel alle fotoprogramma's beschikken daarvoor.

Een ander effect dat ik graag toepas bij een zwart witfoto, is het versterken van de structuur. Ik heb er eerder over geschreven in de SoftwareBus. Het kan een foto beduidend meer kracht en drama geven, en levert vaak ook betere doortekening op in donkere en lichte partijen. Uiteraard geldt ook hier, net als bij verscherpen, dat overdaad kan schaden. Probeer simpelweg proefondervindelijk vast te stellen hoeveel je de structuur het beste en naar smaak kunt versterken.

Nog even in een nutshell hoe ik dat doe in GIMP: (1) maak een tweede laag aan met de optie 'Laag dupliceren', (2) ga vervolgens naar het 'Hoogdoorlaatfilter' dat bevindt zich onder het rolmenu 'Verbeteren', (3) klik op OK en ga weer terug naar de duplicaatlaag, (4) kies uit het lagenmenu voor 'Bedekken', en (5) als het naar je zin is, stapel de lagen op elkaar (kies voor 'Neerwaarts samenvoegen' onder het rolmenu 'Lagen').

Met deze handelingen het vereist enige gewenning, maar doe het gewoon, en bouw zo ervaring op kan je tot een fantastisch eindresultaat van de zwart witfoto komen, waar gevoel en drama bij wijze van spreken kunnen afdruppen.



Foto met een versterkte structuur en verscherpt

NIK Silver Efex Pro

Er is nog een (fantastische en goede) manier om vanuit het fotobewerkingsprogramma een omzetting te doen naar zwart wit, en dat is de plug in NIK. En dan in het bijzonder het onderdeel Silver Efex Pro. Deze plug in gebruik ik in combinatie met Photoshop Elements. Het NIK onderdeel Silver Efex Pro biedt enorm veel opties om naar smaak te converteren naar zwart wit. Met of zonder verhoging van dynamisch bereik, structuur, korreffect van een traditionele zwart witfilm, tintje (bijvoorbeeld sepia), enz. enz.

En dat allemaal traploos in te stellen, op intuïtieve wijze. NIK is niet heel goedkoop, en kan niet in alle fotoprogramma's worden geïnstalleerd (het kan bijvoorbeeld niet met GIMP samenwerken, maar wel met Photoshop) maar het levert adembenemende en kwalitatief zeer goede resultaten op. Als ik met GIMP (waaronder de hiervoor genoemde handelingen) niet helemaal uit de voeten kan, mag ik graag overschakelen naar Photoshop en vervolgens NIK Silver Efex Pro inzetten.





Met NIK Silver Efex Pro kan fantastisch resultaat worden bereikt

Tot zover de methodieken die ik toepas voor het verkrijgen van een foto in zwart wit. Uiteraard zijn er veel meer mogelijkheden en toepassingen om tot een pakkend resultaat te komen. Maar ik hoop dat deze beschouwing een prikkel is voor de lezer om eens wat (meer) te gaan experimenteren met zwart wit. Het is echt heel bijzonder, vooral als het in fine art wordt geprint en gepresenteerd.

Hiermee is nog niets gezegd over de verdere finetuning in fotobewerkingsprogramma's. Denk daarbij bijvoorbeeld aan het optimaliseren van de middentonen, ruis verminderen, duo toon toepassen (wat maar weer eens bewijst hoe belangrijk het behoud van onderhuidse kleurinformatie is), enz.. In een volgend artikel ga ik nader op zulke finetuning.

Het eindresultaat, waarvoor je het doet



● Firefox isoleren in een Linux-cgroup ●

Johan Swenker

Dit artikel is geïnspireerd op een probleem van Jan¹, een van de leden van de HCC Linux-werkgroep in Groningen. Zijn Linux-computer crasht op (on)regelmatige tijden. De computer wordt dan door het swappen zodanig onbruikbaar dat er weinig anders op zit dan die maar uit te zetten. Binnen Linux kun je processen van elkaar isoleren met cgroups. Maar hoe werkt dat?

Linux Weekly News

Ik ben al sinds jaren abonnee van <https://lwn.net>. Dit is een webzine dat over Linux schrijft. Abonnees krijgen elke donderdag een nieuwe uitgave. Voor andere mensen komt dezelfde informatie met een week vertraging beschikbaar. Door LWN kende de kreet *cgroup*; ik herinner me dat ik een keer met cgroups geëxperimenteerd heb die het cpu gebruik van processen beperken. Maar verder heb ik een zoekmachine nodig om me de details te vertellen.

Control group

Het sleutelwoord waar het om gaat is dus control group (cgroup). Met cgroups kun je processen beperken met betrekking tot cpu, cpu gebruik, geheugen en vele andere zaken. Dit mechanisme wordt ook gebruikt bij containers. Voor ons probleem willen we een applicatie niet voor 100% isoleren; isoleren op geheugen zou voldoende moeten zijn. Bij de vorige bijeenkomst richtten we onze aandacht op Wine. Inmiddels gebruikt Jan met veel plezier de Linux applicatie *nomacs*. Afgelopen bijeenkomst vonden we Firefox verdacht. Nee, we begrijpen nog steeds niet wat Jan anders doet dan alle andere Linux gebruikers.

Het plan is nu als volgt: maak een cgroup die aan Firefox een beperkte hoeveelheid geheugen geeft, en zorg ervoor dat je dit niet bij elke reboot handmatig opnieuw moet instellen. Firefox kan er dan niet meer voor zorgen dat de computer onbruikbaar wordt.

Als Firefox toch meer geheugen wil gebruiken dan geconfigureerd is, dan moet het operatiesysteem ingrijpen. Linux heeft hiervoor een *out of memory killer* (*oom killer*). De oom killer breekt processen af als het systeem out of memory is. Dit gebeurt als het systeem fysiek geheugen te kort komt. De oom killer stopt ook processen als de grenzen van een cgroup overschreden dreigen te worden. Binnen Firefox zal dan een tabblad crashen.



Linux Journal

Na enig zoeken vond ik twee websites die ik heel leerzaam vond. Op de website van het Linux Journal staat een meerde

1. Echte naam bekend bij de auteur en alle leden van de Linuxwerkgroep Groningen.

lige serie artikelen over containers en cgroups. In het eerste deel <https://www.linuxjournal.com/content/everything-you-need-know-about-linux-containers-part-i-linux-control-groups-and-process> wordt uitgelegd hoe cgroups werken, het gaat nog niet over containers.

In een voorbeeld wordt een cgroup aangemaakt die een geheugenbegrenzing heeft tot 50 Mbyte. De applicatie is een simpel bash script dat elke minuut 'hello world' afdrukt. Dat past in 50 Mbyte. Als dat proces in die cgroup geplaatst wordt, dan gebeurt er ogenschijnlijk niets. Dat was heel anders toen de cgroup beperkt werd tot 4 kbyte: de oom killer greep direct in.

In de tweede helft van dat artikel staat hoe je dit beter of handiger kunt doen met de programma's uit cgroup tools. Daar staat ook hoe je zo'n cgroup automatisch kunt laten aanmaken bij het booten via systemd.

Voor mij was dit een mooie inleiding om een en ander te leren begrijpen. Maar nu moet ik dat nog toepassen op Firefox en alle componenten die door Firefox opgestart worden. He las is dat wel nog iets wat ik zelf moet doen. De programma's daarvoor zijn er wel, maar Ubuntu heeft niet een simpel knopje 'beperk Firefox tot maximaal 2 Gbyte'.

Microsoft Teams

Op de website van Remy van Elst kwam ik het volgende artikel tegen: https://raymii.org/s/articles/Limit_specific_process_memory_on_desktop_linux_with_cgroups.html Remy wil Microsoft Teams gebruiken onder Linux, maar hij wil er geen last van hebben als dat programma te veel geheugen gebruikt.

Earlyoom

Halverwege zijn artikel maakt Remy eerst een uitstapje naar het pakket *earlyoom*. Earlyoom is een out of memory killer. De manual page van earlyoom geeft aan waarom earlyoom beter is dan de standaard oom killer van de kernel. Je kunt die manual page lezen voordat je earlyoom installeert: <http://manpages.ubuntu.com/manpages/bionic/man1/earlyoom.1.html>

Misschien dat earlyoom het verschijnsel waar Jan last van heeft al oplost.

Het artikel van Remy gaat verder over het beperken van het geheugengebruik van processen met cgroups; hoe je dat moet activeren bij het booten van de computer en hoe je MS Teams, maar ook bijvoorbeeld Firefox, automatisch in de juiste cgroup plaatst.

Raspbian en Ubuntu

Om het geheugen te limiteren moet je, als root, het maximaal toegestane geheugengebruik schrijven naar een bestand ergens diep in `/sys`. Op mijn Raspberry bleek dat be

stand niet te bestaan. Dit is op te lossen door `cgroup_memory=1 cgroup_enable=memory` toe te voegen aan `/boot/cmdline.txt`.

Als je bovendien het gebruik van het swap geheugen wilt limiteren, moet je de totale geheugenlimiet `ram+swap` schrijven naar een bestand met de naam: `memory.memsw.limit_in_bytes`. Dit bestand bestaat in Ubuntu niet. Je wilt toch echt ook het swap geheugen limiteren, anders gaat het systeem swappen en traag worden, voordat het proces dat zich misdraagt gestopt wordt.

Op stackexchange: <https://unix.stackexchange.com/questions/44985/limit-memory-usage-for-a-single-linux-process/125024#125024> staat hoe je dit, als root, kunt oplossen:

```
pas /etc/default/grub aan:
GRUB_CMDLINE_LINUX_DEFAULT="" →
GRUB_CMDLINE_LINUX_DEFAULT="cgroup_enable=memory
swapaccount=1"
```

en voer de volgende commandos uit:

```
apt install cgroup-bin
update-grub
```

Na een reboot kun je nu ook het totale geheugengebruik `ram+swap` limiteren.

Een cgroup van 2 GByte

Ik heb een en ander uitgeprobeerd. De stappen die ik, als root, heb uitgevoerd zijn de volgende:

```
apt install cgroup-tools
cgcreate -t johan:johan -a johan:johan -g
memory:/cgTEST
echo 2048m > /sys/fs/cgroup/memory/cgTEST/
memory.limit_in_bytes
echo 2048m > /sys/fs/cgroup/memory/cgTEST/
memory.memsw.limit_in_bytes
```

Dit betekent dat een cgroup gemaakt wordt, die gekoppeld is aan de Linux gebruiker `johan`² en de Linux groep `johan`. Applicaties in die cgroup mogen samen maximaal 2 GByte `ram` gebruiken. Ook `ram+swap` is gelimiteerd tot 2 GByte. De gewone gebruiker `johan` kan dit testen:

```
cgexec -g memory:cgTEST bash
</dev/zero head -c 2050m | tail
```

2. Denk eraan dat je dit aanpast voor jouw situatie.

Dit betekent dat `bash` wordt uitgevoerd in de cgroup `cgTEST`. Het programma `head` kopieert 2050 MByte nullen vanuit `/dev/zero` naar de `pipe`.

Het programma `tail` zal de laatste tien regels afdrukken, maar krijgt een enkele regel van 2050 MByte aangeboden. Dus `tail` zal meer gebruiken dan de beschikbare 2 GByte. Het proces wordt inderdaad netjes door de kernel gestopt.

Als je `cgTEST` toch enige swapspace wilt geven, dan kun je `memory.memsw.limit_in_bytes` natuurlijk vergroten. Je zult dan zien dat `tail` bijna 2 Gbyte gaat gebruiken, en dat swap space in gebruik genomen wordt, terwijl het geheugen van de computer nog niet vol is.

Permanent maken

Nu we dit getest hebben, kunnen we `cgTEST` permanent maken met `cgconfigparser`. Wederom als root, maak het bestand `/etc/cgconfig.conf` aan met als inhoud:

```
group cgTEST {
    perm {
        admin {
            uid = johan;
        }
        task {
            uid = johan;
        }
    }
    memory {
        memory.limit_in_bytes = 2048m;
        memory.memsw.limit_in_bytes = 3072m;
    }
}
```

Let op, dit is weer gerelateerd aan de gebruiker `johan`, zoals die op mijn systeem bestaat! Kijk voor makkelijk knippen en plakken op de website van Remy. Je kunt dan ook zien wat ik heb aangepast om het voor de gebruiker `johan` te laten werken.

Eerst maar even `cgconfigparser` handmatig opstarten om te controleren dat er geen typfouten in de configuratiefile zitten:

```
cgconfigparser -l /etc/cgconfig.conf
```

`Systemd` is de moderne manier om programma's op te starten bij het booten van de computer.

Om `cgconfigparser` automatisch uit te voeren heb ik het bestand `/lib/systemd/system/cgconfigparser.service` aangeemaakt met de volgende inhoud:

```
[Unit]
Description=cgroup config parser
After=network.target

[Service]
User=root
Group=root
ExecStart=/usr/sbin/cgconfigparser -l /etc/
cgconfig.conf

Type=oneshot

[Install]
WantedBy=multi-user.target
```

Nu nog even aan `systemd` vertellen dat deze service actief moet zijn:

```
systemctl enable cgconfigparser
systemctl start cgconfigparser
```

Nu wordt bij een reboot de cgroup `cgTEST` telkens opnieuw aangemaakt, met limieten van 2 GByte voor `ram` en 3 Gbyte voor `ram+swap`.

Firefox in cgTEST

Voor mijn demonstratie moet ik er nu nog voor zorgen dat Firefox in deze cgroup geplaatst wordt. Er zijn verschillende opties:

- Firefox voortaan opstarten vanuit de commandoregel met `cgexec -g memory:cgTEST firefox`
- in een configuratiefile zoals `/usr/share/applications/firefox.desktop` aangeven dat Firefox moet worden opgestart als `cgexec -g memory:cgTEST firefox`
- een `cronjob` maken die elke minuut alle Firefox processen in de juiste cgroup stopt met `cgclassify -g memory:cgTEST $(pidof firefox)`
- er is een 'control group rules engine daemon': `cgrulesengd`. Dit is iets fraaier, en waarschijnlijk ook efficiënter en effectiever dan de `cronjob`.

Het artikel van Remy eindigt met de beschrijving van `cgrulesengd`. Daar heb ik voor gekozen: ik heb eerst als root de configuratie file `/etc/cgrules.conf` voor `cgrulesengd` gemaakt. Deze file moet uit 1 regel bestaan

```
johan:firefox memory cgTEST/
```

Deze regel zorgt ervoor dat “firefox” van de gebruiker `johan` in de cgroup `cgTEST` komt. De stappen om ervoor te zorgen dat `cgrulesengd` opstart bij het booten van de computer zijn, als root: maak het bestand

```
/lib/systemd/system/cgrulesengd.service
```

```
[Unit]
```

```
Description=cgroup rules generator
```

```
After=network.target cgconfigparser.service
```

```
[Service]
```

```
User=root
```

```
Group=root
```

```
Type=forking
```

```
EnvironmentFile=-/etc/cgred.conf
```

```
ExecStart=/usr/sbin/cgrulesengd
```

```
Restart=on-failure
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Kopieer een standaard configuratiefile, en vertel `systemd` dat `cgrulesengd` als daemon moet blijven draaien:

```
cp /usr/share/doc/cgroup-tools/examples/
    cgred.conf /etc/cgred.conf
systemctl enable cgrulesengd
systemctl start cgrulesengd
```

En inderdaad, na een reboot werden alle Firefox processen in `cgTEST` geplaatst.

Nog een paar losse opmerkingen over deze cgroup.

In `/sys/fs/cgroup/memory/cgTEST/tasks` maar ook in `/sys/fs/cgroup/memory/cgTEST/cgroup.procs` staan de proces ids van alle processen in deze cgroup.

Als een proces al draait, dan kan root het achteraf alsnog in een cgroup stoppen:

```
echo $PID > /sys/fs/cgroup/memory/cgTEST/
    cgroup.procs
cgclassify -g memory:cgTEST lijst_van_proces_ids
```

Games maken (3)

René Suiker

Terugblik

We hebben 2021 overleefd en leven nu in 2022. Ik wens u al len veel gezondheid, geluk, liefde en computerplezier. Intussen zijn we toe aan episode 3 van dit nieuwe feuilleton voor de SoftwareBus en we blikken dus heel even terug.

In 2021 begon ik met een kleine oproep, kreeg contact met de IG Programmeren, en samen schreven we een volgend stukje. Dat wil zeggen, Marco Kurvers legde uit en ik schreef op. En zo gaan we waarschijnlijk nog wel even verder, maar dat terzijde. Intussen heeft het tweede artikel ook weer een nieuwe aanmelding opgeleverd, dus misschien komt een Plat form GameDesign wel in zicht.

Vorige keer zijn we in een uitgebreide inleiding ingegaan op de installatie van Unity, de betekenis van de Unity Hub en hebben we vooral het hoofdscherm, de editor, van Unity bekeken. Ook hebben we het even gehad over de game industrie, welke rollen je allemaal hebt en dat het echt een serieuze industrie is.

Ten slotte ging ik nog in op de samenwerking met de IG Programmeren en dan met name de werkgroep Unity daarbinnen.

Deze keer

Zoals ik eerder aangaf, wilden we beginnen met iets te laten werken nog voordat we er zelf mee gingen programmeren. Gewoon Unity laten doen wat het doet en dan kijken wat we kunnen bereiken. We hebben wel wat elementair programmeren (in C#) in het vooruitzicht, maar we wilden niet al te grote stappen in één keer zetten.

Mijn idee was om een soort deeltjesversneller te bouwen, een (2D) box, waarbinnen een balletje stuitert en dat stuitert steeds harder en op enig moment komt het bij een gat in de doos en schiet daar doorheen.

Overigens ga ik elders in dit nummer ook kijken hoe ik dat met Scratch kan doen. In Scratch moet je alles zelf programmeren, Unity regelt een heleboel zaken zelf en voor dit project hoeven we niet te programmeren. Dat lijkt dus makkelijker, maar persoonlijk vind ik het in Scratch iets een vuidiger. Wel heb je met Unity veel meer mogelijkheden. Maar, alles op zijn tijd.

Natuurwetten

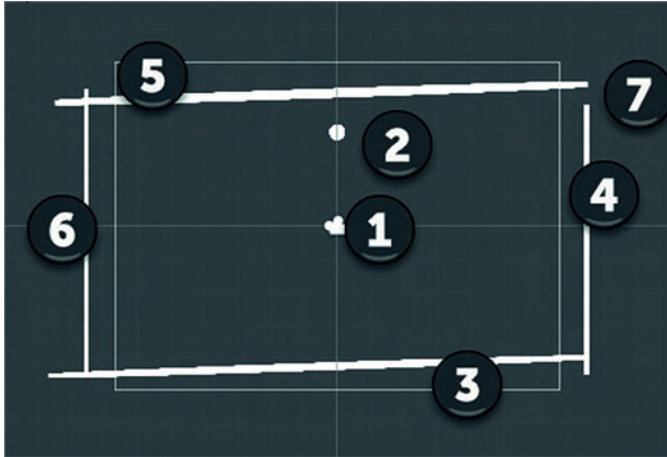
In Scratch moet je dus alles zelf programmeren, in Unity kan je je eigen wereld configureren. Inclusief de wetten van de mechanica die er gelden. Je kunt bij wijze van spreken een negatieve zwaartekracht implementeren en je kunt dus ook zonder verliezen laten stuiteren, zelfs met versnelling. En dat gaan wij ook doen, stuiteren met versnelling. De zwaartekracht laten we wel zijn werk doen, dus wij kunnen een balletje laten vallen en als het de vloer raakt, dan stui tert het omhoog. Daarbij laten we de bal iets harder omhoog komen dan waarmee hij op de grond kwam.

In een ideale wereld, als een balletje naar beneden valt en daar stuitert, dan stuitert het weer recht omhoog. Zonder programmeren kunnen we dat niet veranderen, daarom be ginnen we met het tekenen van de box met een wat schuin plafond en een schuine vloer, dan stuitert het balletje in wat wisselende richtingen.

Vervolgens gaan we een balletje definiëren en bepalen we zowel voor de box als voor de bal de van toepassing zijnde natuurwetten. We plaatsen dan het balletje ergens in de box en als we het programma laten lopen, dan zal het balletje vallen en gaan stuiteren. Dat stuiteren gaat steeds harder en omdat de box niet recht is, stuitert het balletje alle kanten op en op enig moment komt de bal bij het gat en dan schiet hij eruit. Althans, dat is onze theorie. De praktijk blijkt weerbarstiger. Maar, we hopen hiervan te leren.

De start

Goed, laten we maar gaan beginnen. Ik bouw een speelveld op in de 'scene'. Dat is in het midden van het Unity scherm. Ik heb het project 'SoftwareBus deel 3' genoemd, omdat het derde artikel in de reeks dit project beschrijft. De bedoeling is dat ik het iets korter houd dan de vorige keer, want normaal staat er zo'n vier pagina's per onderwerp gepland. Uiteraard kan het incidenteel wel eens iets meer zijn, maar we willen het blad graag afwisselend houden, dus als ik nu al acht pagina's vul met Scratch en Unity, dan is het wel weer genoeg. Het speelveld:



Figuur 1 Het speelveld

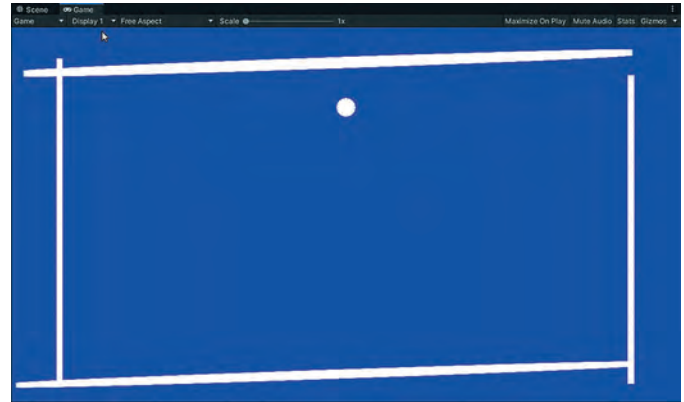
Bij (1) zie je de camera. Deze zorgt ervoor dat er in de spelmodus iets te zien is. Als je, zoals hieronder straks afgebeeld, de spelmodus selecteert, dan zie je de camera niet meer, maar dan zie je wat de camera ziet.

Bij (2) zie je de bal. We werken in 2D, dus je ziet een cirkel en meer is het niet. Ik ga straks laten zien wat er allemaal bij komt kijken om dit zo op te zetten.

Bij (3) zie je de vloer. Zoals ik aangaf, die staat niet recht, want we willen de bal een beetje door de ruimte laten stuiten en als we alles recht houden, dan stuitert hij alleen maar recht op en neer, of ik moet met andere complexe effecten gaan werken en ik wil het in het begin eenvoudig houden.

De muren (4) en (6) heb ik wel recht gehouden: omdat er een zwaartekracht actief is wordt er op de wanden toch niet exact haaks gestuiterd en bovendien is de afwijking via vloer en plafond al genoeg om de bal semi random te laten stuiten.

Bij (5) dus het plafond en bij (7) ten slotte de opening, want het idee is dat het balletje door rond te stuiten overal een keer komt en dan hier eruit schiet. Dat is dus de theorie. Als je dit in de spelmodus bekijkt, ziet het er aldus uit:



Figuur 2 Speelveld in spelmodus

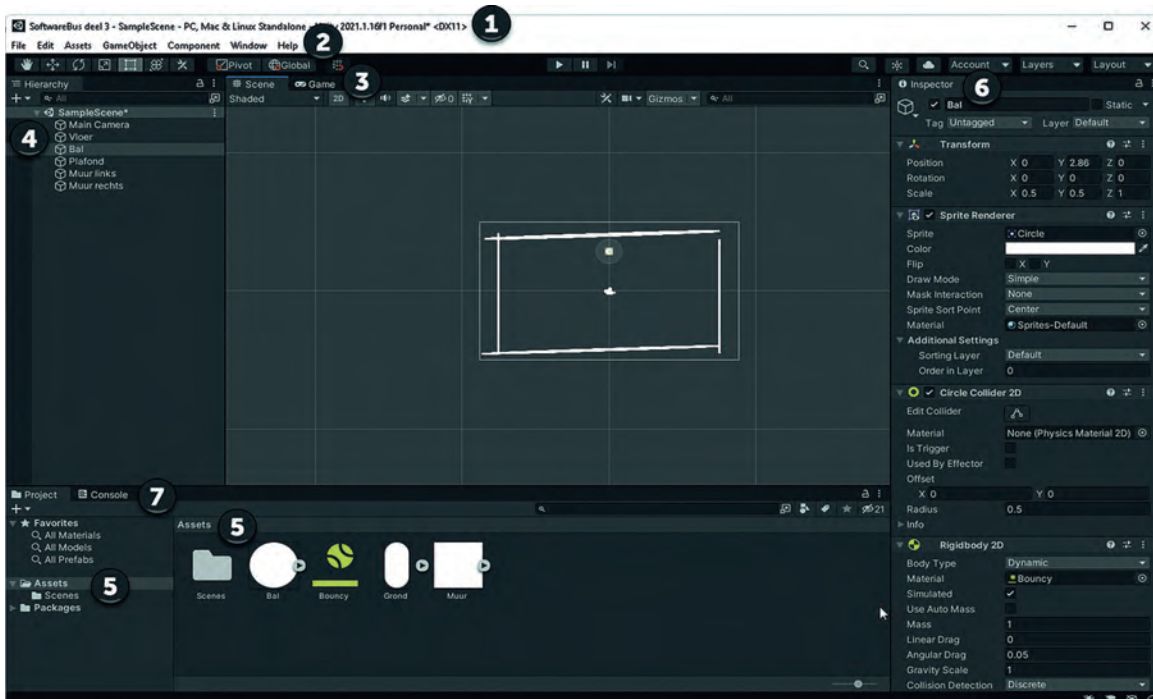
Als je goed linksboven kijkt, zie je dat de 'game' mode is geselecteerd. Links daarvan zie je 'scene' staan, daarin bouw je het scenario op en dat was de basis voor figuur 1.

En als ik de hiërarchie laat zien, dan kom je dezelfde onderwerpen weer tegen, want je kunt zowel via de hiërarchie als via de scene zaken regelen. Daarover straks meer.



Figuur 3 Hiërarchie

Maar dit staat er niet zomaar, daar hebben we iets voor moeten doen, want als je met een nieuw project begint, dan heb je in de 'SampleScene' (het voorbeeldscenario) alleen maar de camera.



Figuur 4
Unity scherm

We gaan het dus opbouwen. Ik laat daarom nog even het Unity scherm zien, want we kunnen op verschillende manieren bereiken wat we willen bereiken en dat wil ik stap voor stap uitleggen. (Zie *Figuur 4 onderaan de vorige pagina*)

Bij (1) zie je de titel van je project. Daarbij zie je ook welke versie van Unity gebruikt is. Zoals ik eerder al aangaf, je kunt meerdere versies van Unity tegelijk in je hub hebben.

Bij (2) zie je de menubalk, daar gaan we straks even wat dieper op in. Ik gaf al eerder aan: vaak zijn er meerdere manieren om iets in Unity te bereiken. Bijna alles wat je wilt doen kun je ook via de menubalk bereiken. We gaan de mogelijk heden straks in enige mate van detail bespreken.

Bij (3) kun je kiezen of het venster eronder de scene weergeeft waar je in kunt editen, of dat het de spelmodus weergeeft, waarbinnen je dus kunt spelen.

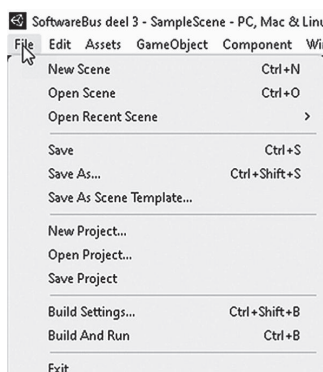
Bij (4) zie je de hiërarchie van de spelobjecten. Zoals ik reeds eerder aangaf is in feite alles in Unity een game object. Op dit moment hebben we één scene en daaronder hangen alle objecten in deze scene, zonder verdere hiërarchie. Je zou je kunnen voorstellen dat we nog een object 'doos' zouden maken, waaronder dan de objecten 'vloer', 'plafond' en de beide muren zouden hangen. Dan kun je sommige bewerkingen op het hele object uitvoeren, maar je kunt ook nog steeds bij de individuele objecten. Hierover volgt in een later stadium nog meer informatie, maar ik wil vooral niet alles tegelijk behandelen. Voorlopig houden we het vooral zo eenvoudig mogelijk.

Bij (5) zie je de 'assets'. Dat zijn in feite de spelobjecten die we ter beschikking hebben. Deze kun je in een scene slepen, maar je kunt ze ook in andere scenes gebruiken. Je ziet dat scenes ook als assets worden geregistreerd en je ziet dat daar een apart folder voor is aangemaakt. Dat is niet echt nodig, maar wel zo makkelijk om straks zaken terug te vinden. Het is aan te bevelen om zaken zo gestructureerd mogelijk op te zetten, met meerdere folders, zodat je zaken snel terug kunt vinden. In een 'spel' met vijf objecten raak je de draad niet zo snel kwijt, maar als we aan een serieus spel beginnen praat je al gauw over honderd of meer objecten. En dan zoekt het een stuk minder makkelijk.

Bij (4) zie je in de hiërarchie, maar ook in het speelveld eronder (3), dat de bal geselecteerd is. We hebben het nu over een object in de scene, dus niet over de asset bal, onder (5).

Bij (6) zie je nu de 'inspector' waarmee je het object 'bal' kunt bekijken. En niet alleen bekijken, je kunt er ook mee aan het werk en waarden veranderen. Als we straks een nieuw bal zouden maken, dan zul je zien dat de waarden die je nu in de inspector ziet, al aangepast zijn op de huidige toevoeging. In feite bepaal je heel veel van het gedrag van je objecten in de inspector.

Bij (7) ten slotte zie je de console. Hier kunnen vanuit het systeem meldingen worden geplaatst die je kunnen helpen bij het onderzoeken van fouten. Voorlopig laten we dit nog even voor wat het is. Van belang voor onze eerste actie zijn de hiërarchie, de assets en de inspector en uiteraard de scene zelf. Laten we eens wat uitgebreider naar de menubalk kijken:



Figuur 5 Menu 'File'

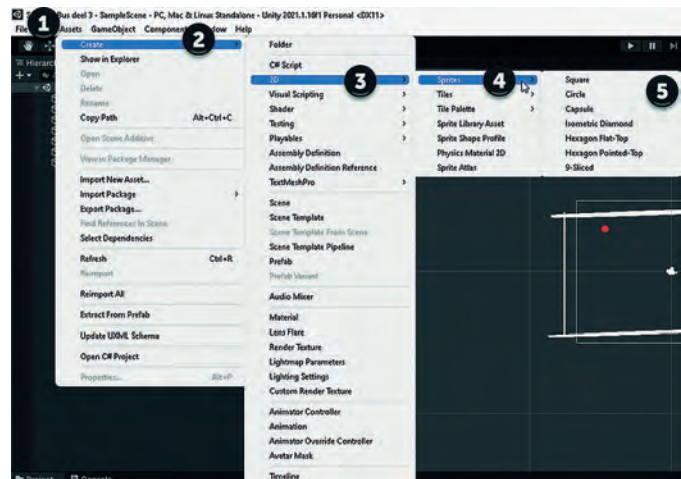
sla je ook je project op. Verder kan je hier wat instellingen vastleggen over 'build en run', waarover later in de reeks

meer. Vooralsnog laten we dit even voor wat het is. Het volgende menu onderdeel is 'Edit' en ook dit is wel redelijk standaard, alhoewel we hier wel veel opties hebben:



Menu 6 Menu 'Edit'

We kijken naar het volgende menu, het menu 'Assets':



Figuur 7 Menu 'Assets'

Dit is ook een heel uitgebreid menu, niet zozeer de eerste set van keuzes, maar het menu gaat heel diep door in submenu's. Zo zie je bij (1) dat we het menu 'Assets' kiezen. Ter herinnering, de assets staan onder het speelveld en zijn in feite de middelen die je ter beschikking hebt. De eerste keuze in het menu is 'Create': we gaan iets maken, zie (2).

En dan zie je dat je bij het aanmaken van een asset een enorme keuze hebt. Vooralsnog zijn we vooral geïnteresseerd in (3) 2D objecten en daarbinnen weer de sprites (4). Sprites zijn figuren die bovenop de achtergrond geplaatst kunnen worden. Zij bedekken de achtergrond, maar als ze zich verplaatsen keert de achtergrond weer terug. Vervolgens kun je bij (5) dus één van de basistypes van sprites kiezen. Voor nu kiezen we even een 'Circle' oftewel een cirkel. In mijn voorbeeld heb ik al een bal, maar voor de lezer is dit de eerste bal. Als je dit gedaan hebt, dan zie je dat onder in beeld een nieuw element wordt toegevoegd:

Een heel uitgebreid menu, met een aantal selectieopties, de standaard knip- en plakopties, maar ook zaken om te dupliceren, de besturing van het spel en wat instellingen. Vooralsnog laten we dit even zo, maar in de loop van de reeks gaan we hier af en toe wel aandacht aan besteden.

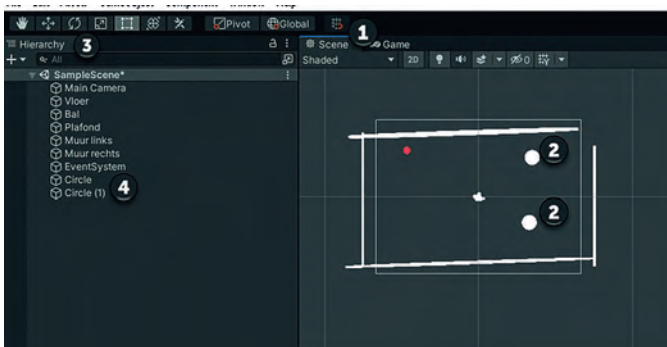
Als er bij de installatie wat zaken mis gingen, dan heb je hier wel de kans dit te herstellen. Zodra we in een volgende aflevering over het maken van scripts gaan beginnen (het toevoegen van c# code aan de elementen in Unity) komen we hier op terug.

Bedenk gewoon dat er al genoeg nieuws op je af komt zonder dat je alle keuzes al hoeft te begrijpen. We zijn voorlopig nog niet uitgeleerd en voor de eerstvolgende tien edities van de SoftwareBus hebben we al ideeën klaar staan.



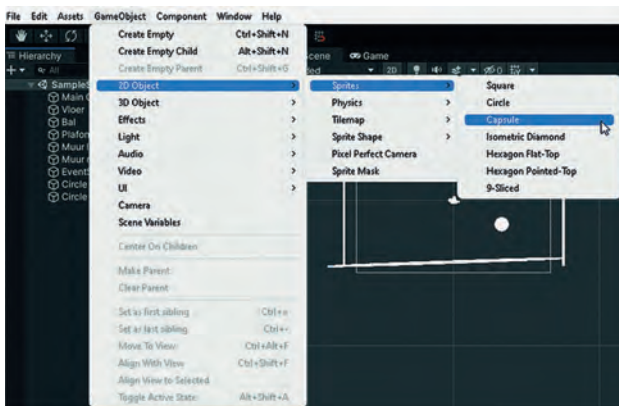
Figuur 8 Nieuw asset

Bij (1) zie je dat we naar de assets kijken. Ik blijf deze term maar gebruiken, want de letterlijke vertaling 'activa' of 'bezitting' voegt niet heel veel toe. Bij (2) zie je een blauw icoontje, met daaronder een naamvak waarin nu 'Circle' staat, maar dat is geselecteerd. Je kunt hier een andere, be tekenisvollere naam kiezen. Ik noem 'm gemakshalve 'Bal 2', maar jullie kunnen 'm rustig 'Bal' noemen. Deze bal staat nu in de assets, maar hij staat nog niet in het speelveld en hij staat ook nog niet in de hiërarchie. Dat gaan we nu veranderen door de bal vanuit de assets naar het speelveld te slepen. De bal blijft dan in de assets staan, maar verschijnt ook op het speelveld. Je kunt namelijk vanuit de asset 'Bal (2)' meerdere instanties in je spel gebruiken. Sleep hem er nog maar een keer in en je ziet dat er nog een bal letje in het speelveld verschijnt. En alles wat je in het speelveld ziet, zie je dus ook in de hiërarchie:



Figuur 9 Nieuwe ballen

Je ziet bij (1) dat we de scene hebben geselecteerd voor het speelveld. Hier hebben we de nieuwe asset twee keer naartoe geslept (2). Bij de hiërarchie (3) zie je dat deze twee balletjes onderin zijn opgenomen. Jullie speelveld ziet er iets anders uit dan dit hierboven, jullie hebben als het goed is alleen de camera en de twee ballen. Geen zorgen, we gaan verder. Via het menu 'Game object' gaan we nu een 'capsule' aanmaken:



Figuur 10 Menu 'Game object'

We kiezen het menu, kiezen dan '2D Object', dan 'Sprite' en dan 'Capsule'. Je ziet dan op het speelveld, in het midden, een figuurtje komen dat eruit ziet als een tablet:

Dit figuurtje is vervolgens ook zichtbaar in de hiërarchie. Het is echter niet zichtbaar bij de assets. Als we de capsule selecteren (dat kan in het speelveld of in de hiërarchie, zoals ik eerder uitlegde, dan kunnen we even naar de hiërarchie gaan en dan op de capsule met de rechtermuis knop klikken en dan 'copy' selecteren. Vervolgens elders in de hiërarchie met rechts klikken en 'plakken' of 'paste' selecteren, afhankelijk van de taal waarin Windows is geïnstalleerd.

leerd. Dan zie je een tweede capsule verschijnen in de hiërarchie.

Toen ik dat deed zag ik de tweede capsule nergens. Ik dacht even dat hij misschien op of onder de eerste stond, maar dat was niet het geval. Als je een object kwijt bent, ga dan naar de inspector (rechts in beeld) en stel daar dan de x en y positie in. Het midden van het speelveld is als (0,0) gedefinieerd.

Met de inspector kun je een heleboel regelen rondom de game objecten. Dat heb ik al verteld en we gaan daar nu een stapje verder mee:

Bij (1) zie je de geselecteerde capsule. Bij (2) zie je de inspector. Bij (3) zie je de naam van het element dat je inspecteert. Overigens is inspector eigenlijk een wat vreemde naam, want je inspecteert niet alleen, je kunt ook aanpassen, zoals onder meer de naam. Je ziet in figuur 9 bijvoorbeeld de hiërarchie zoals die door mij in het voorbeeld was gezet met daaronder een aantal elementen met Nederlandse namen. Die heb ik eraan gegeven, want Unity is zelf Engelstalig.

Bij (4) zie je een blokje dat je bij alle sprites en in misschien wel bij alle game objecten tegenkomt, 'Transform'. Hier kun je de positie, de grootte en de rotatie aflezen, maar dus ook aanpassen. Als je een object kwijt bent, dan kun je 't hier ergens neerzetten en dan kun je zorgen dat je het wel ziet. Overigens, de oplettende lezer heeft nu door dat er niet alleen een X en een Y, maar ook een Z staat. Nu zijn de X en Y as zoals we gewend zijn van de wiskunde: rondom het middelpunt is naar rechts en naar boven positief, naar links en naar beneden is negatief. De Z as staat eigenlijk loodrecht op dit vlak. Voor posities doet dit niet echt ter zake, er zijn in elk geval betere manieren om objecten voor of achter elkaar te plaatsen, maar de rotatie is wel nuttig, want hiermee kunnen we de verticale capsule horizontaal maken, door via de Z as 90 graden te draaien. Probeer het maar, zo heb ik ook mijn vloer en plafond gemaakt.

Bij (5) zie je de 'sprite renderer' waarmee je aangeeft hoe de sprite wordt weergegeven. Hoewel we hier in dit stadium nog niet heel diep op ingaan, kun je wel bijvoorbeeld de kleur hier instellen. Als je op de witte balk naast 'Color' klikt zie je een heel palet ter beschikking komen waarmee je de kleur van je spelobject in kunt stellen.

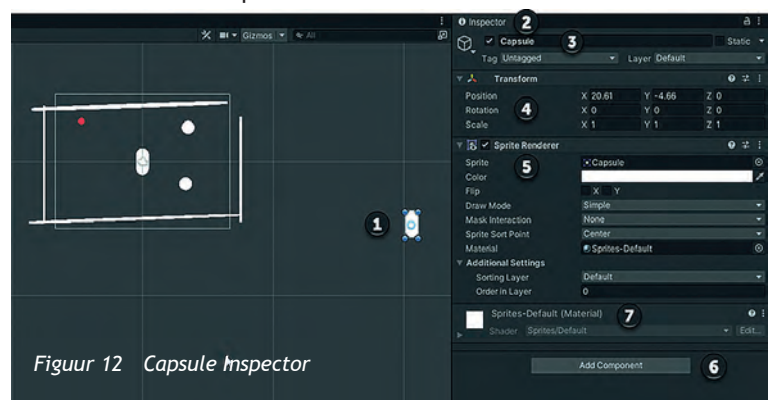
Bij (6) kun je 'Componenten' toevoegen aan de spelobjecten. Dat gaan we de volgende keer dan ook doen, want dat hebben we nodig om ons doel te bereiken.

Bij (7) ten slotte, kun je het materiaal definiëren waarmee het object is opgebouwd. Ook daar moeten we de volgende keer meer mee gaan doen. Voor nu moeten we het even bij deze uitleg laten.

Nog één ding waar ik op wil wijzen, in het kader van het menu nu (dat we ook niet helemaal hebben kunnen behandelen). Er is binnen Unity ontzettend veel hulp verkrijgbaar via de online manual. Wel in het Engels, maar je kunt natuurlijk te genwoordig hele pagina's prima door Google laten vertalen. Ik zou zeggen, kijk er eens rustig naar.

Als je een object in het speelveld geselecteerd hebt, dan is het dus te bewerken in de inspector, maar in de scene zie je ook zogenaamde 'handles' aan het object en hiermee kun je de omvang en dergelijke aanpassen.

Als huiswerk zou ik zeggen, bouw de scene van figuur 1 op. Alle wanden zijn capsules. Dan gaan we in de volgende SoftwareBus verder. Veel plezier!



Figuur 12 Capsule Inspector