

● Games maken (3) ●

René Suiker

Terugblik

We hebben 2021 overleefd en leven nu in 2022. Ik wens u allen 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 Platform 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 eenvoudiger. Wel heb je met Unity veel meer mogelijkheden. Maar, alles op zijn tijd.

Natuurwetten

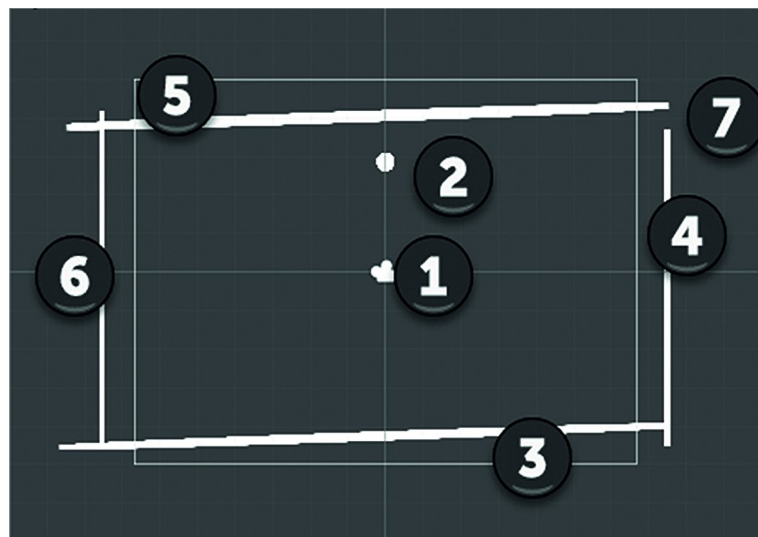
In Scratch moet je dus alles zelf programmeren, in Unity kan je je eigen wereld configureren. Inclusief de wetten van de mechanica die er gelden. Je kunt bij wijze van spreken een negatieve zwaartekracht implementeren en je kunt dus ook zonder verliezen laten stuiteren, zelfs met versnelling. En dat 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 stuitert 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 beginnen 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:



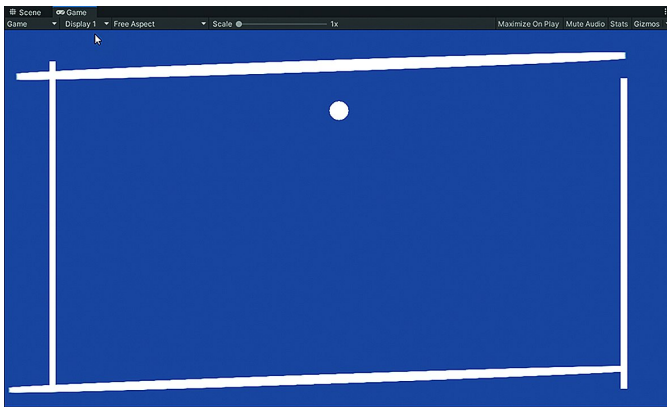
Figuur 1 - Het speelveld

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.

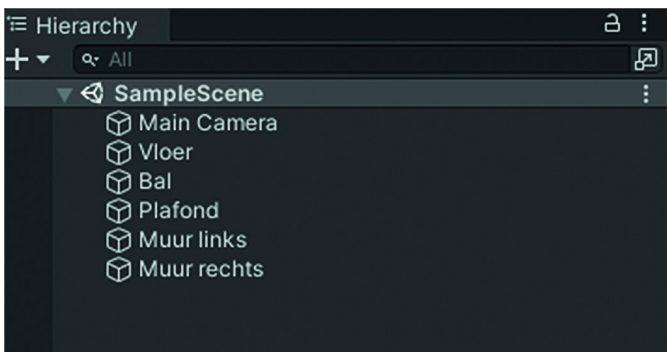
Bij (3) zie je de vloer. Zoals ik aangaf, die staat niet recht, want we willen de bal een beetje door de ruimte laten stuiteren en als we alles recht houden, dan stuitert hij alleen maar recht op en neer, of ik moet met andere complexe effecten gaan werken en ik wil het in het begin eenvoudig houden.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

Bij (5) dus het plafond en bij (7) ten slotte de opening, want het idee is dat het balletje door rond te stuiteren 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 speelmodus

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

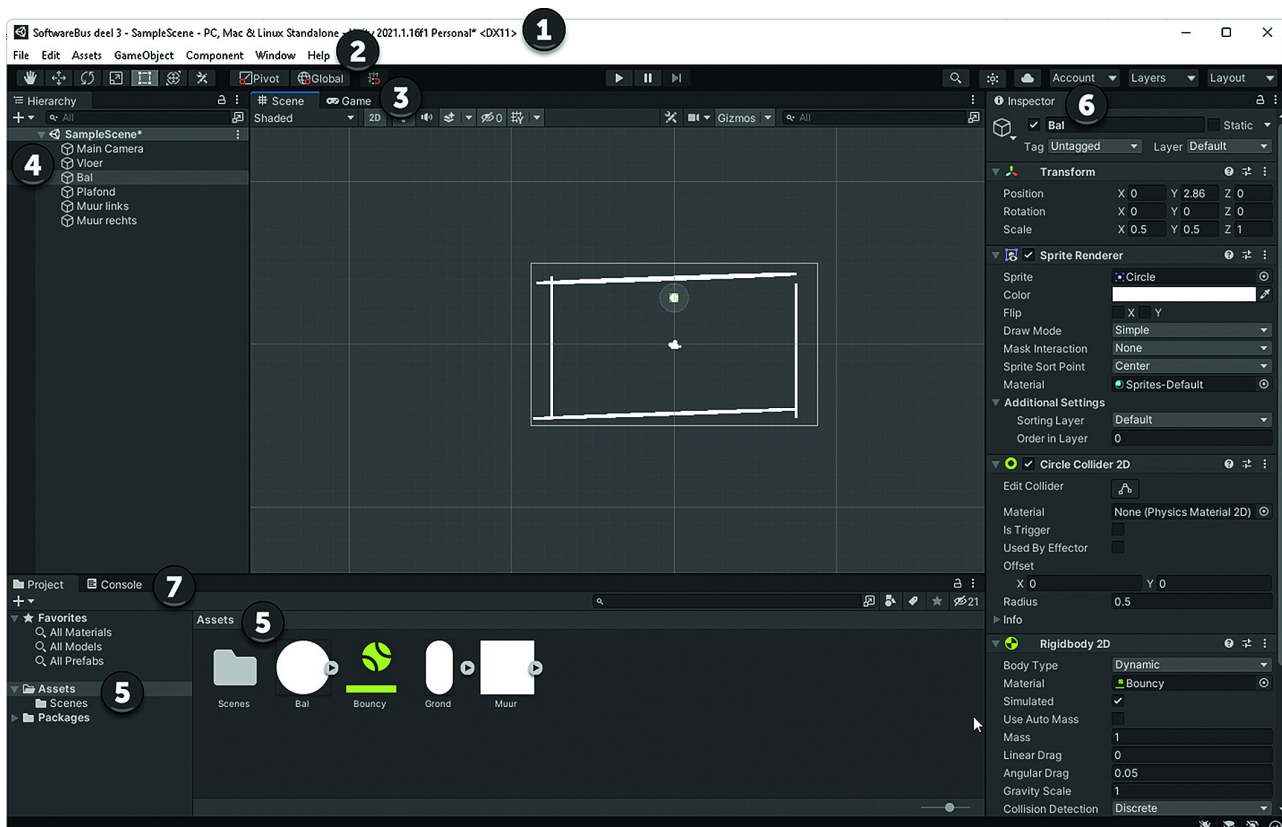
Maar dit staat er niet zomaar, daar hebben we iets voor moeten doen, want als je met een nieuw project begint, dan heb je in de 'SampleScene' (het voorbeeldscenario) alleen maar de camera.

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 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 mogelijkheden 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).



Figuur 4 - Unity scherm

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 nieuwe bal zouden maken, dan zul je zien dat de waarden die je nu in de inspector ziet, al aangepast zijn op de huidige toepassing. In feite bepaal je heel veel van het gedrag van je objecten in de inspector.

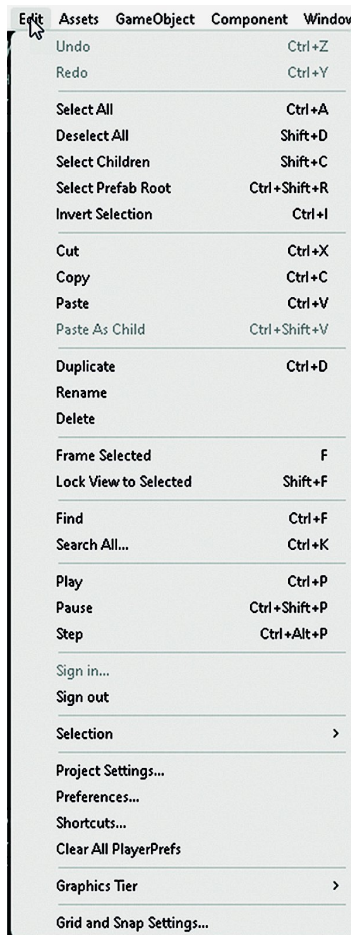
Bij (7) ten slotte zie je de console. Hier kunnen vanuit het systeem meldingen worden geplaatst die je kunnen helpen bij het onderzoeken van fouten. Voorlopig laten we dit nog even voor wat het is. Van belang voor onze eerste actie zijn de hiërarchie, de assets en de inspector en uiteraard de scene zelf.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



Figuur 5 - Menu 'File'

de scenes openen. Hier 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:



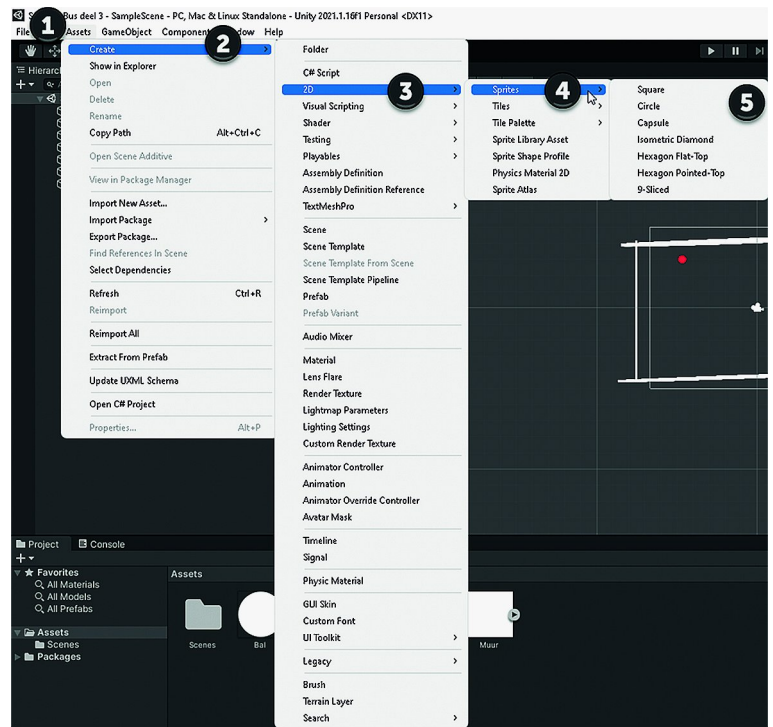
Figuur 6 - Menu 'Edit'

Het eerste onderdeel van het menu is het bekende menu 'file'. Ik zeg bekend, omdat dit in bijna alle applicaties standaard is; elke applicatie begint met een menu 'file', waarmee je de applicatie kunt beëindigen, maar ook zaken kunt opslaan, openen, nieuwe zaken kunt beginnen. Wij beginnen vooralsnog nieuwe projecten vanuit de Hub, maar binnen projecten kan je bijvoorbeeld hier nieuwe scenes maken, of bestaan-

de scenes openen. Hier 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: 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 even 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 klaarstaan.

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).

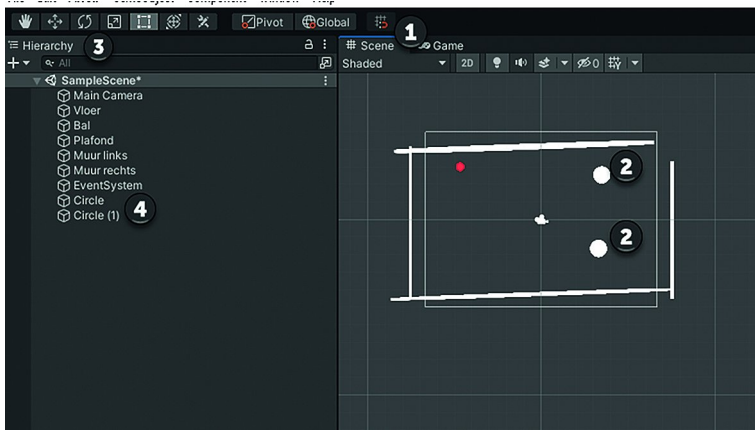
En dan zie je dat je bij het aanmaken van een asset een enorme keuze hebt. Vooralsnog zijn we vooral geïnteresseerd in (3) 2D-objecten en daarbinnen weer de sprites (4). Sprites zijn figuren die bovenop de achtergrond geplaatst kunnen worden. Zij bedekken de achtergrond, maar als ze zich verplaatsen 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:



Figuur 8 - Nieuw asset

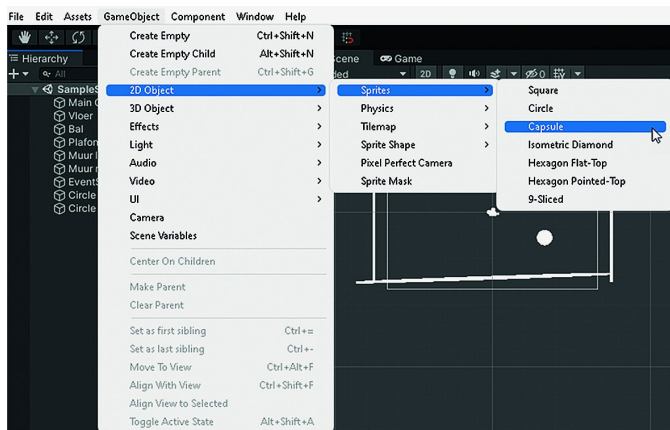
Bij (1) zie je dat we naar de assets kijken. Ik blijf deze term maar gebruiken, want de letterlijke vertaling 'activa' of 'bezitting' voegt niet heel veel toe. Bij (2) zie je een blauw icoontje, met daaronder een naamvak waarin nu 'Circle' staat, maar dat is geselecteerd. Je kunt hier een andere, betekenisvollere 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 balletje 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 gesleept (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:

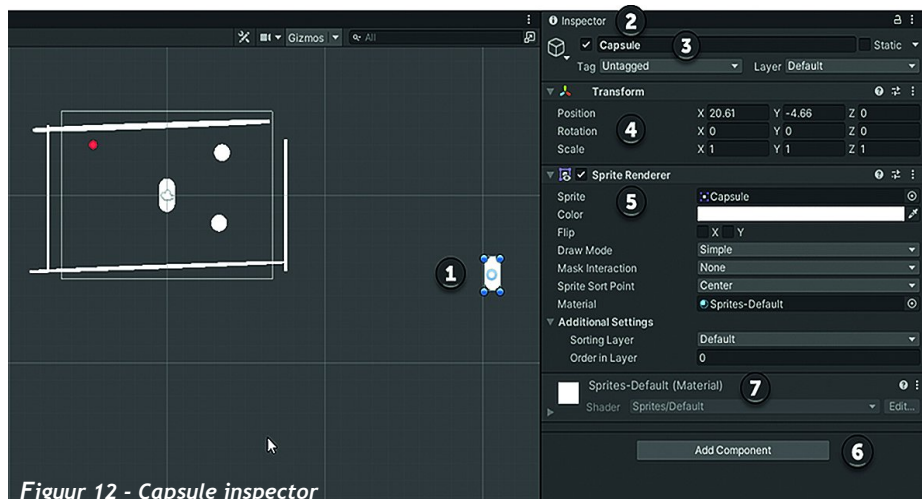


Figuur 11 - Capsule

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 rechtermuisknop klikken en dan 'copy' se-

lecteren. Vervolgens elders in de hiërarchie met rechts klikken en

'plakken' of 'paste' selecteren, afhankelijk van de taal waarin Windows is geïnstalleerd. 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:



Figuur 12 - Capsule inspector

Bij (1) zie je de geselecteerde capsule. Bij (2) zie je de 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 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 opletende 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.

Ik zie tot mijn grote schrik dat het einde van pagina 4 al in zicht komt. En meer pagina's mag ik op dit moment niet gebruiken.

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 (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 tegenwoordig 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!

Slotopmerking: Bij Scratch waren we in die 4 pagina's al een stuk verder. Volgende keer grijpen we weer terug naar het Scratch-verhaal van deze SoftwareBus.