Eindverslag afstudeerwerk

Kusomari

17 augustus 2017

**Auteur** : Jiska Baeten

**Schoolpromotor** : Wim van WeyenBerg, Pieter Jorissen



Opleiding Multimedia en Communicatietechnologie

|  |  |  |
| --- | --- | --- |
|  |  |  |

Inhoud

[Omschrijving 1](#_Toc486925230)

[Technische overzicht 3](#_Toc486925231)

[Conclusie 12](#_Toc486925232)

[Bronnen 13](#_Toc486925233)

Omschrijving

Het idee

Ik wou een pc-game maken waarin ik ongeveer dezelfde spelconcepten had als het bestaande playstation game: katamari. Sinds dit nog niet leek te bestaan, leek het me wel interessant om dit game dan te maken.

Ik wou een variant hierop maken en teruggrijpen naar de roots van het idee zelf. Waarschijnlijk kwam het originele idee van de dierenwereld: een mestkever die een bal van mest vooruitduwt.

Hierbij was het plan om een speciale controller te maken die speciaal ontworpen was om de bal realistisch en makkelijk te besturen.



Concept

In katamari speel je met een klein personage dat een bal vooruitduwt en andere voorwerpen die kleiner dan de bal zijn, opraapt. Deze voorwerpen blijven aan de bal hangen en maken de bal groter. Ik heb dit idee overgenomen, maar beperk wel hoeveel objecten er zichtbaar aan de bal kleven om geheugen uit te sparen.

Bij katamari wordt er ook getoond hoe groot de bal momenteel is en wordt er een timer bijgehouden die aftelt. Ik hou ook bij hoe groot de bal is, maar laat de timer optellen, zodat de speler meer dan genoeg tijd heeft om een level te voltooien. Hierdoor kan de snelste tijd bijgehouden worden en kan de speler steeds proberen om zijn eigen score te verbeteren.

Ik heb ervoor gezorgd dat het level pas kan voltooien als je een bepaalde grootte hebt gehaald en daarmee bv. uit het park kan ontsnappen door de omheining op te rollen met je bal. Dus je rolt altijd een bepaald voorwerp op, om het level te voltooien.

Ik dacht ook wel dat enkele hulpmiddelen zouden kunnen helpen, o.a.: een minimap waarop alles dat kleiner is dan je bal, getoond zou worden. Voorwerpen die het level zouden voltooien worden op een andere manier getoond op deze map. Daarnaast heeft de speler ook de mogelijkheid om een omlijning rond de voorwerpen te tonen die kleiner zijn dan de bal, door een druk op een bepaalde toets.

Ik vond het ook noodzakelijk dat je het spel op verschillende manieren van besturing kon spelen. Zo is het spel dat niet enkel speelbaar met de speciale controller, maar ook door de pijltjestoetsen, de nummertoetsen en zelfs de computermuis. Zo kunnen mensen die de controller niet hebben, het spel ook spelen.

Doelgroep

Ik vond dat iedereen het spel zou moeten kunnen spelen. Daarom is er een timer die optelt.

Technische overzicht

Technieken en tools

Overzicht gebruikte Technieken en Tools (welke heb je zelf gemaakt, welke heb je gebruikt van andere bronnen, vermeld ook duidelijk waar die bronnen vandaan komen)

* Unity
* Arduino
* Icomoon (voor fonts)
* Photoshop
* Illustrator (fonts)
* Verwijzing naar bronnen te vinden op de laatste pagina.

Opbouw eindwerk

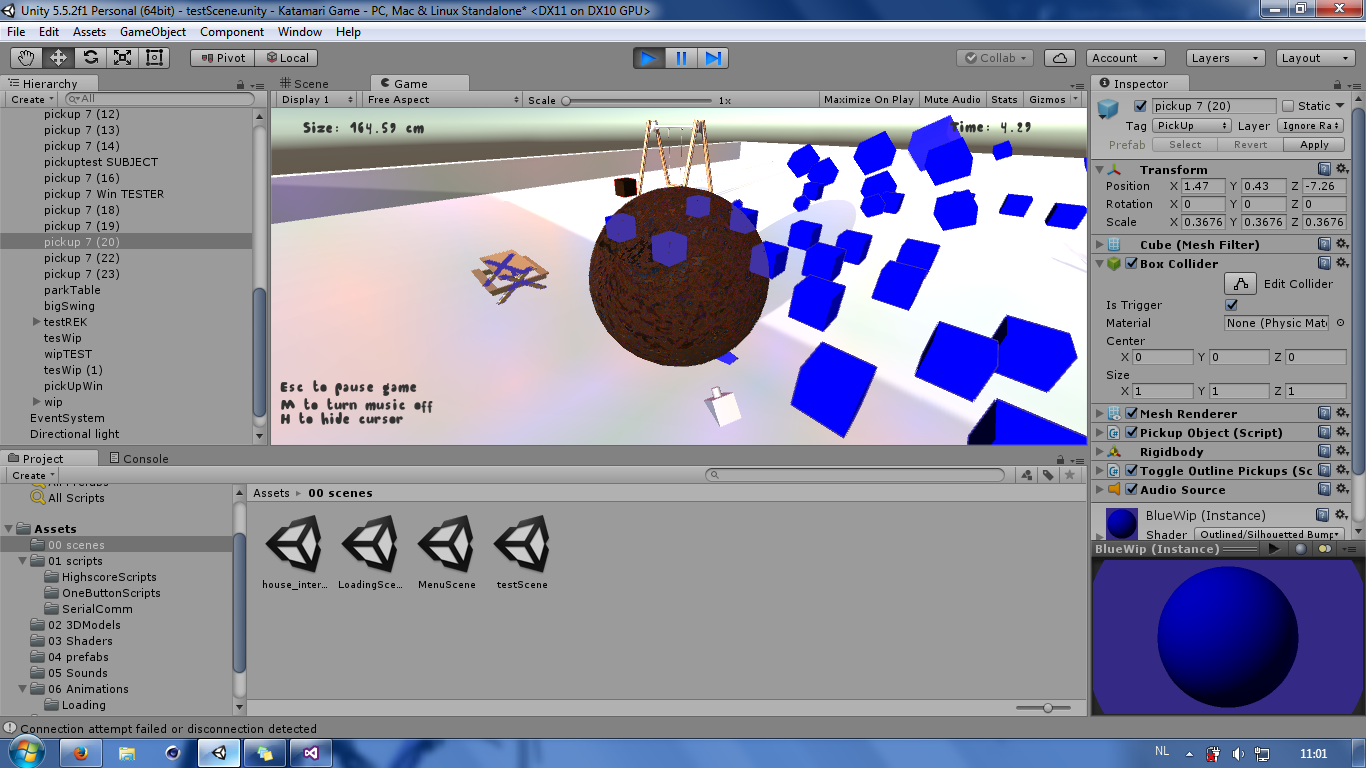
Opbouw eindwerk (hoe ben je gestart, welke stappen heb je genomen)

Screenshots

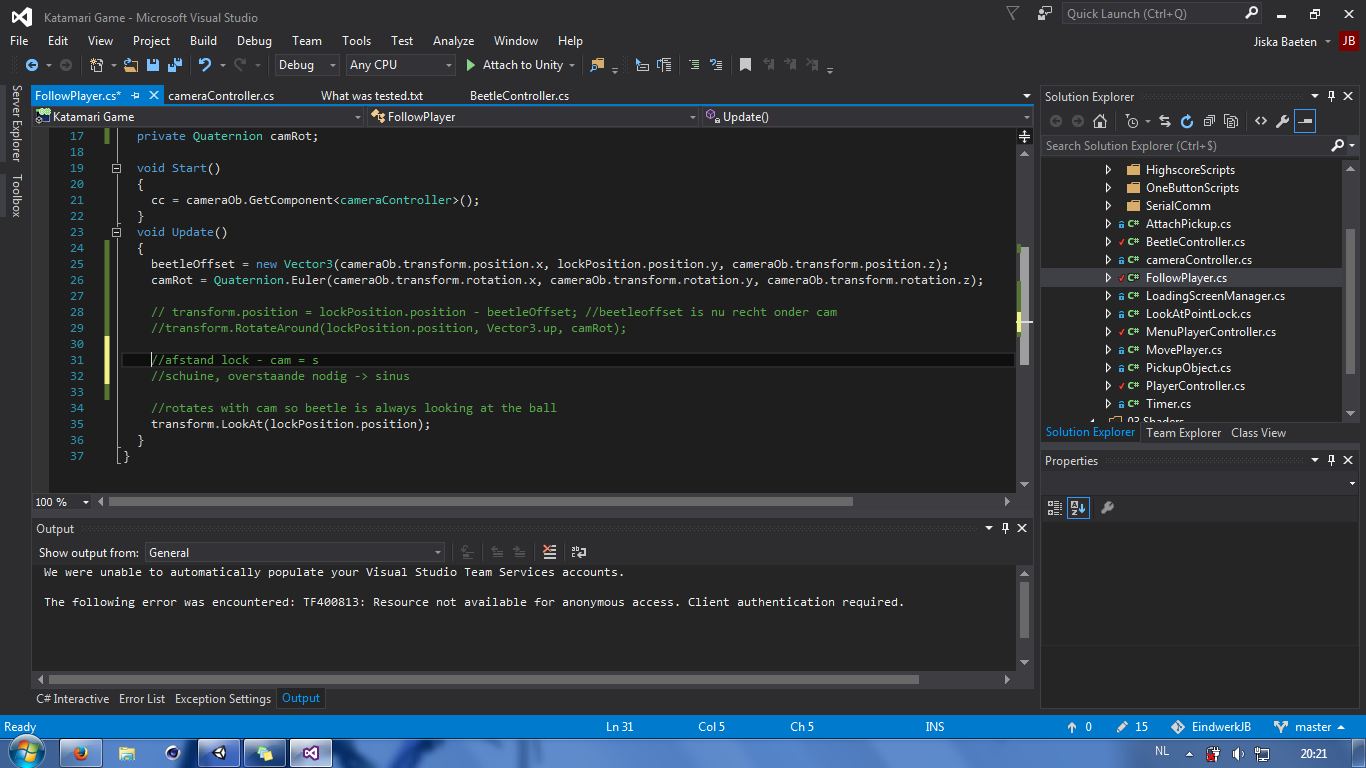
1. Pauzescherm, cursor en icon

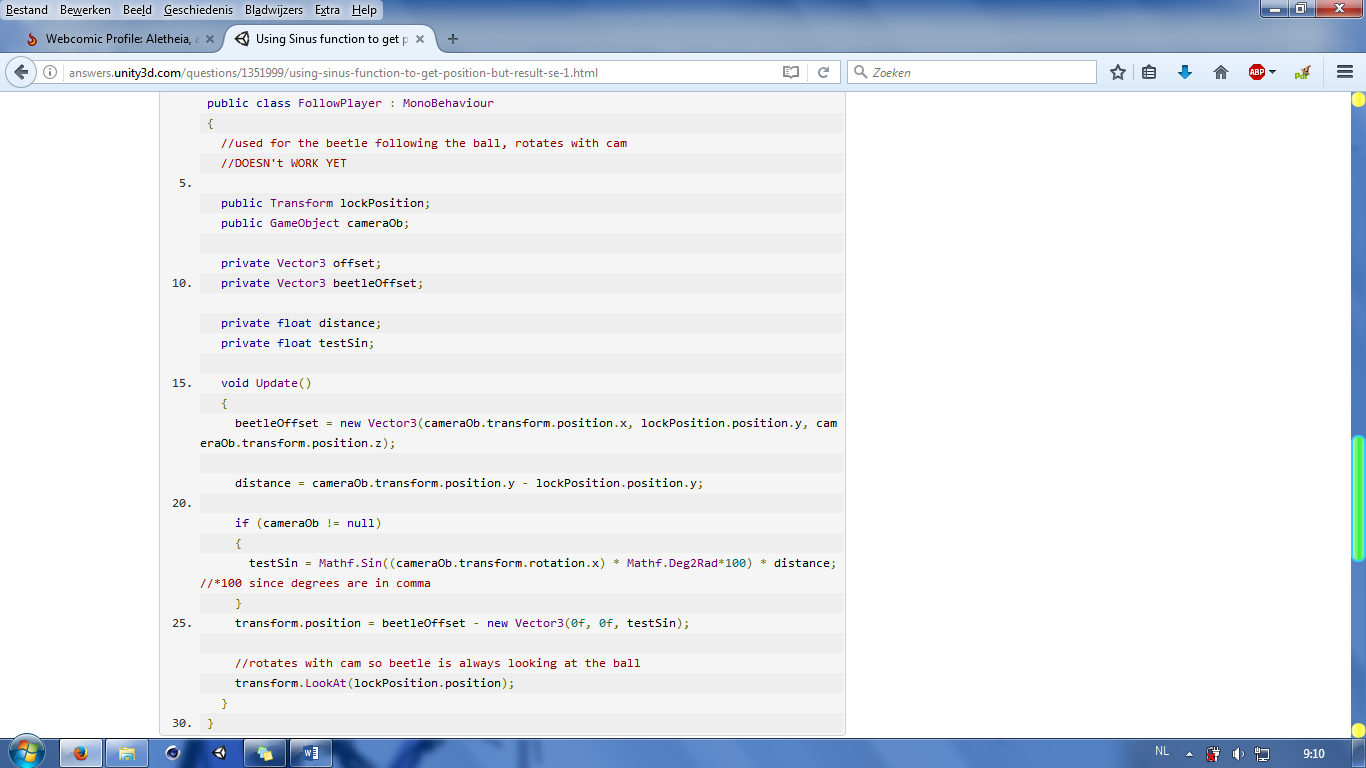


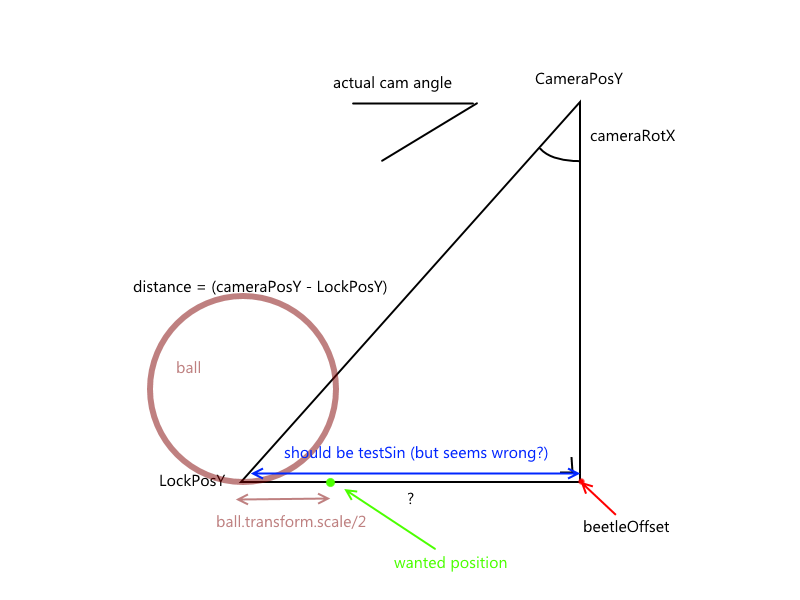
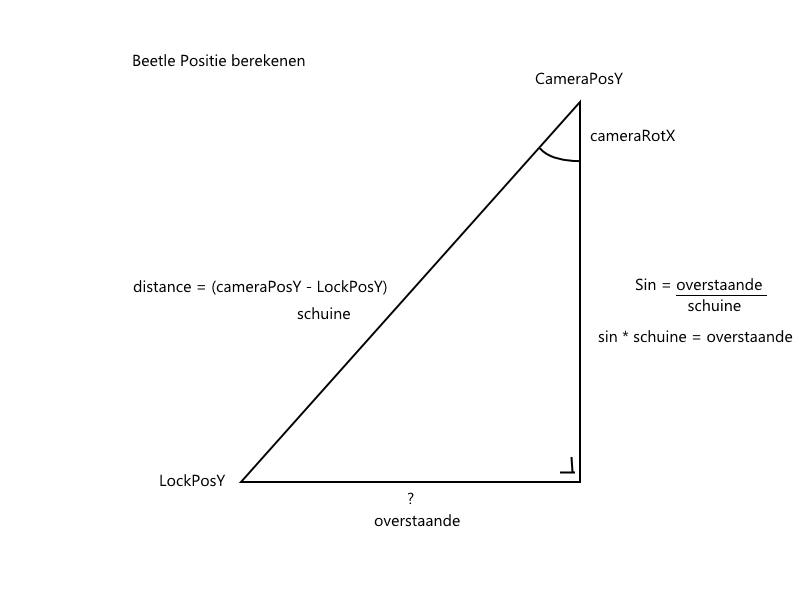
1. Testen outline door bal en testen op onregelmatige meshes



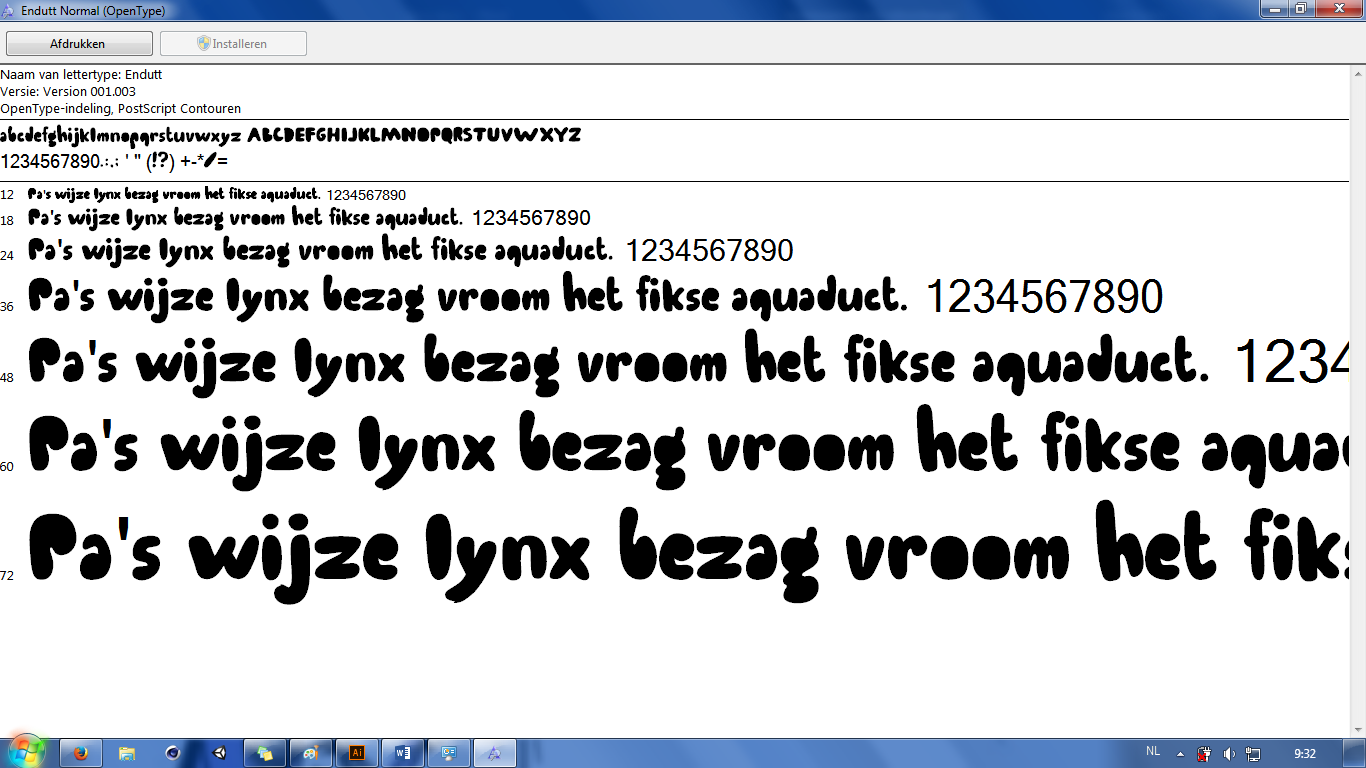
1. Juiste code zoeken voor beetle die de bal achtervolgt en dezelfde rotatie heeft als camera

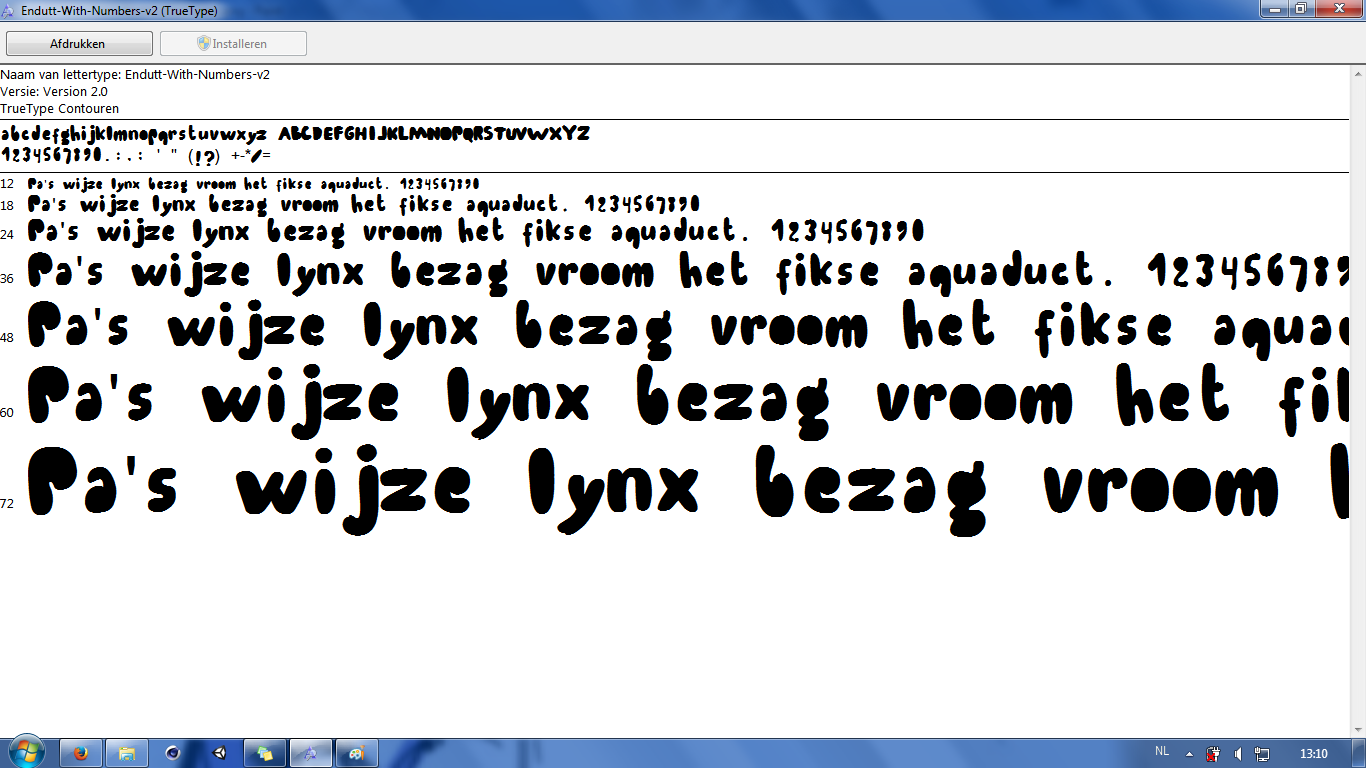


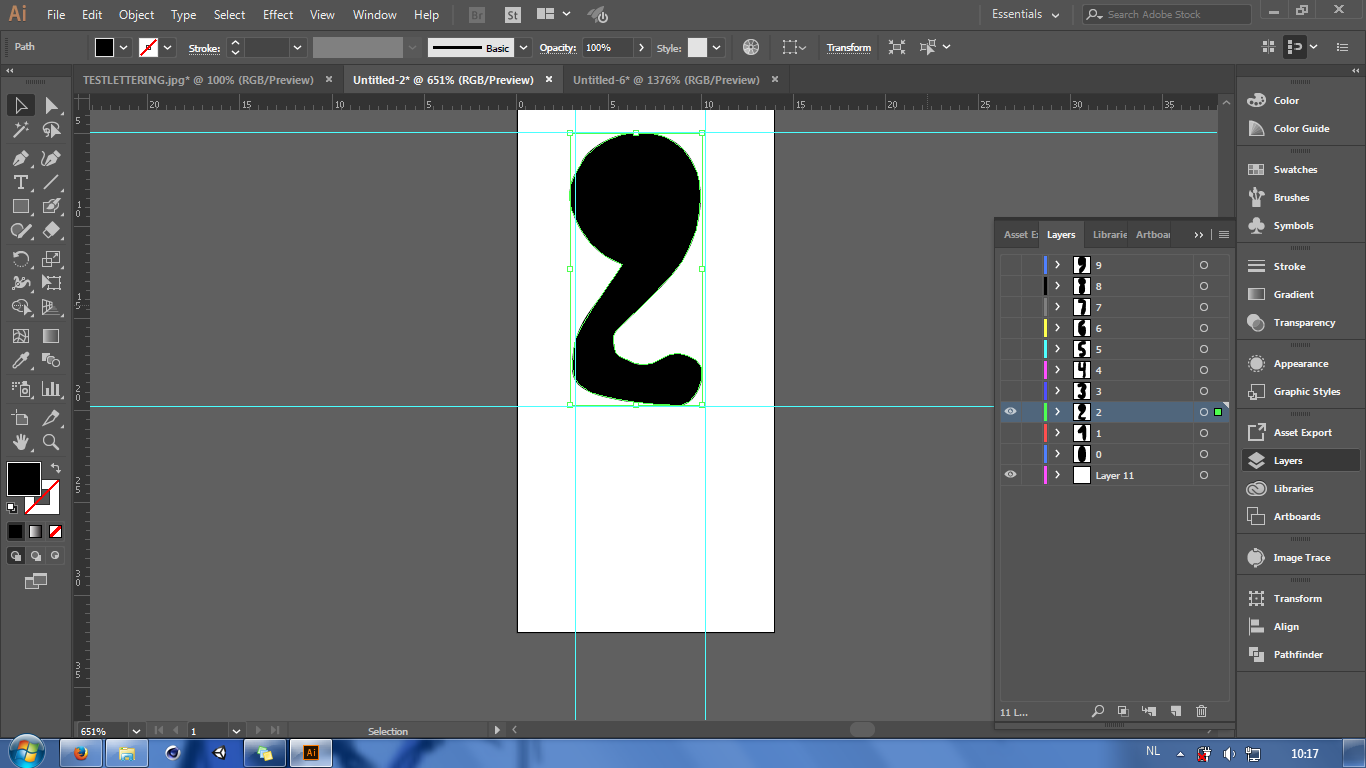


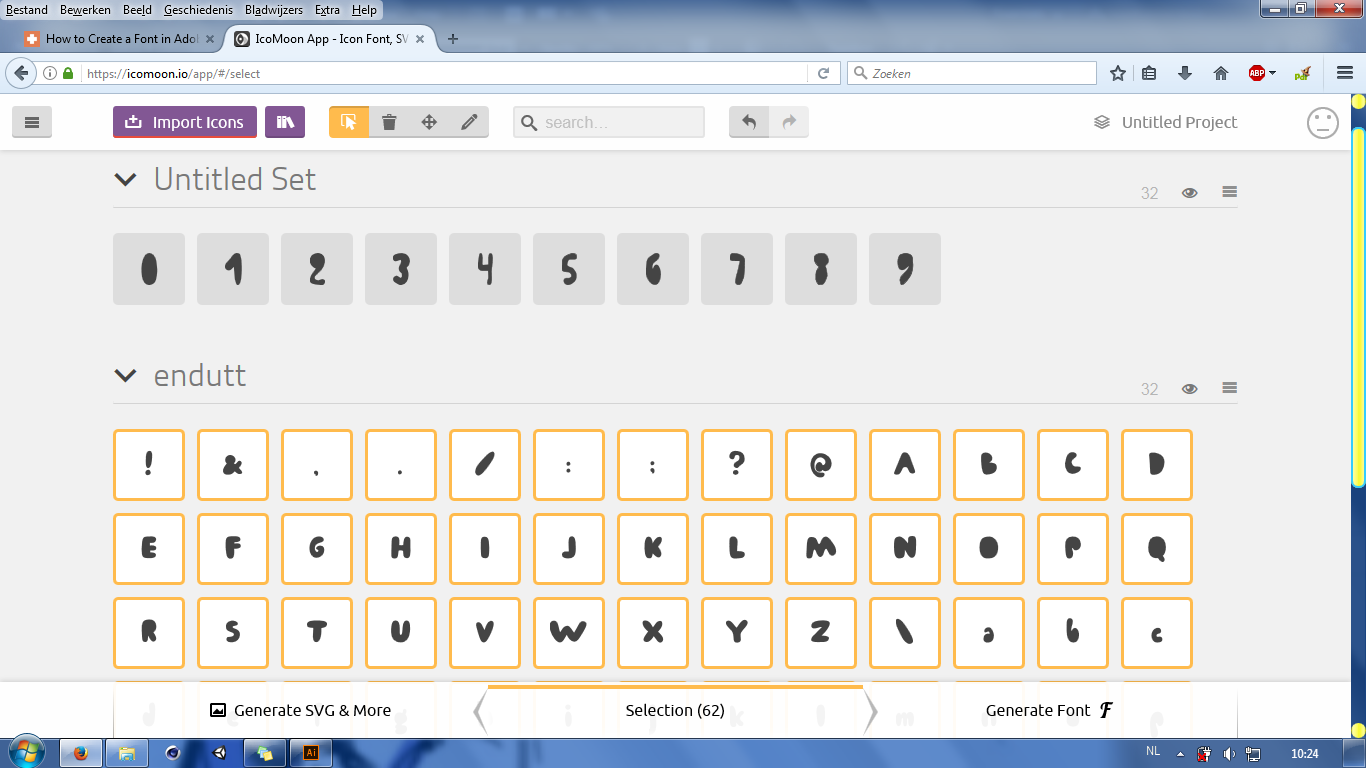


1. Nummers toevoegen aan het gekozen font

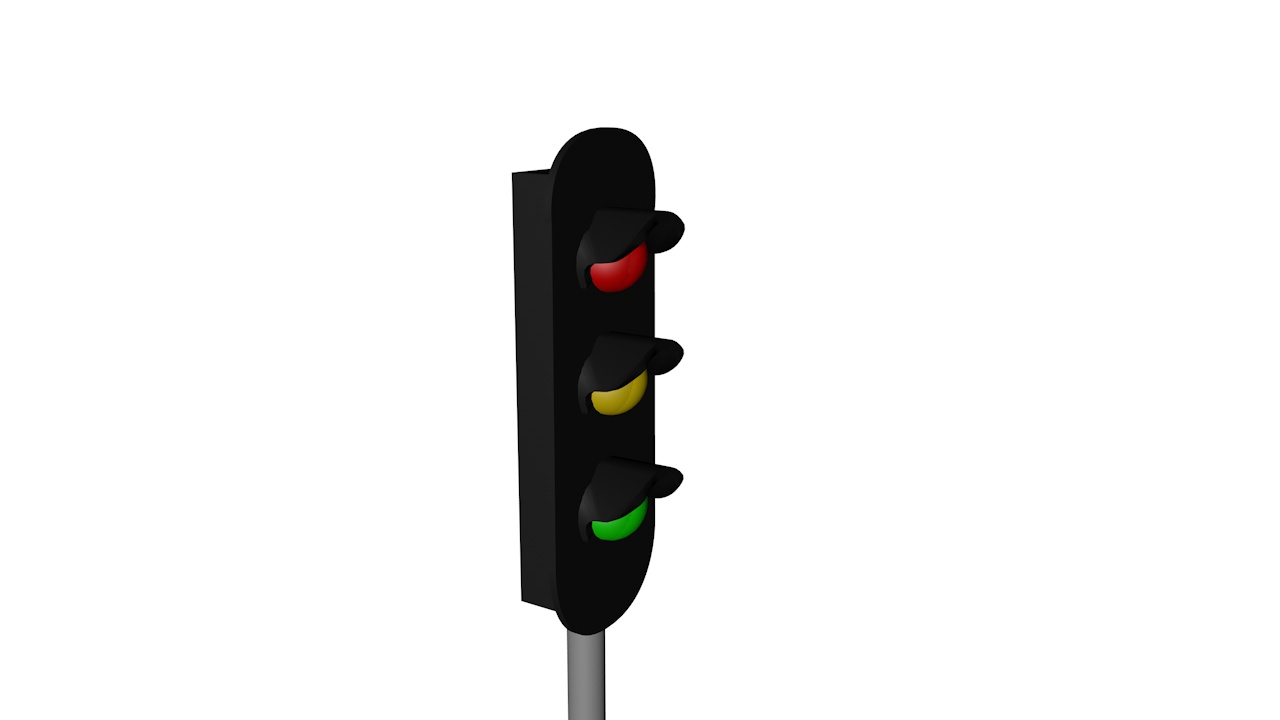
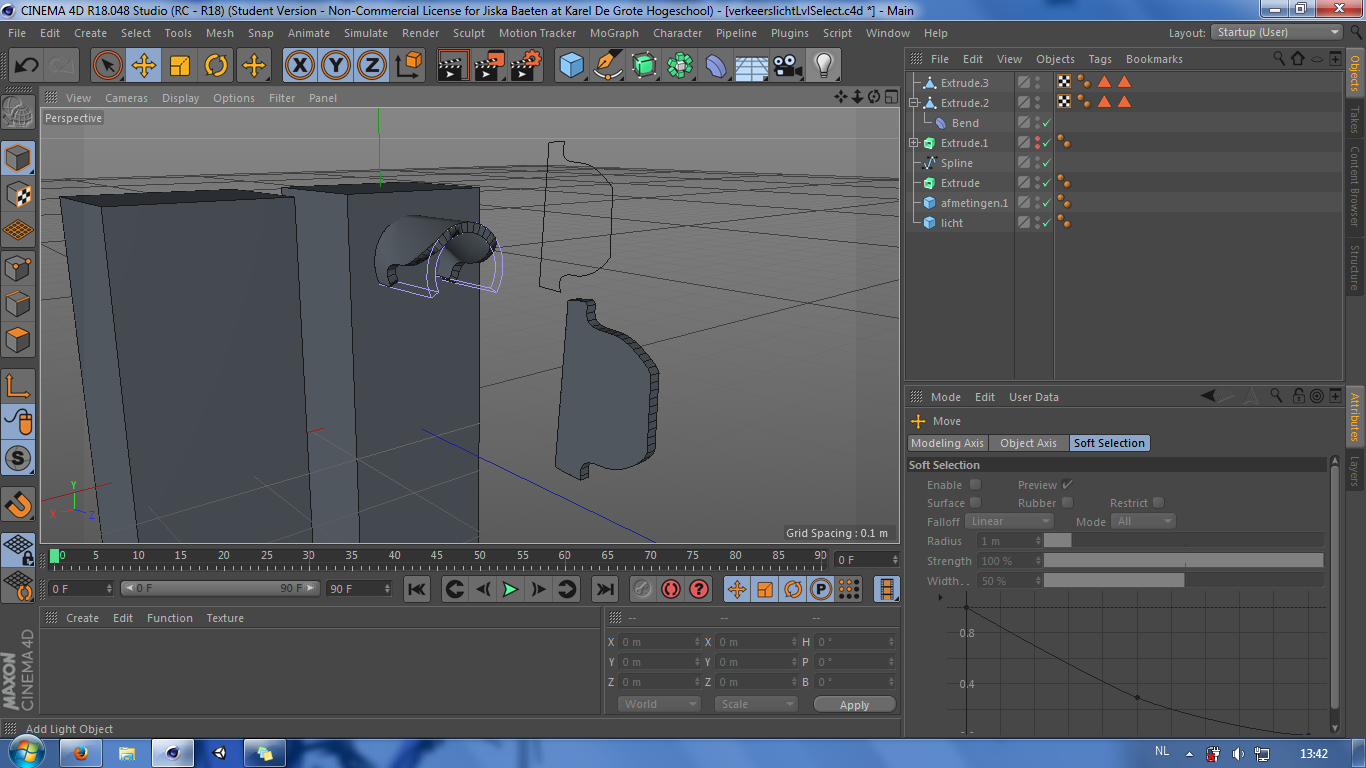




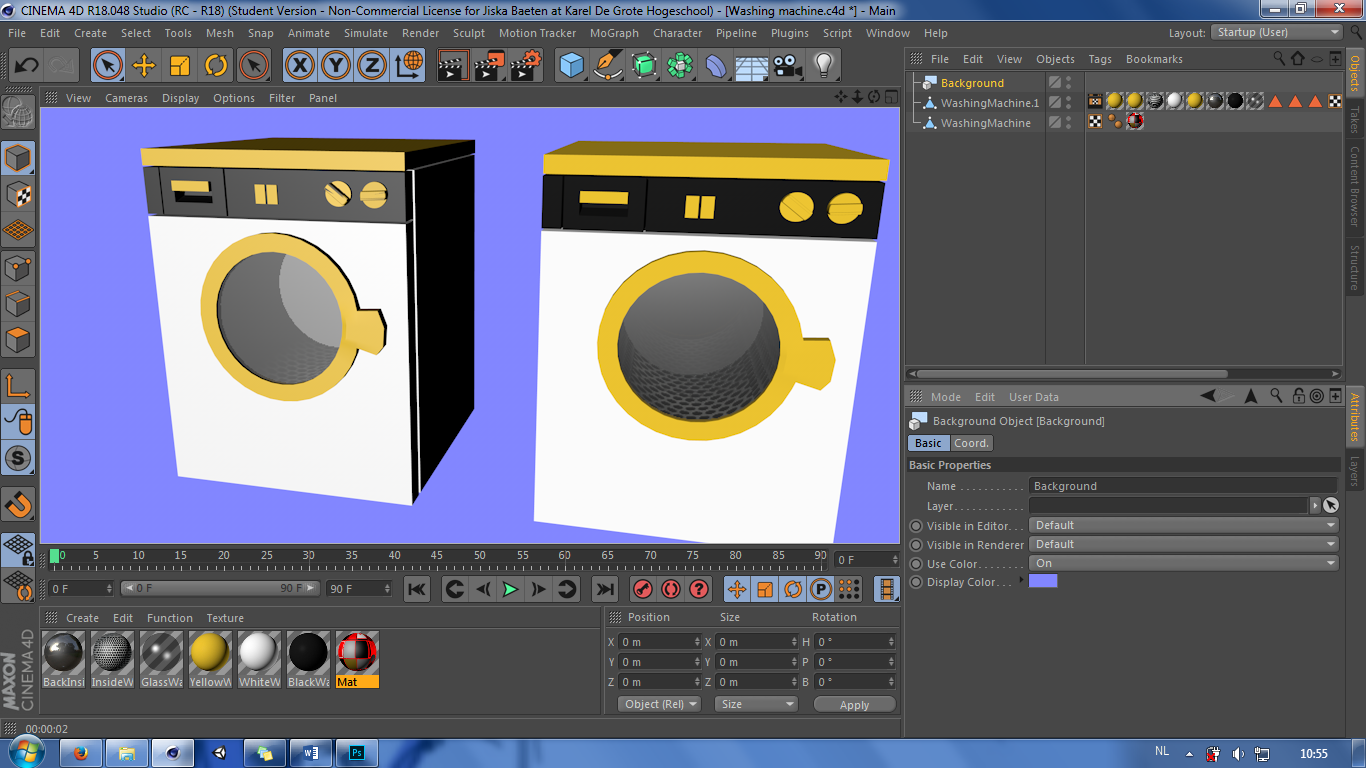




1. Denken hoe het met papier gemaakt zou zijn.



1. Testen of baken van een texture beter is



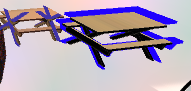




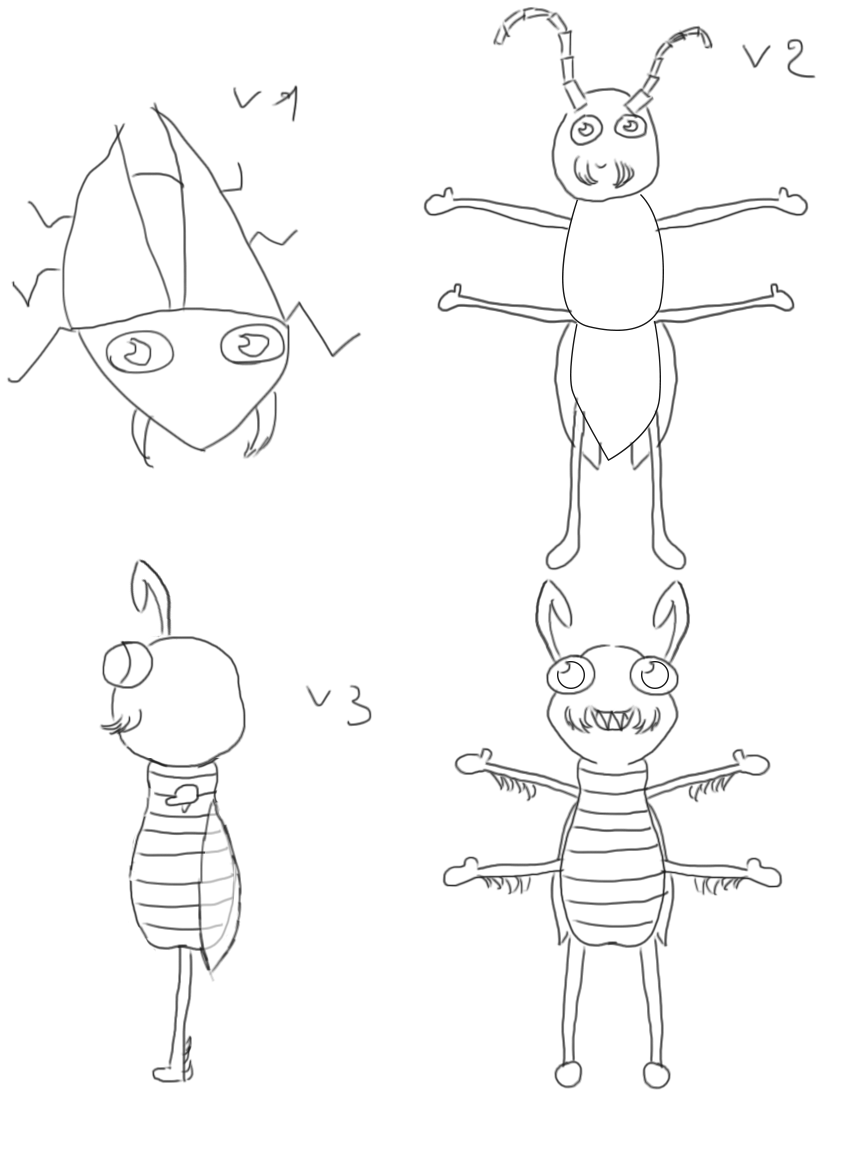
1. Outlines testen op complexe meshes (hier exact hetzelfde model)

Links: outline op voorwerp met verschillende texturen

Rechts: outline op 1 textuur



1. Groei van het karakter



Overzicht overwonnen problemen en oplossingen

* Als een object kleiner wordt dan de speelbal, dan zou de trigger aan moeten springen van een object dat opgeraapt kan worden. Bij mesh colliders leek dit een probleem te zijn. Als bij mesh collider **convex niet aangevinkt is, dan kan je geen trigger gebruiken.**
* **Bij convex trigger kwam er een ander probleem. Als een collider convex is en de bal is kleiner dan het object (bv. bij een parkbark), dan kan de bal niet onder het object rollen. Dit heb ik dan opgelost om tijdelijk meerdere box colliders te gebruiken.**
* **Ik wou de beetle de bal laten volgen en de rotatie laten gebruiken van de camera (zodat de beetle altijd voor de camera staat). Hiervoor wou ik wiskunde en rechthoekige driehoeken voor gebruiken, maar dit zorgde altijd voor een ander resultaat dan gewenst (zien screenshot punt 3). Dus uiteindelijk heb ik een punt aangemaakt onderaan de bal dat altijd op local ypos = 0 blijft t.o.v. de bal. Daaraan zit de beetle bevestigd. De beetle staat op een afstand die gelijk is aan 2/3 van de diameter van de bal, zodat hij niet te dicht of veraf van de bal staat. Het punt dat onderaan de bal staat, kijkt altijd naar de bal.**
* **Toen ik de nummers aan het font wou toevoegen, waren enkele letter afgekapt (bv. l, p, f…). Dit bleek fout zijn gelopen bij de conversie naar het font bij icomoon. Omdat Endutt geen genormaliseerd font was, stonden er geen afstanden en spacing geprogrammeerd zijn. Dus ik heb alle letter en nummers correct moeten positioneren in de ruimte t.o.v. de rest van de letters en cijfers, zodat alles even hoog/laag staat.**
* **Een spatie aanmaken in een font bleek lastiger te zijn dan verwacht. Elke keer als ik dit probeerde, leken alle letters aan elkaar vast te hangen, ondanks dat ik een spatie had getypt. Uiteindelijk heb ik dit opgelost door een lijn te tekenen in Adobe Illustrator en deze een opacity van 0% te geven zodat ze onzichtbaar werd. Dit leek de oplossing te zijn.**
* **Als je een eigen ontwerp voor de cursor maakt, dan moet dat geen .cur-extensie hebben. Een gewone .png afbeelding werkt.**
* **Ik wou minder verschillende textures, die op bepaalde polygonen waren geplaats met polygon selection, herleiden naar 1 texture (zoals je bij UV mapping zou doen). Daarvoor moest ik de texture baken. In cinema4d ziet het eruit zoals verwacht, maar in unity was dat niet zo (zie punt 6 bij screenshots). Ik heb veel verschillende manieren geprobeerd om het gewenste effect te verkrijgen. Dus ik heb ervoor gekozen om transparante texturen op objecten, apart te houden en deze niet te baken. Dus eerst voeg ik alle aparte stukken behalve het glas object samen door connect, dan bake ik alle texturen tot 1 textuur en daarna connect ik het glazen onderdeel. Hierdoor kan ik zelf kiezen hoe groot de texturen zijn in filegrootte en beperk ik alle objecten tot ong. 2 textures (textuur van alles + glastextuur).**
* **Daarnaast heeft het baken van de textuur een oplossing gegeven voor een ander probleem dat ik tegenkwam: de outline rond variërende meshes met meerdere texturen werkt niet zoals ik het zou willen (zie punt 7 bij screenshots). Door dit naar 1 textuur te brengen, werkt het zoals ik het zou willen. Het had nog wel kleine aanpassingen nodig bij de shader zelf, maar het was al vele beter zoals ik het verwachte.**

Conclusie

Wat geleerd?

* **technisch**
* **creatief**
* **persoonlijk**
* Creatief / technisch: bij 3d modellen dacht ik soms: “Hoe zou ik het met papier gemaakt hebben,” en dit paste ik dan toe (zie screenshot punt 5)
* Hoe minimap
* Hoe highscore met keys bijhouden
* Hoe shaders werken (shaders aanpassen)
* Niet altijd de moeilijke weg proberen, soms kan het ook makkelijk
* Je kan verschillende materials toewijzen met polygon selection (tags) en daarna alle aparte stukken samenvoegen tot 1 object. Daarna kan je de verschillende textures tot 1 texture samenvoegen door de texture te baken.
* Een movietexture kan je als een normale texture gebruiken (je kan het bv. in een rawimage steken). Voor sommige extensies kon unity niet de correcte encoder vinden (MPEG2).

Zelfevaluatie

* uhhh….

Verbeteringen

Wat zou je verbeteren/toevoegen als je nog 6 maanden zou kunnen doorwerken.

Bronnen

* **Unity docs algemeen**. <https://docs.unity3d.com/Manual/index.html>. Verschillende momenten.
* **Arduino Rotary Encoder**. <http://playground.arduino.cc/Main/RotaryEncoders>. Geraadpleegd op 3 maart 2017.
* **Meer info en code voor de Arduino Rotary Encoder**. W. Van Weyenberg, lector en schoolpromoter KdG Multimedia en CommunicatieTechnologie, persoonlijke communicatie, 3 maart 2017.
* **Outline met shader**. <http://wiki.unity3d.com/index.php/Silhouette-Outlined_Diffuse>. Geraadpleegd op 1 april 2017.
* **Loading screen**. <https://www.youtube.com/watch?v=xJQXoG3caGc>. Geraadpleegd op 2 april 2017.
* **Highscore met playerprefs**. <https://unity3d.com/learn/tutorials/topics/scripting/high-score-playerprefs>. Geraadpleegd op 10 mei 2017.
* **Minimap**. <http://blog.theknightsofunity.com/implementing-minimap-unity/>. Geraadpleegd op 11 mei 2017.
* **MovieTexture**. <https://forum.unity3d.com/threads/on-movie-end-switch-levels.47929/>. Geraadpleegd op 20 juni 2017.