



HOGESCHOOL ROTTERDAM / CMI

Computer Graphics

TINCGR02

Aantal studiepunten: 4

Cursusbeheerder: dr. W. M. Bergmann Tiest



Cursusbeschrijving

Cursusnaam:	Computer Graphics
Cursuscode:	TINCGR02
Aantal studiepunten en studiebelastinguren:	4 ec Dit studieonderdeel levert je 4 studiepunten op, hetgeen overeenkomt met een studielast van 112 uren. De verdeling van deze 112 uren over de collegeweken is als volgt: <ul style="list-style-type: none"> • 27 × 0,25 uur theorie filmpjes bekijken 7 uur • 14 × 0,5 uur werkcollege 7 uur • 7 × 4 uur onbegeleid werken aan opdrachten 28 uur • zelfstudie voor theoretische toets 68 uur • deelname aan toets 2 uur
Vereiste voorkennis:	<ul style="list-style-type: none"> • Wiskunde 2b - Lineaire algebra (TINWIS012B) • Programmeren 1 - imperatief programmeren (TINPRO02-1) • Programmeren 2 - object-georiënteerd programmeren (TINPRO02-2) • Programmeren 3 - object-georiënteerd programmeren (TINPRO02-3) • Programmeren 4 - object-georiënteerd programmeren (TINPRO02-4) • Programmeren 5b: C++ (TINPRO025B)
Werkvorm:	<ul style="list-style-type: none"> • On-line theorie filmpjes bekijken; • On-line werkcollege met gelegenheid voor vragen; • Zelfstandig werken aan opdrachten; • Zelfstudie
Toetsing:	Opdrachten (50 %), schriftelijke toets (50 %)
Leermiddelen:	<ul style="list-style-type: none"> • Boek: Zhiang Xiang. <i>Computer Graphics</i> (second edition). McGraw-Hill Schaum's Outline Series. ISBN: 0-07-135781-5 (aanbevolen, niet verplicht); • On-line bronnen en slides van colleges; • Je wordt geacht tijdens de werkcolleges over een computer (laptop) te beschikken met een werkende Python-, OpenGL- en Unity-installatie. Voor de opdrachten mag desgewenst ook van Java, C of C++ gebruik gemaakt worden.
Leerdoelen:	<p>De student kan:</p> <ol style="list-style-type: none"> 1. De basisprincipes van de visuele waarneming van vormen, kleuren, diepte en beweging uitleggen. 2. Verschillende technieken voor het weergeven van computerbeelden uitleggen. 3. Kleurruimtes begrijpen. 4. Verschillende basistechnieken met betrekking tot raster graphics toepassen. 5. 2D en 3D transformaties, zoals translatie, schaling en rotatie toepassen. 6. Deze transformaties als matrixoperaties voorstellen en hiermee werken. 7. Het algoritme voor Bézier-krommen toepassen. 8. Werken met de OpenGL-bibliotheek en hiermee 2D en 3D voorstellingen en animaties genereren. 9. De basisprincipes van raytracing uitleggen. 10. Eigenschappen van virtual en augmented reality uitleggen.
Inhoud:	Indrukwekkende computer graphics zijn de basis van veel computerspellen, maar zijn ook essentieel in wetenschappelijke en medische toepassingen. Om deze graphics er goed uit te laten zien en animaties soepel te laten lopen is veel computerkracht, wiskunde en slim programmeerwerk nodig. Deze cursus behandelt de basisprincipes van raster- en vectorgraphics (2D en 3D). Hierbij is er aandacht voor de manier waarop visuele waarneming werkt, en komen zowel de wiskundige achtergrond als de praktische toepassing van computer graphics aan bod. Je zal zelfstandig een programma ontwikkelen dat gebruik maakt van grafische technieken en gebruikersinteractie.
Cursusbeheerder:	dr. W. M. Bergmann Tiest
Datum:	18 november 2020

1 Algemene omschrijving

1.1 Inleiding

Indrukwekkende computer graphics zijn de basis van veel computerspellen, maar zijn ook essentieel in wetenschappelijke en medische toepassingen. Om deze graphics er goed uit te laten zien en animaties soepel te laten lopen is veel computerkracht, wiskunde en slim programmeerwerk nodig. Deze cursus behandelt de basisprincipes van raster- en vectorgraphics (2D en 3D). Hierbij is er aandacht voor de manier waarop visuele waarneming werkt, en komen zowel de wiskundige achtergrond als de praktische toepassing van computer graphics aan bod. De cursus bestaat uit 27 filmpjes waarin stukjes theorie behandeld worden, 14 on-line bijeenkomsten en 5 praktische opdrachten (practicum) waarin deze theorie toegepast wordt. Deze opdrachten moeten ingeleverd worden en maken deel uit van de eindbeoordeling van de cursus. De laatste opdracht heeft een grotere omvang en bestaat uit het maken van een compleet programma (met OpenGL in Python, Java, JavaScript, C, C++ of C#) dat gebruik maakt van de behandelde grafische technieken en gebruikersinteractie (muis en/of toetsenbord). Dit kan een spel zijn of een andere grafische toepassing, naar je eigen inzicht. Je wordt geacht om zowel tijdens de bijeenkomsten als zelfstandig daarbuiten aan de opdrachten te werken.

1.2 Relatie met andere onderwijseenheden

Deze cursus leunt zwaar op het rekenen met matrices zoals behandeld in het vak *Wiskunde 2b - Lineaire algebra* (TIN-WIS012B). Als je hier moeite mee hebt, wordt je geadviseerd om de stof van dit vak te herhalen voorafgaand aan les 7 (Vector Graphics). De opgedane kennis over het maken van visualisaties kan toegepast worden in de projecten, de stage en het afstuderen.

1.3 Leermiddelen

- Boek: Zhiang Xiang. *Computer Graphics* (second edition). McGraw-Hill Schaum's Outline Series. ISBN: 0-07-135781-5 (aanbevolen, niet verplicht);
- On-line bronnen en slides van colleges;
- Je wordt geacht tijdens de werkcolleges over een computer (laptop) te beschikken met een werkende Python-, OpenGL- en Unity-installatie. Voor de opdrachten mag desgewenst ook van Java, C of C++ gebruik gemaakt worden.

2 Programma

De cursus bestaat uit 7 weken met elk 2 on-line bijeenkomsten per week geroosterd. Vóór iedere bijeenkomst wordt je geacht de aangegeven filmpjes bekeken te hebben. Tijdens de bijeenkomsten wordt de behandelde stof besproken en is er gelegenheid tot het stellen van vragen. Bij filmpjes 5, 8, 12 en 18 hoort een inleveropdracht. Deze moet je maken en de week daarop ingeleverd hebben via Teams. De opdracht bij filmpje 23 is de eindopdracht; deze is groter en hier heb je 2 weken de tijd voor. Deze moet ook ingeleverd worden en moet tijdens de laatste on-line les mondeling toegelicht worden. De vier kleinere opdrachten en de eindopdracht bepalen samen het cijfer voor het onderdeel *opdrachten*. In de toetsweek wordt ook nog een schriftelijke toets gehouden.

Het programma is als volgt:

week (CMI)	week (TI)	les	filmpjes	huiswerk
2.2	1	1	1 Inleiding	inleveropdracht 1
			2 Het oog	
			3 Vormen zien	
		2	4 Kleuren zien	
			5 Diepte zien	
			6 Beweging zien	
2.3	2	3	7 Displays	inleveropdracht 2
		4	8 Kleuren afbeelden	
			9 Helderheid afbeelden	



2.4	3	5	10	Raster graphics: Gaussian blur	
			11	Edge detection	
		6	12	Rasteren & anti-aliasing	inleveropdracht 3
			13	Image & video compression	
2.5	4	7	14	Vector Graphics: coördinaten en vectoren	
			15	Polygonen	
			16	Transformaties	
		8	17	Matrixoperaties	inleveropdracht 4
			18	Projecties	
			19	Krommen	
				<i>Kerstvakantie</i>	
2.6	5	9	20	OpenGL: basis	
			21	OpenGL: geavanceerd	
		10	22	OpenGL: Bézier	eindopdracht
			23	OpenGL: shaders	
2.7	6	11	24	Unity	
		12	25	Raytracing	
2.8	7	13	26	Virtual Reality	
			27	Augmented Reality	
		14			demonstratie eindopdracht
2.9	8			Schriftelijke toets	

3 Aanwezigheid

Aanwezigheid is verplicht bij de mondelinge toelichting van de eindopdracht, omdat hiermee vastgesteld wordt of de student eigen werk heeft ingeleverd.

4 Toetsing en beoordeling

4.1 Procedure

De toetsing wordt gevormd door twee deeltolsten: de schriftelijke toets en de opdrachten (practicum). Het cijfer van de schriftelijke toets telt voor 50 % van de totale beoordeling voor de cursus en dient minimaal een 4 te zijn om een voldoende beoordeling voor de gehele cursus te kunnen verkrijgen. De andere 50 % wordt bepaald door de opdrachten (practicum). Compensatie is mogelijk; het gemiddelde van beide deeltolsten moet minstens 5,5 zijn om de cursus te behalen.

Schriftelijke toets De schriftelijke toets is in week 2.9 en bestaat uit een aantal meerkeuze- en open vragen. Voor voorbeeldvragen, zie bijlage 3.

Opdrachten De opdrachten 1–4 moeten uiterlijk 1 week na de bijbehorende les ingeleverd worden via Teams. De eindopdracht moet in de laatste on-line les mondeling worden toegelicht, en voor het einde van week 2.9 ingeleverd worden via Teams. Voor iedere opdracht is een cijfer tussen 0 en 10 te behalen. Opdrachten 1–4 tellen samen voor 50 % mee voor het cijfer van de deeltolst *Opdrachten*. De andere 50 % van de deeltolst wordt bepaald door de eindopdracht. Voor de beoordelingscriteria, zie bijlage 2.

Eindopdracht De eindopdracht luidt: programmeer een Computer Graphics-demo die gebruik maakt van de OpenGL-bibliotheek. Dit kan een spel zijn, een interactief kunstwerk of een ander demonstratie van Computer Graphics-technieken. De uitwerking is vrij te kiezen, maar moet aantonen dat je de geleerde theoretische concepten toe kan passen. Daarbij moet je in staat zijn om toelichting te geven op elk aspect van de code. De code moet geschreven zijn in Python, Java, JavaScript, C, C++ of C# en moet gebruik maken van de OpenGL-bibliotheek. De uitwerking moet uiterlijk tijdens de tweede les in week 2.8 gedemonstreerd en toegelicht worden. Een korte beschrijving (incl. mogelijkheden voor gebruikersinteractie), de (gedocumenteerde) broncode, en (indien van toepassing) een gecompileerde versie dienen ingeleverd te worden via Teams, uiterlijk eind week 2.9. Voor de beoordelingscriteria, zie bijlage 2.

Een voorbeeld van een eindopdracht:



Opdracht: Een bewegend 3D-model van het zonnestelsel, dat met behulp van de muis van alle kanten bekeken kan worden. De planeten hebben een realistische texture map en op de achtergrond zijn sterren zichtbaar.

Beoordeling: De opdracht bevat de volgende 7 aspecten en krijgt daarom het cijfer 7:

- massieve 3D-vormen
- animatie
- gebruikersinteractie
- 3D transformaties (rotatie, translatie en/of schaling)
- perspectief
- belichting (gebruik van lichtbronnen)
- texture mapping

4.2 Herkansing

In het geval dat het gemiddelde cijfer van de twee deoltoetsen onvoldoende is, kan je naar keuze ofwel de schriftelijke toets, ofwel het practicum, of beide herkansen. Als het cijfer voor de schriftelijke toets lager dan een 4 is, dan moet deze in ieder geval herkanst worden. De herkansing van de schriftelijke toets is in week 3.9. Dan moeten ook de opdrachten opnieuw ingeleverd zijn.

Voor hogerejaars gelden dezelfde regels als voor studenten die het vak voor de eerste keer volgen.

5 Changelog

Datum	Wijzigingen
18/03/2017	Eerste versie
29/03/2018	Programma en toetsing iets aangepast, toetsmatrijs toegevoegd
16/01/2019	Raytracing en Virtual Reality in één les, Unity toegevoegd
07/01/2020	Cursuscode veranderd, beoordeling Unity-opdracht toegevoegd
05/11/2020	Stof opgedeeld in 27 onderwerpen, 2 lessen per week. Beoordeling aangescherpt. Unity-opdracht weer verwijderd

Bijlage 1: toetsmatrijs

Toetsonderdeel	Bloom-niveau	Leerdoel
Opdracht 1	3, 6	1
Opdracht 2	3	3
Opdracht 3	3	4
Opdracht 4	3	5, 6
Eindopdracht	3, 6	8
Schriftelijke toets	2, 3	1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Bloom-niveaus:

1. Remember
2. Understand
3. Apply
4. Analyse
5. Evaluate
6. Create

Bijlage 2: beoordelingscriteria opdrachten

Opdracht 1

- Er is een statische tekening ingeleverd: 1 pnt
- Er is een filmpje/animatie ingeleverd: 2 pnt (niet cumulatief met een statische tekening).
- In de tekening/animatie is diepte-informatie aanwezig op basis van:
 - afdekken (occlusie): +1 pnt
 - parallax of beweging: +1 pnt
 - grootte (perspectief): +1 pnt
 - scherptediepte: +1 pnt
 - mistigheid (contrast): +1 pnt
 - schaduw (shading): +1 pnt
 - vergentie: +1 pnt
 - dispariteit: +1 pnt
- Alle aanwezige diepte-informatiebronnen zijn correct benoemd: +1 pnt
- De tekening of het filmpje vertegenwoordigt naar de mening van de docent een uitzonderlijk hoge esthetische, technische of entertainment-waarde: +1 pnt
- Het maximale cijfer is 10.

Opdracht 2

- Het programma maakt geen gebruik van één van de voorgeschreven templates voor Python, Java, C of C++: 0 pnt, ook geen andere punten voor deze opdracht mogelijk.
- Er is een programma ingeleverd dat (compileert en) draait zonder foutmeldingen: 1 pnt
- De functie `RGBtoCMY()` is correct geïmplementeerd: +1 pnt
- De functie `CMYtoRGB()` is correct geïmplementeerd: +1 pnt
- De functie `RGBtoHSL()` is correct geïmplementeerd: +2 pnt
- De functie `HSLtoRGB()` is correct geïmplementeerd: +2 pnt
- De functie `transparency()` is correct geïmplementeerd: +1 pnt
- Het programma is voorzien van informatief commentaar: +1 pnt
- Het programma is naar de mening van de docent zeer elegant of zit technisch zeer slim in elkaar: +1 pnt

Opdracht 3

- Het programma maakt geen gebruik van één van de voorgeschreven templates voor Python, Java, C of C++: 0 pnt, ook geen andere punten voor deze opdracht mogelijk.
- Er is een programma ingeleverd dat (compileert en) draait zonder foutmeldingen: 1 pnt
- De functie `rasterline()` tekent lijnen in de richtingen 0° en 180° correct: +1 pnt
- De functie `rasterline()` tekent lijnen in de richtingen 90° en 270° correct: +1 pnt
- De functie `rasterline()` tekent lijnen in richtingen tussen -45° en 45° correct: +1 pnt
- De functie `rasterline()` tekent lijnen in richtingen tussen 135° en 225° correct: +1 pnt
- De functie `rasterline()` tekent lijnen in richtingen tussen 45° en 135° correct: +1 pnt
- De functie `rasterline()` tekent lijnen in richtingen tussen 225° en 315° correct: +1 pnt
- Alle getekende lijnen zijn lang genoeg (lopen helemaal tot het aangegeven eindpunt): +1 pnt
- Het programma is voorzien van informatief commentaar: +1 pnt
- Er wordt anti-aliasing toegepast op de getekende lijnen: +1 pnt

Opdracht 4

- Het programma is geschreven in een andere taal dan Python, Java, C, of C++: 0 pnt, ook geen andere punten voor deze opdracht mogelijk.
- Er is een programma ingeleverd dat (compileert en) draait zonder foutmeldingen: 1 pnt
- Het programma tekent een kubus in een orthografische projectie door middel van het toepassen van de juiste projectiematrix: +1 pnt
- Het programma tekent een kubus in een parallelprojectie door middel van het toepassen van de juiste projectiematrix: +2 pnt, geen punten voor orthografische projectie
- Het programma tekent een kubus in een isometrische projectie door middel van het toepassen van de juiste projectiematrix: +3 pnt, geen punten voor orthografische of parallelprojectie
- Het programma tekent een kubus in een perspectiefprojectie door middel van het toepassen van de juiste projectiematrix: +4 pnt, geen punten voor orthografische, parallel- of perspectiefprojectie



- De coördinaten van de kubus zijn door middel van het toepassen van een translatiematrix verschoven: +1 pnt
- De coördinaten van de kubus zijn door middel van het toepassen van een rotatiematrix 30° gedraaid rond de verticale as: +1 pnt
- De rotatie-, translatie- en projectiematrices zijn eerst gecombineerd tot één transformatiematrix die vervolgens op de coördinaten van de kubus wordt toegepast: +1 pnt.
- Het programma is voorzien van informatief commentaar: +1 pnt
- Het programma is naar de mening van de docent zeer elegant of zit technisch zeer slim in elkaar: +1 pnt

Eindopdracht

- Het programma is geschreven in een andere taal dan Python, Java, JavaScript, C, C++ of C#: 0 pnt, ook geen andere punten voor deze opdracht mogelijk.
- Het programma (compileert en) draait niet zonder foutmeldingen en is ook niet werkend gedemonstreerd aan de docent: 0 pnt, ook geen andere punten voor deze opdracht mogelijk.
- Het programma maakt gebruik van de OpenGL-bibliotheek om te laten zien:
 - polygonen (draadmodellen of 2D-vlakken): +1 pnt
 - massieve 3D-vormen: +1 pnt
 - krommen: +1 pnt
 - animatie: +1 pnt
 - gebruikersinteractie: +1 pnt
 - 3D transformaties (rotatie, translatie en/of schaling): +1 pnt
 - perspectief: +1 pnt
 - belichting (gebruik van lichtbronnen): +1 pnt
 - shading (kleuren van 3D-objecten): +1 pnt
 - texture mapping: +1 pnt
 - shaders in GLSL: +1 pnt
- Het programma vertoont naar de mening van de docent een uitzonderlijk hoge esthetische, technische of entertainment-waarde: +1 pnt
- Het maximale cijfer is 10.

Bijlage 3: Voorbeeldvragen tentamen TINCGR

- Welke van de volgende begrippen is *geen* bron van diepte-informatie voor het menselijk visueel waarneemsysteem? Kies slechts één antwoord.
 - occlusie
 - vergentie
 - reflectie
 - dispariteit
 - parallax
- Een kleur in de RGB-kleurruimte, waarbij de componenten tussen 0 en 1 liggen, wordt beschreven door de waarden $(R, G, B) = (0.1, 0.2, 0.3)$. Wat zijn de waarden van de componenten van dezelfde kleur in de CMY-kleurruimte?
- Welke van de volgende uitspraken is waar? Kies slechts één antwoord.
 - Door anti-aliasing wordt een afbeelding scherper.
 - Door anti-aliasing kost het opslaan van een afbeelding minder ruimte.
 - Door anti-aliasing worden randen in een afbeelding gladder.
 - Door anti-aliasing kunnen randen in een afbeelding gedetecteerd worden.
- Gegeven de rotatiematrix
$$\mathbf{R} = \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix},$$
 - Bepaal de matrix voor een rotatie van een object over 45° om de oorsprong.
 - Wat zijn de nieuwe coördinaten van het punt $p = (1, 4)$ na de rotatie?
- Wat is het nut van het gebruik van homogene coördinaten? Kies slechts één antwoord.
 - Homogene coördinaten stellen je in staat een translatie in de 3D ruimte voor te stellen als een vermenigvuldiging van een matrix met een vector in plaats van een optelling van twee vectoren. De vermenigvuldiging van een matrix met een vector is efficiënter te berekenen dan de optelling van twee vectoren.
 - Homogene coördinaten stellen je in staat een translatie in de 3D ruimte voor te stellen als een vermenigvuldiging van een matrix met een vector in plaats van een optelling van twee vectoren. Omdat ook rotatie en schaling door vermenigvuldiging van een matrix met een vector kunnen worden voorgesteld, is het mogelijk één matrix te berekenen, die de totale transformatie voorstelt. Dit is efficiënter dan achtereenvolgens toepassen van de afzonderlijke transformaties.
 - Homogene coördinaten zorgen ervoor dat je alle hoekpunten van objecten in de 3D ruimte uitdrukt in hetzelfde coördinatenstelsel. Dit leidt tot een eenvoudige, logische opbouw van de transformaties.
 - Homogene coördinaten zorgen ervoor dat je 2D transformaties kunt schrijven als 3D transformaties door toevoeging van een extra coördinaat die steeds 1 blijft. Dit is efficiënt omdat de berekeningen hierdoor op de grafische kaart kunnen worden uitgevoerd, die gespecialiseerd is in 3D berekeningen.
- Construeer een tweede-orde Bézier-kromme met behulp van de controlepunten $p_1 = (1, 1)$, $p_2 = (3, 1)$ en $p_3 = (3, 3)$.

7. Welke soort projectie wordt in het volgende OpenGL-programma gebruikt? Kies slechts één antwoord.

- (a) orthografische projectie
- (b) parallelprojectie
- (c) isometrische projectie
- (d) perspectiefprojectie

```
from OpenGL.GL import *
from OpenGL.GLUT import *

def display():
    glClear(GL_COLOR_BUFFER_BIT)
    glPushMatrix()
    glFrustum(-1.333, 1.333, -1, 1, 1, 5)
    glTranslate(0, 0, -2)
    glutWireCube(1)
    glPopMatrix()
    glFlush()

glutInit()
glutInitWindowSize(640, 480)
glutCreateWindow("Projectie".encode("ascii"))
glutDisplayFunc(display)
glutMainLoop()
```

8. Waar of niet waar: Bij een afbeelding gemaakt met raytracing, waarbij alleen diffuse lichtbronnen gebruikt worden, hangt de kleurverdeling over een voorwerp af van de kijkrichting.