**HOGESCHOOL ROTTERDAM / CMI**

ComputerGraphics 2

(Inleiding)

**TIRCGR02**

Goedgekeurd door:

**(namens toetscommissie)**

Datum:

|  |
| --- |
| Aantal studiepunten: 4 ects  Modulebeheerder: J. de Hooge |

I**nhoudsopgave:**

[Modulebeschrijving 3](#_Toc351285889)

[1. Algemene omschrijving 5](#_Toc351285890)

[1.1 Inleiding 5](#_Toc351285891)

[1.2 Inhoud theorie, deels behandeld in college, deels te verwerven via zelfstudie (zie literatuur) 5](#_Toc351285892)

[1.3 Inhoud praktikum: 6](#_Toc351285893)

[1.4 Relatie met andere onderwijseenheden 7](#_Toc351285894)

[1.6 Literatuur 7](#_Toc351285895)

[2. Programma 8](#_Toc351285896)

[3. Toetsing en beoordeling 9](#_Toc351285897)

[3.1 Procedure 9](#_Toc351285898)

[Bijlage 1 Toetsmatrijs 10](#_Toc351285899)

[Bijlage 2 Voorbeeldtoets 11](#_Toc351285900)

[B 2.1 Voorbeeldtoets 11](#_Toc351285901)

[B 2.2 Antwoorden voorbeeldtoets 12](#_Toc351285902)

[Bijlage 3 Studielastnormering (ects) 15](#_Toc351285903)

# Modulebeschrijving

|  |  |  |
| --- | --- | --- |
| **Modulenaam:** | Computergraphics 2 (inleiding) | |
| **Modulecode:** | **TIRCGR02** | |
| **Aantal studiepunten en studiebelastinguren:** | Dit studieonderdeel levert de student 4 studiepunten op, hetgeen overeenkomt met een studielast van 112 uren. De verdeling van deze 112 uren over de collegeweken is als volgt:  Hoorcolleges:  gedurende 8 weken: 8 \* 2 uur 16 uur  Overige uren per week:  Literatuurstudie (8 \* 3 uur per week) 24 uur  Uitwerken van vragen/cases (8 \* 4 uur per week) 32 uur  Toetsvoorbereiding en deelname 40 uur  Totaal 112 uur | |
| **Vereiste voorkennis:** | tircps01,tircms02/03,tirlin01,tirana01, tirpro01/03,tirdat01/02,tirebs0, tirpif01, tirrea01 | |
| **Werkvorm:** | Hoorcollege en zelfstudie en praktikum  Aanwezigheid tijdens het praktikum is verplicht, aanwezigheid tijdens de hoorcolleges wordt bijgehouden.  Het kan in uitzonderlijke gevallen voorkomen dat het practicum niet kan worden bijgewoond of dat opdrachten door overmacht (ziekte) niet op tijd kunnen worden ingeleverd. In dat geval dient dit onmiddellijk (dus niet achteraf) bij de docent gemeld te worden. De docent beslist of er inderdaad sprake van overmacht is, tenzij er een doktersverklaring of vergelijkbaar wordt overlegd. | |
| **Toetsing:** | Er wordt een tentamen gehouden over de in het hoorcollege behandelde stof.  Er dienen projecten te worden uitgevoerd en met een groep van max 3 personen.  Het eindcijfer is het rekenkundig gemiddelde van de projecten en het tentamen. Het minimumcijfer voor project of tentamen is een 4 om voor een beoordeling in aanmerking te komen | |
| **Leermiddelen:** | Theorie:  Computer Graphics (second edition) , Auteur : Zhiang Xiang  Uitgever: McGraw-Hill Schaum’s Outline Series , ISBN: 0-07-135781-5  Praktikum:  Dictaat Inleiding grafische computertechnieken , P.J.den Brok | |
| **Draagt bij aan HBO-I competentie:** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | analyse | advies | ontwerp | realisatie | beheer | | gebuikers  interactie |  |  |  |  |  | | bedrijfs  processen |  |  |  |  |  | | software | nivo 3 | nivo 1 | nivo 3 | nivo 3 | nivo 3 | | infra-structuur |  |  |  |  |  | | hardware  interfacing |  |  |  |  |  | | |
|  |  | |
|  |  | |
| **Leerdoelen:** | M.b.t. competentie Software Analyse:  - Puntsgewijs kunnen formuleren van requirements  - Requirements kunnen beoordelen op haalbaarheid en prioriteiten  - Kunnen beoordelen of de requirements binnen de planning/beschikbare tijd kunnen worden gerealiseerd  M.b.t. competentie Software Advies:  - Kunnen onderkennen welke functionaliteit primair is en welke functionaliteit secundair met het oog op het voorstellen van timeboxes  - Kunnen toelichten welke onderliggende infrastructuur/libraries is/zijn aan te bevelen  M.b.t. competentie Software Ontwerp:  - Kunnen uitleggen en toepassen van concepten achter OpenGL  - Kunnen ontwerpen van modules, classes en/of functies en interfaces hiervan  - Te verwachten bottlenecks kunnen onderkennen (real time gedrag, geheugenconsumptie)  - Een acceptabele user interactie kunnen ontwerpen (consequent, met redelijke responstijd)  M.b.t. competentie Software Realisatie:  - Een robuust stapsgewijs traject kunnen uitzetten volgen van ontwerp naar gerealiseerde code, met tests na elke stap, afgeleid uit de requirements  - De oorzaak van foutief programmagedrag systematisch kunnen opsporen  M.b.t. competentie Beheer:  - Aspecten die de veiligheid beinvloeden kunnen benoemen en kunnen aangeven hoe het ontwerp hiermee rekening kan houden | |
| **Inhoud:** | 1 Grafische computertechnieken  1.2 OpenGL  2 Meetkundige basisbegrippen  2.1 Affiene transformaties  2.2 Homogene co¨ordinaten  3 3D Graphics  3.1 2D versus 3D  3.2 3D transformaties  3.3 Grafische objectstructuren  3.4 3D viewing  3.5 Projecties  4 Raster Graphics  4.1 Algemeen | 4.2 Praktische implementatie  4.3 Kleur  4.4 Vlakvullingen met kleuren of grijstinten  4.5 Anti-aliasing  5 Afbeelden in 3D  5.1 Algemeen  5.2 Netstructuren  5.3 Krommen  5.4 Oppervlakken  5.5 Hidden surface removal  5.6 Lichteffecten  5.7 Shading |
| **Opmerkingen:** | Alle programmatuur is vrij verkrijgbaar op Internet (zie dictaat en literatuur) | |
| **Modulebeheerder:** | J. de Hooge | |
| **Datum:** | 17 maart 2013 | |

# 1. Algemene omschrijving

## 1.1 Inleiding

De leerstof bestaat uit theorie afgewisseld en gekoppeld met toepassingen, voorbeelden en vaardigheidsoefeningen. Het een kan niet zonder het ander.

Compententies en leerdoelen zijn geformuleerd in de voorafgaande tabel.

Grafische toepassingen kunnen globaal verdeeld worden in:

* een al dan niet interactieve presentatie van informatie (grafieken, afbeeldingen, cartografie, XV, WWW),
* een interactief ontwerp, CAD of Computer Aided Design (mechanisch ontwerp, elektronische schakelingen, architectuur, electronic publishing, POV-Ray, Blender, SodiPodi, AutoCAD e.d.),
* simulaties en animaties (vluchtsimulatie, tekenfilms, video-spelletjes),
* een gebruikersinterface (windowsystemen, WWW browsers, CAVE),
* computer art (kan iets of niets van elke vorige toepassing hebben).

De grafische basistechnieken zijn het maken van:

* 2D lijntekeningen (gebruik van lijnen),
* 2D beelden (gebruik van vlakken),
* 3D lijntekeningen (lijnen, wire frames met hidden line removal),
* 3D beelden (vlakken, solid modeling met hidden surface removal).
* Binnen deze groepen zijn er nog grote verschillen mogelijk:

1. monochroom, met grijstinten of met kleuren
2. intensiteitsverschillen
3. egale of met niet-egale vlakken .

Bij interactieve grafische toepassingen zijn ondermeer de volgende aspecten van belang:

* interactie met de gebruiker,
* invoer van gegevens om het maken van de gewenste figuren te sturen,
* manipulatie van (deel)figuren,
* vormgeving van complexe figuren.

## 1.2 Inhoud theorie, deels behandeld in college, deels te verwerven via zelfstudie (zie literatuur)

1. Grafische computertechnieken

1.1 Geschiedenis

1.2 OpenGL

1.2.1 Elementaire instructies

1.2.2 Projecties en views

1.2.3 Clipping

1.2.4 Interactie

1.2.5 Voorbeeld van een OpenGL programma

2. Meetkundige basisbegrippen

2.1 Affiene transformaties

2.1.1 Translatie

2.1.2 Rotatie om de oorsprong

2.1.3 Schaalverandering

2.1.4 Shearing

2.1.5 Het groepsgedrag van affiene transformaties

2.2 Homogene coordinaten

2.2.1 Het toepassen van homogene coordinaten

2.2.2 Concatenatie van transformaties van homogene coordinaten

2.2.3 2D transformaties met OpenGL

3. 3D Graphics

3.1 2D versus 3D

3.1.1 Homogene 3D coordinaten

3.2 3D transformaties

3.3 Grafische objectstructuren

3.4 3D viewing

3.4.1 Grafische pijplijn

3.5 Projecties

4. Raster Graphics

4.1 Algemeen

4.2 Praktische implementatie

4.2.1 Het algoritme van Bresenham (ideeen erachter kunnen verwoorden)

4.3 Kleur

4.3.1 Het RGB additieve kleurmodel (praktisch kunnen toepassen)

4.3.2 Het YMC subtractieve kleurmodel (weten wat het is)

4.4 Vlakvullingen met kleuren of grijstinten

4.4.1 Scan line algoritme

4.5 Anti-aliasing

5. Afbeelden in 3D

5.1 Algemeen

5.2 Netstructuren

5.3 Krommen

5.4 Oppervlakken

5.5 Hidden surface removal

5.6 Lichtffecten

5.6.1 De wet van Lambert

5.6.2 Het Phong-model

5.7 Shading

5.7.1 Ray tracing

5.7.2 Flat-shading

5.7.3 Gouraud-shading

5.7.4 Phong-shading

5.7.5 Texture mapping

## 1.3 Inhoud praktikum:

In een praktijkgestuurd project dient een project te worden uitgevoerd :

‘Interactief ontwerpen met OpenGL’

Algemene voorwaarden

* Er dient gewerkt worden werken in groepen van max 3 personen,
* De groep dient met een projectvoorstel te komen met minimaal 5 puntsgewijs geformuleerde requirements
* Een project dient minimaal 5 deelproblemen te bevatten,
* Er dient een korte beschrijving van het onderwerp en een planning gegeven te worden.
* Aangegeven moet worden welke technieken en deelproblemen worden gebruikt
* De docent moet de beschrijvingen goedkeuren en voortgang controleren,
* ( Het is niet nodig om na afloop een uitgebreid verslag te geven. Wat wel nodig is :een analyse, broncode, demonstratie met toelichting.)

De volgende deelproblemen kunnen o.a. opgenomen zijn in het project:

* modelleren met polygonen,
* modelleren met krommen,
* interactie (callbacks),
* 3D transformaties,
* lichtmodellen,
* shading,
* texture mappings.

Waar kan je ideen opdoen?

* http://www.cvgpr.uni-mannheim.de/student projects/cg03/
* http://graphics.stanford.edu/courses/#cs248
* http://www2.cs.pitt.edu/%7Epanos/teaching/1566-repository/#Proj01
* http://www.student.math.uwaterloo.ca/˜cs488/gallery.html.
* googleisyourfriend.

Voorbeeld van een projectvoorstel:

* Namen: . . .
* Doel: modelleren van een bal in een 3D-doos, bal kan bewegen en kaatst tegen wanden. De bal heeft een lichtreflectie. De bal vliegt in de richting aangegeven door de muis.
* Realisatie:

1. bal: draadmodellen met texture mapping,

2. lichtreflectie: een van de OpenGL modellen,

3. vliegen: bal heeft rechte baan, regelmatig verplaatsen,

4. baaneffecten: bal kaatst tegen muren,

5. interactie: met muisbeweging.

## 1.4 Relatie met andere onderwijseenheden

|  |  |
| --- | --- |
| **Voorkennis** |  |
| **Traject** | **modulen** |
| computertechniek | tircps01,tircms02, tircms03, |
| programmeren | tirpro01,tir pro02, tirpro03, tirdat01, tirdat02 |
| problemsolving | tirlin01, tirana01 |
| Technische informatica | tirebs01,tirpif01, tirrea01 |

|  |  |
| --- | --- |
| **Toepassing in andere leerstof en projecten** |  |
| **Projecten** | **modulen** |
| Interfacing | ( spellen maken voor spelcomputers , virtual reality , beeldbewerkingsprojecten , visionsystemen in combinatie met robots, simulaties van robots agvv’s , visualisatie van productieprocessen , multimediapresentaties). |
| Afstuderen | Veel van de behandelde onderwerpen komen tijdens de afstudeerprojecten aan de orde. |

## 1.6 Literatuur

De presentaties en modulewijzer zijn te vinden op http://www.med.hro.nl/hojac/cgr02/

**Verplicht Boek**:

Computer graphics (second edition) , Auteur : Zhiang Xiang

Uitgever: McGraw-Hill Schaum’s Outline Series , ISBN: 0-07-135781-5

**Aanbevolen boeken:**

[1] Edward Angel, *Interactive Computer Graphics*, Addison Wesley, 2003, ISBN 0-321-19044-0.

[2] Leen Ammeraal, *Computer Graphics for Java Programmers*, Wiley, 2002, ISBN0-471-98142-7.

[3] Samuel R. Bush, *3-D Computer Graphics*, Cambridge University Press, 2003,ISBN0-521-82103-7.

[4] Foley, van Dam, Feiner, Hughes, *Computer Graphics, Principles and Practice, second edition in C*, Addison Wesley, 1996, ISBN 0-201-84840-6.

[5] Donald Hearn, M. Pauline Baker, *Computer Graphics with OpenGL*, 3rd edition, Pearson Prentice Hall, ISBN 0-13-015390-7 (hardcover) ISBN 0-13-0120238-3 (paperback)

[6] F.S. Hill, Jr., *Computer Graphics Using OpenGL,* Prentice Hall, 2000, ISBN 0-02-354856-8.

[7] Brian W. Kernigan, Dennis M. Ritchie, *The C Programming Language, 2nd ed.*,Prentice Hall, 1988, ISBN 0-13-110362-8.

[8] GlennW. Rowe, *Computer Graphics with Java*, Palgrave, 2001, ISBN 0-333-92097-X.

**Andere bronnen:**

[9] http://www.cs.unm.edu/˜angel/

[10] http://www.hef.kun.nl/˜pfk/teaching/icg/docs/icghand.pdf

[11] http://www.opengl.org/

[12] http://fl y.cc.fer.hr/˜unreal/theredbook/

[13] http://chromium.sourceforge.net/

[14] http://www.avl.iu.edu/technology/cave/

# 2. Programma

**2.1 Theorie**

De theorie bestaat uit hoofdstukken uit het boek Computer Graphics, Auteur : Zhiang Xiang

**Weekindeling theorie:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week** | **Hoofdstuk** | **Theorie** | **Paragraaf** | **Bladzijden** |
| 1 | 1,2 | Inleiding  Beeldvoorstelling | 1.1,1.2  2.1 t/m 2.8 | 1 t/m 5  6 t/m 16 |
| 2 | 3 | Scan conversie | 3.1 t/m 3.10 | 25 t/m 51 |
| 3 | 4,5 | 2-dimensionale transformaties  2 dimensionale beeldsysteem en clipping | 4.1 t/m 4.4  5.1 t/m 5.5 | 68 t/m 76  89 t/m 99 |
| 4 | 6,7 | 3-dimensionale transformaties  Wiskundige projectie | 6.1 t/m 6.4  7.1 t/m 7.3 | 114 t/m 118  151 t/m 159 |
| 5 | 9 | Geometrische voorstelling | 9.1 t/m 9.11 | 174 t/m 189 |
| 6 | 10 | Verborgen oppervlakten | 10.1 t/m 10.8 | 197 t/m 209 |
| 7 | 11 | Kleuren en schaduw modellen | 11.1 t/m 11.4 | 229 t/m 239 |
| 8 | 12 | Ray-tracing | 12.1 t/m 12.7 | 251 t/m 261 |
| 9 | Bespreken tentamen |  |  |  |
| 10 | Tentamen |  |  |  |

**2.2 Praktikum**

Er dient een graphics project te worden uitgevoerd met OpenGL.

In het dictaat Computergraphics van P.J. den Brok staan aanwijzingen hoe met OpenGL gewerkt dient te worden Het dictaat is in principe volledig.

Daarnaast is het z.g.n. OpenGL Red Book (on line te vinden) een goede praktische informatiebron.

Wel wordt studenten die onder MS-Windows willen werken aanbevolen om een OpenGl-bibliotheek te installeren en tevens GLU en GLUT (gratis te downloaden).

Studenten met Linux en Mac kunnen gebruik maken van bibliotheken zoals Mesa en OpenGL die standaard in hun distributies aanwezig zijn. OpenGL bibliotheken zijn vrij op Internet verkrijgbaar.

Studenten met Windows XP kunnen gebruik maken van bibliotheken met DirectX 8.1.

Windows XP Professional supports the Open Graphics Library (OpenGL) 1.1 specification.

Zie ook:

[http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/enus/Default.asp?url=/resources/documentation/Windows/XP/all/reskit/en-us/prdm\_mtm\_ldiy.asp](http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/Default.asp?url=/resources/documentation/Windows/XP/all/reskit/en-us/prdm_mtm_ldiy.asp) en

<https://www.cs.tau.ac.il/~shulyl/courses/mta/CGraphics-04A/html/help.html>

**Weekindeling praktikum:**

|  |  |  |
| --- | --- | --- |
| **Week** | **Docent** | **Projectgroep** |
| 1 | Introductie project en toelichting geven.  Algemene afspraken maken over spreekuren e.d. | Bestuderen opdracht en samenstellen groepen. Globale analyse van het project. |
| 2 | Administratie groepsindelingen. Maken van groepsafspraken. | Opstellen pakket van eisen + kwaliteitscriteria.  Opstellen werkverdeling + planning.  Verstrekken tussentijdse inleveropdracht 1. |
| 3 | Adviseren, Verwijzen naar literatuur e.d. | Globale analyse en achitectuurontwerp. |
| 4 | Adviseren. Verwijzen naar literatuur e.d. | Gedetailleerde analyse en ontwerpactiviteiten.  Verstrekken tussentijdse inleveropdracht 2. |
| 5 | Adviseren. Verwijzen naar literatuur e.d. | Gedetailleerde analyse en ontwerpactiviteiten. |
| 6 | Adviseren. Verwijzen naar literatuur e.d. | Implementeren en testen van deelprojecten. |
| 7 | Adviseren. Verwijzen naar literatuur e.d. | Implementeren en testen van totaalsysteem. |
| 8 | Adviseren. Verwijzen naar literatuur e.d. | Testen opstelling. Demonstratie. Toelichting |
| 9 | Aanvang beoordelingen | Aanbrengen verbeteringen indien nodig |
| 10 | Beoordeling van het projectverslag, project-management en demonstratie. | Bespreking |

1. Getoetst wordt op het product: welke deelgebieden zijn gebruikt , creativiteit , analyse van de
2. programmasource, enz
3. Een korte demonstratie met toelichting dient te worden gegeven
4. Er wordt een individueel cijfer gegeven, dat mede afhangt van de bijdrage aan de toelichting.
5. Met name dient ieder groepslid de werking van de gehele sourcecode te kunnen uitleggen.
6. Dit zal steekproefsgewijs worden gecontrolleerd.
7. In uitzonderlijke gevallen is een aanvullende opdracht mogelijk, dit ter beoordeling van de docent.

# 3. Toetsing en beoordeling

## 3.1 Procedure

1. De schriftelijke toets is aan het eind van de onderwijsperiode.
2. Het cijfer van het tentamen telt voor 50 % van het totale cijfer voor de module en dient
3. minimaal een 4 te zijn om een beoordeling voor de gehele module te verkrijgen.

De twee tussentijdse opdrachten betreffen het praktische kunnen toepassen van de wiskundige achtergrond van beeldverwerking (transformaties, projecties). Uiterlijk bij aanvang van de practicumsessie twee weken na het verstrekken van een opdracht dient deze te worden ingeleverd. Ingeleverd dienen te worden:

- De sourcecode, op papier, voorzien van doeltreffend commentaar en duidelijk identifiers.

- De uitvoer, op papier, waar het numerieke data betreft voorzien van duidelijke labeling.

De eindopdracht bestaat uit het maken van een programma met OpenGL. Het programma moet aantonen dat de student de geleerde theoretische concepten toe kan passen. De individuele opdracht wordt gekozen in overleg met de docent . Het programma moet gedemonstreerd worden en met documentatie (standaard verslagmethode) worden ingeleverd. De documentatie moet een korte beschrijving bevatten van de functionaliteit, het ontwerp en programmalisting.

Getoetst wordt op het product: welke deelgebieden zijn gebruikt , creativiteit , analyse van de programmasource, enz.

Een kort demonstratie met toelichting dient te worden gegeven

1. Het cijfer voor het programma(praktikum) telt voor 50% van het totale cijfer voor de module en dient
2. minimaal een 4 te zijn om een beoordeling voor de gehele module te verkrijgen.

Het prakticumcijfer is als volgt samengesteld:

1e tussentijdse opdracht: 25%

2e tussentijdse opdracht: 25%

Eindopdracht: 50%

Het eindcijfer voor het gehele vak wordt derhalve als volgt berekend:

0.125 \* cijfer 1e tussentijds opdracht + 0.125 \* cijfer 2e tussentijdse opdracht + 0.25 \* cijfer eindopracht + 0.5 \* cijfer tentamen

# Bijlage 1 Toetsmatrijs

|  |  |
| --- | --- |
| Puntsgewijs kunnen formuleren van requirements | Eindopdracht prakticum |
| Requirements kunnen beoordelen op haalbaarheid en prioriteiten | Eindopdracht prakticum |
| Kunnen beoordelen of de requirements binnen de planning/beschikbare tijd kunnen worden gerealiseerd Kunnen onderkennen welke functionaliteit primair is en welke functionaliteit secundair met het oog op het voorstellen van timeboxes | Eindopdracht prakticum |
| Kunnen toelichten welke onderliggende infrastructuur/libraries is/zijn aan te bevelen | Tentamen  Eindopdracht prakticum |
| Kunnen uitleggen en toepassen van concepten achter OpenGL | Tentamen  Tussentijdse opdrachte prakticum |
| Kunnen ontwerpen van modules, classes en/of functies en interfaces hiervan | Tussentijdse opdrachten en eindopdracht prakticum |
| Te verwachten bottlenecks kunnen onderkennen (real time gedrag, geheugenconsumptie) | Tentamen  Eindopdracht practicum |
| Een acceptabele user interactie kunnen ontwerpen (consequent, met redelijke responstijd) | Eindopdracht practicum |
| Een robuust stapsgewijs traject kunnen uitzetten volgen van ontwerp naar gerealiseerde code, met tests na elke stap, afgeleid uit de requirements | Tussentijdse opdrachten en eindopdracht practikum |
| De oorzaak van foutief programmagedrag systematisch kunnen opsporen | Tussentijds opdrachten en eindopdracht practicum |
| Aspecten die de veiligheid beinvloeden kunnen benoemen en kunnen aangeven hoe het ontwerp hiermee rekening kan houden | Tentamen |

# Bijlage 2 Voorbeeldtoets

## B 2.1 Voorbeeldtoets

Toegestane hulpmiddelen :

* Boek: Computer Graphics (second edition) , Auteur : Zhiang Xiang
* Modulewijzer tircgr02
* Power Point presentaties hoorcollege tircgr02
* Rekenmachine, géén PC/laptop

1. Een RGB kleurenbeeld kan worden geconverteerd naar een grijswaarde beeld met de formule: (10 pt)

0,299R+ 0,587G +0,114B.

Neem aan dat getPixel(x,y,rgb) pixel waarden inleest van een 24 bit beeld en setPixel(x,y,i) een pixelwaarde

toekent aan een outputbeeld dat een grijswaarde lookuptable gebruikt.

Ontwerp een pseudo-code programma om een kleuren beeld te converteren naar een grijswaarde beeld

2. Ontwerp een pseudo-code programma die een DDA-algoritme beschrijft voor scan-converting een (10 pt)

lijn waarvan de helling ligt tussen -150 en +450 (d.w.z . |m| ≤1)

3. Welke stappen zijn vereist om scan-conversie toe te passen op een cirkel die Bresenham’s algoritme (20 pt)

gebruikt.

4. Gegeven de rotatiematrix (10 pt)

a. Bepaal de matrix voor een rotatie van een object over 300 om de oorsprong

b. Wat zijn de nieuwe coordinaten van het punt P(2,-4) na de rotatie?

5. Gegeven de algemene matrix voor verschaling t.o.v. een vast punt P(h,k) met vector **v**=h**I**+k**J** als (20 pt)

Wat worden de nieuwe hoekpunten van een driehoek als deze vergroot wordt met een factor 2.

De bestaande driehoek heeft de hoekpunten A(0,0) , B(1,1) en C(5,2). Het hoekpunt C(5,2)

blijft op zijn plaats.

6. Bepaal de normalisatie transformatie N welke een cirkel gebruikt met straal 5 en centrum (1,1) (10 pt)

als window en een cirkel met straal ½ en centrum(½,½) als viewport.

7. Als P0(0,0) , P1(1,2), P2(2,1), P3(3,-1), P4(4,10), P5(5,5) gegeven datapunten zijn Er wordt (20 pt)

interpolatie toegepast die gebaseerd is op cubic B-splines om een kromme te vinden

die door die datapunten gaat. Bepaal de knopenverzameling t0,….,t9 die gebruikt kan

worden om de cubic B-splines te definieeren.

## B 2.2 Antwoorden voorbeeldtoets

1.

int i,j,c,rgb[3];

for( j=0; j< height; j++)

for ( i=0;i < width i++) {

getPixel(i,j,rgb);

c=0,299\*rgb[0]+0,587\*rgb[1] +0,144\*rgb[2];

setPixel(i,j,c);

}

2. Neem aan dat x1 < x2 voor de 2 eindpunten (x1 , y1) en (x2 , y2)

int x= x1,y;

float xf = y1 ,m=(y1 - y1) / (x2- x1) ;

while( x<= x2 ) {

y=Floor(xf +0,5);

setPixel(x,y);

x++;

xf = xf +m:

}

3. Aantal stappen Bresenham:

1. De coördinaten van het cirkelcentrum zijn (h,k). Zet de beginwaarden van de variabelen x=0;

y=straal r en d=3-2r.

2. Test om te bepalen of de gehele cirkel gedaan is. Als x>y, stop

3. Teken de 8 punten die gevonden zijn door symmetrie t.o.v. het centrum(h,k), voor de lopende

(x,y) coördinaten.

Plot(x+h,y+k) Plot(-x+h,-y+k)

Plot(y+h,x+k) Plot(-y+h,-x+k)

Plot(-y+h,x+k) Plot(y+h,-x+k)

Plot(-x+h,y+k) Plot(x+h,-y+k)

4. Bereken de lokatie van het volgende pixel.

Als d <0 , dan wordt d=d+4x+6 en x=x+1.

Als d ≥ 0 ,dan wordt d=d+4(x-y)+10 en x=x+1 en y=y-1.

5. Ga terug naar stap 2.

4.

a. , invullen =300 geeft

b. De nieuwe coordinaten worden gevonden met matrixvermenigvuldiging:

=

5. Het hoekpunt C(5,2) is het translatiepunt. Dat betekent dat h=5 en k=2.

De translatievector is: **v**=5**i**+2**j**

De verschaling is 2 voor beide zijden. Dit betekent dat a=2 en b=2

Invullen geeft de verschalingmatrix:

De hoekpunten zijn A(0,0) , B(1,1) en C(5,2). Dit kunnen we in een homogene matrix plaatsen:

n.l. A=[0,0,1]T B=[1,1,1]T A=[5,2,1]T .

De nieuwe hoekpunten worden dan

Hieruit volgen de nieuwe hoekpunten A’(-5,-2) , B’(-3,0) en C’(5,2).

6. Bepaal de normalisatie transformatie N van:

cirkel straal 5 en centrum (1,1) als window

cirkel straal ½ en centrum (½,½) als viewport

N wordt bepaald door translatie en verschaling:

De translatie gaat van centrum (1,1) naar centrum(½,½)

De verschaling gaat straal 5 naar straal ½ en wordt dan s=1/10 Zowel voor x en y-richting

N=S1/10, 1/10,C.Tv Hierin is S de verschalings matrix t.o.v. centrum C. Tv de translatie over vector **v** met **v** =-1/2**i**-1/2**j**

De verschalingsmatrix t.ov C is:

Hierin is h=½, k=½ a=1/10 b=1/10

Invullen voor verschaling geeft :

De translatiematrix is:

Hierin is t=-½ u =-½ immers **v** =-1/2**i**-1/2**j**

Invullen voor translatie geeft:

7. Gegeven P0(0,0) , P1(1,2), P2(2,1), P3(3,-1), P4(4,10), P5(5,5) als datapunten

Gevraagd: Bepaal de knopenverzameling t0,….,t9 die gebruikt kan worden om de

cubic B-splines te definieeren.

De knopen kunnen gekozen worden volgens 1 van de 2 schema’s. Als m=3 (cubic splines)

en n=5 (aantal datapunten) geldt:

1. Kies:

t0=t1=t2=t3=-1 (<x0) en t6=t7=t8=t9=6 (>xn) .

De overgebleven knopen worden bepaald volgens:

Hieruit volgt:

2. Een alternatief schama voor cubic splines is:

t0=t1=t2=t3=-1 (<x0) en t6=t7=t8=t9=6 (>xn) .

en de overgebleven knopen worden gekozen volgens

ti+4=xi+2 , met i=0,…,n-4

Hieruit volgt: t4=2 en t5=3

Dat de antwoorden overeenkomen komt doordat er een uniforme verdeling van de datapunten is langs de x-as

|  |  |  |  |
| --- | --- | --- | --- |
| **Studielastnormering (ects)** | |  |  |
| **Betreft:** | **Naam module:** | **Module-code:** |  |
|  | computer graphics | tircgr02 |  |
|  | aantal weken | aantal lesuren van 50 minuten | klokuren |
| Lesuren | 8 | 2 | 13 |
|  |  |  |  |
| Zelfstudie |  |  |  |
| *Leestijd* | aantal pagina's |  |  |
|  | 150 | 3 per uur | 50 |
|  |  | 6 per uur | 0 |
|  |  | 10 per uur | 0 |
|  |  |  |  |
| *Presentaties* |  |  |  |
|  |  |  |  |
| *Overlegtijd* |  |  |  |
|  |  |  |  |
| *Uitzoektijd/ research* |  |  |  |
|  |  |  |  |
| *Niet ingeroosterde lestijd* |  |  | 20 |
|  |  |  |  |
|  |  |  |  |
| Toetsen | voorbereiden |  | 20 |
|  | toets |  | 2 |
|  | nabespreking |  | 1 |
|  |  |  |  |
| Werkstuk, verslag, rapport, scriptie |  |  |  |
|  | uitzoeken |  | 25 |
|  | overleggen |  |  |
|  | schrijven |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Stage, Praktijkopdracht | voorbereiding |  |  |
|  | aanwezigheid |  |  |
|  | overleg |  |  |
|  |  |  |  |
|  |  |  |  |
| Subtotaal in klokuren |  |  | 131 |
| Ruis 5% |  |  | 6,6 |
|  |  |  |  |
|  |  |  |  |
| Totaal in klokuren |  |  | 137,9 |
|  |  |  |  |
| Totaal in studiepunten (ects) |  | hier alleen veelvouden van 28 uur | 4 |

# Bijlage 3 Studielastnormering (ects)