

**Computer Graphics (in2770)**  
28 oktober 2003, 9.00 - 11.00 uur.

**N.B.: Dit tentamen bestaat uit 30 opgaven**  
**Totaal aantal bladzijden: 11**

---

Aanwijzingen bij het invullen van de antwoordformulieren:

- Vul het antwoordformulier bij voorkeur in met **pen** (beslist **geen rode pen** gebruiken en **geen doorhalingen**). Met **potlood** invullen mag ook, maar niet te licht (gummen mag).
  - Vergeet niet je **naam**, **studierichting** en **studienummer** in te vullen.
  - Vul je **studienummer** zowel in cijfers als ook met streepjes in (controleer goed).
- 

**Opgave 1**

Een afbeelding gemaakt met Gouraud-shading ziet er minder goed uit dan een afbeelding gemaakt met Phong-shading. Waarom?

- Bij Gouraud-shading wordt de lichtbron minder goed benaderd
- Bij Gouraud-shading wordt de kleurwaarde (in plaats van de normaal) geïnterpoleerd
- Bij Gouraud-shading wordt geen spiegelende reflectie bepaald
- Bij Gouraud-shading wordt geen rekening gehouden met de ambient-term

**Opgave 2**

De transformatiematrix voor een 3D schaling (in homogene coördinaten) met schalingsfactoren  $s_x$ ,  $s_y$  en  $s_z$  ten opzichte van een punt  $F = (x_F, y_F, z_F)$  noemen we  $S_F$ . Het beeld  $P'$  van een punt  $P$  onder deze transformatie kan bepaald worden uit  $P' = S_F \cdot P$

Hoe ziet matrix  $S_F$  eruit?

a. 
$$S_F = \begin{pmatrix} s_x & 0 & 0 & (1-s_x)x_F \\ 0 & s_y & 0 & (1-s_y)y_F \\ 0 & 0 & s_z & (1-s_z)z_F \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

b. 
$$S_F = \begin{pmatrix} s_x & 0 & 0 & -s_x x_F \\ 0 & s_y & 0 & -s_y y_F \\ 0 & 0 & s_z & -s_z z_F \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

c. 
$$S_F = \begin{pmatrix} s_x & 0 & 0 & s_x x_F \\ 0 & s_y & 0 & s_y y_F \\ 0 & 0 & s_z & s_z z_F \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

d. 
$$S_F = \begin{pmatrix} s_x & 0 & 0 & x_F \\ 0 & s_y & 0 & y_F \\ 0 & 0 & s_z & z_F \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

### Opgave 3

Beschouw de volgende uitspraken over een algemeen 3D viewing systeem met een perspectief projectie:

- (I) Bij het projecteren van een scene op het projectievlak geldt dat indien een object tussen het projectievlak en het Center Of Projection (COP) ligt, dit object kleiner wordt afgebeeld dan wanneer het object achter het projectievlak ligt ten opzichte van het COP.
- (II) Het viewing frustum wordt volledig bepaald door het Center Of Projection, het View Reference Point en de near en far planes.

Welke van deze uitspraken is juist?

- |    | (I)     | (II)    |
|----|---------|---------|
| a. | juist   | juist   |
| b. | juist   | onjuist |
| c. | onjuist | juist   |
| d. | onjuist | onjuist |

### Opgave 4

Waarom worden bij de opslag in de computer van een polygonmodel van een 3D solid object naast een geordende lijst van hoekpunten van ieder polygon ook de coëfficiënten opgeslagen van de vlakvergelijking van het vlak waarin het polygon ligt?

- a. omdat anders het genereren van afbeeldingen van het 3D object meer rekentijd vergt.
- b. omdat anders het 3D object dubbelzinnig is beschreven, d.w.z. bij één model kunnen meerdere geldige objecten horen.
- c. omdat anders het 3D object niet volledig beschreven is, d.w.z. uit een model is de vorm van het bedoelde object niet altijd af te leiden.
- d. omdat anders alleen lijnafbeeldingen van het 3D object kunnen worden gemaakt ten gevolge van het ontbreken van informatie die nodig is voor het berekenen van een kleur voor de zichtbare polygons.

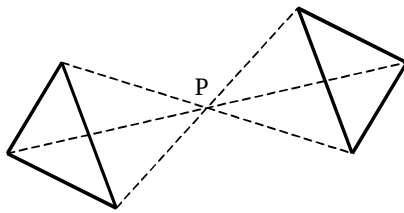
### Opgave 5

Een y-richting shearing transformatie t.o.v. de lijn  $x=0$  beeldt het punt  $P(3, 4)$  af op  $P'(3, 16)$ . Wat is de shear parameter?

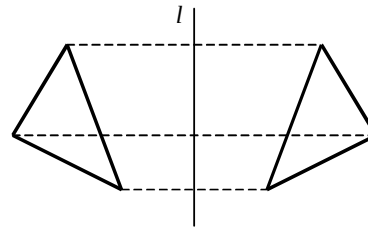
- a. 2
- b. 4
- c. 8
- d. 12

## Opgave 6

De onderstaande figuur toont het onderscheid tussen een *puntspiegeling* en een *lijnspiegeling*.



*puntspiegeling t.o.v. punt P*



*lijnspiegeling in lijn l*

Beschouw de volgende 2D transformatie-matrices:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Hoeveel van deze transformatie-matrices representeren een *puntspiegeling*?

- a. 1
- b. 2
- c. 3
- d. 4

## Opgave 7

We willen een cirkel waarvan de begrenzing in de frame buffer gegeven is, inkleuren door gebruik te maken van een fill algoritme. De achtergrond waarop de begrenzing van de cirkel getekend is, is egaal en heeft een andere kleur dan die van de begrenzing en de vulkleur. Beschouw de volgende beweringen:

- (I) Een cirkel die correct ingekleurd wordt met het 4-connected boundary fill algoritme wordt zeker ook goed ingekleurd met het 8-connected boundary fill algoritme.
- (II) Een cirkel die correct ingekleurd wordt met het 8-connected boundary fill algoritme wordt zeker ook goed ingekleurd met het 4-connected boundary fill algoritme.

Van deze beweringen is:

- |    | (I)     | (II)    |
|----|---------|---------|
| a. | juist   | juist   |
| b. | juist   | onjuist |
| c. | onjuist | juist   |
| d. | onjuist | onjuist |

## Opgave 8

Beschouw de volgende uitspraken over texture mapping.

- (I) Bij een bilineaire mapping worden rechte lijnen (in een willekeurige richting) in de textuuruimte gemapt op rechte lijnen in de schermruimte.  
(II) Bij perspectivische mapping worden rechte lijnen (in een willekeurige richting) in de textuuruimte gemapt op rechte lijnen in de schermruimte.

Van deze uitspraken is

- |    | (I)     | (II)    |
|----|---------|---------|
| a. | juist   | juist   |
| b. | juist   | onjuist |
| c. | onjuist | juist   |
| d. | onjuist | onjuist |

## Opgave 9

Met welke uitspraak wordt het beste de werking van de `glutMainLoop()` functie uit de OpenGL Utility Toolkit beschreven?

De functie `glutMainLoop()` creëert een lus waarbinnen herhaaldelijk

- a. een volgend grafisch uitvoerprimitief wordt afgebeeld op een output device.
- b. alle in een programma gedefinieerde event procedures worden uitgevoerd.
- c. gewacht wordt op het optreden van een event waarna de callback functie van de betreffende event wordt uitgevoerd.
- d. gewacht wordt totdat de gebruiker de Close menu optie kiest om een grafisch window te sluiten.

## Opgave 10

Welk grafisch invoerapparaat is het handigst voor het overnemen in een computer van gegevens vanaf een landkaart?

- a. touch panel
- b. joystick
- c. tablet met pen
- d. muis

## Opgave 11

Het depth sorting algoritme voor hidden surface removal sorteert een polygonlijst door herhaaldelijk voor paren van polygons te bepalen of deze in de polygonlijst in de juiste volgorde staan en deze zo nodig om te wisselen (zoals een "normaal" sorteeralgoritme). De situatie kan zich hierbij voordoen dat twee polygons die reeds zijn verwisseld later nogmaals verwisseld moeten worden (doordat er sprake is van cyclische overlap van twee of meer polygons). Hoe wordt dit probleem opgelost?

- Er wordt bijgehouden of de polygons al verwisseld zijn en als dit zo is dan worden ze een volgende keer niet verwisseld.
- Er wordt bijgehouden of de polygons al verwisseld zijn en als dit zo is dan wordt een volgende keer een van de twee polygons op de snijlijn van het vlak van het andere polygon gesplitst in twee nieuwe polygons.
- Een tweetal polygons kan meer dan een keer verwisseld worden, maar na verloop van tijd zal dit tweetal polygons niet meer getest worden en houdt het verwisselen op.
- De polygons worden (tijdelijk) uit de polygonlijst verwijderd en nadat het sorteerproces beëindigd is met een extra sorteerstap een voor een op de juiste plaats aan de lijst toegevoegd.

## Opgave 12

Een lijn door de punten A en B met  $A=(x_A, y_A)$  en  $B=(x_B, y_B)$  kunnen we representeren met

(I) een *vergelijking*  $y = ax + b$  met  $a = (y_B - y_A) / (x_B - x_A)$  en  $b = (x_B y_A - x_A y_B) / (x_B - x_A)$ .

(II) een *parametervoorstelling*  $(x, y) = (x_A, y_A) + u (\Delta x, \Delta y)$  met  $\Delta x = x_B - x_A$  en  $\Delta y = y_B - y_A$

Hoe clipt het Liang en Barsky lijn clipping algoritme een lijnstuk AB?

- Het algoritme berekent, uitgaande van de *vergelijking*  $y = ax + b$ , de snijpunten van alle doorgetrokken windowgrenzen met de lijn en bepaalt vervolgens of deze snijpunten op de rand van het window liggen en tussen de punten A en B in.
- Het algoritme gaat uit van de *parametervoorstelling* van de lijn en bepaalt het parameterinterval  $[u_1, u_2]$  voor het deel van de lijn binnen het window. Door invullen van  $u_1$  en  $u_2$  in de parametervoorstelling worden de eindpunten van het geclipte lijnstuk bepaald.
- Het algoritme bepaalt eerst 4-bits codes van de eindpunten en berekent, alleen als de lijn niet direct verworpen of geaccepteerd kan worden, met behulp van de *parametervoorstelling* snijpunten van de windowgrenzen met het lijnstuk.
- Het algoritme bepaalt eerst 4-bits codes van de eindpunten en berekent, alleen als de lijn niet direct verworpen of geaccepteerd kan worden, met behulp van de *vergelijking*  $y = ax + b$  snijpunten van de windowgrenzen met het lijnstuk.

### Opgave 13

Een parallelprojectie waarbij alle drie de assen van het coördinatenstelsel van het af te beelden object (d.w.z. alle drie de hoofdrichtingen in het model) een gelijke hoek maken met het projectievlak heet

- a. scheve projectie
- b. isometrische projectie
- c. cabinet projectie
- d. cavalier projectie

### Opgave 14

Het z-buffer algoritme is een hidden surface removal algoritme. Bij dit algoritme moet vele malen bij een pixel  $(x, y)$  de z-coördinaat van een punt in een polygon met vlakvergelijking  $ax+by+cz+d=0$  berekend worden. Voor naast elkaar liggende pixels  $(x, y)$  en  $(x+1, y)$  kan dit met een incrementele berekening:  $z(x+1, y) = z(x, y) + \text{inc}$ . Hierbij betekent  $z(x, y)$  de diepte (dus z-coördinaat) van het polygon voor pixel  $(x, y)$ . Wat is de juiste increment  $\text{inc}$  waarmee uit de diepte  $z(x, y)$  voor pixel  $(x, y)$  de diepte  $z(x+1, y)$  voor pixel  $(x+1, y)$  berekend kan worden?

- a.  $\text{inc} = -a / c$
- b.  $\text{inc} = -a / b$
- c.  $\text{inc} = -b / c$
- d.  $\text{inc} = b / c$

### Opgave 15

Een lijnstuk AB wordt geclipt met het Cohen en Sutherland lijn clipping algoritme. De 4-bits codes van de twee eindpunten van het lijnstuk zijn  $\text{codeA} = 0110$  en  $\text{codeB} = 1001$ . Hierbij geven de 4 bits (van rechts naar links) een ligging links, rechts, onder en boven het window aan. Welke uitspraak is juist?

- a. Lijnstuk AB wordt op grond van de codes  $\text{codeA}$  en  $\text{codeB}$  geaccepteerd.
- b. Lijnstuk AB wordt op grond van de codes  $\text{codeA}$  en  $\text{codeB}$  verworpen.
- c. Lijnstuk AB wordt op grond van de codes  $\text{codeA}$  en  $\text{codeB}$  niet verworpen en niet geaccepteerd, maar het staat vast dat na clippen van een of meer delen van het lijnstuk een laatst overblijvend gedeelte geaccepteerd zal worden.
- d. Lijnstuk AB wordt op grond van de codes  $\text{codeA}$  en  $\text{codeB}$  niet verworpen en niet geaccepteerd, en het staat nog niet vast of na clippen van een of meer delen van het lijnstuk een laatst overblijvend gedeelte geaccepteerd of verworpen zal worden.

## Opgave 16

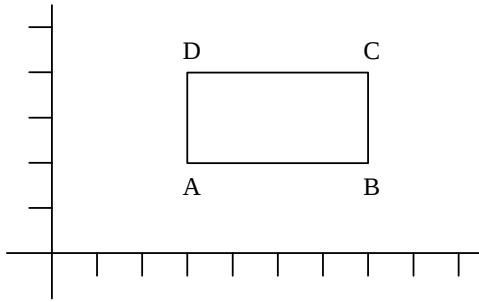
We gebruiken de volgende notatiewijze voor 2D transformaties:

$T_{(a,b)}$  is een translatie over de vector (a, b).

$R_{O,\alpha}$  is een rotatie om de oorsprong over  $\alpha$  graden.

$S_{(sx,sy)}$  is een schaling t.o.v. de oorsprong met schalingsfactoren  $s_x$  en  $s_y$ .

Rechthoek ABCD uit onderstaande figuur kan op verschillende manieren (door een combinatie van spiegeling, rotatie, translatie en schaling) op zichzelf worden afgebeeld.



Welke samengestelde transformatie beeldt ABCD af op BCDA, d.w.z. A wordt op B afgebeeld, B op C, C op D en D op A ?

- a.  $T_{(5,3)} \cdot R_{O,90} \cdot S_{(0.5,2.0)} \cdot T_{(-5,-3)}$
- b.  $T_{(5,3)} \cdot R_{O,90} \cdot T_{(-5,-3)}$
- c.  $T_{(5,3)} \cdot S_{(0.5,2.0)} \cdot T_{(-5,-3)}$
- d.  $T_{(5,3)} \cdot R_{O,180} \cdot T_{(-5,-3)}$

## Opgave 17

Met het DDA lijn scan conversie algoritme wordt een lijn AB met  $A = (200, 200)$  en  $B = (100, 30)$  afgebeeld. Welke pixels krijgen beide de kleur van de lijn?

- a. de pixels (131, 83) en (131, 84)
- b. de pixels (131, 83) en (132, 84)
- c. de pixels (131, 84) en (132, 84)
- d. de pixels (132, 83) en (132, 84)

## Opgave 18

In een VRML virtuele wereld moet een object met een Sphere geometrie geleidelijk van kleur veranderen als de waarnemer in de buurt van het object komt. Tussen welke typen knopen moet een route worden gedefinieerd om dit effect te realiseren?

- a. ProximitySensor => TimeSensor => ColorInterpolator => Material
- b. ProximitySensor => TimeSensor => ColorInterpolator => Sphere
- c. TimeSensor => ProximitySensor => ColorInterpolator => Material
- d. TimeSensor => ProximitySensor => ColorInterpolator => Sphere

### Opgave 19

Welke component van het lichtreflectie model van Phong veroorzaakt highlights op objectoppervlakken?

- a. de spotlight component
- b. de oppervlakte lichtbron component
- c. de diffuse reflectie component
- d. de spiegelende reflectie component

### Opgave 20

Hoeveel impliciete events horen bij een *field* en hoeveel bij een *exposed field* in VRML?

|    | field | exposed field |
|----|-------|---------------|
| a. | 0     | 0             |
| b. | 0     | 2             |
| c. | 2     | 0             |
| d. | 2     | 2             |

### Opgave 21

Beschouw de volgende code (met OpenGL functie aanroepen):

```
glMatrixMode (GL_MODELVIEW);  
glLoadIdentity ();  
glRotatef (theta, vx, vy, vz);  
glScalef (sx, sy, sz);  
glTranslatef (tx, ty, tz);
```

Objecten die hierna gedefinieerd worden ondergaan de transformaties uit de bovenstaande code. In welke volgorde worden de transformaties uitgevoerd?

- a. De volgorde is, onafhankelijk van de volgorde van de calls in de code, altijd eerst transleren, dan roteren en als laatste schalen.
- b. De volgorde is, onafhankelijk van de volgorde van de calls in de code, altijd eerst schalen, dan roteren en als laatste transleren.
- c. De volgorde is afhankelijk van de volgorde van de calls in de code. In dit geval komt dat neer op eerst roteren, dan schalen en als laatste transleren.
- d. De volgorde is afhankelijk van de volgorde van de calls in de code. In dit geval komt dat neer op eerst transleren, dan schalen en als laatste roteren.

## Opgave 22

Een polygon scan conversie algoritme maakt gebruik van een actieve zijden lijst (AZL). Welke zijden zitten er in deze AZL?

- a. Alle zijden van het polygon die niet horizontaal lopen
- b. Alle zijden van het polygon die de huidige scan line snijden
- c. Alle zijden van het polygon die geen andere zijden van het polygon snijden.
- d. Alle zijden van het polygon die niet verborgen zijn achter andere polygons.

## Opgave 23

Bij scan conversie wordt

- a. raster data getransformeerd in de frame buffer
- b. raster data geconverteerd naar een andere resolutie
- c. vector data getransformeerd naar een ander coördinatenstelsel
- d. vector data geconverteerd naar raster data

## Opgave 24

In een computerprogramma dat gebruik maakt van OpenGL komt het volgende programmafragment voor

```
// argumenten xmin, ymin, width, height in deze volgorde
glViewport (200.0, 500.0, 400.0, 200.0);
glMatrixMode (GL_PROJECTION);
glLoadIdentity ();
// argumenten xmin, xmax, ymin, ymax in deze volgorde
glOrtho (100.0, 700.0, 200.0, 600.0, -1.0, 1.0);
```

Op welk punt in het display window wordt het punt met wereldcoördinaten (460.0, 300.0) afgebeeld?

- a. op punt (380.0, 650.0)
- b. op punt (380.0, 550.0)
- c. op punt (440.0, 650.0)
- d. op punt (440.0, 550.0)

## Opgave 25

Welke stap komt niet voor in het afbeeldingsproces als we een afbeelding maken met de ray tracing methode?

- a. bepalen van dichtstbijzijnde snijpunt van straal met objectoppervlak
- b. toepassen van een licht reflectiemodel
- c. uitvoeren van de window to viewport transformatie
- d. uitvoeren van de viewing transformatie

### Opgave 26

Gegeven zijn de punten  $A = (100, 550)$  en  $B = (300, 150)$ . AB is een zijde van een polygon dat met een textuur erop wordt afgebeeld. Hoekpunt A correspondeert met punt  $(0.0, 0.5)$  uit de textuur en hoekpunt B met punt  $(1.0, 1.0)$  uit de textuur. De textuurcoördinaten worden lineair geïnterpoleerd over de zijde van het polygon.

Welke textuurcoördinaten worden gebruikt voor punt  $P = (260, 230)$  op zijde AB ?

- a.  $(0.8, 0.9)$
- b.  $(0.75, 0.875)$
- c.  $(0.5, 0.75)$
- d.  $(0.2, 0.6)$

### Opgave 27

Welke conclusie is juist als de 'bounding volumes' van twee bewegende objecten elkaar overlappen in een frame van een animatie?

- a. De twee objecten zijn met elkaar in botsing.
- b. De twee objecten kunnen met elkaar in botsing zijn. Er is een nauwkeuriger test nodig om te bepalen of de objecten inderdaad met elkaar in botsing zijn.
- c. De twee objecten zijn nog niet met elkaar in botsing, maar zullen spoedig botsen.
- d. De twee objecten zijn nog niet met elkaar in botsing, maar er is een kans dat zij spoedig zullen botsen.

### Opgave 28

Een textuur moet op een polygon worden afgebeeld. Antialiasing gebeurt hierbij met een mip-map. Een hoe groot gedeelte van de mip-map wordt gebruikt voor de opslag van de versie van de textuur op de helft van de maximale resolutie?

- a.  $\frac{1}{16}$
- b.  $\frac{3}{16}$
- c.  $\frac{1}{4}$
- d.  $\frac{3}{8}$

### Opgave 29

De matrix  $\begin{pmatrix} \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2}\sqrt{2} & 0 \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2}\sqrt{2} & 0 \\ -\frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$  is de transformatiematrix van een rotatie om een lijn  $l$

over een hoek van 90 graden. Wat is de correcte parametervoorstelling van de lijn  $l$ ?

- a.  $\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \lambda \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$       b.  $\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \lambda \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}$       c.  $\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \lambda \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$       d.  $\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \lambda \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}$

### Opgave 30

Wat is het voordeel van het gebruik van een acyclische graaf (Directed Acyclic Graph) in plaats van een tree om de scene op te slaan?

- a. Met een acyclische graaf ziet de uiteindelijke rendering op het scherm er beter uit.
- b. Collision checking en scene graph traversal is eenvoudiger met een acyclische graaf.
- c. Caching van display lists is eenvoudiger met een acyclische graaf.
- d. Met een acyclische graaf kunnen meerdere kopieën van identieke objecten in de scene geplaatst worden zonder fysieke kopieën in de scene graph te hangen.

### einde tentamenopgaven