Le raytracer … à la rencontre de l’informatique, des mathématiques et de la physique

Les calculs matriciels dans le *ray tracing*

Par Simon Vézina
Département de physique
Collège de Maisonneuve

http://en.wikipedia.org/wiki/Ray_tracing_(graphics)

Le raytracer … à la rencontre
de l'informatique, des mathématiques
et de la physique

Les calculs matriciels dans le *ray tracing*

Préambule

- Diplômé de l'Université de Montréal en **Physique-informatique** en automne 2002.

- Diplômé de l'Université de Montréal en Microprogramme en enseignement postsecondaire en automne 2003 (stage réalisé au collège en A2003).

- Début de maîtrise en théorie des champs quantiques à l'Université de Montréal en automne 2004 (interrompue en H2005).

- Enseignant au département de physique au Collège de Maisonneuve depuis H2005.

- Coordonnateur du programme de Sciences de la nature par intérim de A2005 à H2006.

- Membre du comité du programme Sciences informatiques et mathématiques (SIM) depuis A2008.

- Projet d'aide à la réussite A2015-H2016 : Le ray tracer … à la rencontre de l'informatique, des mathématiques et de la physique.

Le raytracer … à la rencontre
de l'informatique, des mathématiques
et de la physique

Les calculs matriciels dans le *ray tracing*

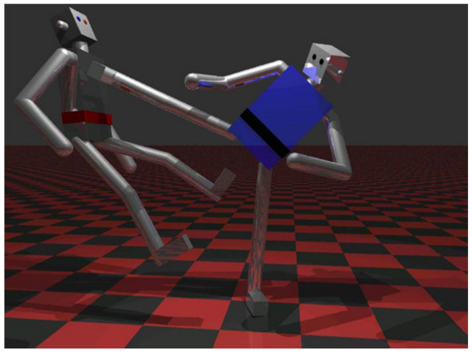
Conférence : 1ier octobre 2015

Les calculs matriciels dans le *ray tracing*

Plan de la conférence :

1) En quelques mots … le ray tracing.
2) Le rayon et l'intersection
3) Pourquoi utiliser des matrices ?
4) Les matrices de transformation en format 4x4
5) L'algorithme de l'intersection d'une géométrie transformée
6) Un exemple …
7) Des questions … et merci !

http://www.cs.utah.edu/~jstratto/state_of_ray_tracing/



IFT3350 : Par Simon Vézina et Yannick Simard (février 2002)

Le raytracer … à la rencontre
de l'informatique, des mathématiques
et de la physique

Les calculs matriciels dans le *ray tracing*

1) En quelques mots … le *ray tracing*

En quelques mots, un ***ray tracer*** est un **logiciel** capable de produire des **images de synthèse** à partir d'un environnement 3D à l'aide d'un algorithme de rayon lancé récursivement depuis la position d'une caméra.

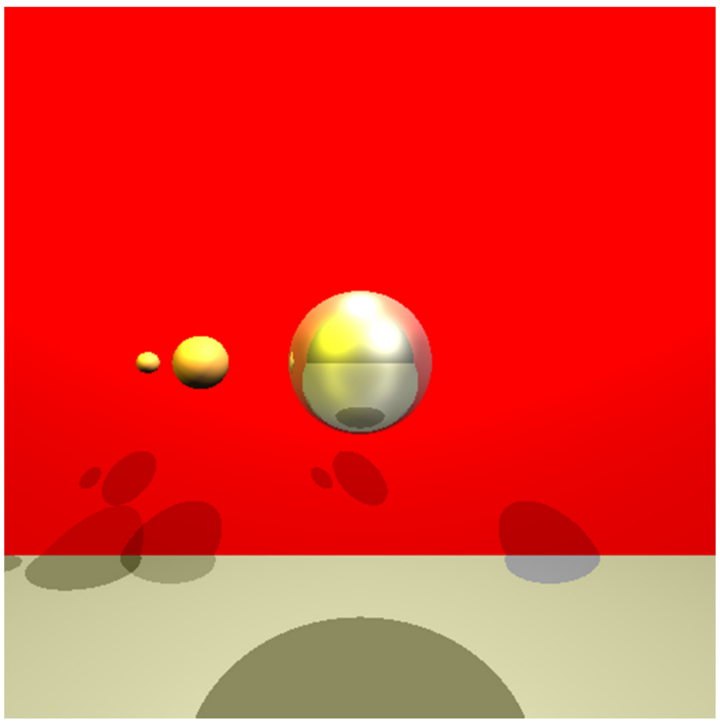


Image sans sphère transparente
(SIMRenderer)

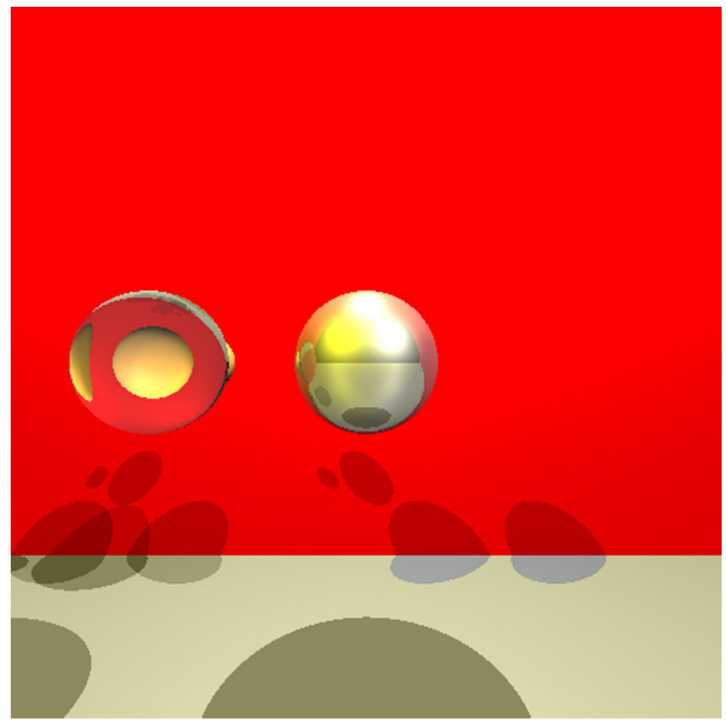
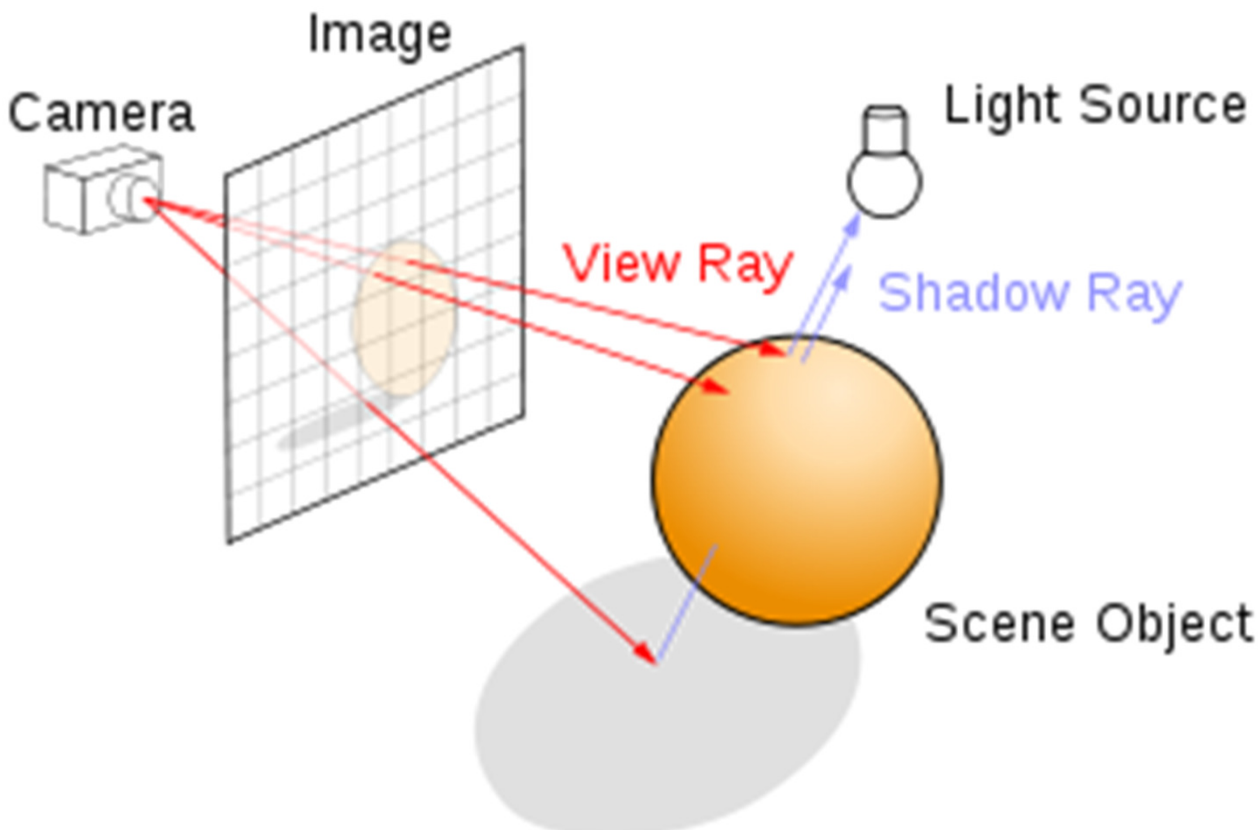


Image avec sphère transparente
devant deux sphères
(SIMRenderer)

Les calculs matriciels dans le *ray tracing*

1) En quelques mots … le *ray tracing*



http://en.wikipedia.org/wiki/Ray_tracing_(graphics)

Un **rayon** est lancé depuis une caméra et **touche** des **objets** de la scène. Selon **l'éclairage** et les obstacles à la lumière, la lumière peut être perçue dans la caméra sous la forme d'une **couleur.** On répète le lancer de rayon pour **l'ensemble des pixels** de l'image.

Les calculs matriciels dans le *ray tracing*

1) En quelques mots … le *ray tracing*

- Pour les nostalgiques …

http://www.dosgamesarchive.com/download/wolfenstein-3d/

http://www.primagames.com/games/wolfenstein-3d/feature/5-ways-wolfenstein-3d-changed-video-games

Avant de produire des images par *ray tracing*, car le calcul était trop exigent pour les PC de l'époque, on utilisait un algorithme de *ray casting* pour évaluer la visibilité des objet 3D (exemple : Jeux Wolfenstein 3-D par id Software).

1) En quelques mots … le *ray tracing*

- Ce que ça implique …

Mathématique :
- **calcul vectoriel**
- **transformation matricielle**
- intersection entre droite et forme géométrique

Informatique :
- architecture informatique
- paquetage multiple
- classe abstraite
- héritage
- Redéfinition de méthode
- récursivité
- base de données
- gestion d'exception

Physique :
- modèle d'illumination
- optique géométrique
- optique physique
- photons et énergie
- interaction matière-lumière

Chimie :
- spectre d'émission et d'absorption des matériaux

Les calculs matriciels dans le *ray tracing*

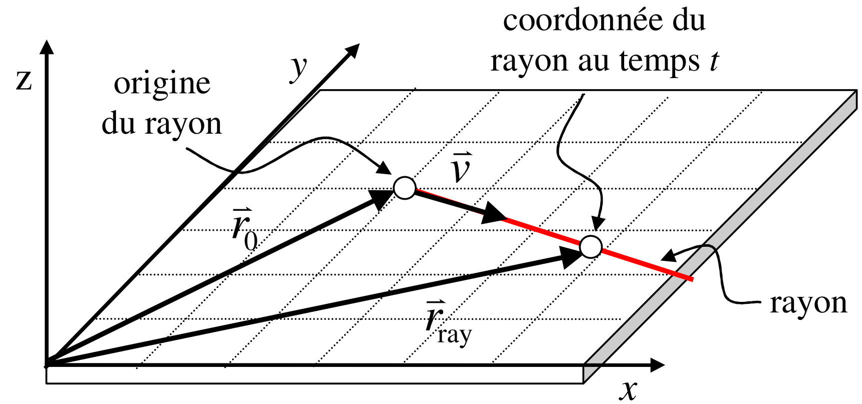
2) Le rayon et l'intersection

- Équation du rayon

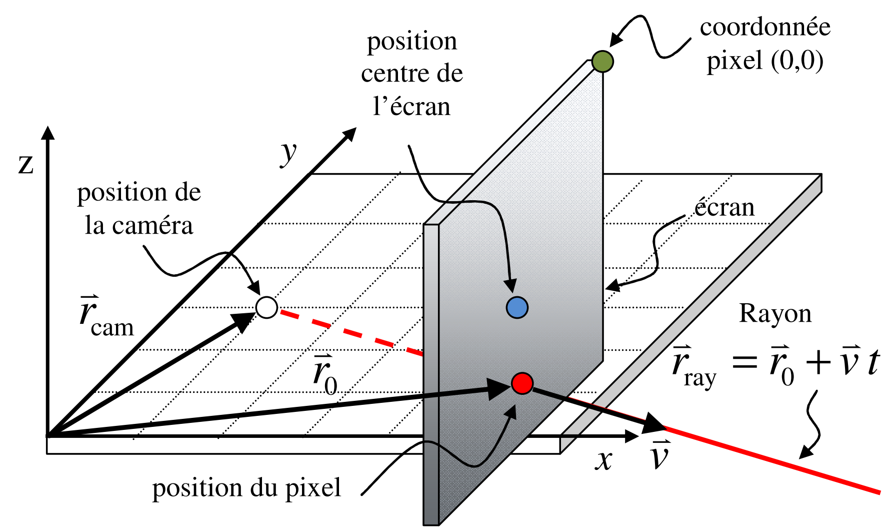
$$\vec{r}_{\text{ray}} = \vec{r}_0 + \vec{v}t$$

- Rayon paramétrisé selon t

$$0 \leq t \leq \infty$$



- Les rayons sont lancés depuis la position d'une caméra dans tous les case (pixel) d'un écran de face



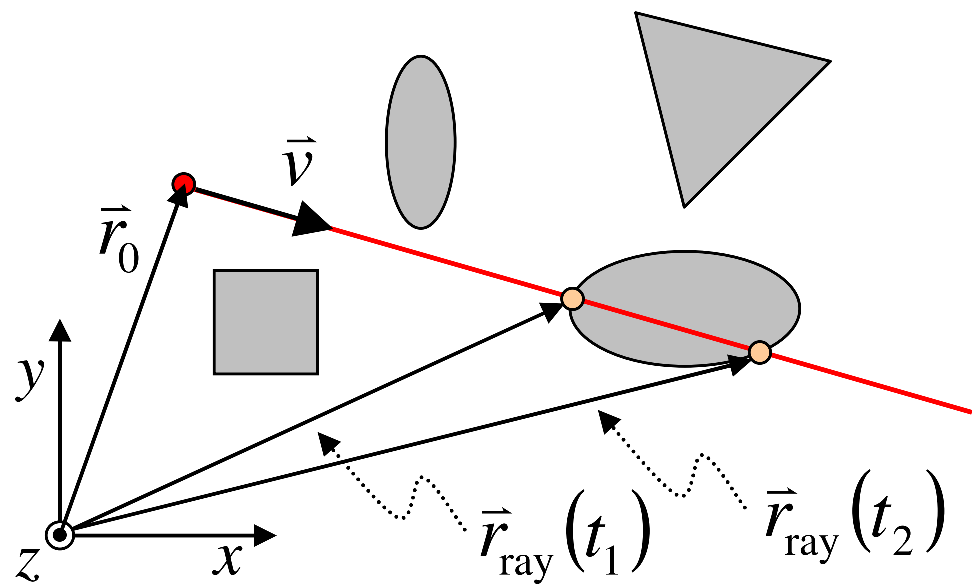
Les calculs matriciels dans le *ray tracing*

2) Le rayon et l'intersection

- Le ray tracing est un algorithme de visibilité.

- Le calcul de l'intersection doit évaluer quelle géométrie est intersecté par le rayon en premier (plus petite valeur de *t*).

- La couleur associée au rayon sera déterminée par un calcul d'illumination.

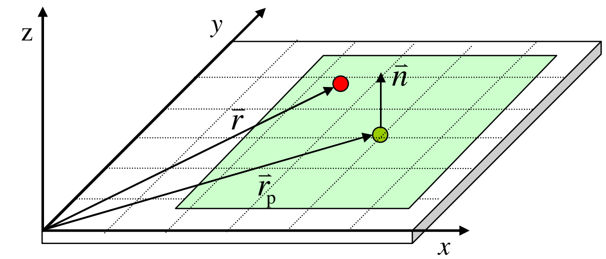


Les calculs matriciels dans le *ray tracing*

2) Le rayon et l'intersection

- Calcul de l'intersection d'un plan :

$$t_{\text{int}} = \frac{\vec{n} \cdot (\vec{r}_{\mathrm{P}} - \vec{r}_0)}{\vec{n} \cdot \vec{v}}$$



- Calcul de l'intersection d'une sphère :

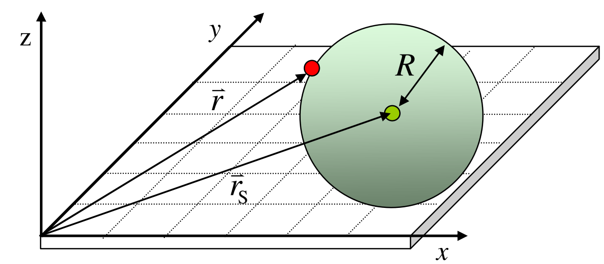
$$At^2 + Bt + C = 0$$

où $$A = \vec{v} \cdot \vec{v}$$

$$B = 2\,\vec{r}_{\mathrm{S0}} \cdot \vec{v}$$

$$C = \vec{r}_{\mathrm{S0}} \cdot \vec{r}_{\mathrm{S0}} - R^2$$

et $$\vec{r}_{\mathrm{S0}} = \vec{r}_0 - \vec{r}_{\mathrm{S}}$$



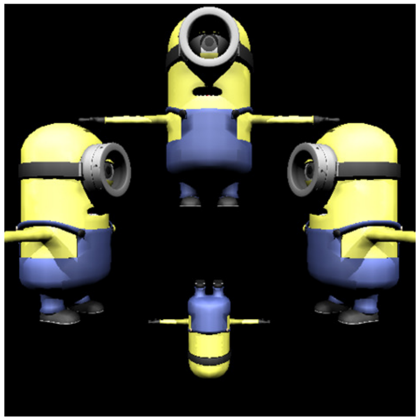
Les calculs matriciels dans le *ray tracing*

3) Pouquoi utiliser des matrices ?

- On utilise les matrices dans un algorithme de *ray tracing* pour

1) Modifier un modèle 3D construit dans un espace unitaire.

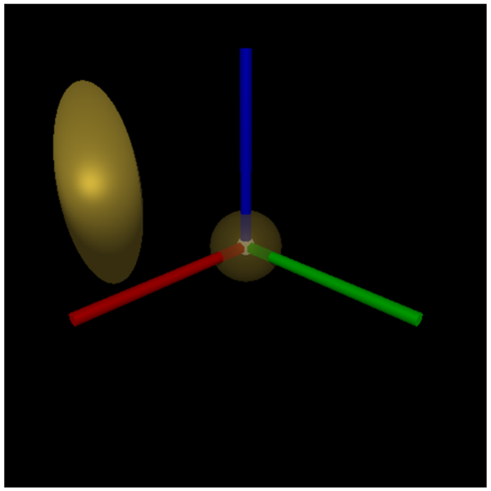
http://www.deskeng.com/de/seeing-ray-tracing-in-a-different-light/



(SIMRenderer)

2) Pour reproduire un modèle 3D afin de le visualiser à différents endroits dans d'autres positions.

3) Pour réaliser un **calcul de l'intersection avec une forme géométrique plus complexe**, mais identique à une forme plus simple sous une **transformation linéaire inversée**.

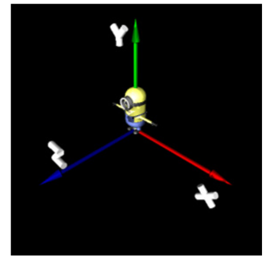


Sphère unitaire à l'origine et sphère transformée (SIMRenderer)

Les calculs matriciels dans le *ray tracing*

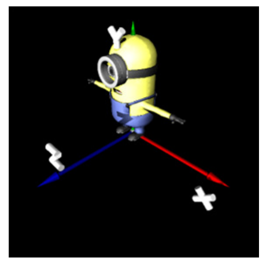
3) Pouquoi utiliser des matrices ?

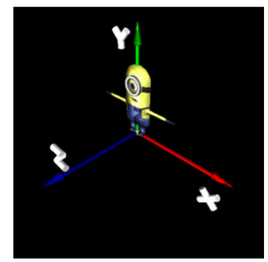
- En appliquant des matrices de transformation à des géométries ou des modèles 3D, nous pouvons



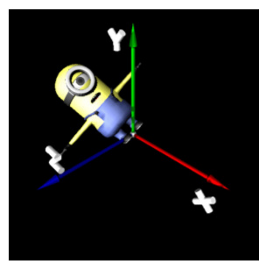
Modèle 3d (obj)
(SIMRenderer)

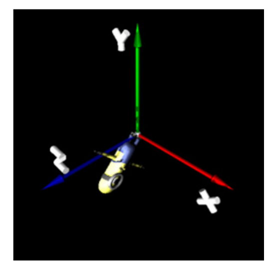
1) les déformer (homothétie)



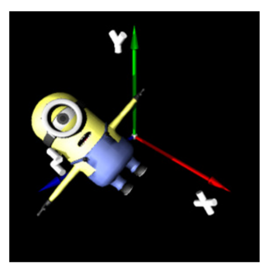


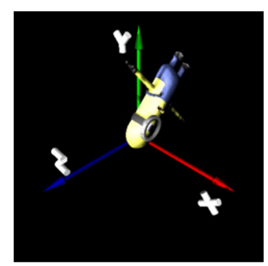
2) les tourner (rotation *xyz*)





3) les déplacer (translation)





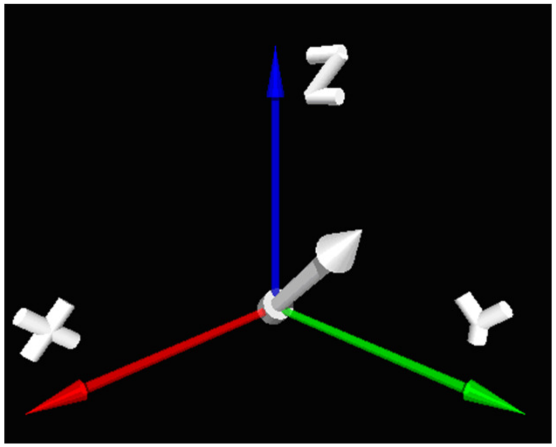
Les calculs matriciels dans le *ray tracing*

4) Les matrices de transformation en format 4x4

- Pour uniformiser les opérations mathématiques de transformation (utiliser uniquement la multiplication matricielle et éviter l'addition pour l'application d'une translation), on utiliser des vecteurs à 4D.

$$\vec{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \qquad \vec{r} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

(3D) vers (4d)



(SIMRenderer)

- On effectue la transformation des vecteurs 4D à l'aide de la multiplication :

$$\vec{r}_{\mathrm{m}} = M_{\mathrm{o}\to\mathrm{m}}\ \vec{r}_{\mathrm{o}}$$

Transformation Objet vers Monde

$$\vec{r}_{\mathrm{o}} = M_{\mathrm{m}\to\mathrm{o}}\ \vec{r}_{\mathrm{m}}$$

Transformation Monde vers Objet

Monde : L'espace de la scène 3D

Objet : L'espace unitaire de la géométrie

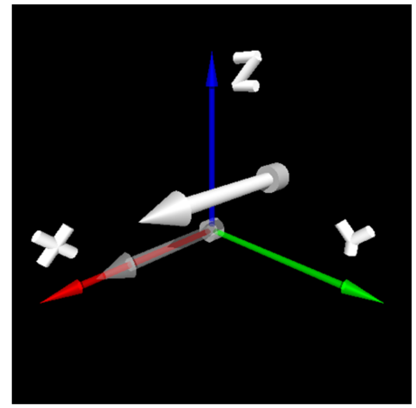
Le raytracer … à la rencontre
de l'informatique, des mathématiques
et de la physique

Les calculs matriciels dans le *ray tracing*

4) Les matrices de transformation en format 4x4

- La matrice de translation

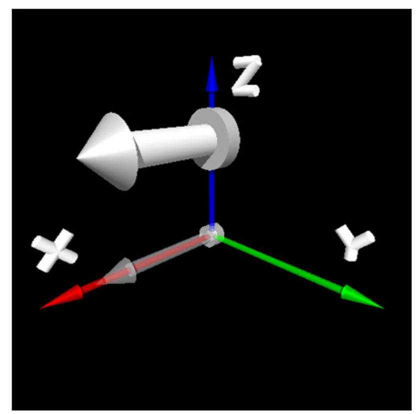
$$Tr = Tr(tr_x, tr_y, tr_z) = \begin{pmatrix} 1 & 0 & 0 & tr_x \\ 0 & 1 & 0 & tr_y \\ 0 & 0 & 1 & tr_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



(SIMRenderer)

- La matrice d'homothétie

$$Sc = Sc(sc_x, sc_y, sc_z) = \begin{pmatrix} sc_x & 0 & 0 & 0 \\ 0 & sc_y & 0 & 0 \\ 0 & 0 & sc_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



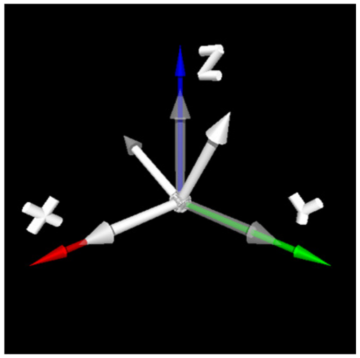
(SIMRenderer)

Les calculs matriciels dans le *ray tracing*

4) Les matrices de transformation en format 4x4

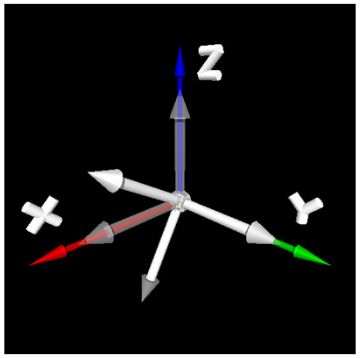
- La matrice de rotation selon l'axe *x*

$$R_x = R_x(\theta_x) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) & 0 \\ 0 & \sin(\theta_x) & \cos(\theta_x) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



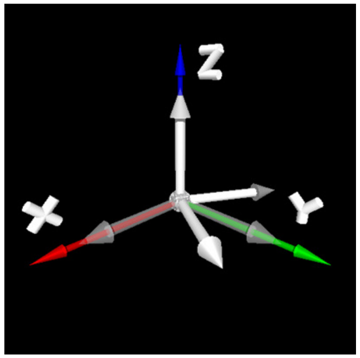
- La matrice de rotation selon l'axe *y*

$$R_y = R_y(\theta_y) = \begin{pmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



- La matrice de rotation selon l'axe *z*

$$R_z = R_z(\theta_z) = \begin{pmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



Le raytracer ... à la rencontre
de l'informatique, des mathématiques
et de la physique

Les calculs matriciels dans le *ray tracing*

4) Les matrices de transformation en format 4x4

- Les matrice inverse (forme semblable)

$$Tr = Tr(tr_x, tr_y, tr_z) = \begin{pmatrix} 1 & 0 & 0 & tr_x \\ 0 & 1 & 0 & tr_y \\ 0 & 0 & 1 & tr_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad Tr^{-1} = Tr(-tr_x, -tr_y, -tr_z) = \begin{pmatrix} 1 & 0 & 0 & -tr_x \\ 0 & 1 & 0 & -tr_y \\ 0 & 0 & 1 & -tr_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Sc = Sc(sc_x, sc_y, sc_z) = \begin{pmatrix} sc_x & 0 & 0 & 0 \\ 0 & sc_y & 0 & 0 \\ 0 & 0 & sc_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad Sc^{-1} = Sc\left(\frac{1}{sc_x}, \frac{1}{sc_y}, \frac{1}{sc_z}\right) = \begin{pmatrix} 1/sc_x & 0 & 0 & 0 \\ 0 & 1/sc_y & 0 & 0 \\ 0 & 0 & 1/sc_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_x = R_x(\theta_x) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) & 0 \\ 0 & \sin(\theta_x) & \cos(\theta_x) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad R_x^{-1} = R_x(-\theta_x) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-\theta_x) & -\sin(-\theta_x) & 0 \\ 0 & \sin(-\theta_x) & \cos(-\theta_x) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_y = R_y(\theta_y) = \begin{pmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad R_y^{-1} = R_y(-\theta_y) = \begin{pmatrix} \cos(-\theta_y) & 0 & \sin(-\theta_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-\theta_y) & 0 & \cos(-\theta_y) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_z = R_z(\theta_z) = \begin{pmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad R_z^{-1} = R_z(-\theta_z) = \begin{pmatrix} \cos(-\theta_z) & -\sin(-\theta_z) & 0 & 0 \\ \sin(-\theta_z) & \cos(-\theta_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

4) Les matrices de transformation en format 4x4

- La transformation Objet-Monde

$$\vec{r}_{\mathrm{m}} = Tr \cdot R_z \cdot R_y \cdot R_x \cdot Sc \; \vec{r}_{\mathrm{o}}$$

Pour transformer une géométrie unitaire
vers l'espace de la scène

$$\boxed{M_{\mathrm{o}\to\mathrm{m}} = Tr \cdot R_z \cdot R_y \cdot R_x \cdot Sc}$$

- La transformation Monde-Objet

$$\vec{r}_{\mathrm{o}} = Sc^{-1} \cdot R_x^{\;-1} \cdot R_y^{\;-1} \cdot R_z^{\;-1} \cdot Tr^{-1} \; \vec{r}_{\mathrm{m}}$$

Pour transformer un rayon de l'espace monde
vers l'espace unitaire d'une géométrie

$$\boxed{M_{\mathrm{m}\to\mathrm{o}} = Sc^{-1} \cdot R_x^{\;-1} \cdot R_y^{\;-1} \cdot R_z^{\;-1} \cdot Tr^{-1}}$$

5) L'algorithme de l'intersection d'une géométrie transformée

- Voici les informations disponibles et non disponibles

Espace monde	Espace objet
$\vec{o}$: Origine de l'espace monde ($\vec{o}=0$). $\vec{r}_0$: Position initiale du rayon dans l'espace monde. $\vec{v}$: Orientation du rayon dans l'espace monde. $M_{\mathrm{m}\to\mathrm{o}}$: Transformation de l'espace monde à l'espace objet.	$\vec{o}$: Origine de l'espace objet ($\vec{o}=0$). $M_{\mathrm{o}\to\mathrm{m}}$: Transformation de l'espace objet à l'espace monde.
➢ Rayon : $\vec{r}_{\mathrm{ray}} = \vec{r}_0 + \vec{v}\,t$ ➢ Calcul de l'intersection non implémenté	➢ Rayon : $\vec{r}_{\mathrm{ray(o)}}$ à obtenir par transformation ➢ Calcul de l'intersection : implémenté

- Voici les informations que l'on désire calculer

t_{int} : Temps afin que le rayon réalise l'intersection avec la géométrie (invariant).

$\vec{n}_{\mathrm{int}}$: Normale à la surface de la géométrie à l'endroit de l'intersection dans l'espace monde.

5) L'algorithme de l'intersection d'une géométrie transformée

Étape #1 : Transformation du rayon $\vec{r}_{\text{ray}} = \vec{r}_0 + \vec{v}\,t$ de l'espace monde vers l'espace objet

a) Transformer de l'origine du rayon $\vec{r}_0$ de l'espace monde vers l'espace objet :

$$\vec{r}_{0\,\text{o}} = M_{\text{m}\to\text{o}}\vec{r}_0$$

b) Transformer de l'orientation du rayon $\vec{v}$ de l'espace monde vers l'espace objet :

$$\vec{o}_{\text{m}\to\text{o}} = M_{\text{m}\to\text{o}}\vec{o} \qquad \text{(origine)}$$

$$\vec{v}_{\text{m}\to\text{o}} = M_{\text{m}\to\text{o}}\vec{v} \qquad \text{(direction à transformer)}$$

$$\vec{v}_{\text{o}} = \vec{v}_{\text{m}\to\text{o}} - o_{\text{m}\to\text{o}} \qquad \text{(nouvelle direction)}$$

P.S. Il est important **de ne pas normaliser** $\vec{v}_{\text{o}}$, car c'est ce qui permet d'avoir t comme invariant.

Le raytracer … à la rencontre
de l'informatique, des mathématiques
et de la physique

Les calculs matriciels dans le *ray tracing*

5) L'algorithme de l'intersection d'une géométrie transformée

Étape #2 : Réalisation du test de l'intersection

a) Construire le rayon transformé $\vec{r}_{\mathrm{ray(o)}} = \vec{r}_{0\,\mathrm{o}} + \vec{v}_{\mathrm{o}} t$.

b) Tester l'intersection entre le rayon transformé $\vec{r}_{\mathrm{ray(o)}}$ et la géométrie dans l'espace objet.

c) Obtenir le temps d'intersection approprié $t = t_{\mathrm{int}}$.

d) S'il y a eu intersection, évaluer la normale à la surface $\vec{n}$ à l'endroit de l'intersection. Sinon, il n'y a pas eu d'intersection.

Les calculs matriciels dans le *ray tracing*

5) L'algorithme de l'intersection d'une géométrie transformée

Étape #3 : Transformation de la normale à la surface

a) Transformer la normale à la surface $\vec{n}$ de l'espace objet vers l'espace monde :

$$\vec{o}_{\mathrm{o}\to\mathrm{m}} = M_{\mathrm{o}\to\mathrm{m}}\vec{o} \qquad \text{(origine)}$$

$$\vec{n}_{\mathrm{o}\to\mathrm{m}} = M_{\mathrm{o}\to\mathrm{m}}\vec{n} \qquad \text{(direction à transformer)}$$

$$\vec{n}_{\mathrm{m}} = \vec{n}_{\mathrm{o}\to\mathrm{m}} - \vec{o}_{\mathrm{o}\to\mathrm{m}} \qquad \text{(nouvelle direction)}$$

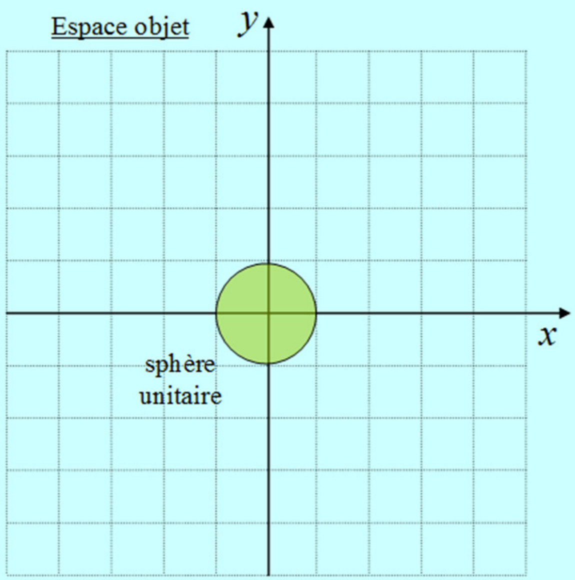
b) Normaliser la normale à la surface :

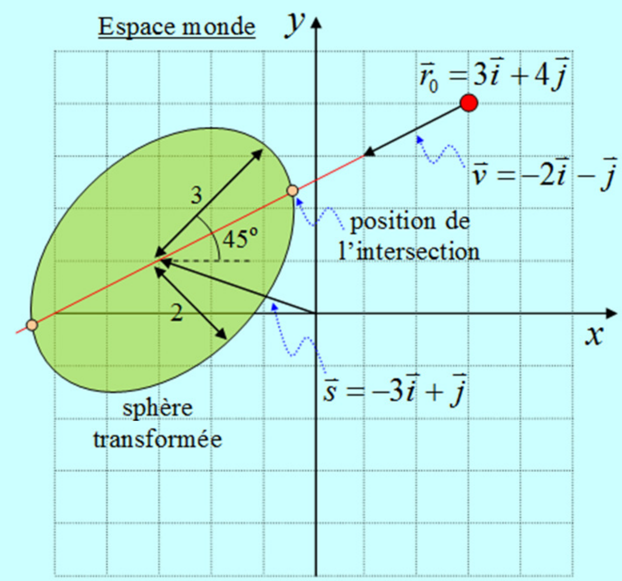
$$\vec{n}_{\mathrm{int}} = \frac{\vec{n}_{\mathrm{m}}}{\left|\vec{n}_{\mathrm{m}}\right|}$$

Les calculs matriciels dans le *ray tracing*

6) Un exemple …

Situation A : ***L'intersection d'une sphère transformée.*** Une sphère unitaire possède un rayon égal à 1 et est centré à l'origine de son espace objet. On désire faire l'intersection de cette sphère dans un espace monde après avoir effectué une transformation d'homothétie $Sc = (3,\ 2,\ 1)$, une rotation autour de l'axe z tel que $\theta_z = 45°$ et une translation $Tr = (-3,\ 1,\ 0)$. Le rayon qui réalisera l'intersection aura une origine $\vec{r}_0 = 3\vec{i} + 4\vec{j}$ et une orientation $\vec{v} = -2\vec{i} - \vec{j}$.

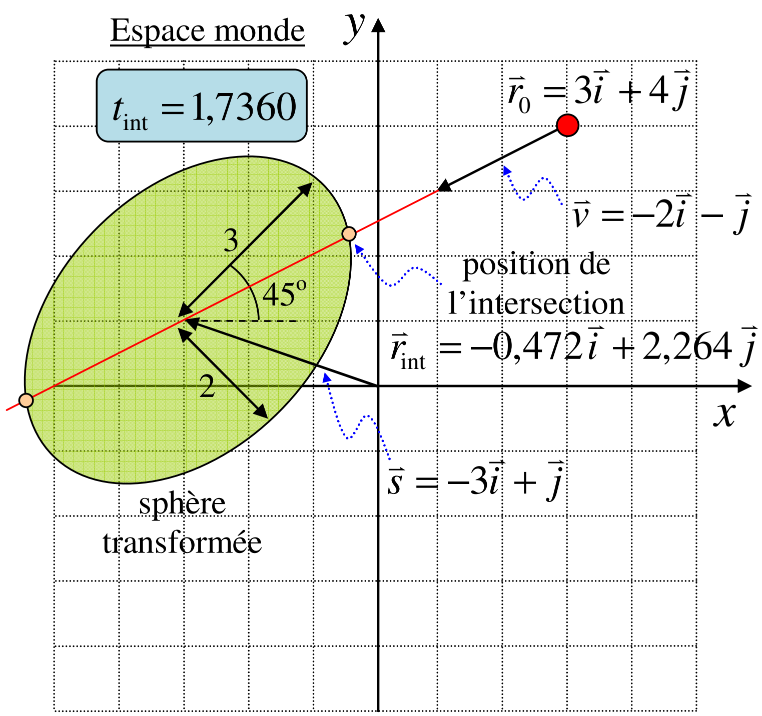


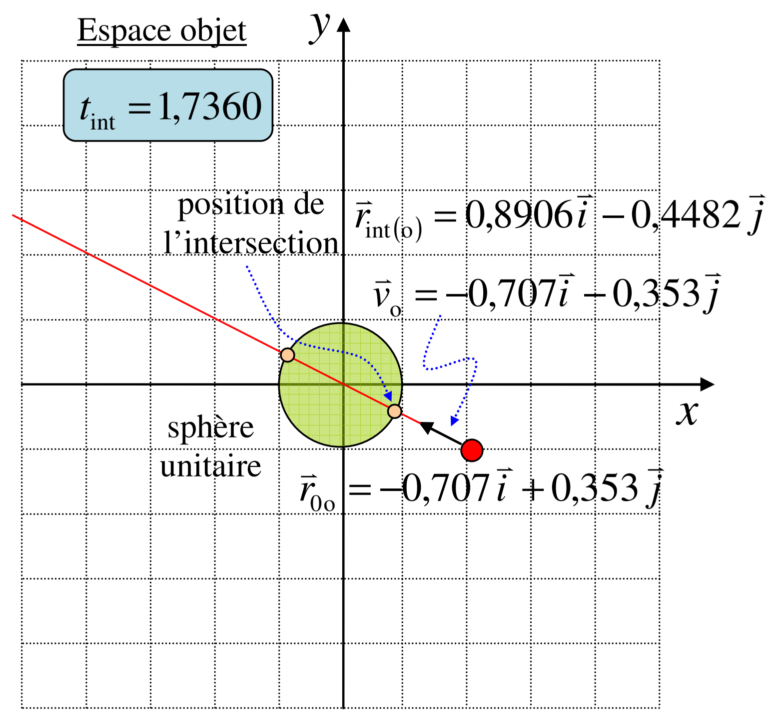


On désire **(a)** évaluer la matrice de transformation de l'espace objet-monde et la matrice de transformation monde-objet, **(b)** évaluer le temps t_{int} requis pour réaliser l'intersection et **(c)** vérifier que le temps t_{int} est un invariant dans les deux espaces.

6) Un exemple …

- Le résultat … rapidement





- Le résultat … en détail (voir NYC XXI – Chapitre 2.X2b)

  http://profs.cmaisonneuve.qc.ca/svezina/nyc/note_nyc/NYC_XXI_Chap%202.X2b.pdf

Les calculs matriciels dans le *ray tracing*

6) Des questions … et merci !