

VPython : exemples d'application 3D

Laurent Pierron

INRIA-Lorraine
LORIA

Rencontres Mondiales du Logiciel Libre 2005

Plan

- 1 Applications en Physique
 - Mécanique pendulaire
 - Thermodynamique
 - Infiniment petit
- 2 Applications en Géométrie
 - Etude des Projections
 - Sphères dans l'espace
 - Animation spatiale
- 3 Applications diverses
 - Jeux
 - Intelligence Artificielle
 - Architecture

Plan

- 1 Applications en Physique
 - Mécanique pendulaire
 - Thermodynamique
 - Infiniment petit
- 2 Applications en Géométrie
 - Etude des Projections
 - Sphères dans l'espace
 - Animation spatiale
- 3 Applications diverses
 - Jeux
 - Intelligence Artificielle
 - Architecture

Mouvement d'un pendule double

Programme : [doublependulum.py](#)

Description du modèle

Mouvement de deux pendules
calculé par des Lagrangiens.

Algorithme

Transcription des formules

- $B = (1./2.) * M2 * L1 * L2$
- $atheta2 = -(A * atheta1 + C * \sin(theta1))/B$

Objets utilisés

- 5 box() : structure
- 2 cylinder() : axes
- 2 frame() : pendules

Astuces

- utilisation de rotate()
- position relative du second pendule
- positions calculées par **VPython**

Plan

- 1 Applications en Physique
 - Mécanique pendulaire
 - **Thermodynamique**
 - Infiniment petit
- 2 Applications en Géométrie
 - Etude des Projections
 - Sphères dans l'espace
 - Animation spatiale
- 3 Applications diverses
 - Jeux
 - Intelligence Artificielle
 - Architecture

Gaz idéal avec collisions

Programme : [gas.py](#)

Description du modèle

- mouvement brownien dans un gaz
- comparaison théorie/modèle

Algorithme

- calcul de vitesse
- calcul de position
- collisions

Objets utilisés

- `sphere()` : atomes
- `curve()` : la boîte
- `gdisplay`, `gcurve`, `ghistogram` : graphique

Astuces

- constantes physiques en valeurs réelles
- accélération via **Numeric**

Plan

- 1 Applications en Physique
 - Mécanique pendulaire
 - Thermodynamique
 - **Infiniment petit**
- 2 Applications en Géométrie
 - Etude des Projections
 - Sphères dans l'espace
 - Animation spatiale
- 3 Applications diverses
 - Jeux
 - Intelligence Artificielle
 - Architecture

Pulsation d'un cristal

Programme : [cristal.py](#)

Description du modèle

Pulsation d'un cristal composé de $N \times N$ atomes.

Objets utilisés

- `sphere()` : atomes
- `helix()` : liaisons élastiques

Algorithme

- 1 Position initiale aléatoire des atomes
- 2 Variation position des atomes
- 3 Position liaisons, fonction des atomes

Astuces

- utilisation de `helix()`
- accélération `Numeric`

Plan

- 1 Applications en Physique
 - Mécanique pendulaire
 - Thermodynamique
 - Infiniment petit
- 2 Applications en Géométrie
 - **Etude des Projections**
 - Sphères dans l'espace
 - Animation spatiale
- 3 Applications diverses
 - Jeux
 - Intelligence Artificielle
 - Architecture

Calculs de projections

Programme : [testcrosspoint.py](#)

Description du modèle

Calcul de projection de points et d'intersection dans un plan.

Algorithme

- Description des droites et des points.
- Etiquetage des intersections.

Objets utilisés

Module `pygeo` : FreePoint, Line, Plane, Slider, Intersect, CrossPoint

Astuces

Tout est calculé par `pygeo`.

Plan

- 1 Applications en Physique
 - Mécanique pendulaire
 - Thermodynamique
 - Infiniment petit
- 2 Applications en Géométrie
 - Etude des Projections
 - **Sphères dans l'espace**
 - Animation spatiale
- 3 Applications diverses
 - Jeux
 - Intelligence Artificielle
 - Architecture

Mouvement d'un pendule double

Programme : [spherestest.py](#)

Description du modèle

Calculs et visualisation
d'intersection de sphères.

Algorithme

- Placer les points
- Créer les sphères à partir des points
- Créer les intersections

Objets utilisés

Toujours **pygeo** : FreePoint,
Sphere, Plane,
SpheresIntersect,
SphereCircle, OrthoSphere,
BiChord

Astuces

- Déplacer points à la souris
- Réinitialiser avec la fenêtre de contrôle

Plan

- 1 Applications en Physique
 - Mécanique pendulaire
 - Thermodynamique
 - Infiniment petit
- 2 **Applications en Géométrie**
 - Etude des Projections
 - Sphères dans l'espace
 - **Animation spatiale**
- 3 Applications diverses
 - Jeux
 - Intelligence Artificielle
 - Architecture

Animation de cercles concentriques

Programme : pencilanimation.py

Description du modèle

Construction de cercles concentriques animés

Algorithme

Description des points, des cercles à partir des points

Objets utilisés

Toujours **pygeo** : Z_FreePoint, Z_Line, Z_Circle, Z_Pencil, U_Pencil

Astuces

Utilisation des **Z_objets** pour l'animation sur l'axe Z

Plan

- 1 Applications en Physique
 - Mécanique pendulaire
 - Thermodynamique
 - Infiniment petit
- 2 Applications en Géométrie
 - Etude des Projections
 - Sphères dans l'espace
 - Animation spatiale
- 3 **Applications diverses**
 - **Jeux**
 - Intelligence Artificielle
 - Architecture

3D TicTacToe

Programme : [tictac.py](#)

Description du modèle

Arriver à aligner 4 pions dans un espace en trois dimensions.

Algorithme

- Créer la grille de jeu
- Créer les objets cachés
- Gérer les évènements
- Tester la fin de partie

Objets utilisés

- `curve()` pour dessiner la grille
- `cylinder()` pour les axes
- `ball()` pour les pions

Astuces

- `project` pour tester clic plan
- `drag` pour déplacer l'objet
- Plusieurs boucles

Plan

- 1 Applications en Physique
 - Mécanique pendulaire
 - Thermodynamique
 - Infiniment petit
- 2 Applications en Géométrie
 - Etude des Projections
 - Sphères dans l'espace
 - Animation spatiale
- 3 **Applications diverses**
 - Jeux
 - **Intelligence Artificielle**
 - Architecture

Déplacement aléatoire de fourmis

Programme : [fourmis.py](#)

Description du modèle

- Déplacement aléatoire de fourmis
- Marquage du chemin

Algorithme

- Création décor
- Création classe fourmi
- Création des fourmis
- Animation des déplacements

Objets utilisés

- `cone()`, `box()` : décor
- `sphere()` : corps et yeux
- `cylinder()` : pattes
- `frame()` : fourmi complète
- `curve()` : trace du chemin

Astuces

- `frame` pour fourmi
- Sélection fourmi par clic

Plan

- 1 Applications en Physique
 - Mécanique pendulaire
 - Thermodynamique
 - Infiniment petit
- 2 Applications en Géométrie
 - Etude des Projections
 - Sphères dans l'espace
 - Animation spatiale
- 3 Applications diverses
 - Jeux
 - Intelligence Artificielle
 - Architecture

L'étrangeté de Stonehenge

Programme : [stonehenge.py](#)

Description du modèle

Représenter scène architecturale avec objets animés et changement de point de vue

Objets utilisés

- sphere, cone, box, cylinder
- frame, text
- scene, scene.mouse

Algorithme

- Décrire scène avec objets de base
- Boucle d'animation pour objets animés
- Gérer souris pour modifier point de vue

Astuces

- Utilisation de `time()` pour créer l'horloge
- Modification des attributs de scene

Conclusion

VPython un outil pédagogique pour

- se concentrer sur les algorithmes plutôt que la visualisation
- diversifier les applications
- réaliser des simulations simples sans gros investissement
- s'initier facilement à la 3D

Remerciements

- **David Scherrer**, D. Andersen, J. Brandmeyer, R. Chabay, A. Heitner, I. Peters, B. Sherwood
- **Arthur Siegel** auteur de PyGeo
- R. Chabay (TicTacToe), F. de Fougérolles (Fourmis)