

Tarea 3-C: Hidrocarburos

Prof: Nancy Hitschfeld K.

Auxiliares: Pablo Pizarro R., Pablo Polanco Galleguillos, Mauricio Araneda H.

Ayudantes: Iván Torres, María José Trujillo Berger

Fecha de entrega: Algún día

Problema

Los hidrocarburos son compuestos orgánicos formados únicamente por átomos de carbono e hidrógeno. La estructura molecular consiste en un armazón de átomos de carbono a los que se unen los átomos de hidrógeno. Un átomo de carbono puede mantener cuatro enlaces, mientras que el hidrógeno solo puede mantener uno.

Se desea simular el comportamiento que tendrían estos átomos al ser depositados en una caja en distintas proporciones, y estudiar cuáles serían las estructuras de hidrocarburos generadas.

En esta ocasión se modelará el problema sin considerar efectos de gravedad, pero considerando las interacciones gravitacionales entre los átomos y moléculas. Esto es, en una caja, se inicialmente se distribuirá uniformemente una cantidad N_C de átomos de carbono, y una cantidad N_H de átomos de hidrógeno. Los átomos pueden ser modelados como esferas que rebotarán inelásticamente en los bordes de la caja.

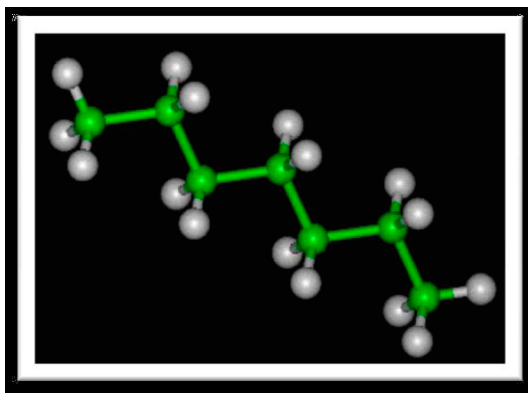


Figura 1: Hidrocarburos.

En esta tarea, debe utilizar OpenGL para dibujar. Como lenguaje base se recomienda utilizar Python.

Modelos (1 punto)

Utilizando OpenGL, dibuje un modelo esférico para el hidrógeno y otro para el oxígeno. Utilice tamaños proporcionales a los radios atómicos (carbono tiene 70 pm de radio medio, mientras que el hidrógeno tiene solo 25 pm). Por otro lado, también debe dibujar una representación para un enlace simple.

Las partículas se moverán al interior de una caja. Esta caja debe ser modelada de forma que en su interior se logre un buen efecto de iluminación (i.e. utilizar una malla de polígonos relativamente fina).

Configure una fuente de luz al interior de la caja. Todos sus modelos deben utilizar sombreado, ya sea FLAT o SMOOTH.

Condición inicial (1 punto)

Al comienzo del programa, se le debe solicitar al usuario que ingrese la cantidad de átomos de hidrógeno y de carbono que se utilizarán en la simulación. Acto seguido, se da inicio a la simulación, donde los átomos son distribuidos uniformemente al interior de la caja. La velocidad inicial para cada partícula es nula. La cámara debe estar inicialmente en una esquina y mirando hacia el interior de la caja.

Cámara móvil (1 punto)

La cámara debe poder moverse con las flechas del teclado, rotando hacia los lados y hacia arriba y abajo. Con las teclas “z” y “x” la cámara debe moverse continuamente hacia adelante y atrás.

Simulación (3 puntos)

Cada átomo debe interactuar con los demás según la ley de gravitación universal, esto es, una fuerza atractiva proporcional al producto de las masas involucradas, e inversamente proporcional al cuadrado de la distancia.

$$\vec{F} = -\frac{Gm_1m_2}{d_{1,2}^2}$$

Puede desprejir la acción de esta fuerza si los átomos están lo suficientemente distantes. La fuerza neta sobre un átomo estará dada entonces por la suma de las interacciones con cada otro átomo o molécula. A partir de esta fuerza neta, puede calcular la aceleración instantánea de dicha partícula. Para las masas utilice magnitudes proporcionales a las masas atómicas de los isótopos más típicos (1 uma para el hidrógeno, y 12 uma para el carbono).

Si dos átomos colisionan, de ser posible se formará un nuevo enlace (i.e. ambos con capacidad para un enlace adicional), sino, cada átomo simplemente seguirá su camino (se traslaparán sus

trayectorias). En el caso de una colisión carbono-carbono, sólo se considerarán enlaces simples.

Asuma que los enlaces se distribuyen uniformemente alrededor de cada átomo de carbono central. Esto es, cada átomo de carbono se localizará al centro de un tetraedro, y los otros átomos en los vértices.

No se complique con la transformación gradual de la molécula, luego de la colisión, modifique la molécula directamente a la forma final. Ignore además las reacciones intramoleculares, es decir, un átomo particular de la molécula no se siente afectado por otros átomos de la misma molécula. Tampoco se complique con que dos átomos de la misma molécula se sitúen en regiones superpuestas.

Bonus Track (1 punto sobre la nota)

Dese algunos supuestos y modele las reacciones intramoleculares. Dos átomos no pueden superponerse en la misma región. Explícite este ítem y sus supuestos en su presentación-informe.

Presentación-Informe

Se espera que en el informe de esta tarea se explique la metodología llevada a cabo para resolver el problema, la estructura de su código, los resultados y una breve discusión sobre los resultados obtenidos.

Haga especial énfasis en la estructura de datos utilizada para modelar las moléculas. Presente varios screenshots mostrando distintas simulaciones con su programa.

Observaciones

- **Plazo de Entrega: Algún día. No se aceptan atrasos.**
- No se reciben tareas por otro medio que no sea la sección habilitada en u-cursos.
- Entregue TODOS los archivos que sean necesarios para la correcta ejecución de su programa.
- Puede reenviar su tarea dentro del plazo. En este caso, procure subir nuevamente TODOS sus archivos. El equipo docente sólo tiene acceso a la última entrega que usted realiza.
- La presentación-informe DEBE estar en formato .pdf, no se revisarán otros formatos.
- Las tareas se revisarán utilizando Python 2.7 y sus librerías (instaladores en u-cursos).
- Para consultas, se ofrece soporte en el lenguaje Python. También puede utilizar C++ o Java (junto a OpenGL), pero va por su cuenta y debe presentar el código entregado funcionando en su computador.

Hints

- Planifique su tiempo y comience su tarea con anticipación. No comience directamente a programar. Identifique fragmentos de algoritmos, formule clases y objetos útiles, si lo necesita,

diseñe máquinas de estados.

- Construya su programa de manera incremental, esto es, vaya respaldado su código de forma que siempre disponga de algo que “funcione”.
- Se recomienda utilizar pequeños programas que simplemente dibujen cada una de las partes del juego (objetos, personajes, escenario, etc..). Del mismo modo, puede construir pequeños programas que vayan añadiendo las interacciones entre los distintos actores.
- Una buena metodología es diseñar una clase por cada tipo de actor que interfiera en su programa.