

Tarea 3-D: Earth Defender 3D

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

En un futuro muy, muy lejano . . .

Una lluvia de meteoritos acecha la tierra. Afortunadamente, un equipo de trabajo de la fcfm la ha detectado a tiempo y los ingenieros han diseñado naves capaces de destruirlos.

Las naves tienen la capacidad de moverse libremente en el sector de la Exosfera. Los meteoritos aceleran en su caída debido a la fuerza de gravedad.

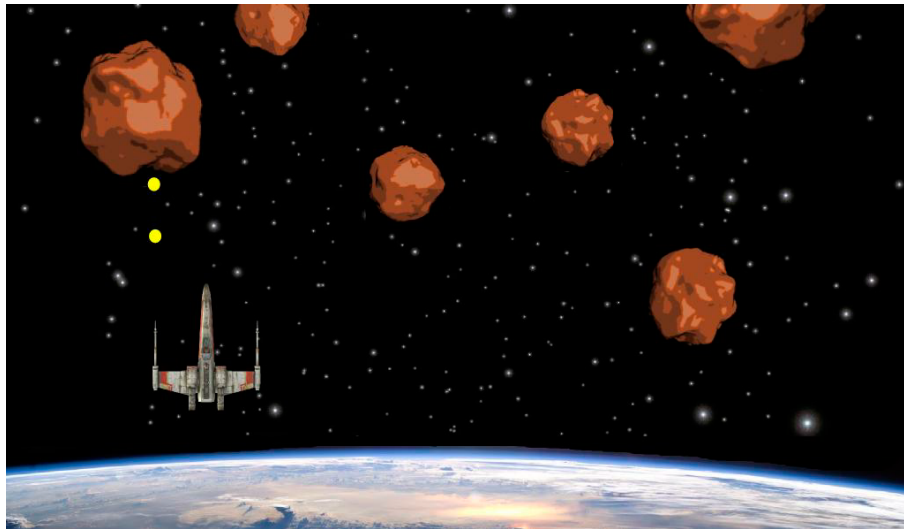


Figura 1: *Imágenes referenciales, pues sus modelos deben ser tridimensionales.

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

Modelos (3 puntos)

Dibuje 3 naves espaciales utilizando OpenGL y distintas estrategias:

- Una nave construida en base a figuras GLUT, utilizando al menos unas 4.
- Una nave generada a partir de un fichero STL, OBJ u otro formato (Puede encontrar figuras en internet).
- Una tercera nave dibujada por usted en base a especificación manual de vértices y polígonos.

Dibuje un meteorito, una bala y la superficie del planeta tierra como usted estime conveniente. Todos sus modelos deben tridimensionales y utilizar sombreado, ya sea FLAT o SMOOTH. Implemente un programa que permita visualizar todos sus modelos.

El juego (3 puntos)

Para implementar correctamente el juego debe considerar lo siguiente:

- Pantalla de selección **(0,5 puntos)**:
 - Al comienzo del juego se inicia una pantalla de selección donde es posible escoger una de las tres naves espaciales. Con las flechas del teclado \leftarrow , \rightarrow debe ser posible cambiar la nave seleccionada. Con “espacio” se escoge la nave y comienza la partida.
- Control e interacciones **(1,5 puntos)**:
 - De la parte superior de la pantalla comienzan a caer meteoritos (respetando la fuerza de gravedad) de tamaño arbitrario hacia la tierra (ubicada en la parte inferior de la pantalla). - No se preocupe de las colisiones entre meteoritos. - Con las flechas del teclado es posible mover la nave por toda la pantalla. Al presionar “espacio” la nave dispara hacia arriba. - Si una bala alcanza un meteorito este es destruido. - Las naves espaciales deben poseer al menos una fuente de luz direccional, la que puede ser encendida o apagada al presionar “z” (el efecto de esta luz debe ser claramente visible).
- Puntuación y aumento de dificultad **(1 punto)**:
 - Al destruir un meteorito el jugador gana 1 punto, el que debe ser impreso en la consola.
 - Si un meteorito impacta la tierra, se debe imprimir un aviso en la consola. La tierra soporta un máximo de 10 impactos.
 - Cada 10 meteoritos destruidos, aumenta la velocidad de generación de meteoritos. Imprima en consola un aviso.

Si bien los modelos deben ser en 3D. Todos los movimientos del juego se ubican en un mismo plano, por lo que la física puede ser modelada en 2D. Localice la cámara de forma tal que se visualice de forma similar a la imagen presentada. Configure una fuente de luz (el sol).

¡El destino de la tierra está en tus manos!

Bonus Track (máximo 1 punto sobre la nota)

Elija uno o dos de los siguientes (0,5 puntos cada uno):

- Cuando el meteorito sea destruido, genere animación que represente el proceso.
- Si un meteorito relativamente grande es destruido, se divide en dos meteoritos pequeños.
- Evite que los meteoritos se traslapen, implemente un choque elástico o inelástico entre ellos.
- La nave resiste un máximo de 3 impactos con meteoritos. A medida que vaya siendo dañada, debe mostrar daños, cambiando de color y ralentizando su movimiento.
- Implemente un ataque alternativo y un ataque especial (munición limitada).

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.

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 respaldando 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.