

Proyecto 2 (15%)

Fecha tope de entrega: 8 de noviembre (Jueves S.8) 11:59 PM

Se deducirá 20% de la nota por cada día de retraso.

La entrega es vía correo electrónico a la dirección **esaulgd+usb@gmail.com**.

El título del correo debe contener el string **ci4321sd12p02**. De no ser así se considerará no entregado.

El proyecto se realiza por parejas. El texto del mensaje debe contener los nombres y números de carnet de los autores.

Organice los archivos del proyecto mediante un *makefile*. Incluya todos los recursos necesarios en un archivo comprimido. El texto del correo debe contener instrucciones para su ejecución.

El proyecto debe correr en las máquinas del laboratorio donde se realiza la práctica.

Se evaluará la organización, limpieza y legibilidad del código, y que esté documentado donde sea necesario.

El proyecto consiste en implementar el videojuego especificado en el **documento de diseño** que se coloca a continuación.

StarTroupial

StarTroupial es un juego de disparos con naves en 3D. Podría referirse al mismo como parte del género *on-rails shooter*. El jugador controla una nave que constantemente avanza hacia el fondo de la pantalla. El jugador no puede cambiar la dirección de la nave, pero sí puede controlar su posición en un plano paralelo a la pantalla. La cámara observa a la nave desde atrás y conserva una distancia constante del plano de movimiento.

Un ejemplo de un videojuego en este género puede verse en <http://www.youtube.com/watch?v=A7TF5evojYA>

StarTroupial forma parte del regreso al paradigma *arcade* que se está llevando a cabo por, entre otras razones, la prominencia de los juegos en dispositivos móviles. En vez de poseer una historia que se desarrolla a lo largo de una serie de niveles, *StarTroupial* proporciona un único escenario, con elementos aleatorios, a objeto de que el jugador lo repita múltiples veces para mejorar su puntuación.

Reglas básicas:

A medida que la nave avanza, se encontrará con dos tipos de objetos con los que puede interactuar: blancos y anillos. La posición inicial de los objetos es estática, por lo que la nave se irá acercando a ellos, y de no interactuar con ellos los dejará atrás.

Los **blancos** son objetivos a los que el jugador debe disparar. El control de los disparos se realiza mediante el mouse. El jugador obtiene puntos al acertar a los blancos pero pierde puntos si choca contra los mismos. Los **anillos** otorgan puntos por cada uno que la nave logra atravesar.

La colocación de los objetos en el mundo es parcialmente aleatoria, pero deben existir patrones que permitan apreciar los efectos gráficos del juego. También puede incluir patrones que considere agradables y/o interesantes, por ejemplo, series de anillos sucesivos, o blancos colocados en dúos y tríos.

Una ronda del juego tiene una duración aproximada de **1 minuto**. El jugador inicia con cero puntos y no es posible una puntuación negativa. En todo momento debe mostrarse en la pantalla la puntuación actual y el tiempo restante. Al acabar el juego se muestra la puntuación final y se invita al jugador a intentarlo de nuevo.

Nave:

La nave es el objeto principal del juego y el único controlado directamente por el jugador. El movimiento de la nave se puede realizar mediante las teclas direccionales, teclas WASD o ambas opciones.

El modelo de la nave debe tener una la complejidad poligonal igual o mayor a los ejemplos colocados en <http://ldc.usb.ve/~esaulgd/pregrado/ci4321/sd12/p02/>. La nave debe tener una apariencia metálica, por lo que se recomienda que tenga reflejos especulares.

Anillos:

Los anillos deben tener una forma similar a la de un toro y también tener una apariencia metálica. Los segmentos del mismo pueden estar conectados o separados, a gusto del programador. Al pasar a través de un anillo, esto debe indicarse visualmente (p.ej. con un cambio de color) y el jugador obtiene **1 punto**.

Láser y blancos:

Al hacer click con el mouse, la nave dispara un rayo láser que se mueve a una velocidad alta y constante. De haber hecho click sobre un blanco, el láser se mueve en línea recta hacia el mismo. En caso contrario, se mueve directamente hacia el frente. Se puede limitar la tasa de fuego del láser para no sobrecargar el sistema. El láser, naturalmente, emite luz, la cual se puede representar como una fuente puntual.

El modelo de los blancos se deja a elección del programador, pero no pueden consistir en una única primitiva. Cuando un rayo láser alcanza un blanco, este último debe explotar, con lo cual el jugador **obtiene un punto**. La explosión no necesita ser realista, pero debe iluminar brevemente los objetos a su alrededor.

Si la nave choca contra un blanco, el jugador **pierde 3 puntos**. La detección de colisiones no necesita ser perfecta, sólo lo suficientemente buena para preservar la jugabilidad. Las imperfecciones y aproximaciones en la colisión deben errar a favor del jugador. Cuando haya una colisión, esto debe ser comunicado **claramente** al jugador con algún efecto visual.

Los disparos del láser no hacen mella en los anillos ni el láser ve su trayectoria afectada por los mismos.

Iluminación:

Aparte de las fuentes de luz puntuales arriba mencionadas, el mundo debe contar con iluminación global, la cual puede ser ambiental, direccional, o una combinación de ambas. La prioridad está en que el jugador pueda visualizar clara y fácilmente todos los objetos de la pantalla, sobre todo los más cercanos.

Pausa y funcionalidad *debug*:

El juego se pausa al presionar la tecla P. La acción en la pantalla y el temporizador se congelan y toda manipulación de los controles se descarta hasta que la misma tecla sea presionada de nuevo.

Para realizar pruebas del juego, se debe poder modificar la velocidad de la simulación, y junto con

ella el *frame rate*. Cada vez que se presione (-), la velocidad debe reducirse aproximadamente a la mitad. La tecla (+) debe tener el efecto contrario, hasta regresar a la velocidad original. El temporizador debe ralentizarse en la misma proporción o detenerse. El nuevo *frame rate* o la proporción en que ha sido modificado deben mostrarse en la pantalla.

Detalles técnicos:

Se desea que el juego corra al menos a 30 cuadros por segundo (*FPS*), idealmente a 60. Los objetos del juego pueden ser tanto importados como modelados en código, a conveniencia del programador. Una tutorial de OpenGL, incluyendo iluminación, se encuentra en <http://www.arcsynthesis.org/gltut/index.html>.

Punto extra:

Incluya música de fondo y efectos de sonido.
Si por cualquier razón es difícil la comprobación del sonido en las máquinas del laboratorio, se puede realizar la revisión de este punto en otro hardware.