

### Proyecto 3 (15%)

**Fecha tope de entrega:** 6 de diciembre (jueves S.12) 11:59 PM

No se aceptarán retrasos.

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

El título del correo debe contener el string **ci4321sd12p03**. 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.

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

### StarTroupial World

Luego del gran éxito de *StarTroupial*, se le ha encargado a su equipo el desarrollo de su secuela, la cual hará uso de tecnología gráfica más avanzada.

El concepto y reglas del juego son las mismas. El jugador controla una nave que avanza hacia el fondo de la pantalla y obtiene puntos por atravesar anillos y disparar blancos. A menos que se especifique lo contrario, se mantienen todos los elementos del proyecto anterior. Las características nuevas se indican a continuación.

#### Escenario:

En *StarTroupial World*, la nave del jugador vuela a poca altura sobre la superficie de un planeta, no en el espacio exterior. Dicha superficie se puede representar como un plano que se extiende hasta el horizonte. La nave no debe hacer contacto con la superficie.

Sobre el planeta se encuentran múltiples edificaciones de gran tamaño, las cuales representan obstáculos para la nave. La geometría de cada edificio puede consistir en un paralelepípedo, aunque puede usar formas más complejas si así lo desea. La frecuencia de aparición de los edificios se deja a juicio del programador, pero ésta debe ser alta y en ningún caso debe haber una distancia mayor a 3 segundos entre ellos. También deben estar distribuidos de manera que ocasionalmente sea necesario esquivar más de un edificio a la vez.

El ancho y alto de los obstáculos (desde el punto de vista del jugador) también varía. En ocasiones las edificaciones, individualmente o en conjunto, son fáciles de esquivar, ocupando sólo un 10% del área de movimiento de la nave. En otros deben ocupar hasta el 60% de la pantalla. En general los escenarios planteados deben ocupar el espectro entre estos dos extremos.

También debe haber variabilidad en la profundidad de los edificios. Al menos dos veces por partida deben presentarse edificios cuya extensión sea tal que tome a la nave al menos tres segundos pasar junto a ellos. Análogamente a lo indicado en el párrafo anterior, deben presentarse edificios de tamaños variados, no sólo los extremos.

La generación de las edificaciones debe ser aleatoria y distinta para cada partida, siempre dentro de los parámetros arriba mencionados.

Si la nave impacta contra cualquiera de las edificaciones, el jugador **pierde 5 puntos**. De producirse una colisión, es importante que esto sea comunicado de manera **inmediata y muy explícita** al jugador. Se sugiere efectuar un cambio de color contrastante y llamativo sobre la nave, o incluso en toda la pantalla,

por un número de frames adecuado.

Tras una colisión, ya sea contra una edificación o un blanco, la nave será invulnerable a toda futura colisión por un lapso de **2 segundos**. Esto también debe comunicarse claramente al jugador. El efecto tradicional es hacer parpadear la nave durante este tiempo.

Otros elementos del juego, como anillos y blancos, no deben intersectar a los edificios, sobre todo de manera desordenada o visualmente desagradable.

#### **Texturas:**

Todos los elementos del juego deben estar texturizados, sobre todo la nave del jugador, los edificios, la superficie del planeta y el cielo. El horizonte no debe ser sólo una línea recta sino que debe presentarse la ilusión de objetos en la lejanía.

Se evaluarán desfavorablemente los objetos que posean un aspecto desagradable o discordante debido a texturas distorsionadas, mal colocadas, con discontinuidades o debido a la ausencia intencional o no de las texturas.

#### **Sombras:**

La escena está iluminada por una luz direccional que apunta hacia abajo, hacia un lado y hacia el fondo de la pantalla con cierto grado de inclinación. Deben dibujarse las sombras correspondientes a esta luz sobre la escena.

La mayor importancia la tienen las sombras generadas por la nave sobre la superficie del planeta y sobre el frente, techo y uno de los lados de las edificaciones.

También son deseables las sombras de los blancos y anillos sobre, igualmente, el planeta y las edificaciones. Idealmente, cada edificio también debe proyectar sombras sobre el planeta y los otros edificios.

Todas las superficies sobre las que se proyectan sombras son planos, por lo que se pueden implementar mediante el mecanismo para sombras planares visto en clase. Algunos recursos para implementar sombras planares en OpenGL pueden encontrarse aquí:

<http://www.ia.hiof.no/~borres/cgraph/explain/shadow/p-shadow.html>

<http://www.moddb.com/tutorials/cheap-shadows>

Si lo desea, puede aplicar sus propios mecanismos y optimizaciones. Por ejemplo, puede aprovechar el hecho de que no varía el ángulo de la luz ni respecto a los objetos ni respecto a las superficies sobre las que se proyecta.

No se necesitan considerar las sombras que podrían generar blancos, anillos y la nave entre sí mismos. Opcionalmente se puede reducir la iluminación sobre la nave cuando esté a la sombra de un edificio, de manera exacta o aproximada.

En el caso de los blancos y anillos, es aceptable generar las sombras en base a un modelo de menor complejidad poligonal. Si por razones de desempeño también es necesario hacerlo en el caso de la nave, puede hacerlo si la sombra generada resulta convincente.

Recuerde que el vector de la luz direccional debe ser consistente con la generación de las sombras. No se necesitan sombras correspondientes a cualquier otra fuente de luz en la escena.

#### **Movimientos de la nave:**

Los movimientos de la nave deben ser fluidos, sin discontinuidades. En otras palabras, la nave no debe "saltar" de un punto a otro del espacio.

#### **Funcionalidad extralúdica:**

Como en el proyecto anterior, el juego se pausa al presionar la tecla **(P)**. La acción en pantalla debe congelarse y **permanecer visible**.

Por defecto, el juego debe correr a un *frame rate* o velocidad de cuadro fija, idealmente de 60 o 30 imágenes por segundo.

El *frame rate* se puede modificar haciendo uso de las teclas (+) y (-). Cada vez que se presione (-), el *frame rate* debe reducirse a aproximadamente la mitad, hasta llegar a 1 FPS. La tecla (+) debe tener el efecto contrario, hasta llegar a la velocidad original. Cuando el juego esté corriendo con un *frame rate* modificado, éste debe **mostrarse en pantalla**. Esta funcionalidad es idéntica a la requerida en el proyecto anterior.

Como funcionalidad nueva, se debe detener el temporizador al presionar la tecla **(O)**, permitiendo jugar de manera infinita.

Puede sustituir las teclas mencionadas en esta sección por otras que considere más convenientes, siempre que el cambio esté documentado.

#### **Punto extra:**

Capture un video consistiendo en una partida de su juego y colóquelo en un sitio de visualización de videos vía web como *YouTube* o *Vimeo*. Este video será enlazado desde el sitio web del curso. El título y la descripción del video son libres mientras no incluyan contenido ofensivo.

Adicionalmente, los equipos que no obtuvieron el punto por música y efectos de sonido en el proyecto anterior, pueden hacerlo en esta entrega de cumplir con dichos requisitos.

#### **Punto de chequeo:**

Si le es de utilidad, puede utilizar la librería *trimesh*:

<http://www.cs.princeton.edu/gfx/proj/trimesh2/>

Se recomienda implementar las funcionalidades empezando por aquellas de mayor prioridad.

Se realizará un chequeo previo el viernes de la semana 11. Para el momento del chequeo debe estar implementada como mínimo un tercio de la funcionalidad requerida.

#### **Corrección y revisión:**

Se juzgará su criterio para ajustar los parámetros del juego de manera de lograr el impacto visual deseado. Se evaluará en base a los resultados que se vean en pantalla durante la ejecución del juego, no respecto a si técnicamente están incluidos en el código. Al existir criterios parcialmente subjetivos, en caso de duda consulte con anterioridad.

Si considera que alguno de los requerimientos especificados es contrario a la usabilidad o jugabilidad, consulte (con tiempo) sobre su posible modificación. Son bienvenidas las variaciones temáticas o creativas, siempre y cuando posean la misma complejidad técnica. De nuevo, ante la duda consulte.

La corrección es presencial durante el día 7 de diciembre (viernes S.12) en el horario de laboratorio, de 7:30 AM a 9:30 AM, en orden de llegada. De no alcanzar el tiempo, la corrección continuará inmediatamente después, hasta las 11:30 AM, en MYS-210C.