

Universidad Católica Andrés Bello

Facultad de Ingeniería

Escuela de Ingeniería Informática

Introducción al diseño y programación de videojuegos (Octubre - Febrero 2011)

Proyecto 1: Space Invaders/Tetris

El primer proyecto de este curso está pensado para romper el hielo con el lenguaje Python y la librería PyGame. Debido a que estaremos haciendo juegos a un nivel muy bajo, es decir, sin contar con librerías de interfaces o similares, este primer proyecto presenta dos alternativas de juegos muy simples, que ya presentan un diseño bien conocido, y del cual no deberían desviarse mucho.

Para este proyecto deben escoger uno de estos dos juegos: **Space Invaders** o **Tetris**, cuyo diseño está especificado a continuación, implementarlo según las restricciones establecidas en este documento, entregarlo y presentarlo para la fecha de presentación, y elaborar un post-mórtem del juego. Los equipos serán de dos (2) estudiantes.

El objetivo del proyecto es que puedan aprender a usar Python y PyGame, y al mismo tiempo saber cómo es la estructura interna de un videojuego. La fecha de entrega de este proyecto es el **10 de noviembre de 2010**, día en el que se hará una presentación ante el salón del juego, y se entregará un informe de máximo 2 páginas en el que se hará un post-mórtem del juego, reflexionando sobre el proceso de desarrollo del mismo.

Space Invaders

Space Invaders es un arcade diseñado por Tomohiro Nishikado en 1978, para la Taito Corporation. El objetivo del juego es obtener el mayor puntaje posible, a través de la eliminación de olas de aliens, que se acercan a la Tierra con un patrón de movimiento muy particular, con un cañón movable. Es un juego emocionante y a la vez con reglas muy sencillas, por lo que ha sido escogido para ser una de las alternativas para este primer proyecto.

Reglas de inicio

Al iniciar una partida, la ola de enemigos se dispondrá en la esquina superior derecha del campo. El cañón del jugador comenzará en la parte baja de la pantalla, centrado horizontalmente. Justo encima del cañón habrán cuatro bloques que

conforman las defensas del jugador. El jugador contará con 3 vidas en cada partida.

Reglas de progreso

Movimiento del jugador

El cañón del jugador puede moverse solamente por la parte baja de la pantalla, de un extremo de la pantalla hasta el otro. Puede disparar una bala de su cañón, la cual saldrá impulsada hacia arriba, y manteniendo la posición en el eje x que tenía el cañón en el momento que fue disparada. El cañón no podrá disparar de nuevo hasta que la bala impacte a un enemigo, a una defensa, o escape por la parte superior de la pantalla.

Si la bala de un enemigo impacta al cañón, el cañón se destruye, mostrándose gráficamente la destrucción del cañón, mientras se pausa el movimiento de la ola de enemigos. En caso de que el jugador cuente con una vida extra, se consumirá una vida, y el reposicionará al jugador en la parte central de la pantalla. Si el jugador no cuenta con una vida extra, se mostrará un texto de que el juego ha acabado.

Movimiento de la ola de enemigos

Los enemigos estarán dispuestos en una grilla de 11 enemigos de ancho por 5 de alto. Los enemigos se moverán desde su punto de inicio hasta el otro lado de la pantalla, bajan un espacio hacia abajo, y reinician el movimiento horizontal hacia el lado contrario, y así hasta llegar al nivel del cañón. El punto en el que se determina cuándo bajan está establecido por el enemigo que esté más al extremo de la pantalla hacia donde se dirigen los enemigos. Esto es: el jugador puede retrasar el momento en el que los enemigos se mueven hacia abajo eliminando la columna de enemigos que esté más al extremo.

Existen 3 tipos de enemigos con diferente cara, pero que funcionan de igual manera. Cada cierto tiempo, que se determina por una parte al azar, y por otra parte en función del nivel del juego (es decir el número de ola de enemigos), se escoge a un alien al azar, y éste disparara una bala hacia abajo, manteniendo la posición en x de la bala. La bala desaparece y es destruida si pasa del borde inferior de la pantalla. Si la bala impacta una defensa, quedará a elección del estudiante la deficiencia de este comportamiento. Si la bala impacta al cañón, ocurre lo descrito anteriormente en el apartado del movimiento del jugador. La destrucción de cada enemigo valdrá 10 puntos.

Si los enemigos logran llegar hasta la altura de las defensas, todas las defensas desaparecerán.

Existe un tipo extra de enemigo, que saldrá al menos dos veces en cada nivel, y tiene forma de platillo volador. La destrucción de este tipo especial de enemigo

valdrá 100 puntos.

Cada uno de los 3 enemigos básicos tiene dos frames de animación, los cuales cambian cada vez que se mueven un espacio. Queda de parte del estudiante determinar si quiere animar al enemigo especial.

Bonificaciones

Por cada 1000 puntos que obtenga el jugador, se le otorgará una vida extra.

Reglas de resolución

Si los enemigos básicos llegan a la altura del cañón, la partida terminará y el jugador pierde, independientemente de la cantidad de vidas que posea en ese momento.

Si el jugador logra eliminar a toda la ola de enemigos antes de que éstos toquen al cañón, se inicia una nueva ola de enemigos tal como lo especifican las reglas de inicio, con una velocidad de movimiento más rápida, y una rata de disparos más rápida.

Corolario: el jugador nunca gana.

Interfaz

En todo momento la pantalla debe mostrar el puntaje del jugador, el puntaje máximo que ha logrado en esa sesión, el número de vidas del jugador y, cuando mata a un enemigo, un número en el espacio donde murió indicando la cantidad de puntos que gana el jugador, además del campo de juego.

Al iniciar el videojuego, se debe mostrar un campo vacío, sin cañón o enemigos, solamente las defensas completas, invitando a iniciar el juego con la tecla espacio. El movimiento del cañón se hará con las teclas direccionales, y el disparo se hará con la tecla espacio. La tecla escape cerrará el juego.

Gráficos

El estudiante cuenta con una librería de gráficos descargable para poder desarrollar el juego sin atención a la parte gráfica, aunque es libre de usar gráficos alternativos si así lo prefiere.

El juego debe utilizar una resolución de 434x496 pixeles, o al menos utilizar una resolución que respete la relación de alto/ancho (es decir, puede usar una resolución de 217x248 pixeles). El juego debe caber dentro de una pantalla de 1024x768 pixeles.

Sonido

El estudiante puede generar los sonidos del juego a partir del programa sfxr, de DrPetter: http://www.drpetter.se/project_sfxr.html

Tetris

Tetris es un juego diseñado por Alexey Pajitnov en 1984, en la Unión Soviética. Debido a sus gráficos simples y su alto nivel de abstracción, está disponible para virtualmente cualquier plataforma y sistema operativo que ha surgido desde entonces. Por lo general es un gran juego para comenzar a aprender el desarrollo de videojuegos, porque contiene todos los elementos básicos que puede tener un videojuego, simplificados hasta su esencia.

Reglas de inicio

Al iniciar una partida, se dispone de un espacio cuadrulado, de 10 bloques de ancho por 14 de alto. Una pieza de las 7 disponibles se escoge, y se coloca en el tope del espacio, centrado horizontalmente.

Reglas de progreso

Piezas disponibles

El jugador cuenta con 7 tipos de piezas, cada una con un color característico: el bloque cuadrado (2x2 bloques), la S (3x2 bloques), la S invertida (3x2 bloques), la L (3x2 bloques), la L invertida (3x2 bloques), la T (3x2 bloques), y la línea (4x1 bloques).

Movimiento de los bloques

El jugador puede controlar un solo bloque a la vez. El bloque puede ser rotado en sentido de las agujas del reloj o al contrario libremente y en cualquier momento por el jugador. El bloque cada cierto tiempo baja un espacio, dependiendo del nivel en el que se encuentre el jugador (a mayor nivel, menor es el tiempo en el que baja un espacio).

Cuando el bloque toca el fondo del espacio, o toca un bloque ya fijo, pasado cierto tiempo el bloque se fijará como parte del espacio, se escogerá una nueva pieza al azar y se colocará al tope de la pantalla, como está especificado en las reglas de inicio.

El bloque puede rotarse libremente mientras haya espacio para hacerlo. Si no hay espacio para rotarse, el bloque se mantiene en su orientación previa. Cuando el bloque entra en contacto con el fondo de la pantalla o algún bloque ya fijo, el bloque puede rotar todas las veces que el jugador quiera antes de fijarse.

Después de fijado un bloque, se debe comprobar en cada fila del espacio si todos los espacios de esa fila están cubiertos por un bloque. En tal caso, se desaparecen todas las líneas que hayan sido completadas por ese bloque, y se bajan todas las

filas superiores un número de espacios iguales a la cantidad de filas desaparecidas.

Bonificaciones

Una línea desaparecida otorga al jugador 10 puntos, dos líneas al tiempo otorgan 30 puntos, tres líneas otorgan 80 puntos y cuatro líneas otorgan 200 puntos. Al llegar a 10 líneas desaparecidas, se aumenta en uno el nivel de dificultad del juego y se hace la caída de los bloques más rápida.

Reglas de resolución

Si al surgir un bloque nuevo en el tope-centro horizontal del espacio, ya existe un bloque fijado, se da por terminada la partida.

Interfaz

En todo momento, el juego debe mostrar, además del espacio de juego, el puntaje de la partida actual, el número de líneas completadas, el nivel de la partida actual (comenzando desde 1), y el máximo puntaje obtenido en una sesión. También debe mostrar cuál será la siguiente pieza a salir.

Gráficos

El estudiante cuenta con una librería gráfica que le evitará el trabajo de diseño gráfico para este juego, pero está en la libertad de elaborar otros si así lo prefiere. El juego debe utilizar una resolución de 800x600 pixeles.

Sonido

El estudiante puede generar los sonidos del juego a partir del programa sfxr, de DrPetter: http://www.drpetter.se/project_sfxr.html

Post-mortem

El post-mortem es un procedimiento usual en el desarrollo profesional de videojuegos. En un post-mortem describe a grandes rasgos la envergadura del proyecto (ej. número de líneas de código, o de clases, o tamaño en MB del proyecto) y se listan todas las cosas que fueron bien en el desarrollo (ej. el diseño caló bien la primera vez, ya conocíamos la tecnología, no hubo sorpresas desarrollando, no hubo cambios de diseño, tuvimos una buena relación con el publisher y cumplimos los tiempos, elegimos un buen control de versiones, etc.) y todas las cosas que fueron mal, o que pudieron ir mejor (nos dedicamos demasiado tiempo a hacer gráficos, nos concentramos demasiado en la historia y echamos a perder la jugabilidad, no probamos lo suficiente, nuestro publisher nos recortó el presupuesto a última hora, se quemó el repositorio de código, etc.).

La idea del post-mortem es reflexionar acerca del proceso de desarrollo, para

mejorar lo malo en los siguientes proyectos. Por ello, debe estar plasmado en un informe corto, de 1 página, o máximo 2 si hay mucho de qué hablar. El informe no debe tener portada, solamente un encabezado como el de este proyecto, fechado, con los integrantes del proyecto, y el contenido a continuación.

Evaluación

Al entregar este proyecto, se les pasará una encuesta anónima a cada uno, en el que evaluarán el desempeño de su compañero en base a ciertos parámetros. En los siguientes proyectos no volverán a repetir compañeros, sino que escogerán a otros, y se volverá a aplicar esta evaluación. Aunque no tiene peso dentro de su nota, al final de este curso obtendrán los resultados de esta encuesta, y podrán saber cómo son apreciados en su trabajo por los demás compañeros.

Copia

Aunque la conversación entre equipos para enriquecer los juegos es completamente bienvenida y alentada, la copia textual de juegos entre equipos no está permitida. Cualquier caso de copia detectada ameritará una nota de cero uno (01) en el proyecto para todos los involucrados y se aplicarán las sanciones que permita el reglamento de la UCAB.

¿Preguntas?

Todas las preguntas se harán públicamente a la lista de correos del curso en <http://groups.google.com/group/disena-juegos> y se responderán públicamente en la página del curso en <http://www.ciroduran.com/disenajuegos>. Los próximos días de clase se dedicará un tiempo para atender los proyectos de cada pareja.