

Computer Engineering

(CEAB Substantially Equivalent Program and

AAPIA-CFIA Accredited Seals)

Project II:

Computer Engineering

Star Force Remake, May the Force Be With You
and help Conquer the Stars

Professor:
Milton Villegas Lemus

Trying to read our DNA is like trying to understand software code - with only 90% of the code riddled with errors. It's very difficult in that case to understand and predict what that software code is going to do.

Elon Musk

To make an embarrassing admission, I like video games. That's what got me into software engineering when I was a kid. I wanted to make money so I could buy a better computer to play better video games - nothing like saving the world.

Elon Musk

My vision is for a fully reusable rocket transport system between Earth and Mars that is able to re-fuel on Mars - this is very important - so you don't have to carry the return fuel when you go there.

Elon Musk

Introduction

The second computer programming project is based on the game Star Force. The game was released for arcade machines, by Video Ware as Mega Force [1], in 1984. It was also published in 1985 by Hudson Soft, both for the home computer (home desktop, nowadays) MSX, and for the Family Computer (Famicom) in Japan. The

North American version for Nintendo Entertainment System (NES), was released in 1987 by Tecmo. [2].

Introducción

El segundo proyecto programado está basado en el juego Star Force. El juego fue lanzado para las máquinas Arcade, por Video Ware como Mega Force [1], en 1984. También fue publicado en 1985 por Hudson Soft, tanto para la computadora del hogar (home desktop, hoy día) MSX, como para la Family Computer (Famicomp) en Japón. La versión norteamericana para Nintendo Entertainment System (NES), fue liberada en 1987 por Tecmo. [2].



Figura 1. Versión de Tekhan del juego Star Force, corría en un MP Z80 de 4MHz [3].

Descripción

El juego estará compuesto por una pantalla principal donde aparece el nombre del juego, el tema y un sonido de fondo agradable con un volumen agradable, desde donde se puede ir ya sea a escoger al piloto que se desea utilizar, escoger modo de juego o a una pantalla de configuración. En esta última se pueden cargar nuevos pilotos, se actualiza información de pilotos o donde se eliminan (foto, nombre del piloto, puntaje obtenido, logo del patrocinador y país). En la pantalla principal, el usuario deberá escoger el modo de juego de su preferencia. Los modos de juego son: destrucción de asteroides y la prueba de maniobras, este último es el modo por defecto.

Se tienen un total de 12 pilotos diferentes disponibles por defecto para escoger, 5 hombres 5 mujeres. Inicialmente se les asignará un puntaje aleatorio a cada uno de los pilotos el cual no debe superar el 30% del valor del mayor puntaje posible. Estos valores se deben almacenar en un archivo de forma que estén disponibles cada vez que se reinicie el juego.

El resultado siempre actualiza la tabla de mejores puntajes (ranking) cada vez que se termine un juego, mediante alguno de los algoritmos vistos en clase. No se pueden utilizar funciones propias de Python para ordenar los resultados. Los valores generados aleatoriamente deben estar en un intervalo razonable, de forma que no se den valores muy altos.

Una vez se ha seleccionado el modo, se debe continuar guardando las mejores 7 puntuaciones de los usuarios. Esto quiere decir que se actualizan los puntajes asignados aleatoriamente. Estos puntajes se deben visualizar de forma ordenada de mayor a menor en una ventana definida de altos puntajes que se accede desde la pantalla principal y también aparece si y solo si el puntaje obtenido está entre uno de los mejores 7.

Un jugador puede personalizar el nombre del piloto desde la pantalla de configuración. La nave que utilizan los pilotos, en ambos modos de juego, necesita de energía para seguir funcionando. Fuentes de energía deben aparecer de forma aleatoria en la pantalla, cada cierto tiempo. Conforme el jugador avanza el tiempo de juego más demora en aparecer la energía. El jugador debe pasar sobre la energía para obtenerla.

Los modos de juego se explican a detalle en la sección de pantalla de juego.

A continuación, se explican las diferentes pantallas del juego:

Pantalla principal

Esta pantalla es la primera que se muestra al usuario una vez se ejecuta el programa, en la misma se debe poder acceder a las pantallas de: acerca de, altos puntajes, configuración y al modo de juego seleccionado. Además, debe permitir al usuario cerrar el programa.

Debe tener una buena presentación, una imagen alusiva al juego y una melodía acorde.

Pantalla About

En esta pantalla se debe incluir nombre de la institución, carrera, nombre del profesor, estudiante, carnet, versión, fecha de última modificación, una fotografía del o los autores, indicaciones para el juego entre ellas las teclas y sus acciones en los dos modos. Se debe poder retornar a la pantalla principal.

Pantalla altos puntajes

En esta pantalla se deben de mostrar los 7 mejores puntajes. Los puntajes y el nombre del piloto deben ser almacenados en un archivo de texto (.txt). Se debe poder retornar a la pantalla principal.

Pantalla de Configuración

En esta pantalla se podrán personalizar los nombres de los pilotos que se leen por defecto. Los pilotos se deben mostrar por grupos de mínimo 4. Cuando se desea eliminar un piloto o modificar la información asociada a un piloto. desde el grupo de pilotos, se debe permitir escoger uno, Una vez se abandona esta ventana se debe actualizar automáticamente y reportar este hecho mediante un mensaje de notificación. Se debe poder retornar a la pantalla principal.

Pantalla de juego

Como se mencionó anteriormente existen 2 modos de juego, destrucción de asteroides y la prueba de maniobras.

En ambos modos de juego el usuario puede ver la cantidad de energía restante. La misma disminuye con el paso del tiempo, situación que debe verse en pantalla mediante algún indicador (analógico o de barras). Además del piloto seleccionado y el puntaje obtenido.

Destrucción de asteroides

En esta ventana entre otras cosas se debe mostrar: Puntaje actual, Tiempo de juego, Cantidad de vidas restantes, Nombre del jugador

La nave se encuentra en el espacio, en medio de un cinturón de asteroides, el usuario debe poder esquivar o destruir los asteroides. Un ejemplo se puede apreciar en la

Figura 1, los asteroides vienen de frente a la nave. Cada asteroide destruido suma puntos, los que se esquivan no suman, la cantidad de asteroides restantes debe aparecer en pantalla en un sitio adecuado para no interferir con el juego. El máximo de asteroides es de 9.

Si un asteroide golpea a la nave el juego termina y pasa a mostrar el resultado. Se debe devolver a la pantalla inicial para continuar jugando.



Figura 2. Idea de visualización de la pantalla con asteroides [2]

Maniobras de Prueba

Como se puede ver en la figura 2, en este modo de juego aparecen aros por los cuales el jugador debe pasar. A medida que la nave se aproxima el aro, éste se hace más grande, pero no tan grande que cubra el área de la pantalla. Si se choca con el borde del aro, se termina el juego. Si no se pasa en medio del mismo ese aro no sumará puntaje.

Maniobras de Prueba

Debe haber tres niveles mínimos de dificultad y se deben basar en la cantidad de obstáculos por unidad de tiempo que se muestran. Los puntajes deben ser proporcionales a la cantidad de objetos (Asteroides) esquivados, destruidos o atravesados por la mitad (Aros).



Figura 3. Ejemplo de una nave y los aros por los que debe pasar en su ruta, para obtener puntos. [2]

Como juego su objetivo es que sea divertido, fluido y jugable.

Aspectos generales

- Se debe poder perder y ganar el juego mediante una notificación específica. Su dificultad **NO** debe ser imposible ni nula.
- Cualquier duda, omisión o contradicción en la especificación se debe aclarar con el profesor y se difundirá a través del asistente.
- El proyecto es **INDIVIDUAL o en PAREJAS**. Se deberá entregar el día: Aug 6 2020. Cualquier indicio de copia o plagio significará un cero en la nota y la aplicación de las sanciones específicas en reglamento RREA.
- La documentación archivo(s) formato **.doc** consta de dos partes: Documento Técnico Ejecutivo y Documento Administrativo del Proyecto (Aplica solo para grupos).
 - El Documento Técnico Ejecutivo (archivo **.pdf**) deber contar con lo siguiente: Introducción, Conclusiones, Recomendaciones, Diagramas de la Arquitectura, Diagramas de Módulos, Plan de Pruebas, Análisis de Resultados y literatura o fuentes consultadas.
 - El documento Administrativo del Proyecto, es donde se establecen claramente las reglas del grupo, los roles de cada integrante, asignaciones o actividades, responsable, fecha preliminar y efectiva de entrega, la completitud de la misma. En este documento se entregan los rubros de evaluación para los integrantes y la evaluación justificada por semana respectiva.
- Se debe enviar el software el día **Jueves 6 de Agosto de 2020**, por aparte en una asignación específica enviar la documentación (incluirla bitácora) vía el Classroom, a más tardar a las 23:30. Debe entregar todos los componentes software necesarios para probar todos los comandos de forma que demuestre pleno funcionamiento.

- Debe indicarse un archivo readme. txt con la versión de Python (≥ 3.3) a utilizar para la revisión, alguna otra indicación que se considere importante. Este archivo contiene el nombre y la descripción de una función o procedimiento especial designado `mi_auto_doc()`, que genera de forma automática (con print de Python), todas las autodocumentaciones de los principales módulos (mínimo 3).
- Cada archivo base debe enviarse el nombre construido de la siguiente forma: Iniciales de los nombres (del autor) en mayúsculas, guion bajo primer apellido, guion bajo segundo apellido y extensiones txt y doc.
- Toda la documentación formará parte de la defensa.
- Cualquier duda, omisión o contradicción en la especificación, se debe aclarar con el profesor y esta se difundirá, a través del Classroom de Google.
- Para la realización de la parte gráfica se pueden usar bibliotecas complementarias Python (Pygame o Tkinter) otras que considere de ayuda deben consultarse con el profesor.
- El código debe estar suficientemente documentado de tal forma que Ud. se pueda orientar en el fácilmente durante la defensa.
- El proyecto se debe defender previa cita con el profesor y el asistente. El profesor o asistente publicará unas fechas de defensa. Ud debe inscribirse para defender el proyecto, de no asistir a la defensa, únicamente obtendrá la nota correspondiente a la documentación del proyecto.
- Las funciones de repetición dentro de la lógica del programa deben ser desarrolladas **recursivamente, no se aceptará con iteración**. Desviarse de ésta directriz implicará la anulación de esa parte del código mediante comentarios en él y su correspondiente pérdida de puntos.

- Cualquier clase de copia de código será sancionada de acuerdo con el reglamento vigente y se llevará hasta la consecuencia de amonestación con carta al expediente. Código adoptado para el manejo de interfaz debe especificarse claramente la fuente y reconocer los créditos. Está prohibida la copia de código que involucre la solución lógica general del algoritmo.
- Si en la defensa no demuestra su autoría con el dominio propio de esa calidad solo se le otorgarán los puntos correspondientes a los obtenidos en la documentación.
- Las funciones de repetición dentro de la lógica del juego deben ser desarrolladas **recursivamente, no se aceptará con iteración**. Desviarse de ésta directriz implicará la anulación de esa parte del código mediante comentarios en él y su correspondiente pérdida de puntos.

Literatura

[1] juegos.com. (2018). Star Force. [online] disponible en: <http://www.juegos.com/juego/star-force>. S. F. Obtenido 7 Jul 2020.

[2] "Hudson - Shooting game [NES] (Archive)". Hudson. Hudson. S. F. Obtenido el 16 Mayo 2018.

[2] <https://www.arcade-history.com/?n=star-force&page=detail&id=2614> Obtenido Jul 8 2020.

Tabla de evaluación

Rubro	Valor	Obtenido
Aspectos Generales (25 pts)		
Pantalla Inicial <ul style="list-style-type: none"> • Captura del nombre • Acceso a otras pantallas • Botón/acceso a inicio de juego • Selección de dificultad inicial 	4	
Pantalla Información complementaria <ul style="list-style-type: none"> • Información completa • Retorno a pantalla de inicio 	4	
Pantalla de Mejores Puntajes <ul style="list-style-type: none"> • Muestra mejores puntajes • Se actualiza correctamente • Mejores puntajes se mantienen al cerrar y abrir el juego 	9	
Pantalla de Configuración	8	
Aspectos de juego (43 pts)		
Pantalla de escenario de batalla <ul style="list-style-type: none"> • Muestra fondo de la temática escogida • Muestra puntaje • Muestra tiempo de juego • Muestra nombre del jugador • Se le comunica al jugador cuando pierde o gana • Si el puntaje del jugador es uno de los mejores 7, se lo comunica inmediatamente al perder o ganar y se le dice en cual posición quedó 	12	
Movimiento de jugador dos modalidades <ul style="list-style-type: none"> • Jugador puede moverse en ambas direcciones • Jugador no se sale de los límites de la pantalla • Jugador puede esquivar cuando se encuentra destrucción de asteroides • Jugador no puede saltar mientras se encuentra Maniobras de Prueba 	8	
Destrucción de Asteroides <ul style="list-style-type: none"> • Asteroides reciben daños al ser impactado por un proyectil del jugador • Jugador pierde una vida al ser impactado por un asteroide 	6	
Maniobras de Prueba <ul style="list-style-type: none"> • Se crean aros de manera dinámica • Se implementan los patrones de movimiento especificados. 	8	

<ul style="list-style-type: none"> Jugador pierde una vida al impactar el Aro. La cantidad de Aros se distribuye adecuadamente en la duración del nivel 		
Dificultad (3 niveles mínimo) <ul style="list-style-type: none"> Aumenta cantidad de obstáculos al subir el nivel Frecuencia de disparo limitada (no hay efecto “rayo láser” al lanzar un proyectil) Tiempo mínimo de 30s entre cada nivel 	9	
Documentación 38 pts		
Introducción	1	
Conclusiones	6	
Recomendaciones	5	
Análisis de resultados <ul style="list-style-type: none"> Diagrama de módulos Plan de pruebas 	4	
Bitácora	4	
Literatura o Fuentes	1	
Módulo de Generación de Autodocumentación de los Módulos Principales	2	
Auto documentación de módulos sigue formato	3	
Documentación Interna en partes de los módulos más importantes	2	
Documentación de Reglas de Grupo y Roles	5	
Actividades, fechas de entrega y coevaluación	5	

Nota: Para las personas que realicen el proyecto individual los puntos de los dos últimos rubros se distribuirán de la siguiente forma, en: conclusiones (4 para un total de 10), recomendaciones (4 para un total de 9) y bitácora (2 para un total de 6).