Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Arquitectura de Computadoras y Ensambladores 1
Primer Semestre 2023
Ing. Álvaro Obrayan Hernández García
Tutor Académico: Ronald Marín

Proyecto 2

Descripción

Para la presente actividad se le solicita el desarrollo de un juego sencillo empleando las características gráficas que brinda DOS y el conjunto de interrupciones que este provee.

Junto a este juego, se desarrollará la interfaz que permitirá manejarlo, el manejo de puntajes más altos, carga de niveles y configuración de controles.

El juego a desarrollar es el juego japonés Sokoban, en el cual el jugador tiene como principal objetivo empujar una serie de cajas hasta conseguir que éstas se ubiquen en ciertas posiciones. Cuando el jugador consigue lo anterior se le permite avanzar de nivel y acumular más puntos.

También se contabilizará el tiempo del jugador en cada partida, dato que también será asociado a la puntuación obtenida por el jugador.

Diseño

El usuario manejará el juego por medio de menús interactivos, los cuáles deberán verse de forma similar a la mostrada en la Figura 1. Con las teclas direccionales hacia

arriba, y abajo, el usuario cambiará de opción en opción. Con la tecla F1 el usuario seleccionará la opción marcada y podrá ir a esa sección del programa.

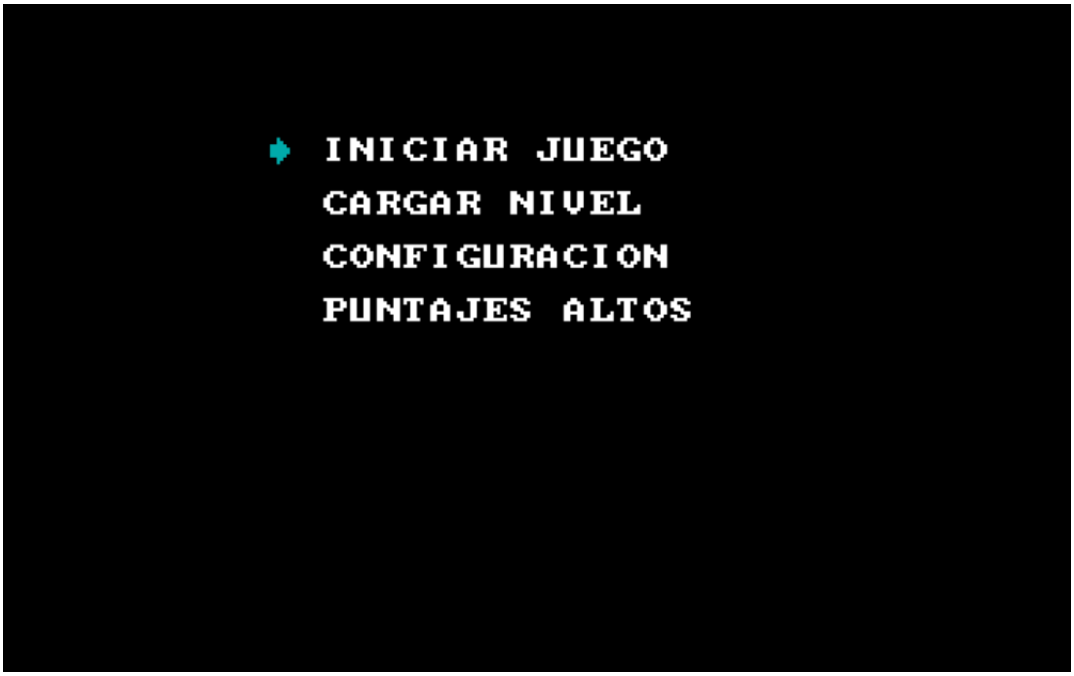

Figura 1

Cuando el jugador tenga una partida activa se deberá mostrar en la última línea de la pantalla las iniciales del desarrollador y en carné, del lado izquierdo, y el tiempo que ha transcurrido desde que el usuario inició la partida, del lador derecho.

En la fila superior, del lado derecho, se mostrará el punteo actual acumulado de otras partidas. El espacio restante en pantalla será ocupado por el juego. En la Figura 2 se presenta un ejemplo con los elementos que se esperan durante el desarrollo de una partida.

Pantalla inicial

Antes de mostrar el menú principal, deberá presentarse una pantalla con sus datos. Esta pantalla deberá mostrarse un

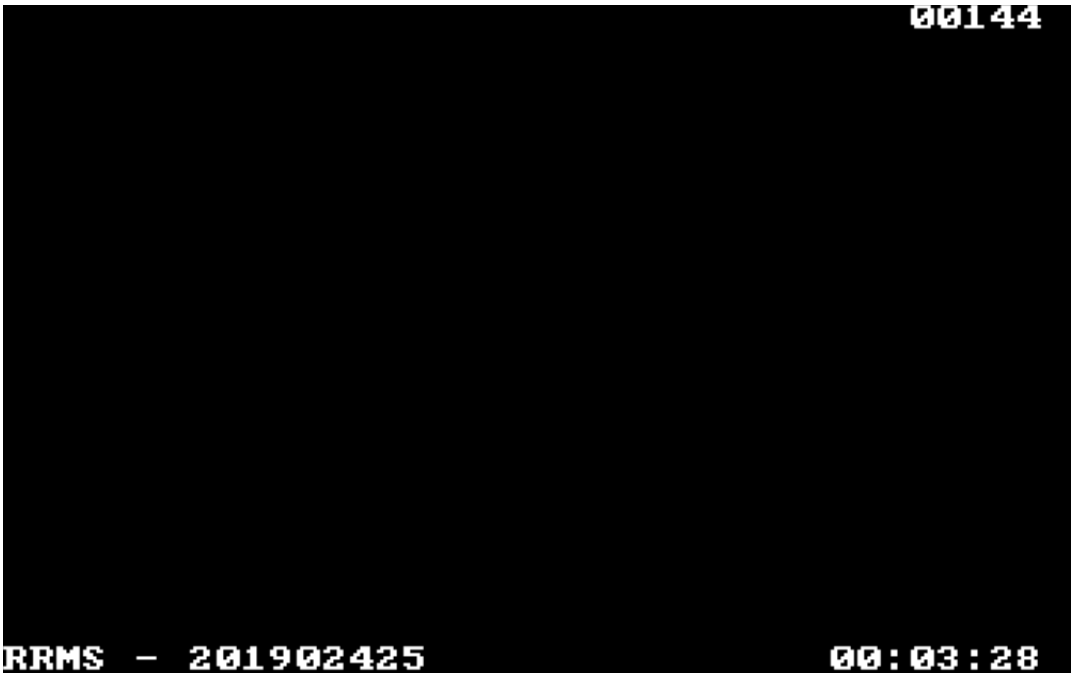

Figura 2

momento para que pueda leerse para luego mostrar el menú principal.

Los datos a incluir quedan a su criterio, pero deben incluir como mínimo su nombre y carné.

Juego

El juego a desarrollar, basado en Sokoban, empleará las mismas mecánicas que el juego antes mencionado. El jugador comenzará la partida observando que las cajas u objetos a mover se encuentran en una parte y las casillas en donde éstos deberían quedar finalmente en otra.

El jugador se podrá mover, con las flechas direccionales o los controles que se configuren posteriormente, por todo el nivel. Normalmente, se le presentará al jugador un área delimitada por paredes de la cuál no podrá salir y constituirá el nivel a resolver.

El jugador tendrá que empujar los objetos para moverlos y ubicarlos en dónde deberían terminar. Se deberá determinar de manera automática cuando el jugador haya logrado resolver el nivel y pasar al siguiente.

Se deberá contar cada uno de los movimientos que haga el jugador, tal número se debe ir acumulando ya que constituirá el puntaje del jugador.

El diseño de las paredes, objetos, jugador y otros elementos queda a discreción del desarrollador. El único detalle a considerar es que éstos deben tener dimensiones de 8x8 pixeles.

Se recomienda dividir la pantalla en una cuadrícula de 40x25. Restando las dos líneas donde se mostrará la información mencionada antes, se tendría un espacio para el juego de 40x23.

Niveles

El juego constará de tres niveles por defecto y una opción para cargar un nivel arbitrario. En ese caso luego de finalizar el nivel se retornará al menú principal.

La ubicación de los elementos del juego en los niveles se obtendrá de un archivo dependiendo del nivel en que se encuentre el jugador. Para el nivel 1 el nombre de tal archivo será “NIV.00” para el segundo “NIV.01” y para el tercero “NIV.10”.

Cuando se inicie una partida normal, el juego cargará el primer nivel, cuando este sea completado cargará el siguiente y continuará de esta forma hasta completar los tres niveles incluidos.

Los archivos de definición de niveles tendrán una sintáxis sencilla, en cada línea del archivo habrá una definición de un elemento del juego. La primera palabra de que aparezca en la línea debería ser: “pared”, “suelo”, “jugador”, “caja” o “objetivo”. Después de eso vendría, separado por uno o más espacios un par de números separados por coma, éstos serían las coordenadas de objetos dentro del espacio disponible para el juego. Las coordenadas tendrán su origen en la esquina superior izquierda y los ejes positivos estarán hacia la derecha y hacia abajo.

Se debe verificar que la coordenada brindada esté dentro del rango posible, de lo contrario se rechazará el archivo y se retornará al menú principal. Un ejemplo de una porción del archivo a analizar se muestra en la Figura 3.

```
pared   02, 09
pared   02, 10
pared   02, 11
pared   02, 12
pared   03,  12
pared   04, 12
jugador 10, 02
caja   10,   02
objetivo  11, 02
```

Figura 3

Puntajes

El “puntaje” en este caso será la cantidad de movimientos que el jugador emplee para resolver el nivel. Mientras menos puntos tenga más arriba estará en el ranking.

Éstos puntajes serán almacenados en un archivo llamado “HIGHSC.BIN”. Se guardarán, en este archivo, como máximo 10 puntajes. Cuando se finalice una nueva partida, se debe verificar si ese puntaje puede ingresar al archivo de puntajes máximos. De ser así se insertará en el lugar correcto dentro del archivo. Si el archivo no existiera deberá ser creado automáticamente.

Al finalizar una partida, y si el puntaje obtenido puede entrar al archivo, se le pedirá al usuario que ingrese un nombre de tres caracteres. Esta cadena también se guardará en el archivo de puntajes.

En el menú principal debe haber una opción para consultar los puntajes en el archivo. Cuando el usuario seleccione tal opción el juego mostrará los puntajes válidos, y demás datos, de forma tabular en pantalla.

Pausa

Durante una partida, el jugador podrá presionar la tecla F2 para entrar al menú de pausa. En este menú simplemente se dará la opción para continuar o salir hacia el menú principal. Mientras se esté en pausa, el reloj del tiempo de juego deberá detenerse y continuar cuando el jugador regrese a la partida.

Configuración

Se dará la opción, desde el menú principal a acceder a un menú de configuración. Este menú servirá principalmente para configurar los controles del juego. En la Figura 4 se ejemplifica el diseño de este menú.

Se deben mostrar los controles actuales y opciones para cambiar controles o regresar al menú principal.

Si se decide cambiar, el programa preguntará por la tecla con la que el jugador avanzará en cierta dirección.

El orden en que se solicitará la nueva tecla será para carnés impares:

- Abajo
- Arriba
- Derecha

- Izquierda

Para el caso de carnés pares el orden sería:

- Derecha
- Abajo
- Arriba
- Izquierda

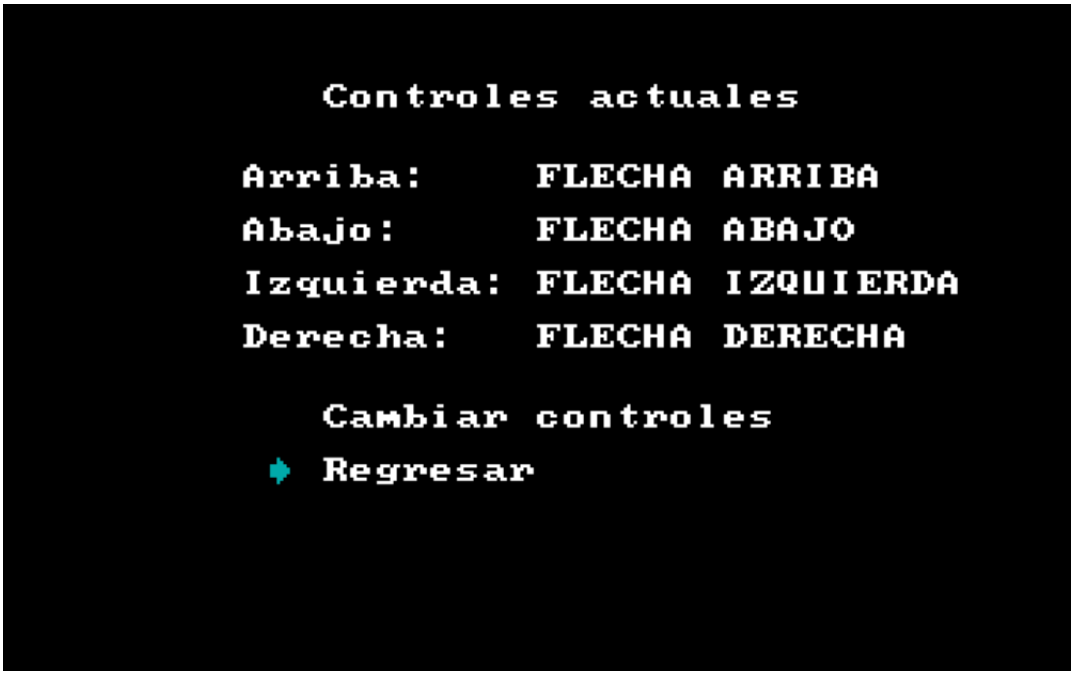

Figura 4

Entregables

- Manual técnico
- Manual de usuario
- Código funcional
- Link al repositorio donde trabajó

Entrega

- La entrega, en UEDI, será el link al repositorio que utilizó.
- El nombre del repositorio responde a la siguiente estructura:

    - ACE1-
    - Los últimos dos dígitos del año en curso
    - La cadena “VJ”
    - Código del curso
    - Su carné
    - PROY2
- Ej. ACE1-23VJ0778202301234PROY2
- Realizar el último commit y hacer su entrega en UEDI antes de 23:59 horas del 01 de julio.
- Se ejecutará un checkout hacia el último commit hecho antes de la fecha de entrega.
- Cuidar que su repositorio no contenga ningún archivo *.EXE o *.OBJ, de lo contrario habrá penalización por ello.
- El ejecutable será generado al momento de la calificación.
- Se debe agregar al auxiliar como miembro del repositorio
    - @romasa000

Observaciones y Restricciones

- Los reportes se abrirán de manera manual una vez generados.
- Debido a la naturaleza del lenguaje ensamblador, solo se evaluarán salidas (resultados en pantalla o reportes) y no código.
- El código del programa debe ser estrictamente ensamblador, no se permite el uso de alguna librería.
- Se debe presentar el proyecto en Dosbox.
- Lenguaje ensamblador a utilizar: MASM 6.11
- No está permitido el uso de estructuras de control if, if else, while, repeat, for.
- No está permitido el uso de STRUCT.
- Es requerido el uso de la directiva .RADIX 16
- Los números de fecha y tiempo de los reportes, o salidas

en pantalla y en caso de ser menores a 10, deben incluir un cero antes del número (ej. 9 -> 09).
- Se debe entregar los manuales técnico y de usuario, de lo contrario se asumirá que el estudiante copió.
- La realización de la práctica es de forma individual.
- El día de la calificación se harán preguntas, modificación de código sobre aspectos utilizados en la elaboración del proyecto, las cuales se considerarán en la nota final.
- Copias parciales o totales tendrán una nota de 0 puntos y los involucrados serán reportados a la Escuela de Ciencias y Sistemas