Universidad Nacional

Estructura de Datos

EIF 207

Profesor: Máster. José Pablo Calvo Suárez

Estudiantes: Leonardo Hidalgo Fonseca  
 Enzo Quesada Rojas

*“Proyecto de Estructuras de Datos # 1.”*

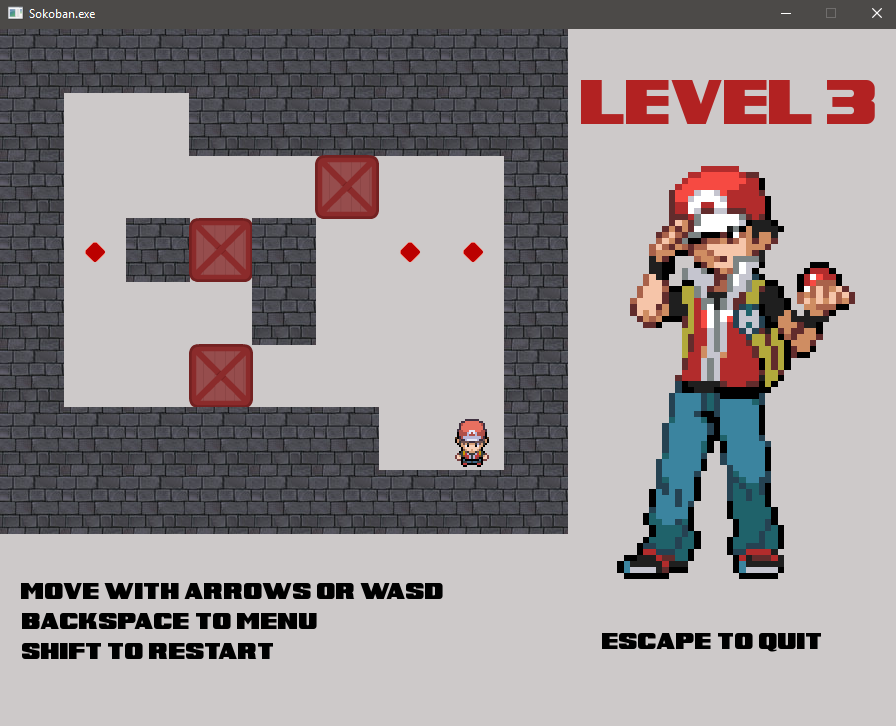
Fecha de entrega: 17 de octubre

2017

II Ciclo

**Introducción.**

El presente escrito resumirá el funcionamiento, características y desarrollo del proyecto de Estructura de Datos (EIF207), dicho proyecto realizado por los estudiantes Leonardo Hidalgo Fonseca y Enzo Darío Quesada Rojas. El proyecto fue realizado en el IDE Visual Studio Community 2017 utilizando la librería SFML para la construcción del modo gráfico, manejo de eventos y total desarrollo de la API. Se implementó el juego conocido como *“Sokoban”*, el cual consiste en un personaje movible el cual empujará cajas hasta llegar a un punto específico, el presente juego desarrollado cuenta con 5 niveles distintos, se pueden cargar por aparte o recorrerlos en orden, avanzando consecutivamente según nivel ganado.

**** El trabajo descrito anteriormente cuenta con un menú paran iniciar o cargar nivel, una vez iniciado el primer nivel, el usuario podrá moverse con las teclas de dirección al igual que con las teclas “WASD”, posee un pequeño menú dentro de la ventana de juego para reiniciar la partida de ser el caso de quedar atrapado y no poder hacer ningún movimiento válido, para esto es necesario presionar la tecla ”LShift”, se regresa el personaje y cajas a su lugar inicial, también está la opción de volver al menú principal con la tecla “Backspace” y como lo indica el enunciado una opción para salir del juego y cerrar el programa inmediatamente, para esto se necesita presionar la tecla “Escape”.

**Limitaciones del proyecto.**

Basado en el enunciado, el proyecto posee 2 limitaciones, a la hora de reproducir la repetición al final de cada nivel, si así lo desea el usuario, la forma de hacerlo es imprimiendo en pantalla cada movimiento uno a uno, según fueron realizados anteriormente para completar el nivel, si se reinicia el nivel o se devuelve al menú principal la repetición será borrada y solo contará a partir de los movimientos que se utilizaron para completar el nivel. Se intentó realizar la reproducción en la ventana del modo gráfico, sin embargo, estaba daba error por una o dos excepciones en un if, como por ejemplo (temp->next->next = nullptr) a diferencia de cómo lo realiza el método para mover al personaje en tiempo real, por lo tanto, se optó por imprimir de esta forma.

La otra limitación es a la hora de cargar la partida actual, se carga la partida por niveles desde 0 a la hora de cargar, ya que se daba un error de debugger y había problemas a la hora de ya sea cargar el archivo alterado que fue una de las opciones y de guardar los nodos tal y como estaban, por este motivo se carga el mapa desde 0 solamente.

C:\Users\Leonardo Hidalgo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\CharacterDer.png**Desarrollo del programa.** Sprites utilizados para la creación de la API.

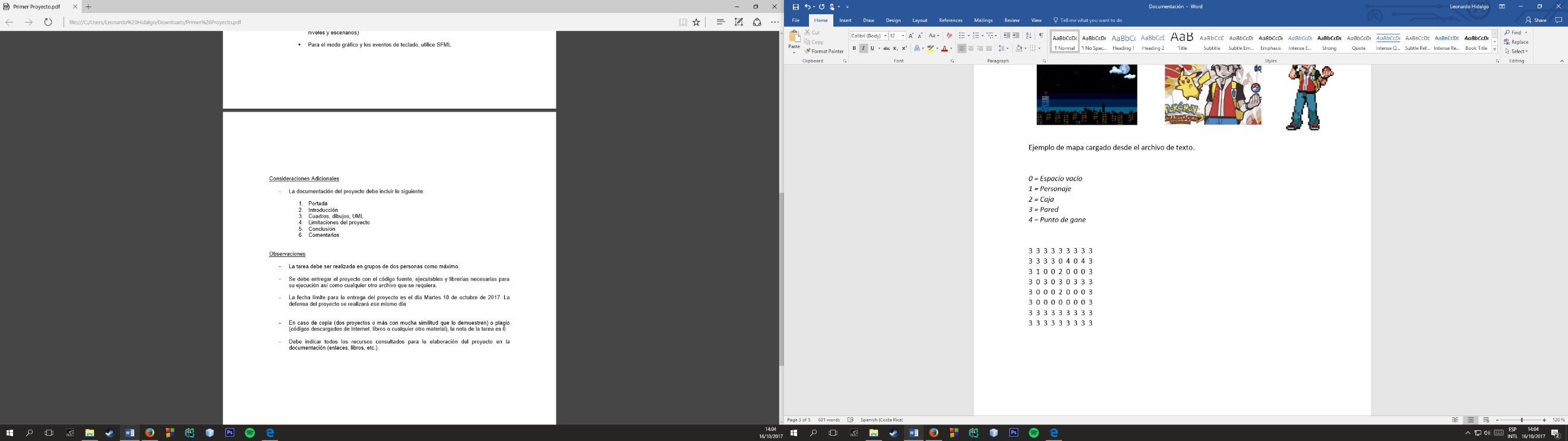








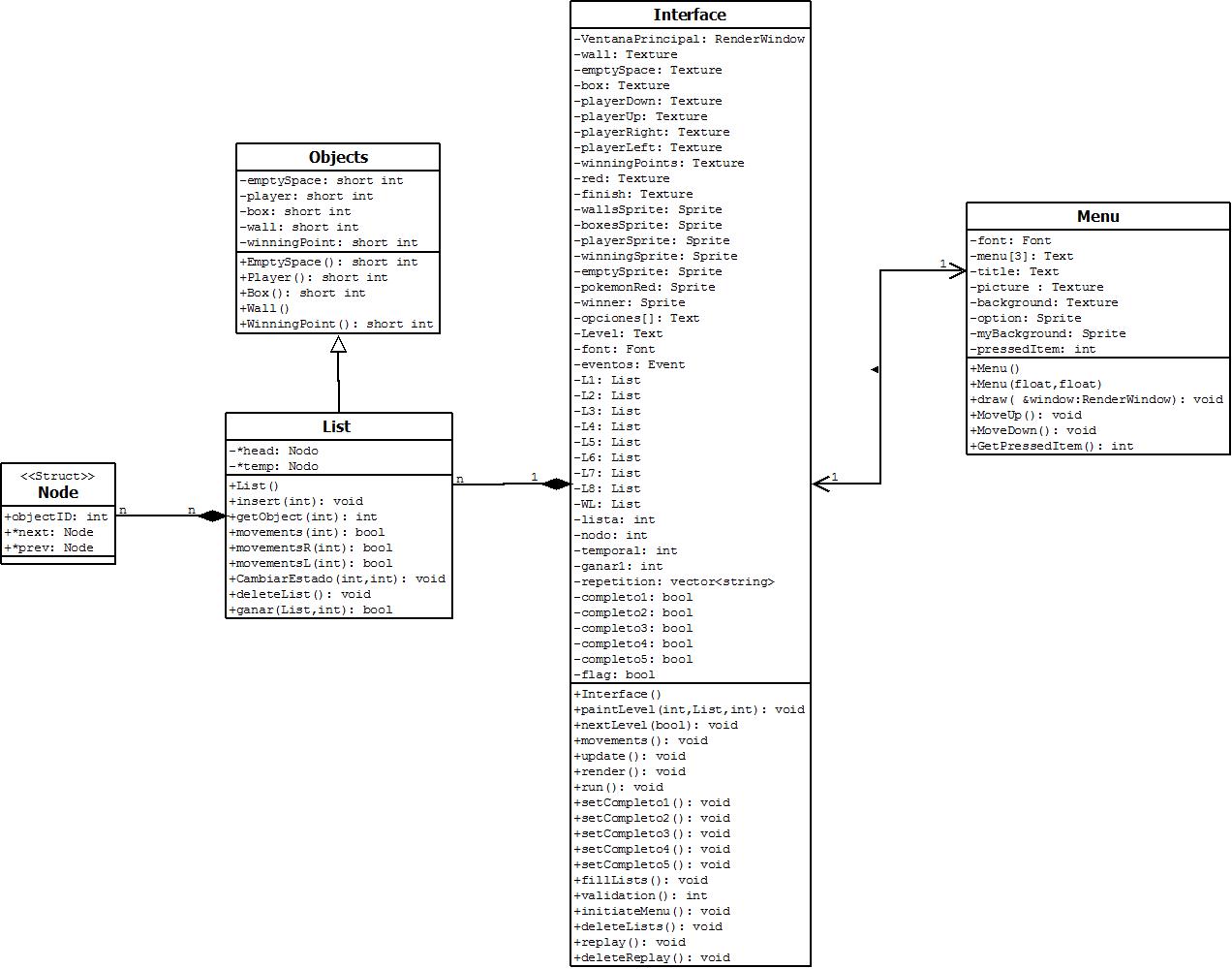
Ejemplo de mapa cargado desde el archivo de texto.

**

*0 = Espacio vacío  
1 = Personaje  
2 = Caja   
3 = Pared  
4 = Punto de gane*

**

Tipo de fuente para los textos en el menú y distintas etiquetas en la API del programa

**UML**

**Conclusión**

**Comentarios.**

El programa fue desarrollado con 8 listas de 9 nodos cada una, concatenadas para reconocer el valor del nodo el cual se le fue asignado por la lectura del archivo de texto.

Se creó una clase Interface que es quién invoca todo lo relacionado con la interfaz gráfica e invoca al menú principal para ser ejecutado, está compuesta de listas para la revisión de eventos del teclado entre otras funciones del programa.

Una clase Objects que contiene los ids los cuales serán usuados en la clase List, por esta razón List hereda de ella y probará los valores para saber si lo que debe dibujar la lista en la clase es interfaz es una pared, una caja, etc.

**Referencias.**

El proyecto se realizó consultando distintas páginas, video tutoriales, según lo requería el problema en cuestión.

Para SFML principalmente canales de youtube, con tutoriales de uso de sprites, movimiento, reconocimiento de teclado y por supuesto, la documentación propia de la librería SFML. Principales fuentes:

*Sonar Systems https://www.youtube.com/user/sonarsystemslimited  
Hilze Vonck https://www.youtube.com/channel/UC8C7ncaMYnXyu-pRU0S9FLg  
Documentation of SFML 2.0 https://www.sfml-dev.org/documentation/2.0/index.php*

Para evacuar dudas con listas se investigó según la tarea requerida en el momento en foros de StackOverflow y también de uno de los libros sugeridos en la bibliografía del curso.

(Joyanes L) 2015. “*Programación en C, Metodología, Algoritmos y estructura de datos”.*

*https://stackoverflow.com/*