MANUAL TÉCNICO
ARKANOID

Nelson Alexander Évora Díaz, 00066819

Gerson Javier Quintanilla Sánchez, 00254719

José Armando Argueta Portillo, 00129619

INDICE

ASPECTOS GENERALES

OBJETIVO DEL DOCUMENTO

El objetivo de este documento es el de explicar el diseño del software del juego Arkanoid, mostrando a detalle las herramientas utilizadas y a utilizar en futuras versiones y para mayor comprensión del funcionamiento.

DESCRIPCIÓN GENERAL

Para el desarrollo del software se implementó el uso del Modelo Vista Controlador, sus siglas MVC. Se desarrolló el clásico juego Arkanoid, el cual consiste en romper todos los bloques mostrados con la ayuda de una pelota y una barra de rebote, con un máximo de 3 vidas las cuales se restan cuando el jugador deja que la pelota toque el borde inferior de la pantalla. Al eliminar todos los bloques se sube el resultado del puntaje final de los jugadores a una base de datos con el objetivo de compararlos entre sí y mostrar los diez mejores.

SOFTWARE UTILIZADO

Para la creación del juego se utilizó Visual Studio Community 2019, junto al gestor de bases de datos PostrgreSQL 12 que se administró con pgAdmin4 para la creación de la base de datos designada para el juego. El complemento adicional para Visual Studio utilizado ha sido NuGet Npgsql para la conexión a la base de datos.

MODELOS UTILIZADOS

UML DIAGRAMA DE CLASES

El diseño de nuestro juego y el de la base de datos están basados en los siguientes diagramas:

- Diagrama de clases (Figura 1).
- Diagrama Entidad-Relación extendido (Figura 2).
- Diagrama relacional (Figura 3).

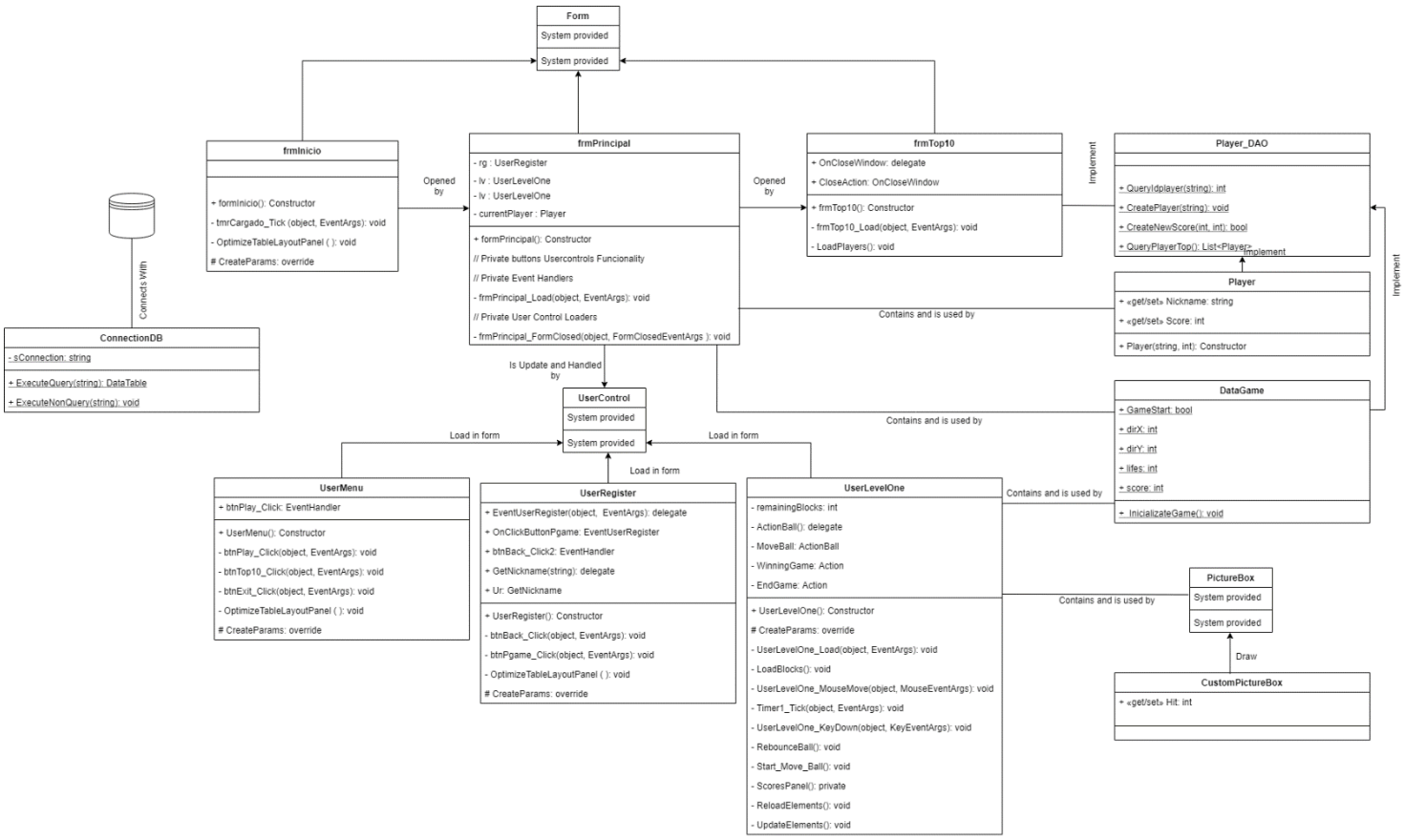

Figura 1 (clic para abrir)

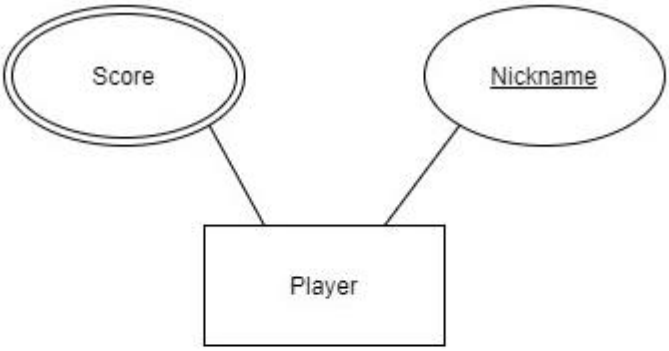

Figura 2

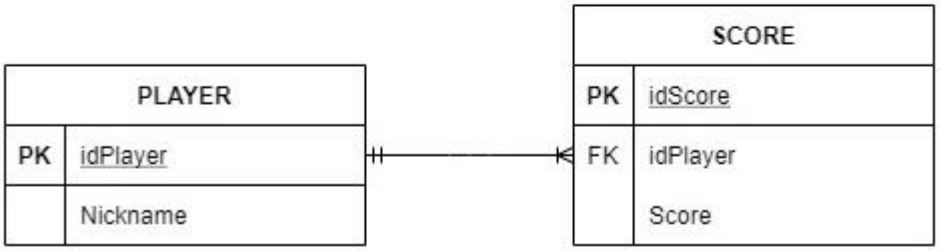

Figura 3

En el diagrama presentado en la Figura 1 se pueden apreciar las diferentes clases, Forms, User controls y entre otros elementos gráficos utilizados en el desarrollo del código del juego.

En la Figura 2 y Figura 3 se presenta la forma en la que se relacionan las entidades tanto en las clases del código del juego como en la base de datos.

CONCEPTOS TÉCNICOS

IMPLEMENTACIÓN DE INTERFAZ GRÁFICA

Los elementos de interfaz gráfica empleados en el juego son:

Forms

1. frmInicio.
2. frmPrincipal.
3. frmTop10.

User Controls

1. UserMenu.
2. UserRegister.
3. UserLevelOne.

Otros controles que se implementan internamente en los forms y User controls son:

1. Labels
2. tableLayoutPanels
3. Panels
4. PictureBox
5. RichTextBox
6. Buttons
7. ProgressBar

El form frmInicio es una pantalla de carga que implementa un timer el cual al completarse muestra al form frmPrincipal el cual es el encargado de albergar a todos los user controls que se ocupan en el programa (UserMenu, UserLevelOne y UserRegister). En primera instancia se carga el user control UserMenu en el que se cargan los botones:

1. btnPlay: es el encargado de llamar y mostrar el user control UserRegister.
2. btnTop10: es el encargado de llamar y mostrar el frmTop10
3. btnExit: su función es cerrar el programa completamente.

El user control UserRegister presenta al jugador un espacio creado con un RichTextBox en donde puede ingresar su nombre de jugador y a través de los Event Handlers de frmPrincipal se ejecuta una consulta a la base de datos establecida previamente en la clase Player_DAO la cual mediante una bandera verifica si existe un registro del jugador y en caso contrario registra al jugador para después cargar y mostrar en el frmPrincipal el user control UserLevelOne además de cargar las sentencias de código asignadas a los Actions: WiningGame y EndGame.

El UserLevelOne inicialmente carga los controles gráficos para el diseño del nivel del juego el cual se compone de una matriz de PictureBox de [10, 5] que son los bloques y dos PictureBox más, uno para la pelota y el otro para la barra de rebote que controla el jugador.

El juego inicia en el momento que el jugador presiona espacio en el teclado y la pelota inicia a moverse y rebotar en los bordes de la pantalla y en los bloques. La barra de rebote del jugador se controla mediante el movimiento del mouse, consecuentemente con todos los controles del UserLevelOne entra en función el concepto del juego anteriormente explicado.

En caso de que el jugador pierde se invoca el action EndeGame el cual muestra un mensaje al jugador indicando que ha perdido y remueve el UserLevelOne y vuelve a mostrar el UserMenu. En caso contrario de que el jugador elimina todos los bloques y gana se invoca el action WiningGame que muestra un mensaje al jugador indicándole que ha ganado y luego procede a ejecutar los métodos necesarios establecidos en Player_DAO para guardar el registro del puntaje del jugador y luego remover el UserLevelOne y vuelve a mostrar el UserMenu.

El frmTop10 se presenta en pantalla cuando el jugador da clic al btnTop10 en UserMenu, dicho form como su nombre lo describe muestra una lista de los diez más altos puntajes registrados los cuales se consultan a través de uno de los métodos de consulta definido en Player_DAO.

MANEJO DE CLASES EN MODELO Y CONTROLADOR

Las clases implementadas en el juego son:

1. Player
2. Player_DAO
3. ConnectionDB
4. DataGame
5. CustomPictureBox

La clase Player es una entidad que permite la creación de objetos de tipo Player que eventualmente representan al jugador.

La clase Player_DAO es una clase estática en la se definen los métodos necesarios para realizar la consulta de los jugadores y sus puntajes, guardar un nuevo registro del puntaje de un jugador y consultar la existencia de un jugador indicando a través de una bandera, entre otras consultas y acciones que se ejecutan mediante la conexión a la base de datos del juego, dicha conexión es tarea de definida de la clase estática ConnectionDB la cual consiste en la creación de una cadena de conexión y mediante las librerías de NuGet NpgSql poder implementar los métodos para ejecutar consulta y para ejecutar comandos SQL .

DataGame es una clase estática que brinda sus atributos como datos para la ejecución del nivel del juego y algunos eventos.

CustomPictureBox es una clase estática que contiene los atributos necesarios para que el UserLevelOne cargue y dibuje la matriz de PictureBox que forman el diseño del nivel con respecto a los bloques.

PLATAFORMA BASE

Sistema operativo: multiplataforma

Tecnologías: Visual Studio Community 2019

Lenguaje: C#

Gestor de Base de Datos: PostgreSQL

NOMENCLATURAS

ABREVIATURAS

Para los elementos del entorno gráfico se implementa la siguiente normativa de nombramiento:

<Abreviatura de tipo>_<Nombre Descriptivo>

Las abreviaturas se presentan a continuación:

pgb: ProgressBar

lbl: Label

tmr: Timer

rtb: RichTextBox

btn: Button

pic: PictureBox

EVENTOS Y EXCEPCIONES

EVENTOS

Para poder llevar a cabo la ejecución de la mayor parte de la funcionalidad del juego es necesario definir los siguientes eventos:

Eventos del frmInicio

- tmrCargado_Tick: se encarga de llenar la barra de cargado al inicio del programa.

Eventos del frmPrincipal

- frmPrincipal_Load: se encarga de cargar el user control UserMenu para interactuar con él y realizar una de las tres opciones que se muestran en éste.

- frmPrincipal_FormClosed: este evento se encarga de terminar la solución al momento de cerrar el frmPrincipal.

Eventos del frmTop10

- frmTop10_Load: se encarga de solicitar a la base de datos los 10 mejores puntajes y mostrarlos al jugador.

Eventos de UserMenu

- btnPlay_Click: llama y muestra a UserRegister.
- btnTop10_Click: llama y muestra a frmTop10.
- btnExit_Click: cierra la solución.

Eventos del UserRegister

- btnBack_Clcik: se encarga de regresar al UserMenu, escondiendo el user control.
- btnPgame_Click: revisa si el campo donde se ingresa el nombre está vacío, sino busca en la base de datos si existe el jugador o no (si existe logea, sino lo registra). Después pasa al UserLevelOne.

Eventos del UserLevelOne:

- UserLevelOne_Load: carga todos los bloques, la pelota y la barra de rebote.
- Timer1_Tick: se encarga de controlar el movimiento de la pelota
- UserLevelOne_KeyDown: espera a que el usuario aprete la tecla espacio para activar una bandera que indica el estado del juego (si está iniciado o no).

EXCEPCIONES

Las excepciones reguladas en el código del juego son:

1. EmptyNickNameException: es la encargada de notificar si en el UserRegister el usuario haya ingresado el nickname.

2. ExceededMaxCharactersException: es la encargada de notificar si en el UserRegister el nickname no pase de 15 caracteres.

3. NoRemainingLifesException: se encarga de notificar que ya se han acabado las 3 vidas brindadas en el juego.

4. OutOfBoundsException: se encarga de notificar si por alguna razón la pelota sale de los límites del campo de juego.

5. WrongKeyPressedException: se encarga de notificar al usuario que no ha apretado la tecla espacio, la cual es la encargada de iniciar el juego.

POSIBLES ERRORES DEL JUEGO.

Problema de Consumo: Al correr el juego, en el momento en que se registra el jugador empieza a consumir una cantidad excesiva de RAM (1 GB) que al perder o ganar y luego remover el UserLevelOne no disminuye debería.

Bugs de Impacto: En ocasiones cuando la pelota impacta con algún bloque o con la barra (picPlayerBar) en alguna de las esquinas esto genera movimientos o rebotes muy inusuales los cuales pueden llegar causar un rebote hacia abajo haciéndote perder una vida o el juego (al tener solo 1 vida) lo cual ocurre en ocasiones muy puntuales (Que la barra intercepte a la pelota por la mitad).