
Manual técnico

Arkanoïd

Autores:

- Raúl Ernesto Guillén Hernández, 00012119
- Maria Jose Jimenez Palacios, 00018619
- Emely Paola Barahona Lazo, 00098619

CONTENIDO

Manual técnico Arkanoid

ASPECTOS GENERALES	3
Objetivo del documento	3
Descripción General	3
Software Utilizado	3
MODELOS UTILIZADOS	3
UML diagrama de clases	3
Diagrama Entidad Relación	5
Diagrama relacional de la Base de datos utilizada.	5
CONCEPTOS BASE	6
Implementación de interfaz gráfica	6
Manejo de Clases en Controller	6
Plataforma Base	7
NOMENCLATURAS	7
Abreviaturas	7
EVENTOS Y EXCEPCIONES	8
Eventos	8
Excepciones	9

ASPECTOS GENERALES

Objetivo del documento

El objetivo de este documento es orientar y explicar el diseño del software creado, explicando las herramientas utilizadas.

Descripción General

Para la creación del software se utilizó el MVC o Modelo-Vista-Controlador. El programa consiste en una recreación del juego clásico Arkanoid.

Software Utilizado

Para la creación del software se hizo uso de JetBrains Rider 2019.3, en conjunto con PostgreSQL 12 para la creación de la Base de Datos. También se hizo uso de herramientas adicionales como Npgsql para la conexión a la base de datos y LiveCharts para gráficos.

MODELOS UTILIZADOS

UML diagrama de clases

El diseño del código está basado en el siguiente diagrama de clases.

Para visualizarlo mejor:

<https://drive.google.com/file/d/1RZHRaSIY9tzglfS-lvurR5-qhdJXnuub/view?usp=sharing>

Diagrama Entidad Relación

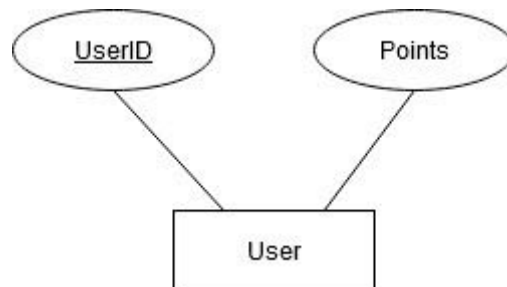


Diagrama relacional de la Base de datos utilizada.

Gráficamente la base de datos utilizada se ve de la siguiente manera:

User	
PK	<u>UserID</u>
	Points

Se hizo uso de una única tabla donde el nickname del Usuario es el UserID, por lo que es único. Además el puntaje de los jugadores es acumulativo.

CONCEPTOS BASE

Implementación de interfaz gráfica

La interfaz gráfica del programa consiste en dos ventanas o formularios, y dos controles de usuarios(UserControl). Una de las ventanas, llamada FrmMain utilizada como menú principal con diferentes botones y un panel en el que se cargar los dos controles de usuarios. La otra ventana ventana, llamada FrmGame formada con todos los materiales? necesarios para que el juego se realice normalmente, además de un panel superior en el que se muestra las vidas y el puntaje del jugador. Los controles de usuarios existentes son:

Play.cs

Score.cs

Manejo de Clases en modelo

Para manejar lo fundamental del modelo del programa se cuenta con las siguientes clases:

User.cs

Pic.cs

GameData.cs

Manejo de Clases en Controller

Connection.cs

ControllerGame.cs

Plataforma Base

Sistema operativo: Multiplataforma.

Tecnologías: JetBrains Rider 2019.3

Lenguaje: C#

Gestor de BD: PostgreSQL 12

NOMENCLATURAS

Abreviaturas

Para una mejor organización en el entorno gráfico del programa se utilizó las siguientes normativas de nombramiento:

<Abreviatura de tipo>Descripción

Las abreviaturas utilizadas:

Label	lbl
Panel	pnl
TextBox	txt
PictureBox	pic
Button	btn
DateTimePicker	dtp

EVENTOS Y EXCEPCIONES

Eventos

Load Event: Carga todos los recursos del controlador.

- **Score_Load:** Carga la gráfica para el Top de jugadores.
- **Game_Load:** Carga todos los controladores de la ventana encargada de ejecutar el juego.
- **Play_Load:** Al ejecutarse el UserControl de Play se carga una pequeña ventana con un mensaje que te informa que para entrar al juego introduzcas un nombre ya registrado previamente O crear un nuevo usuario.

Button.Click Event: Registra los clicks sobre los botones.

- **BttnClosed_Click:** Se ejecuta cuando presiona el botón “Salir” para cerrar la aplicación.
- **BttnPoint_Click:** Se ejecuta cuando se presiona el botón “Puntaje” y muestra el top de jugadores.
- **BttnPlayy_Click:** Se ejecuta cuando se presiona el botón “Jugar” y inicia el juego.

From.FormClosing Event: Ocurre antes de que se cierre la ventana.

- **FrmGame_FormClosing:** Permite cerrar y detener la ejecución del programa al dar click en el botón x que se encuentra en la esquina superior derecha, si está en juego pregunta si quiere guardar su proceso o no.
- **FrmMain_FormClosing:** Detiene la ejecución del programa.

Control.KeyDown Event: Se produce cuando se presiona una tecla.

- **Game_KeyDown:** Permite poder comenzar el juego a partir de presionar la tecla de espacio.

- **frmMain_KeyDown:** Permite poder regresar a la ventana menú si se está previamente en el User del Top 10.

Time.Tick Event: Se produce cuando ha transcurrido el intervalo de temporizador especificado y el temporizador está habilitado.

- **TimerForMovements_tick:** Permite lograr hacer posible que la pelota pueda moverse actualizando la ventana cada 10 milisegundos.

MouseMove Event: Permite controlar el movimiento de un puntero en 2 dimensiones en una interfaz gráfica.

- **Game_MouseMove:** logramos hacer que el jugador pueda mover la plataforma para pegarle a la pelota con el mouse

Excepciones

- **UserException.cs**

Usada para verificar la existencia de un jugador

- **NoAccounts.cs**

Por no existen cuentas registradas.

- **NameLength.cs**

Por si se excede la cantidad de caracteres en el NickName.

- **GameClosedInProcess.cs**

Por si se cierra el juego con una partida en proceso.

- **TopTenException.cs**

Por si hay menos de 10 players se carga un top de menos de 10.