

ARKANOID

C#

MANUAL TECNICO

BY

HERSON ALEXANDER REYES RIVAS 00216119
ROBERTO CARLOS CABEZAS MORENO 00087619
MAURICIO JOSE ZALDAÑA FERNANDEZ 00155619



The recreation of the classic Arkanoid game

CONTENIDO:

Aspectos generales.....3

- Objetivo del documento
- Descripción general
- Subtitulo software utilizado

Modelos utilizados.....4

- UML diagrama de clases
- Diagrama Relacional normalizado de la base de datos utilizada

Conceptos técnicos.....8

- Implementación de interfaz grafica
- Manejo de clases en modelo
- Plataforma base

Nomenclaturas.....9

- abreviaturas

Excepciones.....10

A decorative graphic of pink and yellow smoke or dust clouds, with faint circular patterns, positioned behind the title.

Aspectos generales

Objetivo del documento


El objetivo de este documento es orientar y explicar el diseño del juego creado, explicando las herramientas utilizadas y a utilizar para futuras versiones.

Descripción general del proyecto

El programa consiste en la recreación del juego arkanoid. El juego consiste en destruir bloques mediante una bola que va rebotando, siendo el objetivo del jugador no dejar caer la bola, haciéndola rebotar en una plataforma, fue creado utilizando el modelo vista-controlador.

Software utilizado:

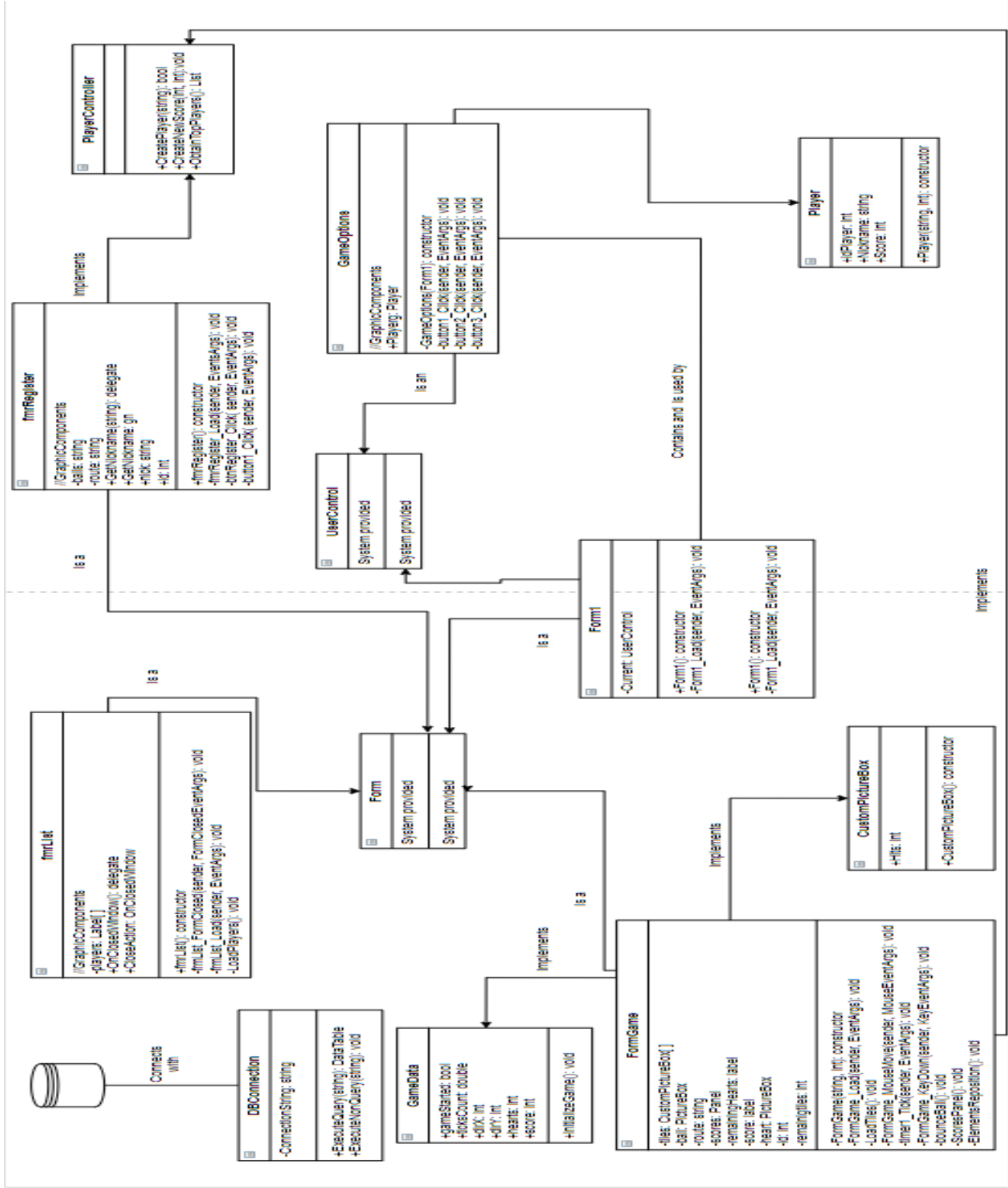
Para la recreación del juego se utilizó como IDE el software Rider, un entorno de desarrollo para el lenguaje de programación c # de la empresa JetBrains, una compañía dedicada a crear herramientas para desarrolladores y gerentes de proyectos, junto a PostgreSQL que se utilizó para la creación de la base de datos. Como complemento de Rider se utilizó Npgsql para establecer la conexión del programa con la base de datos, también fue utilizado el sistema de control de versiones Git.

The background of the slide is decorated with several 3x3 grids of colored squares. These grids are arranged in a way that they appear to be floating or falling. The colors include shades of green, yellow, orange, blue, and purple. The central text is overlaid on a light pink rectangular background.

Modelos utilizados

UML Diagrama de clases

El diseño del código del juego está basado en el diagrama de clases siguiente (ver siguiente página):



En el diagrama UML se muestran las distintas clases utilizadas en el programa, algunas de ellas como los form, tienen una interacción directa con el usuario, mientras que otras se encargan del funcionamiento interno del juego, sin interactuar con el usuario.

La clase DBConnection es una clase que no interactúa con el jugador, pero funciona como enlace entre el programa y la base de datos, contiene los métodos necesarios para poder realizar inserciones y consultas.

GameData como su nombre lo explica contiene los datos del juego, posiciones en x, posiciones en y, si el juego se ha iniciado o no, los corazones que representan las vidas del jugador, y otros datos que son utilizados en formGame, en formGame es donde se encuentra lo que el jugador puede ver al momento de jugar, los ladrillos, la plataforma, la pelota, sus vidas, es la ventana donde el juego se lleva a cabo con los datos obtenidos de GameData.

frmList es la clase donde se cargan 10 de los jugadores existentes en la base de datos, los cuales son los que poseen los mayores puntajes, posicionando en el primer lugar el que tenga la mayor cantidad de puntos al ganar.

El form1 es la clase donde se encuentran las opciones para que se pueda cambiar entre UserControl y donde se pueden abrir las demás ventanas, lo primero que se muestra es un UserControl que contiene los botones con las diferentes opciones, iniciar el juego, ver el top 10 de jugadores y salir, al dar click en jugar, se cambiara a una pantalla de registro llamada frmList, donde se agregara el nombre del jugador y habrá una lista de las distintas pelotas que se pueden utilizar, para lo cual solo se debe seleccionar una y cargarla por medio del botón, se validara que el nombre ya exista en la base de datos o no, auxiliándose de la clase PlayerController, en caso de existir al terminar de jugar se almacenara un nuevo puntaje para ese jugador, la clase player es la que funciona como molde para los registros de un jugador, lo que permite almacenar datos como el IdPlayer, Nickname y Score.



Diagrama Entidad - Relación extendido

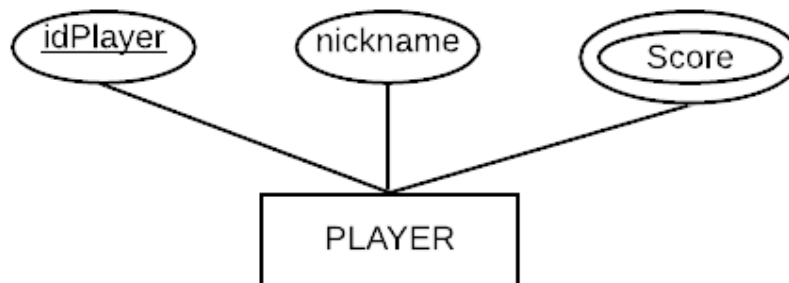
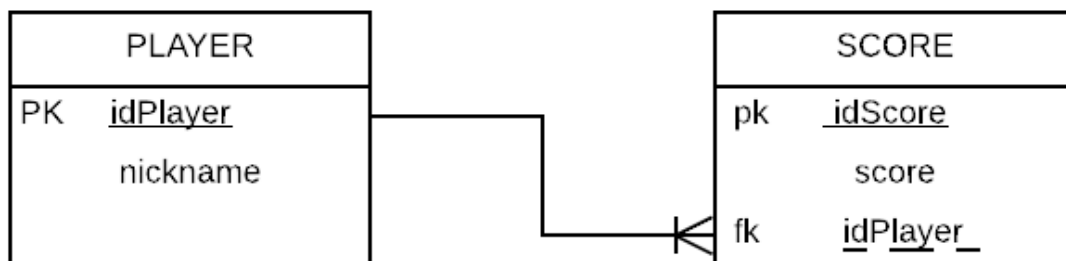


Diagrama relacional normalizado de base de datos utilizada



Conceptos técnicos

Implementación de interfaz grafica

La interfaz gráfica consiste en tres forms y un UserControl:

- El form1 es el menú principal donde se encuentran los botones para ir a los otros form, y la opción para salir del programa.
- El form fmrRegister consiste en una pantalla que permite al jugador registrarse, agregar un nombre de usuario, el cual irá a una base de datos. En la misma pantalla puedes elegir la pelota con la que deseas jugar, dando a elegir al usuario entre varios diseños de bolas.
- El form FormGame es donde se ejecuta el juego, carga los bloques a destruir, la plataforma y la pelota.
- El UserControl es un top 10, donde se muestran los jugadores con los puntajes más altos.

Manejo de clases en modelo

Para manejar la parte fundamental del modelo del programa, se cuenta con las siguientes clases:

- DBconnection es la clase estática que sirve como intermediaria entre el software y la base de datos.
- GameData: Se encarga de iniciar el juego, conteniendo la información de este, como el puntaje, las vidas restantes y la velocidad del movimiento de la pelota.
- PlayerController : Se encarga de realizar distintas operaciones con los jugadores, agregar, guardar puntaje y listar.

Plataforma Base

Sistema Operativo	Windows 10
Tecnologías	Jetbrains Rider
Lenguaje	C#
Gestor de BD	PostgreSQL

Nomenclaturas

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

<Abreviatura de tipo>descripción

Las abreviaturas se presentan a continuación:

Label	lbl
Panel	pnl
TextBox	txt
PictureBox	pb
Button	btn

Excepciones

Constan de un constructor que recibe un string con el mensaje de error. Se cuentan con las siguientes:

- `EmptyNicknameException.cs` : Se lanza cuando el campo para un nombre se encuentra vacío.
- `ExceededMaxCharactersException.cs`: Se lanza cuando la cantidad de caracteres del nombre de usuario es mayor a 15.
- `UnchargedBallException.cs`: Se lanza cuando el usuario quiere jugar sin antes haber hecho en el click de cargar bola.
- `OutOfBoundsException.cs`: Se lanza cuando la pelota sale de la ventana del juego
- `WrongKeyPressedException.cs`: El juego se inicia con la tecla Space, si se presiona una tecla diferente, se lanza esta excepción.