Manual Técnico

Arkanoid, recreación

María José Mendoza Villicaña, 00198319

Josue David Hurtado Argueta, 00156919

Ruben Alexander Rivera Miron,00029916

Junio del 2020

Contenido...

Aspectos Generales

Objetivo del documento

El objetivo del presente documento es explicar el desarrollo y diseño del juego. Se busca lograr dicho objetivo explicando las herramientas utilizadas para la realización del programa, de igual manera mostramos los recursos utilizados para el óptimo desarrollo del proyecto.

Descripción General

El juego desarrollado tiene como nombre “Arkanoid”. Es considerado un juego de tipo arcade que consiste en una pequeña plataforma o player que hace su recorrido de forma horizontal; esta plataforma impide que una bola, que aparece en la parte superior del player, salga de la zona de juego, haciéndola rebotar en las “paredes”. En la parte superior de la pantalla se encuentran bloques ordenados de forma uniforme, estos desaparecen al ser tocados por la bola, cada bloque tiene un dinamismo diferente. El objetivo del juego es evitar que esta bola caiga y, además, eliminar todos los bloques. Para la realización del programa se hizo uso del modelo vista controlador, sus siglas MVC.

Software utilizado

Para la creación del juego se utilizó JetBrains Rider 2019 , para hacer la creación de base de datos se utilizó PostgreSQL 10. Adicional se utilizó una herramienta llamada trello donde se organizaron las diferentes ideas de los integrante del grupo. Además se ocupó la aplicación TeamViewer para terminar el proyecto .

Modelos Utilizados

UML diagrama de clases:

El diseño arquitectónico del código presente en el juego de “Arkanoid” está basado en el diagrama de clases presentado a continuación

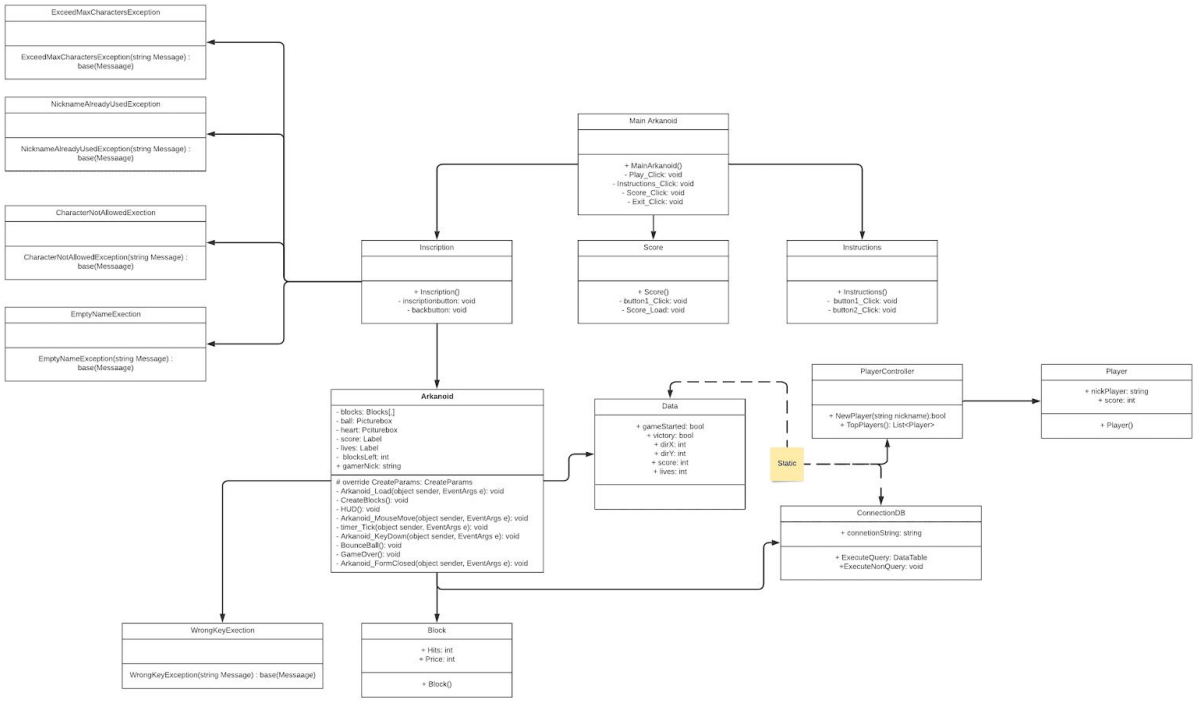

Figura 1. UML de clases

Diagrama Relacional Normalizado de bases de datos utilizada:

A continuación, presentamos el diagrama de relación de la base de datos utilizado. La primera tabla contiene el nombre de player que incluye el nickname de los usuarios que juegan la aplicación, de igual manera contiene score del puntaje de los usuarios. La segunda tabla contiene score final de los jugadores, posteriormente muestra el nickname del usuario y su puntaje.

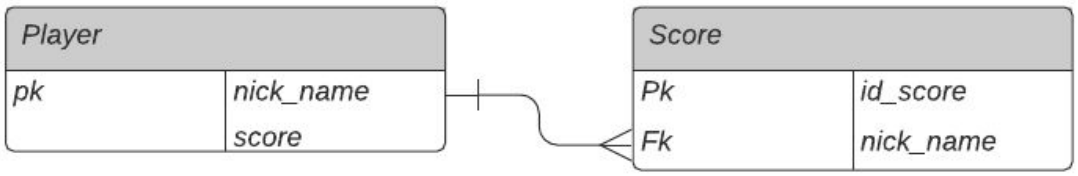

Figura 2 . Diagrama UML de la base de datos

Conceptos técnicos

Implementación de gráfica de interfaz

El juego cuenta con varias forms que son las pantallas que se le muestran al usuario. La forms van variando según la necesidad del juego.

La primera form que mostramos es el menú principal del juego, donde se le dan varias opciones al usuario en un conjunto de botones que presentan las siguientes opciones: Play, Instructions, Score, Exit.

Figura 3. Fotografía del Menú principal del juego, donde se le muestra al usuario las opciones que puede seleccionar.

Cada botón nos muestra un nuevo form, excepto el de Exit, que es utilizado para salir del juego, mostrando, antes, un MessageBox con el mensaje “Vuelva pronto :D” . Al momento de seleccionar el botón deseado se abre la nueva form que da paso a la acción solicitada. Cuando el usuario selecciona el botón de Play se muestra una ventana de inscripción donde el usuario

tiene que ingresar el nickname o el nombre con el cual desea jugar, con ello será registrado para poder almacenar su **score** en la base de datos, para así tener un registro de los puntajes . Una vez inscrito el usuario comienza el juego.

Figura 4 . Fotografía de la inscripción que se le pide al usuario antes de almacenarla en la base de datos

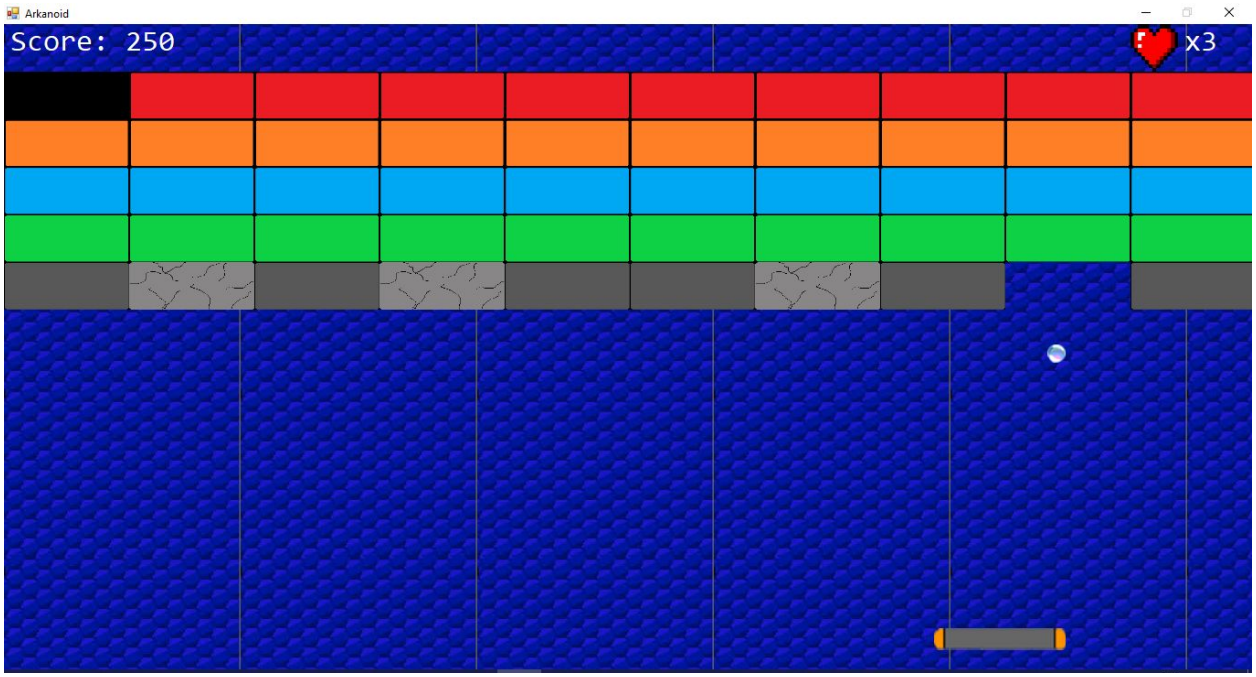

Figura 5. Fotografía del juego ya comenzado, justo después de la inscripción

Cuando el usuario selecciona la opción de Instrucciones se le abre una nueva form donde se explican los controles y el objetivo del juego, esto con el fin de poder hacer más comprensible la forma en la que trabaja el programa.

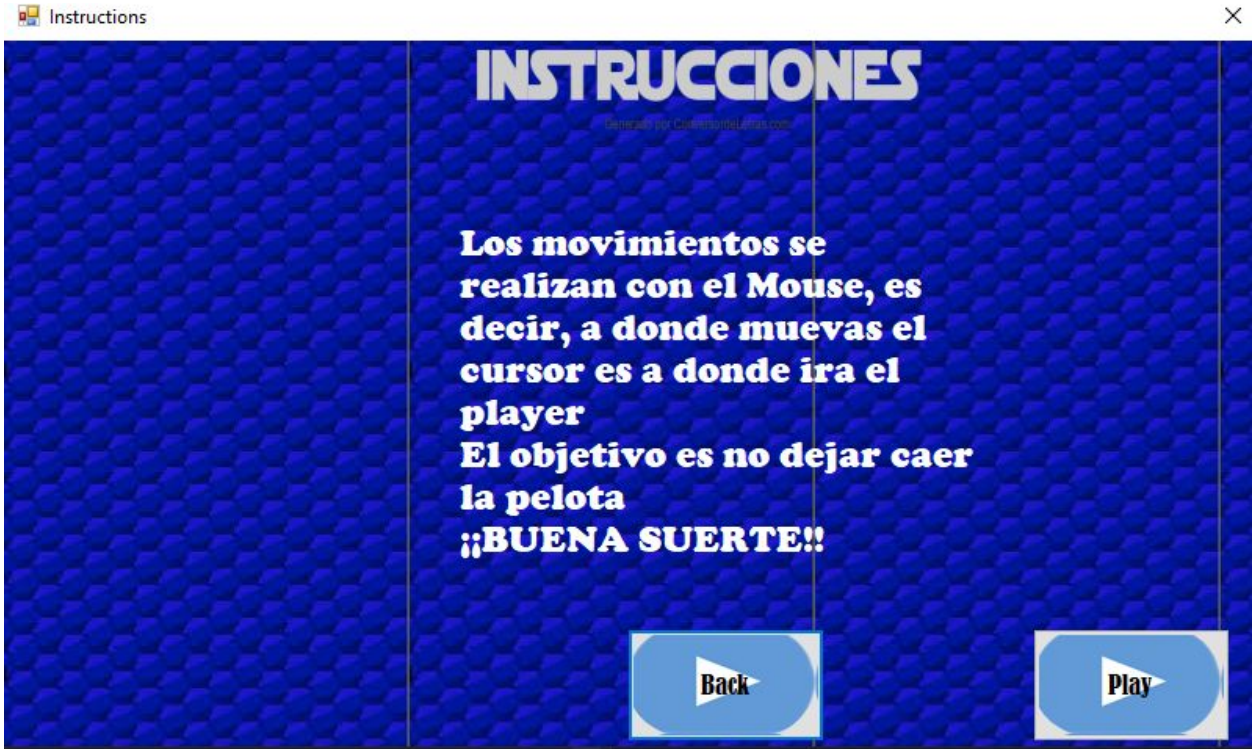

Figura 6. Imagen que se le muestra al usuario donde se le explica las instrucciones del juego, así como el objetivo de este.

Por último cuando el usuario presiona el botón de score se le muestra una pantalla o form nueva donde se observa un datagrid para que pueda visualizar los datos almacenados en la base de datos ya con los scores de cada usuario. Es importante mencionar que únicamente se muestra el "TOP 10 de jugadores", es decir, las 10 **Scores** de valor más alto almacenadas en la base de datos.

Figura 7. Ventana de Score que muestra a los usuarios los datos de la base de datos

Manejo de clases en modelo

Modelo:

- Blocks: Almacena los atributos de cada bloque creado en el juego.
- Data: Contiene la información de una partida.
- Player: Contiene la información del jugador

Controlador:

- ConnectionDB: Es la conección a la base de datos central (base de datos local).
- PlayerControler: Obtiene datos de los jugadores desde la base de datos

Plataforma Base

Plataforma	Multiplataforma
Tecnología	JetBrains Rider 2019.3.4
Lenguaje	C#
Gestor de BD	PostgreSQL
Diagramas	Lucidchart

Nomenclaturas

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

<Abreviatura de tipo>_ descripción _<id correlativo>

Las abreviaturas se presentan a continuación:

Label	*lbl*
Picture Box	*pic*
Text Box	*txt*
Datagrid	*dgv*
Timer	*tmr*
Button	*bttn*
Head up display	*hud*

Eventos y excepciones

Excepciones

CharacterNotAllowedException: No permite que en el nombre de usuario se escriba “ñ”
ConnectionDB: Hace la conección a la base de datos donde se guarda tanto el nombre de usuario como su puntaje
EmptyNameException: No permite dejar vacío el campo de inscripción al momento de jugar
ExceedMaxCharacterException: No permite que excedas la cantidad de caracteres en un nombre de usuario
NicknameAlreadyUsedException: No permite que repitas el nombre de otro usuario
PlayerController: Controlador del player, revisa si ya existe, extra al Top 10 de los jugadores
WrongKeyException: Cuando el usuario no presiona la tecla enter para comenzar. Indica el error

Eventos

Mouse Move: Mueve el mouse
Load: Carga la pantalla
Timer_tick: Cuando sucede un “tick” en el timer
Button_click: Carga una ventana, dependiendo del botón.
PressKey: Detecta la tecla que se presiona
Form_closed: Cierra la ventana actual

Enlaces

1. https://drive.google.com/file/d/1S8yLMKEFbsg9ggnzKjPcJsKibETG_x2X/view?usp=sharing, UML de clases
2. https://drive.google.com/file/d/1aaV9trP4c3wrhbJ68MuBhGn7AjLsJpgw/view?usp=sharing, UML normalizado
3. https://drive.google.com/file/d/1lw-a7JHzK18X0EYTgsetHnQkvxuV1yft/view?usp=sharing, documento de Manual Técnico
4. https://drive.google.com/file/d/1b_9dPj_BkYXPZnHqYUafOJgKmE4AF4gu/view?usp=sharing, Script de la base de datos

Créditos

1. ***Documento, Manual técnico y código:*** Josue David Hurtado Argueta, Rubén Alexander Rivera Mirón, María José Mendoza Villicaña.
2. ***Bloque y Player:*** https://blogdeirenee.wordpress.com/2018/02/18/scratch/
3. ***Bloque editado por:*** Carlos López
4. ***Fondo:*** http://retoscratch.weebly.com/nivel-vii.html
5. ***Corazón:*** https://opengameart.org/content/heart-pixel-art
6. ***Título:*** https://www.pngwing.com/en/free-png-nmcpe
7. ***Pelota:*** https://xtremeretro.com/arkanoid-parte-2/