

Manual Técnico

Arkanoid Project



Realizado por:

- Julio Josué Chávez Flores 00049419
- Daniel Alberto Rodríguez Méndez 00035919
- Dennys Alberto Rivera Pascasio 00026919
- Juan Daniel Escobar Rivera 00138218
- Rafael Humberto Figueroa Acevedo 00118519

Fecha de entrega: Martes 23 de junio de 2020

Contenido del Manual Técnico

❖ Aspectos generales	
❖ Objetivo del documento	3
❖ Descripción general	3
❖ Software utilizado	3
❖ Modelos utilizados	4
❖ Diagrama de clases UML	5
❖ Diagrama relacional normalizado de base de datos implementada	7
❖ Conceptos técnicos	8
❖ Implementación de interfaz gráfica	9
❖ Manejo de clases modelo	10
❖ Plataforma base	10
❖ Nomenclaturas	11
❖ Abreviaturas	11
❖ Eventos y Excepciones	12
❖ Eventos	12
❖ Excepciones	13

Aspectos generales

★Objetivo del documento

El objetivo del presente documento es explicar brevemente el diseño de software llevado a cabo, dando a explicar qué herramientas se utilizaron al momento de organizar el proyecto.

★Descripción general

Para crear este programa se llevó de base el famoso juego “Arkanoid” el cual fue lanzado en muchas plataformas de la época. El programa muestra funciona como medio de entretenimiento y competitividad entre varios usuarios, al manejar un registro por medio de puntuación en lista de las diez puntuaciones más altas.

★Software utilizado

Para la creación del programa se ha hecho uso de dos programas distintos con lenguaje C# los cuales son Visual Studio Community 2019 y JetBrains Rider 2019, al mismo tiempo, haciendo uso para la creación de la base de datos Heroku. De forma adicional, se ha usado para implementar gráficamente Windows Forms.

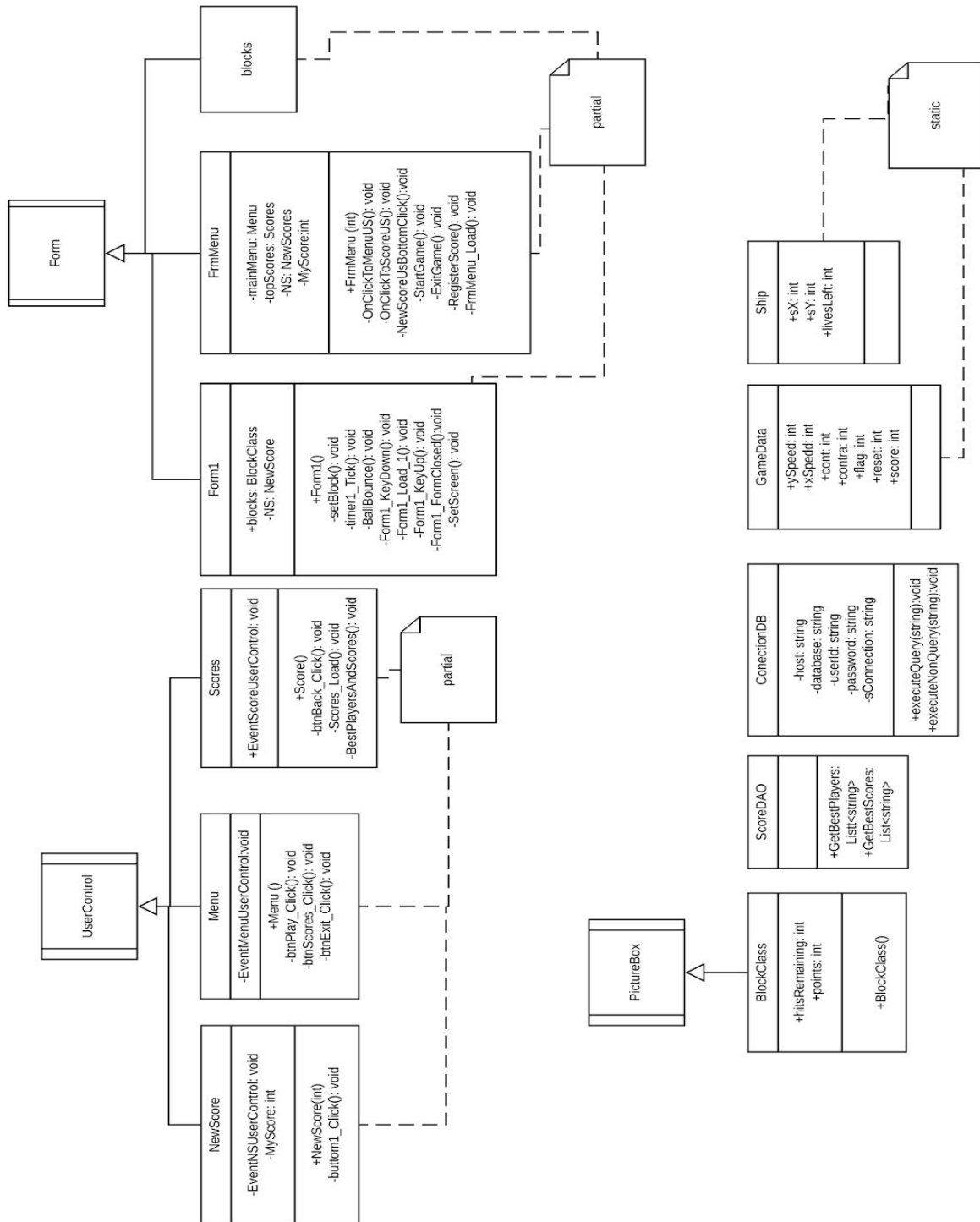
Modelos utilizados

★Diagramas de clases UML

El diseño que se ha seguido para estructurar cada porción del programa por medio del siguiente diagrama de clases (ejemplificado en la siguiente página):

Se deja de adjuntado el url para vista más detallada del diagrama de clases:

https://github.com/Danii920/Co-op_Mode_Arkanoid/blob/master/Proyecto%20Final%20POO/Documentacion/UML%20%2002%20.pdf



Explicando un poco todo lo que se presenta en el diagrama de clases:

Se ha implementado funciones con un timer, que le da funcionalidad al movimiento de la pelota, en movimiento horizontales y verticales de forma consecutiva, para los rebotes que ejerce en las esquinas de la pantalla, en los bloques y por último en la nave. Las presentes PictureBox funcionan como la base clave para todos los elementos dentro del juego, los cuales han sido utilizados para reemplazar el funcionamiento de los sprites, otorgando la posibilidad de poder movilizar un PictureBox alrededor del Form, sin una limitación estática dentro del mismo.

Los buttons sirven para el cambio de ventanas (Forms). Tenemos también implementado un sistema de organización por medio de la cantidad más alta de puntaje obtenido, presentando la inscripción de un nombre como propietario de ese puntaje obtenido, sin limitación, al usar el mismo nombre varias veces, ya que nos hemos basado en la versión arcade del juego Arkanoid.

Diagrama relacional normalizado de base de datos implementada

Agregando como dato adicional, se presenta el esquema con el que se generó el registro de puntaje y nombre del jugador:

PLAYERS
<u>nameplayer</u>
score

Conceptos técnicos

★ Implementación de interfaz grafica

La interfaz gráfica se basa en la implementación de tres ventanas globales, una de ellas conteniendo distintos botones que cargan faltantes dos ventanas globales y controles de usuarios (User Controls en inglés), cada control representan opciones que puede implementar el programa, los controles de usuario y ventanas globales que se implementan son:

- MenuUserControl.cs
- NewScoreUserControl.cs
- ScoresUserControl.cs

★Manejo de clases modelo

Para controlar gran parte del programa modelo, cuenta con las siguientes clases:

- **GameData.cs**
- **Program.cs**
- **Ship.cs**
- **ScoreDAO.cs**
- **ConnectionDB.cs**

★Plataforma Base

Sistema operativo	Windows 10
Tecnologías	Visual Studio Community 2019 y JetBrains Rider 2019
Lenguaje	C#
Gestor de DB	Heroku

Nomenclaturas

★Abreviaturas

Para los elementos en la interfaz gráfica se implementa la siguiente forma de nombramiento:

<Abreviatura de tipo>_descripción_<id correlativo>

Las abreviaturas se presentan a continuación:

Label	lbl
TableLayout	tlp
PictureBox	pic
Buttons	btttn
MessageBox	msb

Eventos y Excepciones

★Eventos

Para poder realizar ciertas acciones que se reflejan en los controles de usuario y las ventanas globales fueron necesarias las implementaciones de eventos. Los eventos implementados fueron:

- Ship.cs

Maneja el movimiento de la nave en direcciones derecha e izquierda, al igual que la hitbox que tiene la nave en el momento que la pelota toca la nave.

- BlockClass.cs

Implementa la hitbox de los bloques y la destrucción de los mismos en el momento que la pelota los toca, al igual que la dirección tomará la pelota cuando toque un bloque.

- Form1.cs

Maneja el movimiento constante de la pelota de forma horizontal y vertical.

★Excepciones

Las excepciones, cumplen como función, detectar un error, dar alerta de ella, arreglarla sin detener el funcionamiento del programa ni cerrarlo abruptamente. Las excepciones que han implementado son:

- FieldIsEmptyException.cs
- finishGameException.cs
- GetDataException.cs
- ScoreIsEmptyException.cs
- UserExistsException.cs