

Manual técnico Arkanoid

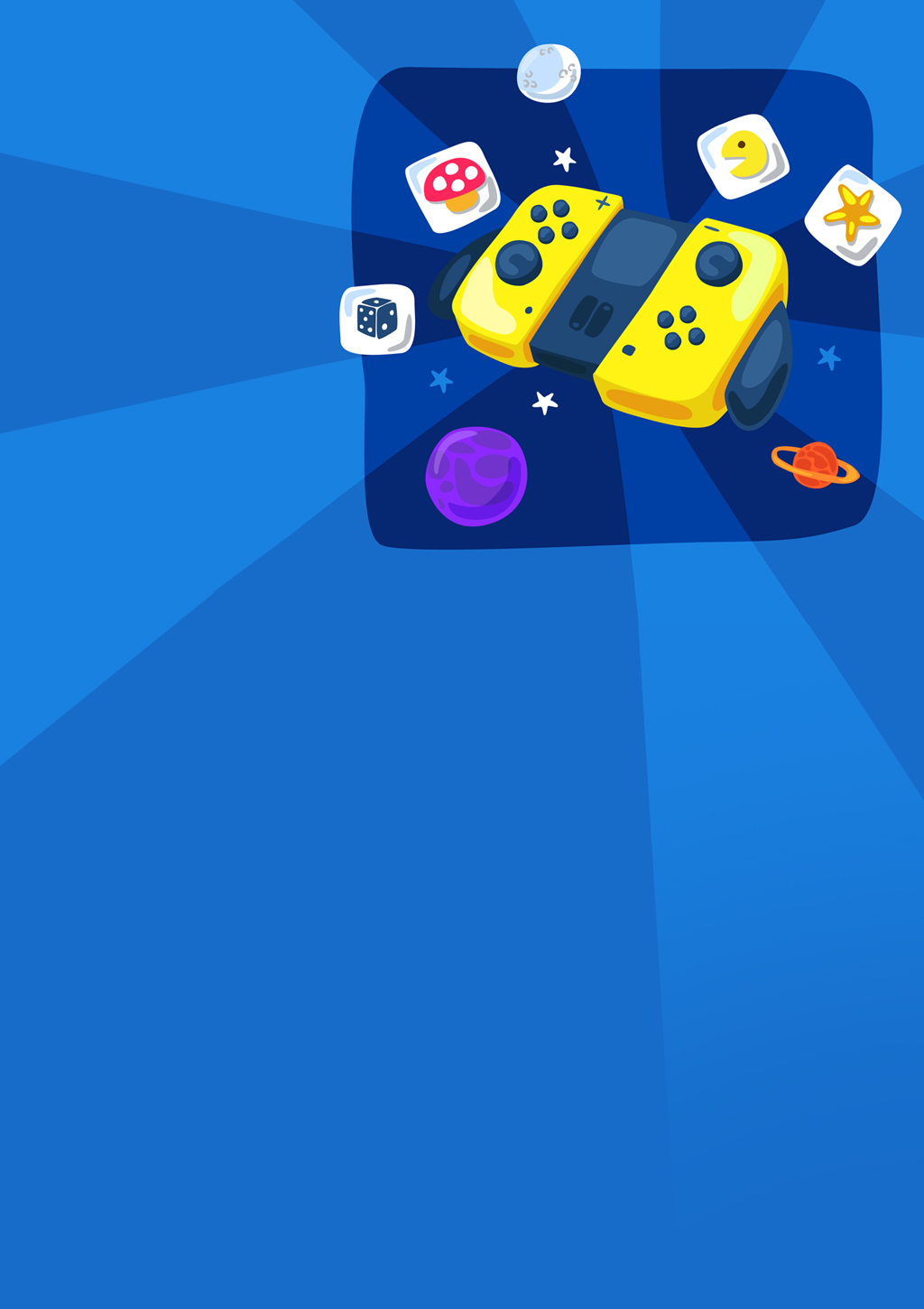
Realizado por:

Omar Rafael Flores Alas 00038619

Fernando Rene Cárcamo Aguilar 00084219

Jasson Alexander López Soriano 00303319

Geovany Javier Ardón Vega 00118618

Contenido

Manual técnico………………………………… 1

Aspectos generales………………………….. 3

Objetivos del documento……………. 3

Descripción general…………………….. 3

Software utilizado……………………….. 3

Conceptos técnicos…………………………… 4

Implementación de interfaz…………. 4

Distintos tipos de error………………… 5

Manejo de clases en modelo……….. 6

Plataforma base…………………………… 7

Nomenclatura………………………………. 7

Excepciones y eventos………………….. 8



Aspecto Generales

Objetivos del documento

El objetivo de este documento es la explicación del proyecto para la asignatura de Programación Orientada a Objetos, dando a conocer los diferentes aspectos impartidos en el proyecto.

Descripción general

El proyecto presente es la recreación de un juego clásico llamado Arkanoid que es un videojuego de árcade desarrollado por Taito en 1986.Que está basado en Breakout de Atari de 1976.

**Software utilizado**

**Para la creación del proyecto se utilizó el IDE desarrollado por JetBrains llamado Rider, en conjunto con pgAdmin 4 para la creación de la base de datos.**

Conceptos Técnicos

**Implementación de interfaz grafica**

**La interfaz gráfica del proyecto consta de varias ventanas, la primera muestra el menú del juego (menú.cs) donde se implementaron tres botones, el cual el usuario puede elegir entre tres botones los cuales son:**

* **BtnPlay (“Jugar”): Abre el form de Game.cs**
* **BtnScore (“Score”): Abre la ventana del Score**
* **BtnExit (“Exit”): El cual se implementa para salir del programa**

**Después que el usuario ejecute el BtnPlay el programa abre el form de Game.cs que consta de dos botones y un TextBox los cuales son:**

* **TxtName: El usuario deberá introducir su nombre que ya este almacenado en la base de datos**
* **BtnBack: Manda al usuario al form menú.cs**
* **BtnPlay: El cual inicializa el juego**

**Los controles del usuario son:**

* **Menu.cs**
* **Game.cs**
* **ScoreControl.cs**
* **ArkanoidControl.cs**
* **ScoreControl.cs**

Distintos Tipos de Errores

**Tipos de errores**

**Al momento de realizar el proyecto surgieron diferentes tipos posibles errores que se pudieran ocasionar al momento de ejecutar el programa los cuales son:**

* **Que el usario dejara vacío el Textbox**
* **Que el usario ponga un nombre mayor a 16 caracteres**
* **Al momento que el usuario se acaba sin vidas**
* **Presionar una letra diferente a Space cuando se inicia el juego**
* **La destrucción de todos los bloques**



**Manejo de clases en modelo**

**Para la realización de programa se ocuparon diferentes clases, se cuenta en el proyecto con las siguientes clases:**

* **ScoreModel.cs: Es el encargado de crear objetos de tipo Attemps**
* **GameData.cs: Verifica si el juego a iniciado, cuánto va incrementar el tick del timer, se encarga de direccionar el rebote de la pelota, establece las vidas que se pueden tener en el juego y el score al inicio del juego**
* **DBConnetion.cs: Se conecta a la base de datos local para poder hacer las consultas y acciones requeridas**
* **CustomPictureBox.cs: Se encarga de establecer los hits para tiles.**



**Plataforma base**

|  |  |
| --- | --- |
| Sistema Operativo | Multiplataforma |
| Tecnologias | JetBrains Rider |
| Lenguaje | C# |
| Gestor de DB | pgAdmin 4 |

**Nomenclaturas**

**Para la elaboración de los componentes gráficos se implementó la siguiente normativa de nombramiento:**

|  |  |
| --- | --- |
| Label | Label |
| Panel | Pnl |
| TextBox | Txt |
| PictureBox | Picture |
| Button | Btn |

 Excepciones y Eventos

**Excepciones**

**Para la elaboración del programa se tuvieron que hacer varias excepciones entre las cuales están:**

* **EmpyNicknameException: Esta excepción se ejecuta cuando el usuario no ingresa un nombre**
* **ExceededMaxCharactersException: Excepción ejecutada cuando el usuario pasa un máximo de 16 caracteres en su nombre**
* **NoRemainingLivesException: Esta excepción se ejecuta cuando el usuario se queda sin vidas**
* **OutOfBoundsException: Esta excepción se ejecuta cuando el usario se mueve fuera de los limites**
* **WrongKeyPressedException: Esta excepción se ejecuta cuando el usuario introduce una tecla que no corresponde en el programa**



**Eventos**

**Para la elaboración de programa se tuvieron que hacer varios eventos que el usario podía ejecutar, unos de estos son:**

* **BtnPlay\_Click: Evento que se ejecuta cuando el usuario da Click en el botón de jugar, mandando al usario a la siguiente ventana.**
* **BtnScore\_Click: Evento que es ejecutado cuando el usuario da Click al Score madandolo a la ventana de los puntajes.**
* **BtnExit\_Click: Evento que es ejecutado cuando el usuario da Click al botón del salir, haciendo que el programa finalice.**
* **BtnBack\_Click: Evento que se ejecuta cuando el usuario da Click al botón volver al menú, mandado al usario al menú principal.**
* **button1\_Click: Evento que es ejecutado desde la ventaja de los puntajes para volver al menú.**
* **GetNickName: Delegate para pasar el nombre de Form al ArkanoidControl**
* **Game\_MouseMove: Evento implementado al momento que el usario mueve su mouse en el juego.**
* **GameWon: Action que se ejecuta al momento en el que el usuario destruye todos lo bloques para guardar su puntaje.**
* **GameEnded: Action que se ejecuta al momento en que el usuario ha perdido sus 3 vidas y lo regresa al menú principal.**