



IDLE MASTER A LA CARTA

ÁLVARO MILLÁN MEJÍAS

ÍNDICE

1.- Introducción.....	2
2.- Tecnología Usada.....	2
3.- Explicación del Proyecto.....	3
4.- Utilidades del Proyecto.....	4
5.- Implementación.....	5
6.- Manual de Uso.....	8
7.- Localización.....	13
8.- Bibliografía.....	13

1.- Introducción.

Dentro de la Asignatura Software Libre y Compromiso Social, he ideado este Proyecto como complemento a otro ya existente en el campo de los Juegos de PC, ya que hoy día es un campo ampliamente utilizado por gran parte de la sociedad y que día a día nos sorprende con los avances que se producen.

El proyecto original es Idle Master, un proyecto de Software Libre que complementa a la conocida plataforma de Steam, de la empresa Valve. Para quien no la conozca, es la mayor plataforma de distribución digital de juegos (también vende software, pero su éxito son los juegos en formato digital).

Me ha parecido útil añadirle una funcionalidad más a dicha aplicación para poder seleccionar tú mismo qué juego quieres “idlear” (ejecutar en segundo plano).

2.- Tecnología Usada.

Idle Master usa el lenguaje C# como lenguaje de codificación. Es un lenguaje orientado a objetos estandarizado por Microsoft para su framework .NET.

Utiliza la API Windows Forms, integrada en el framework .NET. Permite crear aplicaciones de escritorio en las que se puede navegar mediante ventanas.

Su licencia es GNU GPL, que garantiza a los usuarios finales la libertad de usar, estudiar, distribuir y modificar el software.

Como mi proyecto es una ampliación/adición de Idle Master, usaré el mismo lenguaje y API que el proyecto base, así como su licencia.

3.- Explicación del Proyecto.

Para explicar mi proyecto, hay que empezar hablando de Idle Master y su funcionalidad. Idle Master se trata de un proyecto de software libre que intenta ayudar a los usuarios de Steam, concretamente, permitiéndoles ejecutar juegos en segundo plano (o más exacto, simular la ejecución) para poder recibir los cromos pendientes de dichos juegos sin tener que jugarlos.

En mi caso, he visto un pequeño vacío en esta aplicación, y es que sólo simula la ejecución de aquellos juegos que tengan cromos pendientes de recibir, y en un orden determinado que se le puede especificar (entre alfabético, por mayor valor de los cromos en el mercado de Steam o por mayor o menor número de cromos restantes por recibir).

Por ello he decidido añadirle una funcionalidad que amplíe las posibilidades de esta aplicación, permitiéndote decidir manualmente el juego que quieres poner en ejecución, sin depender de la forma de ordenar de Idle Master (que mantendrá su funcionamiento habitual en la pantalla inicial).

4.- Utilidades del Proyecto.

En internet existen algunas páginas web que dan “premios” por conseguir logros o por jugar un determinado número de horas a un juego en un cierto tiempo, es ahí donde entra mi funcionalidad, ya que podrías poner en ejecución dicho juego sin tener siquiera que instalarlo incluso aunque ya hayas cobrado todos sus cromos.

Otra posible aplicación sería conseguir algunos logros por jugar un cierto número de horas a un juego, bastaría con seleccionar el juego y se jugaría solo mientras puedes estar haciendo otras cosas.

Por último y complementándose con el funcionamiento habitual de Idle Master, sería útil si tienes especial interés en recibir antes los cromos de un determinado juego, podrías ir a la ventana de búsqueda y elegir el juego concreto que quieres poner en ejecución, sin tener que esperar o saltar juegos hasta llegar al juego deseado.

5.- Implementación.

Para añadir esta funcionalidad, hay que añadir una nueva opción en la ventana principal.

- frmMain.Designer.cs:
 - o Añadir el nuevo elemento al menú Archivo.

```
//  
// fileToolStripMenuItem  
//  
this.fileToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {  
this.settingsToolStripMenuItem,  
this.searchGameToolStripMenuItem,  
this.blacklistToolStripMenuItem,  
this.toolStripMenuItem1,  
this.exitToolStripMenuItem});  
this.fileToolStripMenuItem.Name = "fileToolStripMenuItem";  
this.fileToolStripMenuItem.Size = new System.Drawing.Size(37, 20);  
this.fileToolStripMenuItem.Text = "&File";
```

- o Configurar la nueva opción.

```
//  
// searchToolStripMenuItem  
//  
this.searchGameToolStripMenuItem.Image = global::IdleMaster.Properties.Resources.imgView;  
this.searchGameToolStripMenuItem.Name = "searchGameToolStripMenuItem";  
this.searchGameToolStripMenuItem.Size = new System.Drawing.Size(117, 22);  
this.searchGameToolStripMenuItem.Text = "&Search Game";  
this.searchGameToolStripMenuItem.Click += new System.EventHandler(this.searchGameToolStripMenuItem_Click);
```

- frmMain.cs
 - o Abrir la nueva ventana al pulsar dicha opción.

```
private void searchGameToolStripMenuItem_Click(object sender, EventArgs e)  
{  
    var frm = new frmGames();  
    frm.ShowDialog();  
}
```

La nueva ventana tendrá un formato similar a otras ventanas que abre la aplicación original, como la de la lista negra.

- frmGames.Designer.cs

```
/// <summary>
/// Required designer variable.
/// </summary>
private IContainer components = null;

/// <summary>
/// Clean up any resources being used.
/// </summary>
/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
protected override void Dispose(bool disposing)
{
    if (disposing && (components != null))
    {
        components.Dispose();
    }
    base.Dispose(disposing);
}

#region Windows Form Designer generated code

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(FrmGames));
    this.lstGames = new System.Windows.Forms.ListBox();
    this.txtName = new System.Windows.Forms.TextBox();
    this.label1 = new System.Windows.Forms.Label();
    this.label2 = new System.Windows.Forms.Label();
    this.btnIdle = new System.Windows.Forms.Button();
    this.SuspendLayout();
    //
    // lstGames
    //
    this.lstGames.FormattingEnabled = true;
    this.lstGames.Location = new System.Drawing.Point(15, 77);
    this.lstGames.Name = "lstGames";
    this.lstGames.Size = new System.Drawing.Size(289, 277);
    this.lstGames.Sorted = true;
    this.lstGames.TabIndex = 0;
    //
    // txtName
    //
    this.txtName.Location = new System.Drawing.Point(15, 35);
    this.txtName.Name = "txtName";
    this.txtName.Size = new System.Drawing.Size(242, 20);
    this.txtName.TabIndex = 0;
    this.txtName.TextChanged += new System.EventHandler(this.txtName_TextChanged);
    //
    // label1
    //
    this.label1.AutoSize = true;
    this.label1.Location = new System.Drawing.Point(12, 22);
    this.label1.Name = "label1";
    this.label1.Size = new System.Drawing.Size(67, 13);
    this.label1.TabIndex = 0;
    this.label1.Text = "Game to Idle";
    //
    // label2
    //
    this.label2.AutoSize = true;
    this.label2.Location = new System.Drawing.Point(12, 362);
    this.label2.Name = "label2";
    this.label2.Size = new System.Drawing.Size(31, 13);
    this.label2.TabIndex = 0;
    this.label2.Text = "Currently In-Game";
    //
    // btnIdle
    //
    this.btnIdle.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom | System.Windows.Forms.AnchorStyles.Right)));
    this.btnIdle.Location = new System.Drawing.Point(237, 385);
    this.btnIdle.Name = "btnIdle";
    this.btnIdle.Size = new System.Drawing.Size(75, 23);
    this.btnIdle.TabIndex = 0;
    this.btnIdle.Text = "Idle";
    this.btnIdle.UseVisualStyleBackColor = true;
    this.btnIdle.Click += new System.EventHandler(this.btnIdle_Click);
    //
    // FrmGames
    //
    this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.None;
    this.ClientSize = new System.Drawing.Size(324, 415);
    this.Controls.Add(this.btnIdle);
    this.Controls.Add(this.label1);
    this.Controls.Add(this.label2);
    this.Controls.Add(this.lstGames);
    this.Controls.Add(this.txtName);
    this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle;
    this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
    this.MaximizeBox = false;
    this.MinimizeBox = false;
    this.Name = "FrmGames";
    this.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent;
    this.Text = "Idle Specific Game";
    this.Closed += new System.EventHandler(this.FrmGames_Closed);
    this.Load += new System.EventHandler(this.FrmGames_Load);
    this.ResumeLayout(false);
}

#endregion

private ListBox lstGames;
private Button btnIdle;
private TextBox txtName;
private Label label1;
private Label label2;
```

- frmGames.cs
 - o La forma de cargar la lista de juegos es la misma en la aplicación original, esto es, lee el html obtenido de una URL relacionada con el usuario logueado y de ahí extrae la lista de juegos.
 - o Al modificar el texto, recargar la lista de juegos.

```
private void txtName_TextChanged(object sender, EventArgs e)
{
    lstGames.Items.Clear();
    if (txtName.Text.Length > 0)
    {
        foreach (var game in AllGames)
        {
            if (game.Name.ToUpper().Contains(txtName.Text.ToUpper()))
                lstGames.Items.Add(game.Name);
        }
    }
}
```

- o Al pulsar el botón, parar el juego en ejecución, si lo hubiera, y ejecutar el juego seleccionado.

```
private void btnIdle_Click(object sender, EventArgs e)
{
    if (lstGames.SelectedItem != null)
    {
        if (CurrentGame != null)
            CurrentGame.StopIdle();
        CurrentGame = AllGames.FirstOrDefault(b => b.Name == lstGames.SelectedItem.ToString());
        if (CurrentGame != null)
        {
            label2.Text = localization.strings.currently_ingame + @" " + CurrentGame.Name;
            CurrentGame.Idle();
        }
    }
    txtName.Text = string.Empty;
    txtName.Focus();
}
```

- o Al cerrar la ventana de búsqueda, parar el juego en ejecución, si lo hubiera.

```
private void frmGames_Closed(object sender, EventArgs e)
{
    if (CurrentGame != null)
    {
        CurrentGame.StopIdle();
    }
}
```

Las modificaciones en los archivos de localización son triviales (simplemente añadir los literales nuevos a los idiomas deseados).

6.- Manual de Uso.

Para poder usar la aplicación hay que cumplir los mismos requisitos que para usar Idle Master, esto es, tener una cuenta de Steam, tener abierta la aplicación de Steam en el PC y vincular Idle Master con Steam.

Una vez validado Idle Master para revisar tu cuenta, para acceder a la funcionalidad de nuestro proyecto hay que ir a la opción Buscar juego, añadida al menú Archivo.



Esto abrirá una nueva ventana donde podremos buscar el juego que queremos ejecutar.



Al iniciar la escritura en la caja de texto, se mostrarán en la lista, los juegos que contengan la cadena introducida.

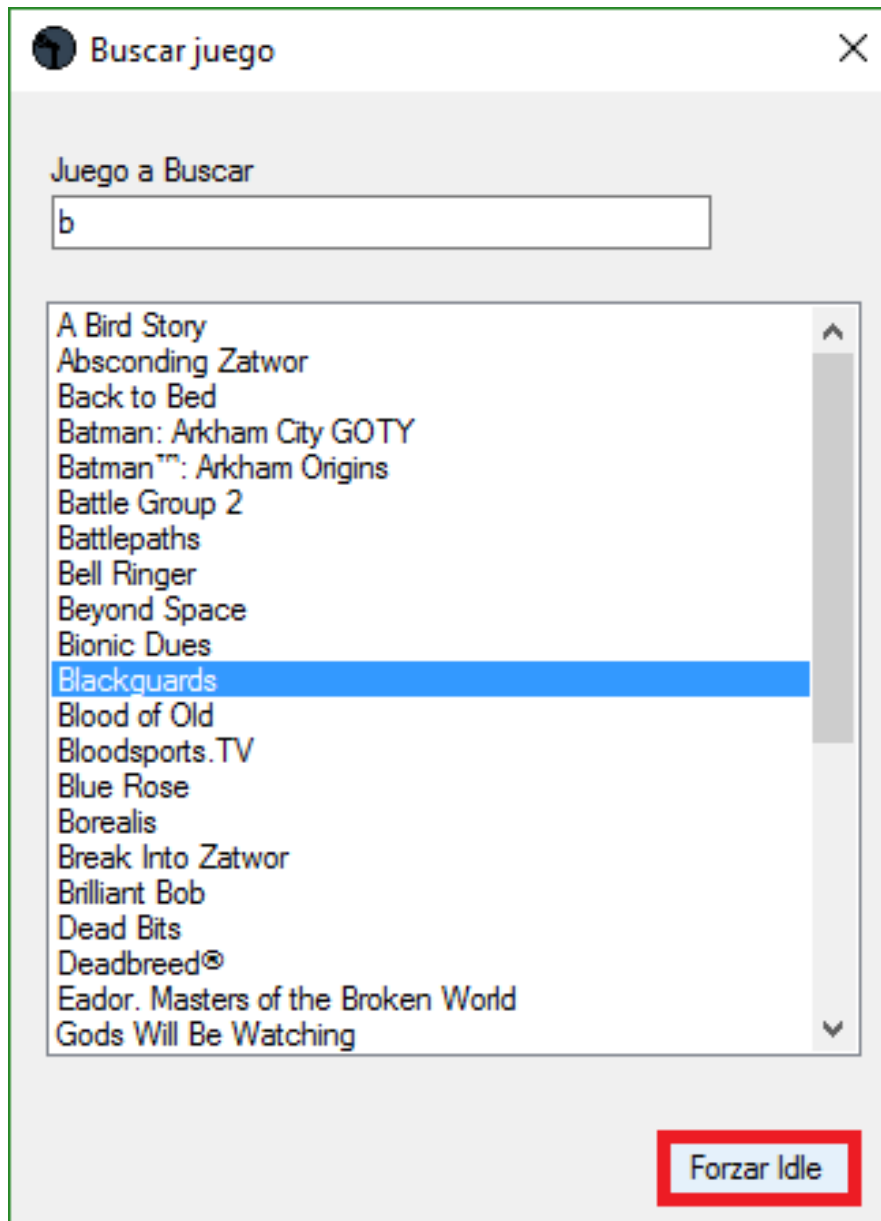
Buscar juego

Juego a Buscar

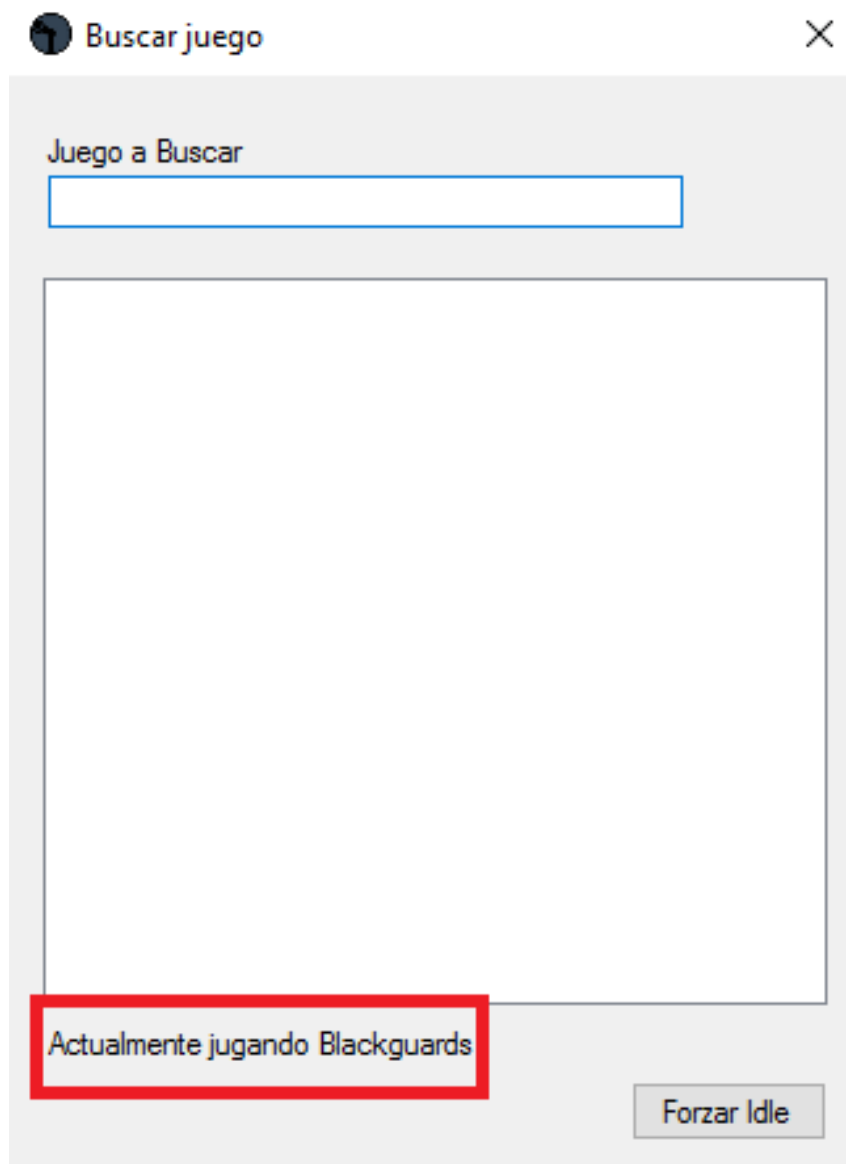
- A Bird Story
- Absconding Zatwor
- Back to Bed
- Batman: Arkham City GOTY
- Batman™: Arkham Origins
- Battle Group 2
- Battlepaths
- Bell Ringer
- Beyond Space
- Bionic Dues
- Blackguards
- Blood of Old
- Bloodsports.TV
- Blue Rose
- Borealis
- Break Into Zatwor
- Brilliant Bob
- Dead Bits
- Deadbreed®
- Eador. Masters of the Broken World
- Gods Will Be Watching

Forzar Idle

Una vez encontrado el juego deseado, se selecciona en la lista y se pulsa en Forzar Idle. Si anteriormente hubiera algún otro juego ejecutándose en esta ventana, se pararía y se sustituiría por el nuevo.



Una vez ejecutado, la lista se limpia y se muestra un mensaje indicando el juego que está actualmente ejecutándose



7.- Localización

Idle Master está localizado a múltiples idiomas: chino, alemán, griego, español, inglés,... Mi ampliación contiene algunos literales que sólo he localizado al inglés y al español.

Si mi funcionalidad es aceptada en una pull request para la aplicación Idle Master, la comunidad se encargaría de localizarlo a los demás idiomas.

8.- Bibliografía

- Página web de Idle Master:

<http://www.steamidlemaster.com/>

- Repositorio de Idle Master:

https://github.com/jshackles/idle_master

- Repositorio del proyecto:

<https://github.com/Razzielsoul/Idle-master-a-la-carta>

- GNU GPL:

<http://www.gnu.org/licenses/gpl-3.0.html>