

# Estructura de software Launcher 1.0.0

Jurgen Heysen

May 11, 2014

## **Contents**

---

<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>2</b>	<b>Estructura</b>	<b>3</b>
2.1	User Interface . . . . .	3
2.2	Utilidades . . . . .	3
2.3	Game Management . . . . .	3
2.4	Service Management . . . . .	3
<b>3</b>	<b>Ideas Inconclusas</b>	<b>4</b>
3.1	Sistema Ball . . . . .	4
3.2	Shell . . . . .	4
3.3	Eventos . . . . .	4

# 1 Introducción

Este documento busca indicar la estructura básica del software al momento de la release de esta versión, así como de informar a los futuros encargados de darle mantenimiento sobre los problemas conocidos e ideas inconclusas.

## 2 Estructura

El software en si se compone de distintos módulos que se comunican entre sí buscando tener el mínimo acoplamiento posible. Estos módulos son:

1. User Interface
2. Utilidades
3. Game Management
4. Service Management

Los módulos de Game Management y Service Management se unen en la Facade del Backend, pero pueden, con mínimas modificaciones, funcionar el uno sin el otro.

### 2.1 User Interface

Esta parte del software tiene relación con la interfaz que se muestra al usuario. No tiene mucha relación con el núcleo del launcher más que lo que extrae desde la facade del sistema.

Actualmente, se tienen dos interfaces:

1. Graphic User Interface
2. Command Line Interface

La GUI se ha programado usando TkInter y Pillow para el manejo de imágenes, pygame para el manejo de joysticks. La línea de comandos se ha programado sólo usando python puro.

Actualmente, la interfaz es escogida por el script de inicio. Hacia el futuro se pensaba que era posible que las interfaces pudieran ser tan complejas como se necesitara, siendo piezas de software casi independientes del núcleo del Launcher, así como poder tener alguna forma centralizada en el módulo que escogiese la interfaz a utilizar, y permitiera instalar más. Así mismo, la GUI actual no se relaciona con información proveniente de los servicios y las posiciones de los elementos son casi fijas, lo que pudiera ser de interés cambiar.

### 2.2 Utilidades

Esta zona del software está pensada básicamente para pequeñas funciones que se utilicen mucho y que puedan ser reutilizadas por diversos módulos, como el manejo de juegos y las interfaces de usuario.

Actualmente, se cuenta con `run_parts` que ejecuta todos los scripts que encuentre en una carpeta y `rel_prep` que elimina todos los `.pyc` en el árbol de directorios del Launcher. En el futuro, `run_parts` puede ser revisado para asegurarse que es lo más general posible.

### 2.3 Game Management

Parte principal del Launcher, es el módulo encargado de detectar y ejecutar los juegos registrados. Contiene objetos para representar la información de un juego (parsing de `GameInfo.xml`), mantención de la lista de juegos y el código para ejecutar el juego en un Thread separado, en este caso un Process separado.

Si en un futuro se piensa cambiar el estándar de información de juegos, se deberá alterar este módulo. No olvidar que debería considerarse retrocompatibilidad con la versión actual del estándar, por lo que juegos que utilicen nuevas versiones del estándar deberán indicarlo de alguna forma.

### 2.4 Service Management

Este módulo contiene todo el código orientado a descubrir, inicializar y administrar los accesos a los servicios instalados. Cabe mencionar que el esquema actual de servicios hace que estén 100% desacoplados del software, ya que son descubiertos en tiempo de ejecución.

Los desafíos futuros de este módulo son el mejorar el manejo de permisos para el acceso a los servicios y un desafío de refactoring por similitudes de partes de su código con `run_parts`.

## 3 Ideas Inconclusas

Al momento de la liberación de esta release, quedaron algunas ideas inconclusas que pueden ser interesantes de implementar:

### 3.1 Sistema Ball

Este sistema estaría encargado de la administración de paquetes del Launcher, no sólo juegos, si no que también de servicios y otros add-ons, posiblemente permitiendo actualizaciones por internet y actualizaciones automáticas, además de búsqueda de paquetes para su descarga e instalación. Se puede pensar en algo como apt-get y dpkg, la idea era que los paquetes estuvieran basados en archivos zip.

### 3.2 Shell

Este componente sería la columna vertebral del Launcher reestructurandolo completamente, ya que todo giraría en torno a él, por ejemplo, las interfaces ya no se comunicarían directamente con el módulo de administración de juegos si no que entregarían comandos a este componente, y este, con consideraciones de permisos de acceso y similares mediaría la respuesta y comunicación. Se puede pensar en algo como bash en linux.

### 3.3 Eventos

La posibilidad de definir un módulo encargado de eventos de forma centralizada, de tal suerte que cualquier módulo podría solicitar la creación de un evento y suscribirse a otros, cosa que cuando se gatillara el evento, pudiera reaccionar a él. Esto es de especial utilidad en las Interfaces de usuario ya que permitiría la adición simple de add-ons.