Especificación de juegos Arcade 1.0

Jurgen Heysen

January 24, 2014

Contents

1 Introducción

El presente documento tiene como objetvo instruir a los desarrolladores de videojuegos sobre los estándares que deben cumplir para ser compatibles con el Launcher del Aracde

2 Estándar

2.1 Código

Los juegos deben exponer un archivo .py con una clase que contenga el método `Go`, que debe recibir como parámetro un objeto del tipo `ArcadeServiceInterface` (ArcadeServiceInterface.py), este objeto será tratado mas adelante y es una interfaz a los servicios solicitados por el juego.

2.2 XML

Los juegos deben exponer un archivo .xml con información relevante sobre ellos. El elemento principal debe llamarse Game y opcionalmente de argumento puede llevar un string llamado version con el valor "1.0". El Titulo del juego va en el elemento `Title` como contenido. Es importante agregar un código corto único para el juego como contenido del elemento `Code`
El elemento `Description` tiene como contenido la descipción del juego que se mostrará en la GUI.
La versión se incluye como contenido de `Version` , se recomienda que sea un entero
La fecha de compilación puede ser añadida con un elemento `Date`
El elemento `MainClass` es obligatorio y debe llevar como atributos name con el nombre de la clase que contiene el método `Go(...)` y path con la ruta relativa a este archivo, extensión incluida.
Se puede incluir la lista de autores mediante un elemento `Authors` que contenga una lista de elementos `Author`, con atributo `name` para el nombre del autor cada uno .
Otro aspecto a considerar es la lista contenida en `Screenshots`, para incluir las screenshots del juego que se mostraran en el Launcher. Cada una se indica mediante un elemento `Screenshot` con un atributo `src` que indica el path a la imagen.
La lista denotada por `Services` indica los servicios que solicita el juego al Launcher, cada servicio se solicita mediante un elemento `Service` con atributo `name` que es el nombre del servicio solicitado, además se deben incluir los parámetros requeridos por el servicio según su propio estándar.
Adicionalmente se puede crear una lista denotada por `AdditionalData` para incluir datos adicionales. No hay especificaciones sobre los contenidos de esta lista.
A continuación se presenta una tabla resumen sobre la información que debe contener el xml:

Table 1: Resumen de documento Xml

Tag	Parent	Args	Num	Descripción
Game	-	-	1	Root del documento
Title	Game	-	1	Título del juego
Code	Game	-	1	Código del juego
MainClass	Game	name: Nombre file: archivo	1	Nombre de la clase de inicialización del juego
Description	Game	-	1	Descripción
Version	Game	-	1	Versión del juego
Date	Game	-	1	Fecha de release
Authors	Game	-	1	Encierra lista de Authores
Author	Authors	name: Nombre	*	Identifica un autor
Screenshots	Game	-	1	Encierra lista de capturas de pantalla
Screenshot	Screenshots	src: path a la imagen	*	Identifica una captura de pantalla
Services	Game	-	1	Encierra lista de servicios
Service	Services	name: Nombre del servicio parámetros adicionales del servicio	*	Identifica un servicio que solicita el juego
AdditionalData	Game		1	Encierra una lista de datos adicionales

3 Servicios

El Launcher ofrece a los juegos diversos servicios que pueden ser instalados. Estos son agegados a la instalación base y siguiendo la documentación de servicios, los desarrolladores pueden incluir los suyos propios.

En esta sección se abordará la interfaz común de acceso a servicios.

3.1 Clase ArcadeServiceInterface

La clase `ArcadeServiceInterface` provee el acceso a los servicios del Arcade que son declarados en el .xml de información del juego. Los servicios contenidos en él se encuentran ya inicializados para el juego, se provee dos formas de acceder a los servicios.

3.1.1 Método getServices()

Este método devuelve un diccionario con las clases que representan cada servicio, donde las llaves son los nombres de estos. Si faltara algún servicio, signiica que este no se registró correctamente en la inicialización o que no se encuentra instalado, pero para todos los efectos no está disponible.

3.1.2 Método getService(servicio)

Este método recibe como string el nombre del servicio y retorna el objeto que representa a este servicio si se encuentra disponible o None en caso contrario.

4 Instalación de juegos

En una distribución normal del Launcher, los juegos pueden ser instalados siguiendo los siguientes pasos:

1. Descomprimir los contenidos del juego en una carpeta de elección.
2. Navegar a carpeta del Launcher.
3. En consola de comandos, ejecutar `python game_install.py -i path_a_xml` [1]
4. Verificar con `python game_install.py -l`

Puede desinstalar[2] un juego con `python game_install.py -u CODIGO` donde CODIGO es el codigo del juego que desea desinstalar. Puede ser visto ejecutando antes `python game_install.py -l`

[1] Ajustar a python 2.7 si no fuera el caso
[2] No borra los archivos, solo quita el juego del registro del arcade