

# LIBGDX Fontes avanzado

## UNIDADE 2: Fontes - avanzado

### Sumario

- 1 Introducción
- 2 Proceso de uso
  - ◆ 2.1 Descarga da ferramenta Hiero
  - ◆ 2.2 Uso da ferramenta Hiero
- 3 Exemplo de código
- 4 TAREFA OPTATIVA A FACER

### Introdución

**Nota:** Esta explicación está relacionada coa sección de Fontes.

Información na Wiki: <https://github.com/libgdx/libgdx/wiki/Hiero>

Como comentamos no punto relacionado, a clase BitmapFont ten un tamaño de 15 puntos e é Arial. Normalmente imos querer que o tamaño da fonte sexa máis grande para que cando a visualicemos non teña que escalar o seu tamaño e que aparezan dentes de serra.

Para xerar estas fontes (con varias efectos que podemos engadir optativos) imos utilizar unha ferramenta de nome Hiero.

### Proceso de uso

#### Descarga da ferramenta Hiero

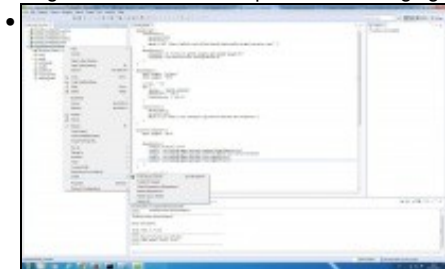
Información na wiki: <https://github.com/libgdx/libgdx/wiki/Dependency-management-with-Gradle>

Imos ver unha das formas que temos de obter a ferramenta. Esta ferramenta se atopa nas GDX-Tools.

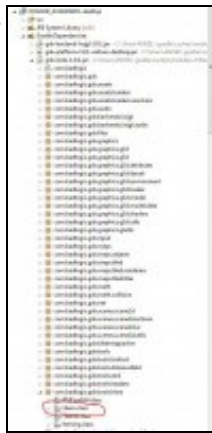
- Descarga das GDX-TOOLS



Debemos editar o arquivo **build.gradle** do **PROXECTO PRINCIPAL** (o que non é DESKTOP, nin HTML, nin CORE nin ANDROID). Engadiremos a liña: `compile "com.badlogicgames.gdx:gdx-tools:$gdxVersion"`.



Debemos refrescar o proxecto principal premendo có botón dereito do rato e escollendo a opción Gradle=>Refresh All.



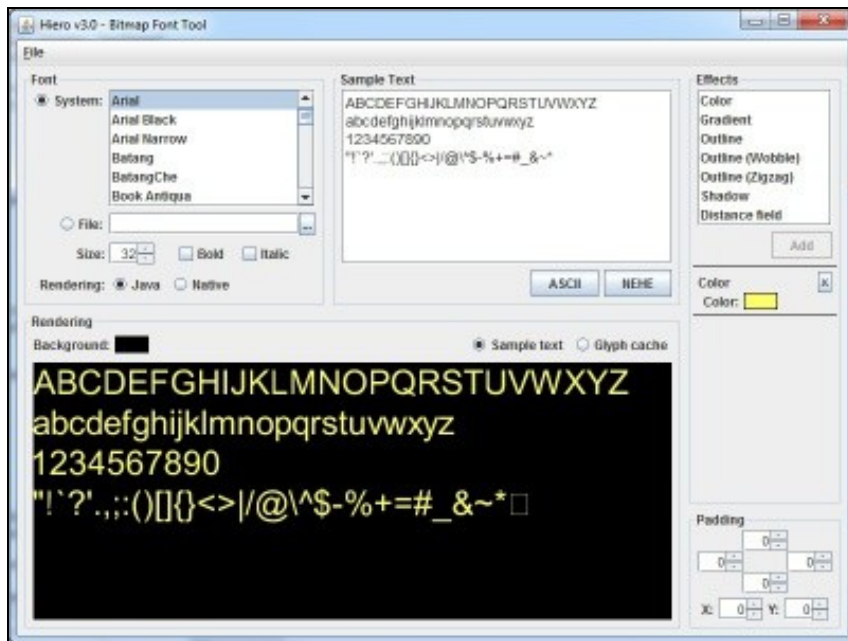
Na versão DESKTOP teremos a classe Hiero.class.



Prememos o botão direito sobre a classe e a executamos.

## Uso da ferramenta Hiero

Unha vez instalada aparecerá a seguinte pantalla:



- Información na wiki: <https://github.com/libgdx/libgdx/wiki/Hiero>
- Para executala directamente da web: <https://wiki.libgdx.googlecode.com/git/jws/hiero.jnlp>
- Un paso a paso: <http://singletechgames.com/2012/04/20/crear-archivos-fnt-con-hiero/>

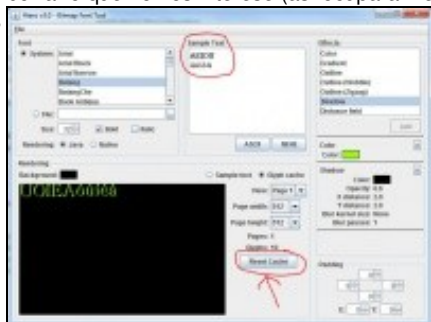
A idea é crear un arquivo png con todas as letras que se necesiten para o noso xogo (lembrade poñer as letras acentuadas).

- Debemos **premer sobre a opción Glyph cache** se queremos
- Se non necesitamos todos os caracteres podemos borrarlos (ventá SampleText) e poñer os que queiramos.
- Na parte da dereita temos unha ventá **Effects** na que podemos conseguir diferentes efectos, dende cambiar de cor, ata darlles unha aparencia 3D....
- Cando teñamos os caracteres que necesitemos escolleremos a opción 'Save BMFont files' do menú File.

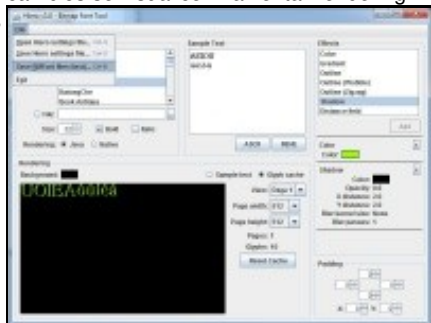
- Exemplo de uso da ferramenta Hiero



Podemos cargar un xogo por defecto de caracteres premendo ASCII ou un xogo reducido premendo NEHE. Unha vez cargado podemos borrar o que no nos interese (así ocupará menos espazo) e engadir o que non estea como por exemplo os caracteres con til.



Unha vez temos os caracteres que nos interesen, o tipo de letra, tamaño e efectos desexados prememos o botón 'Reset cache' para que os cambios se visualicen na ventá Rendering.



Cando teñamos os caracteres que necesitamos escolleremos a opción 'Save BMFont files' do menú File

## Exemplo de código

- Unha vez feito o paso anterior debemos copiar os dous arquivos xerados (un bmp e un fnt) ó cartafol assets da versión Android (dentro de assets podemos organizar como queiramos os arquivos).

Deberedes de cambiar a clase co que inician as diferentes plataformas pola seguinte:

- Deberedes copiar o gráfico dos peixes animados ó cartafol assets do proxecto Android.
- Crear unha nova clase.

### Código da clase FontesAvanzadas

**Obxectivo:** Amosar como usar unha fonte xerada con HIERO.

```
import com.badlogic.gdx.ApplicationAdapter;
import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.graphics.GL20;
import com.badlogic.gdx.graphics.g2d.BitmapFont;
import com.badlogic.gdx.graphics.g2d.SpriteBatch;

public class FontesAvanzadas extends ApplicationAdapter {
    SpriteBatch batch;

    private BitmapFont bitMapFont;

    @Override
    public void create () {
        batch = new SpriteBatch();

        bitMapFont = new BitmapFont(Gdx.files.internal("fontes.fnt"), false);
    }

    @Override
    public void render() {
        Gdx.gl.glClearColor(1, 1, 1, 1);
        Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT);
        batch.begin();

        bitMapFont.draw(batch, "EXEMPLO de código", 10, 100);

        batch.end();
    }

    @Override
    public void dispose() {
        batch.dispose();
        bitMapFont.dispose();
    }
}
```

Comentamos o código:

- Liña 10: Creamos un obxecto da clase BitMapFont.
- Liña 16: Instanciamos o obxecto pasándolle como parámetro o arquivo fnt xerado coa ferramenta. O segundo parámetro indica que debemos voltear as letras á hora de cargalas (facer flip). Isto pode ser necesario se a ferramenta que xera as letras as xera de forma volteada.
- Liña 25: Debuxamos unha cadea utilizando o novo xogo de caracteres.

## TAREFA OPTATIVA A FACER

Modifícade o xogo para que cargue unha fonte xerada por vos utilizando a ferramenta Hiero.

-- Ángel D. Fernández González -- (2014).