Kezdőlap

Témabejelentő: a szakdolgozat bekötve kell, hogy tartalmazza a kitöltött es jóváhagyott (az Informatikai Kar dékánja által aláírt) Szakdolgozat-téma bejelentőt. Ennek a lapnak a helyére kell bekötni.

Tartalomjegyzék

[1. Bevezetés 4](#_Toc454972628)

[2. Felhasználói dokumentáció 5](#_Toc454972629)

[3. Fejlesztői dokumentáció 6](#_Toc454972630)

[3.1. A fejlesztői környezet 6](#_Toc454972631)

[3.1.1. Projektgenerálás 6](#_Toc454972632)

[3.1.2. Android Studio, JDK, Android SDK 7](#_Toc454972633)

[3.1.3. PC build konfiguráció 7](#_Toc454972634)

[3.1.4. Külső könyvtár hozzáadása 8](#_Toc454972635)

[3.2. A projekt mappa tartalma 8](#_Toc454972636)

[3.2.1. doc mappa 8](#_Toc454972637)

[3.2.2. Gradle modul mappák 8](#_Toc454972638)

[3.2.3. core 8](#_Toc454972639)

[3.2.4. android 8](#_Toc454972640)

[4. Irodalomjegyzék 10](#_Toc454972641)

# Bevezetés

# Felhasználói dokumentáció

# Fejlesztői dokumentáció

## A fejlesztői környezet

A játék Java nyelven, a libGDX keretrendszer segítségével készül, az Android Studio integrált fejlesztői környezetben, Gradle használatával.

Az animációk a Spriter xml-alapú animációs programmal készülnek, a pályákat pedig a Tiled általános célú, 2D, csempe alapú pályaszerkesztővel készítem.

A Spriter animációk libGDX játékban való lejátszásához a Spriter közösség egy tagja, TrixtOr által készített Spriter implementációt használom. (1) (2)

A környezet következő, részletesebb leírásában szereplő beállítások és menük a dolgozat írása idejében érvényesek, és a jövőben változhatnak.

### Projektgenerálás

A projekt létrehozására a libGDX weboldaláról letölthető (3) projektgeneráló eszköz használható. Az eszközben megadható a projekt neve, a csomag, a főosztály neve, a célkönyvtár, valamint be kell állítani az Android SDK helyét. Kiválaszthatók a célzott platformok is, valamint kiegészítéseket is megadhatunk, például a Box2d fizika könyvtárat.

Az eszköz futtatásának eredménye egy Gradle projekt, amelyben a különböző platformok, valamint a közös kód (core) külön Gradle modulokba kerülnek. Ezen kívül az Android projekteknél megszokott fájlok is létrejönnek, mint a Gradle wrapper fájlok, és a .gitignore.

### Android Studio, JDK, Android SDK

3.1. ábra. libGDX projektgeneráló eszköz

A projektgeneráló eszköz által létrehozott projektet Android Studioba az „Import project (Eclipse ADT, Gradle, etc.)” paranccsal lehet importálni.

A használni kívánt JDK-t és Android SDK-t a „Module Settings/SDK Location” menüben érhetjük el, valamelyik modulra való jobb kattintással. Célszerű Java 6-ot vagy 7-et választani, mert a Java 8-cal létrehozott külső könyvtárak esetén a Gradle hibát adhat. Ugyanis a Java 8 támogatás az Android N verzióval érkezik, ami a dolgozat írásakor developer preview állapotban van.

### PC build konfiguráció

A projekt importálását követően csak egy build konfiguráció érhető el: „android”, amivel egy Android szimulátoron vagy csatlakoztatott eszközön lehet futtatni a játékot. Ezért létre kell hozni egy build konfigurációt, amivel PC-n is futtatható a program. Ehhez az „Edit Configurations…” menüben adjunk hozzá egy „Application” típusú konfigurációt, a következő beállításokkal:

* Name: pc
* Main class: a „desktop” modul főosztálya
* Working directory: az „android” modul „assets” könyvtára
* Use classpath of module: a „desktop” modul kiválasztása

### Külső könyvtár hozzáadása

Külső könyvtárat, mint a libGDX Spriter implementációt, hozzáadhatunk a projekthez a „File/New/New Module…” menüben, a „Java Library” opciót választva. Így a függőség egy új modulként jelenik meg a projektben. Ezután be kell másolni a modulba a .java fájlokat.

## A projekt mappa tartalma

A projekt létrehozását és beállítását követően a mappa tartalma a következő:

### doc mappa

A dokumentációt és a hozzá tartozó fájlokat tartalmazza.

### Gradle modul mappák

Minden platformhoz tartozó Gradle modul saját könyvtárat kap, ezen kívül külön modult képez a core, azaz a közös, platformfüggetlen kód.

A modul mappákban platformfüggő állományok, IDE és Gradle fájlok, build könyvtárak, illetve kiegészítő könyvtárak is megtalálhatók a forráskódon kívül („src”):

### core

A core modulban a nyers erőforrásokat találjuk az „assets\_raw” mappában. Ebben a modulban van a játék forráskódjának legnagyobb része.

### android

* libs: natív függőségek.
* res: az Androidnál megszokott funkcióval rendelkezik, jelen esetben a játék ikonját tartalmazza, különböző felbontásokban.
* assets: a játékhoz szükséges erőforrások (képek, animációk stb.) találhatók itt. Minden platform ebből a könyvtárból éri el az erőforrásokat (így a „core” modul is).
* AndroidManifest.xml: Android konfigurációs fájl.
* ic\_launcher-web.png: a Google Play Store-ban használt ikon.
* proguard-project.txt: az Androidnál használt ProGuard optimalizációs beállításokat tartalmazza.

# Irodalomjegyzék

1. **BrashMonkey.** Spriter API’s | Spriter from BrashMonkey. *Spriter.* [Online] BrashMonkey LLC, 2015. [Hivatkozva: 2016. 06. 09.] https://brashmonkey.com/spriter-api/.

2. **Trixt0r.** Trixt0r/spriter: A Generic Java importer for Spriter animation files. *GitHub.* [Online] [Hivatkozva: 2016. 06. 29.] https://github.com/Trixt0r/spriter.

3. **Zechner, Mario.** libgdx. *libgdx.* [Online] [Hivatkozva: 2016. 06. 28.] https://libgdx.badlogicgames.com/download.html.