

libGDX

Autor: Martin Hochstrasser

Überblick

OpenSource-Spielerframework für



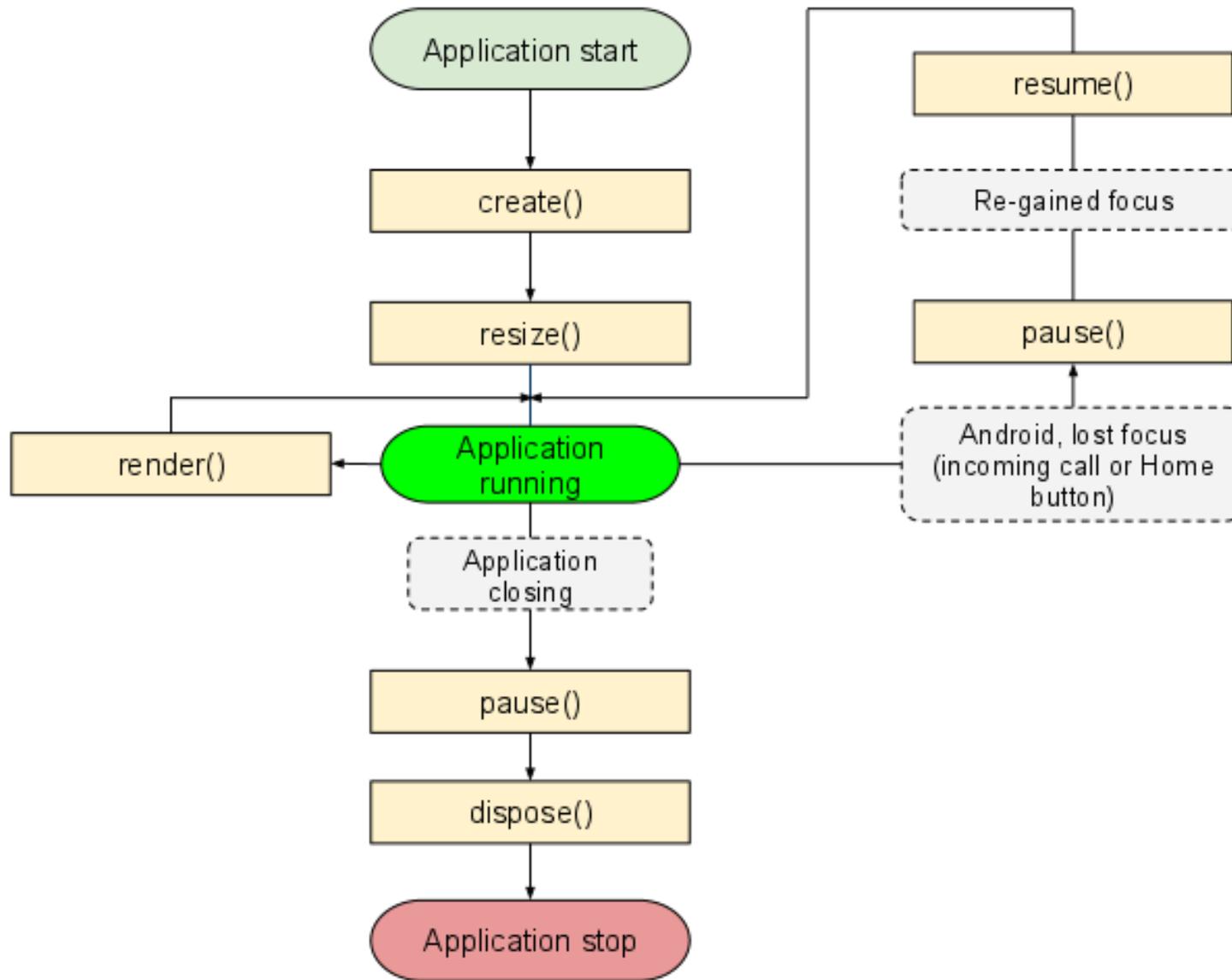
Apache 2 Lizenz



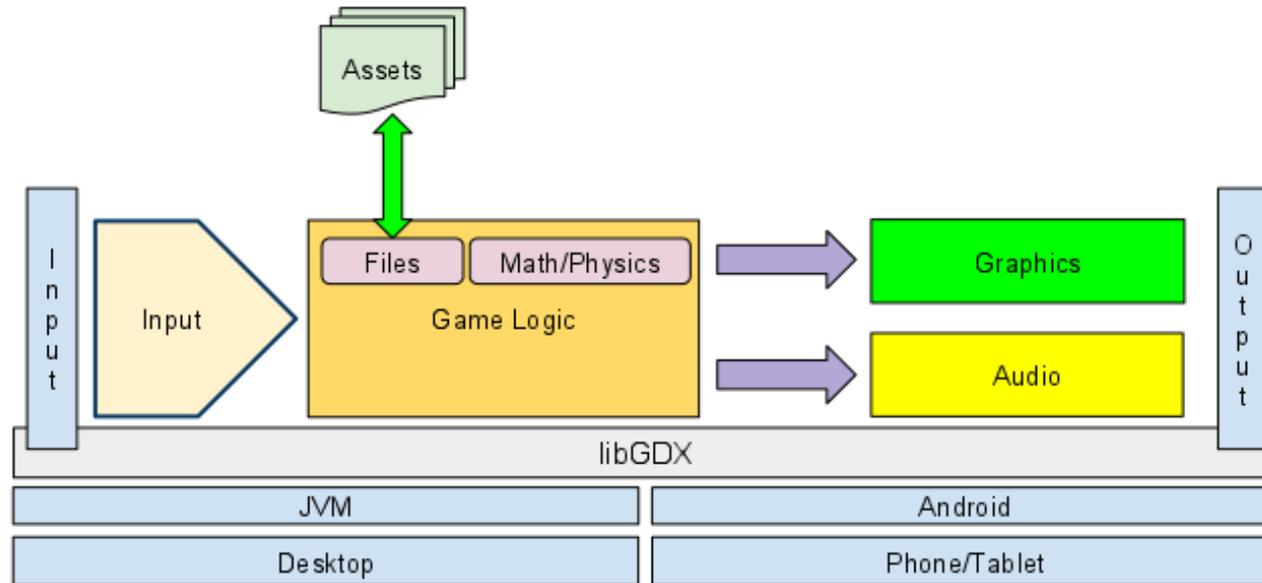
Lauffähig auf



Verfügbar unter <https://github.com/libgdx/libgdx>



Module



Werden durch Plattform implementiert

Zusammengefasst in der Klasse `Gdx` (statische Felder)

Projektaufbau

- MyGdxGame
 - core/
 - src/
 - build.gradle (Einstellungen für core)
 - desktop/
 - android/
 - ios/
 - build.gradle (Einstellungen für alle Projekte)
 - settings.gradle



erstellt durch gdx-setup.jar

Wo gehört was hin?

In `core/` das eigentliche Spiel

Eine Klasse implementiert `ApplicationListener`

In `desktop/`, `android/`, `ios/`, ..
spezifischerer Code für die Plattform

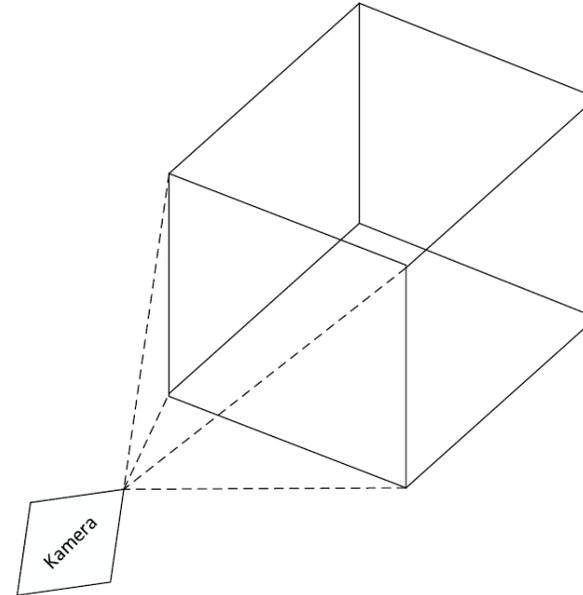
Realisiert als Schnittstelle in `core/`

Zeichenfläche

1. „Aufstellen“ der Kamera
2. Reinigen
3. Spiel zeichnen (immer wieder)

OrthographicCamera

```
Gdx.gl.glClearColor (...)
```



Rendern von Texturen und Texten

SpriteBatch

Zeichnet 2D-Texturen unter Zuhilfenahme einer Kamera

```
spriteBatch.setProjectionMatrix(camera.combined);
```

```
spriteBatch.begin();
```

...

```
spriteBatch.end();
```

Rendern von Texturen und Texten

Texture

Stellt eine Grafik dar, die aber bereits auf der GPU liegt (wurde schon dekodiert)

```
logo = new Texture(Gdx.files.internal („logo.png“))
```

Sprite

Speichert zusätzlich zu einer Textur die Position, Drehung und die Skalierung

```
sprite = new Sprite();
```



Rendern von Texturen und Texten

BitmapFont

Kann Texte zeichnen mithilfe eines `SpriteBatches` und speichert für jedes Zeichen eine Textur

```
font.draw(spriteBatch, "fhLUG", 320, 200);
```

fhLUG

FreeTypeFontGenerator

Erzeugt BitmapFonts aus TTF oder OTF-Schriften

```
font = fontGenerator.generateFont(para);
```

```
abcdefghijklmnopqrstuvwxyZ ABCDEFGHIJKLMNOPQRSTUVWXYZ  
1234567890.:,; ' " (!?) +-*/=
```

Verwaltung von Texturen

TextureAltals

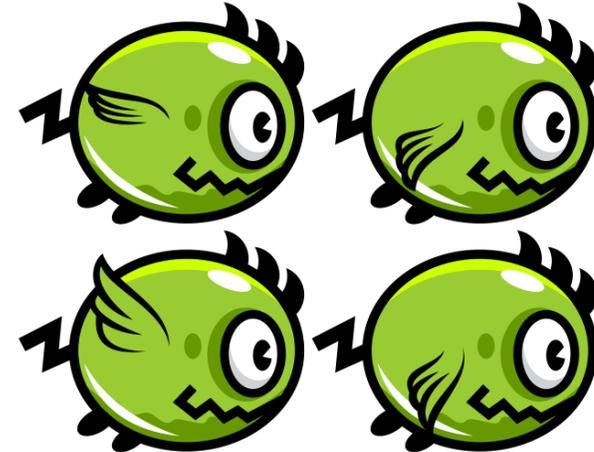
Verwaltet mehrere Bilder in einer einzigen Texture

```
atlas = new TextureAtlas(packFile);
```

TexturePacker

Erstellt aus einem Verzeichnis aus Bilder einen Textureatlas

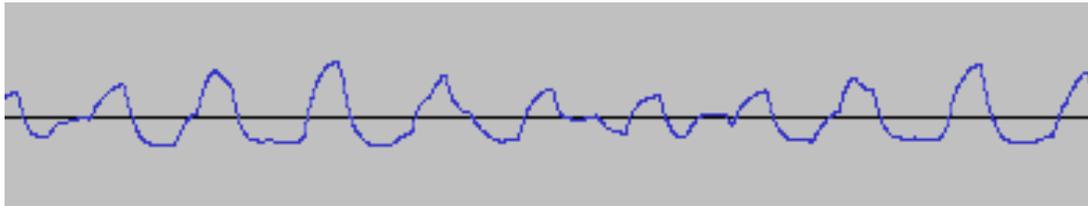
```
atlas = new TextureAtlas(packFile);
```



Wiedergabe von Audio

- Sound
 - Kürzer, wird im Speicher gehalten
- Musik
 - Länger, wird gestreamt

Formate: OGG, WAV, MP3



Eingabehandlung

Tastatur

```
Gdx.input.isKeyPressed(Input.Keys.SPACE)
```

Maus / Touch

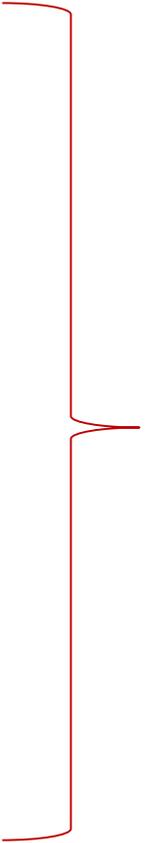
```
Gdx.input.getX();
```

```
Gdx.input.getY();
```

```
Gdx.input.isButtonPressed(Input.Buttons.LEFT)
```

Orientierung

```
Gdx.input.getAzimuth(); Gdx.input.getPitch();
```



Polling

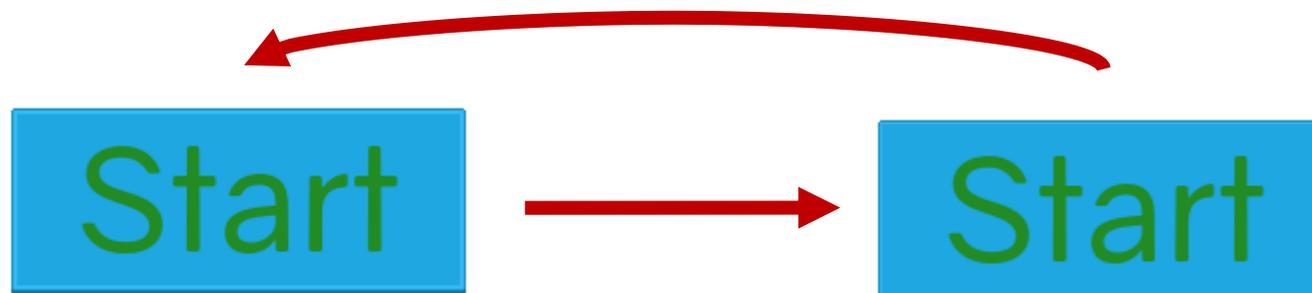
Eingabehandlung

InputProcessor

Behandlung mithilfe des Observer-Patterns (event handling)

Erlaubt eine zeitliche Mitverfolgung

Besser geeignet für GUIs (scene2d)



Verwaltung von „Assets“

AssetManager

Erlaubt die Verwaltung aller möglichen Arten von „Assets“

Gibt diese auch einheitlich frei

Erlaubt das asynchrone Laden (→Ladebildschirm möglich!)

```
assetManager.load("char.atlas", TextureAtlas.class);
```

...

```
assetManager.get("char.atlas");
```

Scene2D

Stage

Dreh- und Angelpunkt jeder Szene

Enthält alle Akteure

Verarbeitet Eingaben und gibt diese weiter

```
stage = new Stage(new ScreenViewport(camera), spriteBatch);
```

```
Gdx.input.setInputProcessor(stage);
```

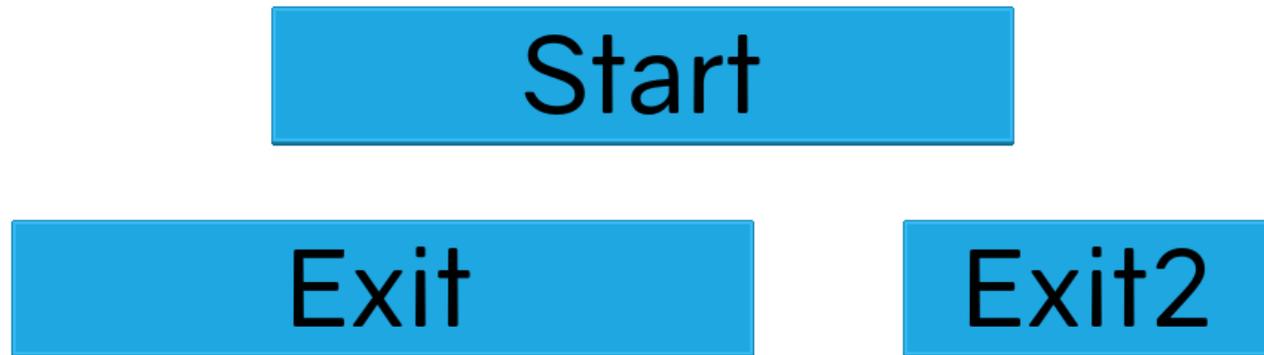
...

```
stage.act(delta);
```

```
stage.draw();
```

Scene2D

- Button
- Tabellen (universell)
- Label
- Image
- CheckBox
- List
- ...



Physik (Box2D)

Wrapper um Box2D (C++ Bibliothek)

`World`

Behälter für Körper

Grundstruktur einer Physik-Welt (Gravitation)

Alles in Meter / Kg und Rad!

`Body`

Dargestellt durch eine Fixture

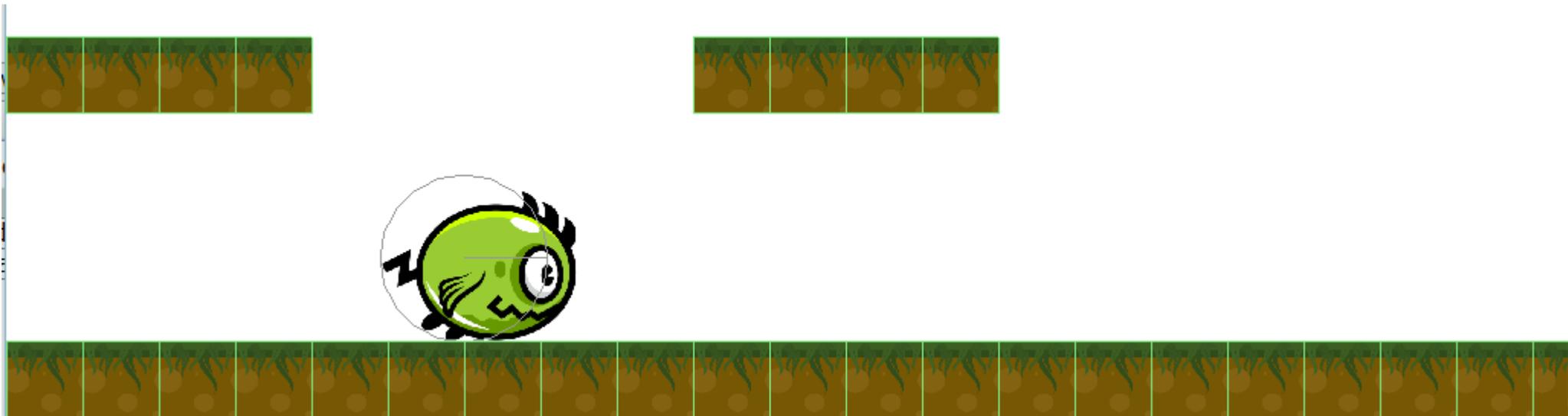
Hat Masse, Position, Rotation und Geschwindigkeit

Physik (Box2D)

Statische Körper: Fixe verankert

Dynamische Körper: Variable Position

(Kinematische Körper)



GameJam von RoboVM

Prizes & Judging

We are happy to have RoboVM and Robotality as sponsors for the following prizes:

1. **Grand Prize:** Mac Mini, sponsored by RoboVM
2. **Silver:** iPad, sponsored by RoboVM
3. **Bronze:** iPod Touch, sponsored by RoboVM
4. **For 20 random submissions:** Steam keys for [Halfway](#), sponsored by Robotality
5. **For another 5 random submissions:** libGDX Jam t-shirt, by yours truly 😊

Timetable

Theme Voting round 1	Nov. 22nd – Dec. 11th
Final Theme Voting	Dec. 11th – Dec. 18th
Jam	Dec. 18th – Jan. 18th
Judging	Jan 19th – Feb. 2nd