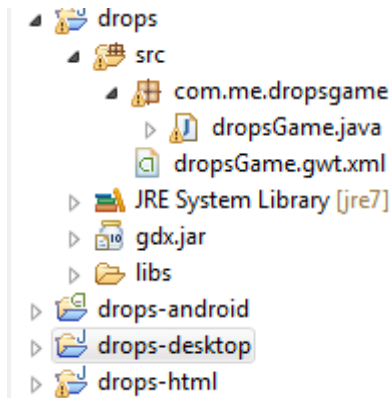


LIBGDX Projekt bearbeiten

Die Texturen und Bilder kommen in den Ordner: **projektname-android > assests/data**
Dieser Ordner ist zum Datenordner des Desktop Projekts und des HTML Projekts verlinkt.
In diesem Fall die Datei **texturebucket.png**

Aus dem von gdx-setup-gui erstellten und in Eclipse importierten Projekt die plattformübergreifende dropsGame.java Datei öffnen und bestehenden Quellcode bis auf unten gezeigten Zeilen entfernen und ergänzen.



```
package com.me.dropsgame;
import com.badlogic.gdx.ApplicationListener;

public class dropsGame implements ApplicationListener {
    OrthographicCamera camera;
    Stage stage;
    Texture texturebucket;
    TextureRegion regionbucket;
    Actor bucketActor;

    @Override
    public void create() {
        float w = Gdx.graphics.getWidth();
        float h = Gdx.graphics.getHeight();
        camera = new OrthographicCamera(1, h/w);

        texturebucket = new Texture(Gdx.files.internal("data/texturebucket.png"));
    }

    @Override
    public void dispose() {
        texturebucket.dispose();
    }

    @Override
    public void render() {
        Gdx.gl.glClearColor(0, 0, 0.2f, 1);
        Gdx.gl.glClear(GL10.GL_COLOR_BUFFER_BIT);

        stage.act(Math.min(Gdx.graphics.getDeltaTime(), 1/60f ));
        stage.draw();
    }

    @Override
    public void resize(int width, int height) {
        stage.setViewport(width, height, true);
    }
}
```

Folgende Bibliotheken importieren:

```
import com.badlogic.gdx.scenes.scene2d.Actor;
import com.badlogic.gdx.scenes.scene2d.InputEvent;
import com.badlogic.gdx.scenes.scene2d.InputListener;
import com.badlogic.gdx.scenes.scene2d.Stage;
import com.badlogic.gdx.scenes.scene2d.ui.Image;
```

In der Methode create() die Stage und den Actor einrichten:

```
texturebucket = new Texture(Gdx.files.internal("data/texturebucket.png"));
regionbucket = new TextureRegion(texturebucket, 0, 0, 256, 256);
bucketActor = new Image(regionbucket);
bucketActor.setX(50);
bucketActor.setY(10);

stage = new Stage(Gdx.graphics.getWidth(), Gdx.graphics.getHeight(), false);
Gdx.input.setInputProcessor(stage);

stage.addActor(bucketActor);
```

In der Methode render() die Stage zeichnen lassen.

```
@Override
public void render() {
    Gdx.gl.glClearColor(0, 0, 0.2f, 1);
    Gdx.gl.glClear(GL10.GL_COLOR_BUFFER_BIT);

    stage.act(Gdx.graphics.getDeltaTime());
    stage.draw();
}
```

Auf Mausclick und Finger reagieren. Dazu in der create() Methode den Listener einfügen.

```
bucketActor.addListener(new InputListener() {
public boolean touchDown (InputEvent event, float x, float y, int pointer, int button)
{
    return true;
}

public void touchDragged (InputEvent event, float x, float y, int pointer) {

    Vector2 touchPos = new Vector2();
    Vector2 stagePos = new Vector2();

    touchPos.set(x, y);
    stagePos = bucketActor.localToStageCoordinates(touchPos);

    bucketActor.setPosition(stagePos.x, 10);

    if(bucketActor.getX() < 0) bucketActor.setPosition(0, 10);
    if(bucketActor.getX() > Gdx.graphics.getWidth() - 48)
    bucketActor.setPosition(Gdx.graphics.getWidth() - 48, 10);

}
});
```