Computer Games 2014
Selected Game Engines

Dr. Mathias Lux

Klagenfurt University

pixi.js

- Web based rendering engine
- Programming with JavaScript
- 2D, but hardware accelerated

pixi.js

- 2D WebGL based renderer
  - With Canvas based fallback
- Multi-touch capable
- WebGL filters, tinting, blending
- Sprite Sheets, asset loader, …

pixi.js Setup

- Embedded in a simple HTML page:

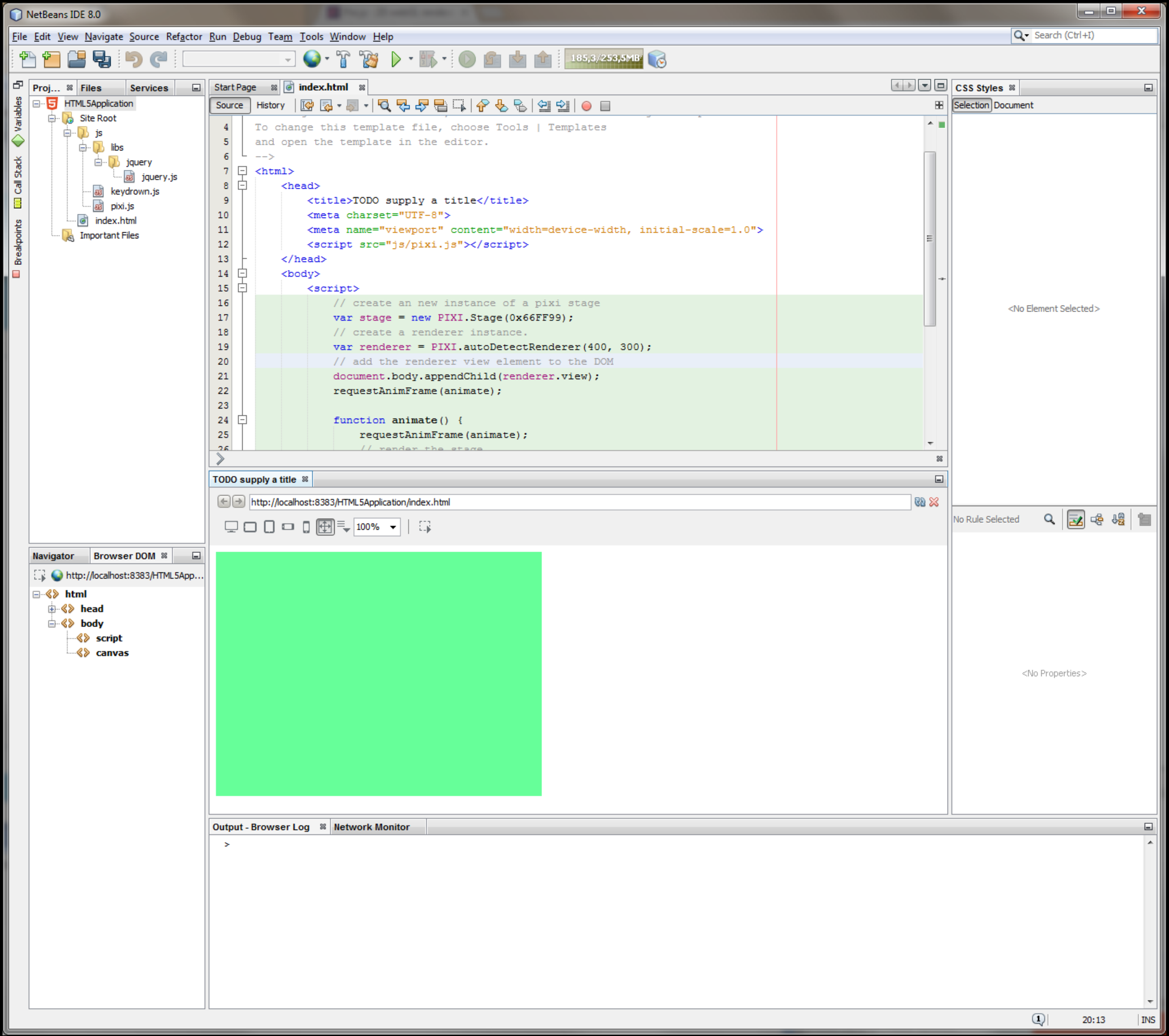
```
<html>
<head>
    <title>ComputerGames - WegGL Example 1</title>
    <style>
        body {
    background-color: #000000;
        }
    </style>
    <script src="pixi.js"></script>
</head>
<body>
```

pixi.js Setup

```
<html>
<head>
    <title>ComputerGames - WegGL Example 1</title>
    <style>
        body { margin: 0; padding: 0; background-color: #000000;}
    </style>
    <script src="pixi.js"></script>
</head>
<body>
    <script>
    // create an new instance of a pixi stage
    var stage = new PIXI.Stage(0x66FF99);
    // create a renderer instance.
    var renderer = PIXI.autoDetectRenderer(400, 300);
    // add the renderer view element to the DOM
    document.body.appendChild(renderer.view);
    requestAnimFrame( animate );

    function animate() {
        requestAnimFrame( animate );
        // render the stage
        renderer.render(stage);
    }
    </script>
    </body>
</html>
```

How to develop with pixi.js?

- WebGL is not available on local pages!
- Web server is needed.
- Consider NetBeans as an option
  - Built-In web server
  - Javascript editor & debugging



```
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <script src="js/pixi.js"></script>
    </head>
    <body>
        <script>
            // create an new instance of a pixi stage
            var stage = new PIXI.Stage(0x66FF99);
            // create a renderer instance.
            var renderer = PIXI.autoDetectRenderer(400, 300);
            // add the renderer view element to the DOM
            document.body.appendChild(renderer.view);
            requestAnimFrame(animate);

            function animate() {
                requestAnimFrame(animate);
                // render the stage
```

http://localhost:8383/HTML5Application/index.html

```
>
```

Adding a sprite

```
// create a texture from an image path
var texture = PIXI.Texture.fromImage("img/bar.png");
// create a new Sprite using the texture
var bar = new PIXI.Sprite(texture);

// center the sprites anchor point
bar.anchor.x = 0.5;
bar.anchor.y = 0.5;

// move the sprite t the center of the screen
bar.position.x = 15;
bar.position.y = 50;
bar.rotation = Math.PI/2;

stage.addChild(bar);
```

Interaction

- Using the Keydrown library

```
kd.P.down(function () {
 console.log('The "P" key is being held down!');
});

kd.P.up(function () {
 console.log.clear();
});

// This update loop is the heartbeat of Keydrown
kd.run(function () {
  kd.tick();
});
```

In our pixi.js game ...

```
   // add the function keys:
   function moveBarDown() {
        if (bar.position.y < 300) bar.position.y += 5;
   }
   function moveBarUp() {
        if (bar.position.y > 0) bar.position.y -= 5;
   }

   kd.DOWN.down(moveBarDown);
   kd.UP.down(moveBarUp);

    function animate() {
        // tick keydrown
        kd.tick();
        requestAnimFrame( animate );
        // render the stage
        renderer.render(stage);
    }
```

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ

Add a ball sprite & collision

- Sprite adding as usual
- Physics should make sure that …
  - the ball is not stuck in the wall or paddle
  - a collision is detected before the actual event
  - appropriate actions are taken at the right time
  - the **outcome** is visualized
- See *game04.html* file …

pixi.js: What is not there?

- Audio & Sound
- Input
- Physics
- Network Layer

JavaScript and HTML 5?

- CocoonJS
  - HTML5 & WebGL for mobile devices
  - Can be packaged in an app
- Crafty
  - Pure JavaScript & Canvas
  - Scene-based, collision detection ...
- Cocos-2D JS
  - Cocos-2D based, supports many tools

libGDX features

- High-performance, cross-platform game development framework.
  - Android, iOS, Blackberry
  - Windows, Linux & MacOS (Java)
  - HTML 5, WebGL, Javascript
- Basis for engines and games.
  - e.g. AndEngine

libGDX features

- Multiple backends
  - Jogl, LWJGL, Android APIs, JavaScript
- Rendering through OpenGL ES
  - Low level & high level 3D

libGDX features

- High level 2D
  - Batched and cached sprite rendering
  - Bitmap fonts
  - Particle systems
  - TMX tile map rendering
  - UI system
  - and many more features, …

libGDX features

- **Audio**
  - Music and SFX from WAV, MP3 and OGG
  - Audio decoding of OGG and MP3
- **File I/O**
  - Abstracting Android assets, classpath and file-system
  - JavaScript binary files, …

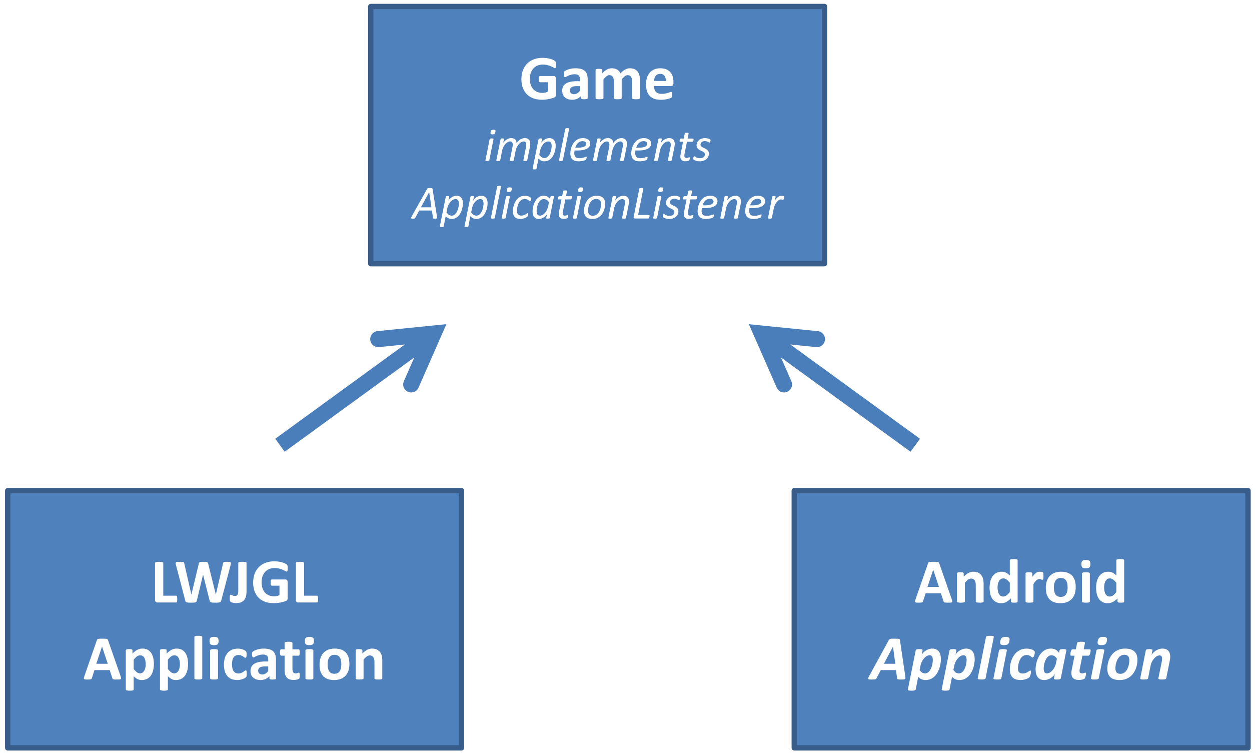
libGDX features

- **Input**
  - Polling and event-based access to touch-screen/mouse and keyboard.
  - Polling access to compass and accelerometer
  - Vibration support
  - Remote input event processing
- **Physics**
  - Full JNI wrapper of box2d

libGDX features

- Tools & Extensions
  - Particle editor
  - Bitmap font generator
  - Texture packer

libGDX - Multiplatform



HelloWorld

```
public class HelloWorld implements ApplicationListener {
    SpriteBatch spriteBatch;
    Texture texture;
    BitmapFont font;
    Vector2 textPosition = new Vector2(100, 100);
    Vector2 textDirection = new Vector2(1, 1);

    @Override
    public void create () {
        font = new BitmapFont();
        font.setColor(Color.RED);
[…]
```

src. http://code.google.com/p/libgdx

HelloWorld LWJGL

```
public class HelloWorldDesktop {
    public static void main(String[] argv) {
        GdxTestGame game = new GdxTestGame();
        new LwjglApplication(new HelloWorld(),
                "Hello World", 480, 320, false);
    }
}
```

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ

src. http://code.google.com/p/libgdx

HelloWord Jogl

```
public class HelloWorldDesktop {
   public static void main (String[] argv) {
        new JoglApplication(new HelloWorld(),
              "Hello World", 480, 320, false);
   }
}
```

src. http://code.google.com/p/libgdx

HelloWorld Android

```
public class HelloWorldAndroid extends
    AndroidApplication {
    @Override
    public void onCreate (Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        initialize(new HelloWorld(), false);
    }
}
```

src. http://code.google.com/p/libgdx

libGdx 2D Sprites

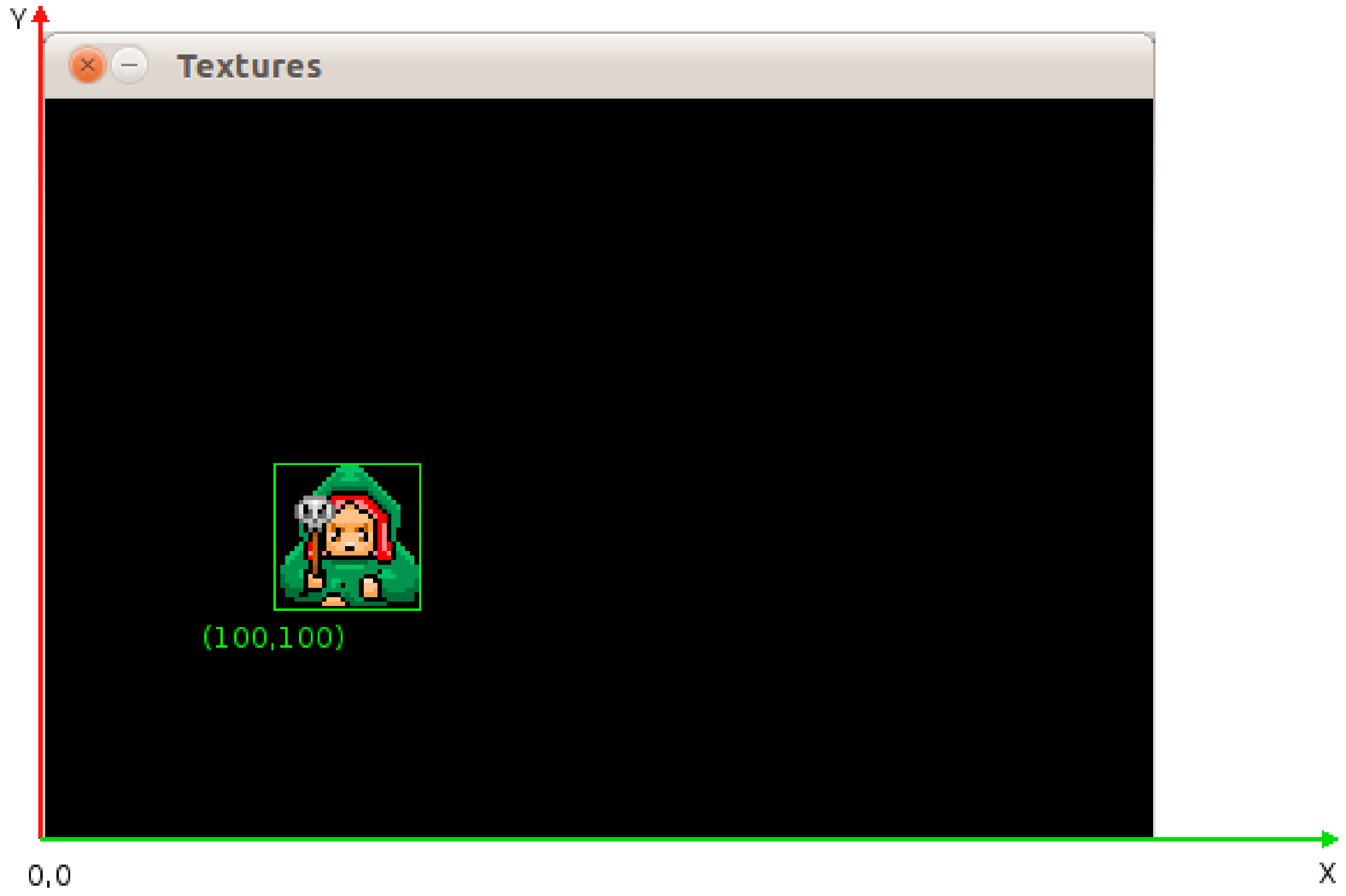
- Textures
  - OpenGL handles all sprites as Textures
  - Textures are decoded images in memory
- SpriteBatch
  - Takes care of displaying textures
  - Texture mapping and creation of rectangles

Drawing a Single Texture

```
public class TextureFun implements ApplicationListener {

        private Texture druidTexture;
        private SpriteBatch batch;

        @Override
        public void create() {
                druidTexture = new Texture(Gdx.files.internal("druid.png"));
                batch = new SpriteBatch();
        }

        @Override
        public void render() {
                batch.begin();
                batch.draw(druidTexture, 100, 100);
                batch.end();
        }

        // … rest of methods omitted … //
}
```

src. http://code.google.com/p/libgdx/wiki/TexturesTextureRegionSpriteBatch

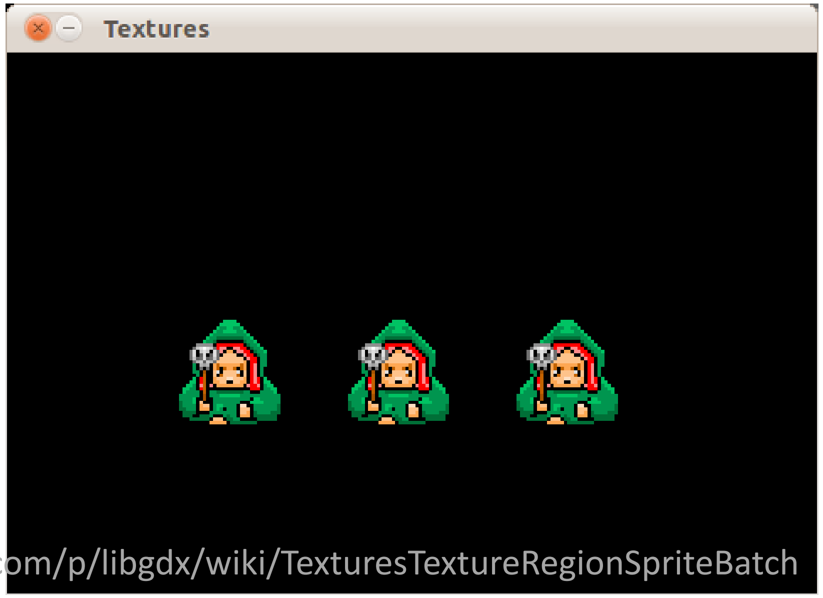
Drawing a Single Texture



src. http://code.google.com/p/libgdx/wiki/TexturesTextureRegionSpriteBatch

Multiple Calls

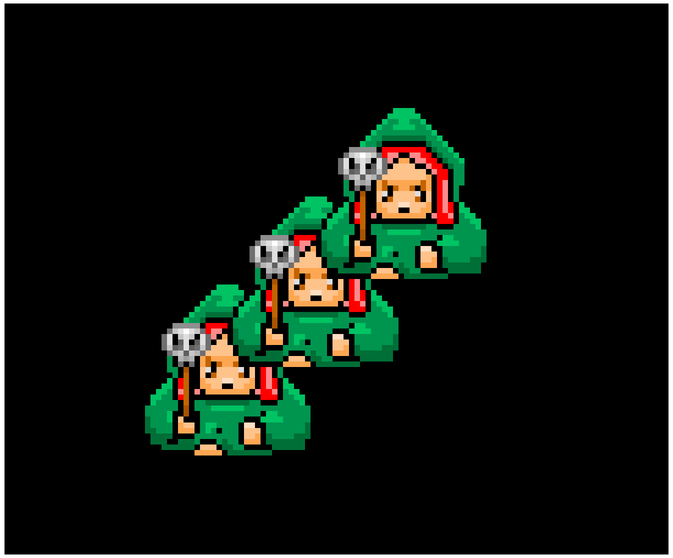
```
public void render() {
        batch.begin();
        batch.draw(druidTexture, 100, 100);
        batch.draw(druidTexture, 200, 100);
        batch.draw(druidTexture, 300, 100);
        batch.end();
}
```



src. http://code.google.com/p/libgdx/wiki/TexturesTextureRegionSpriteBatch

Drawing the textures on top of each other

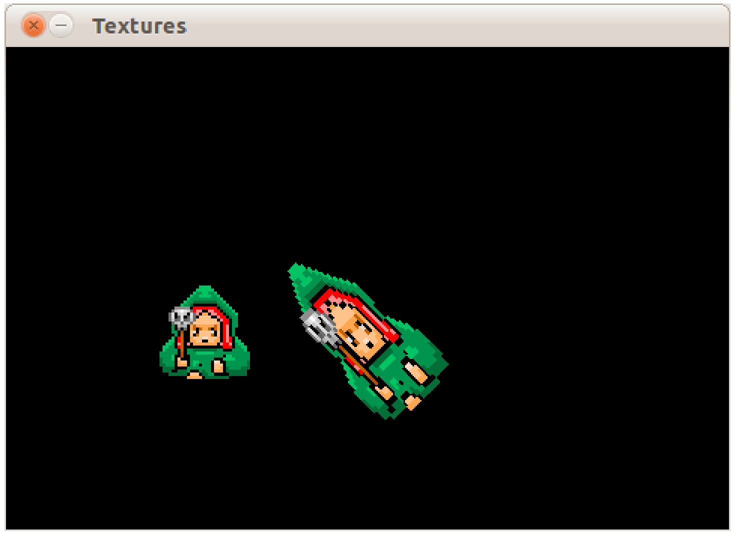
```
public void render() {
        batch.begin();
        batch.draw(druidTexture, 100, 100);
        batch.draw(druidTexture, 132, 132);
        batch.draw(druidTexture, 164, 164);
        batch.end();
}
```



src. http://code.google.com/p/libgdx/wiki/TexturesTextureRegionSpriteBatch

Rotation & Scaling

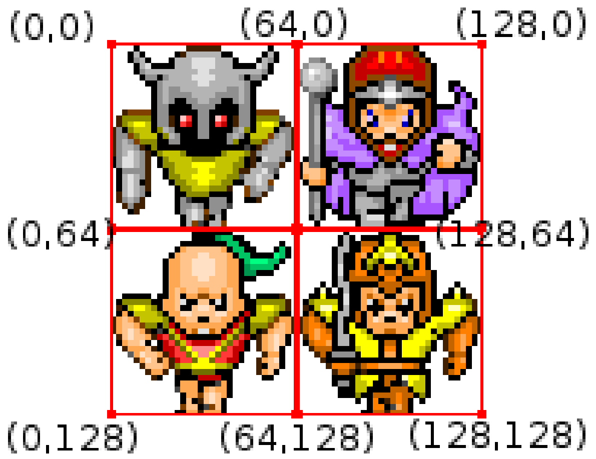
```
public void render() {
        batch.begin();
        batch.draw(druidTexture, 100, 100);
        batch.draw(druidTexture, 200, 100, 32, 32, 64, 64,
                        1f, 2.0f, 45f, 0, 0, 64, 64, false, false);
        batch.end();
}
```



src. http://code.google.com/p/libgdx/wiki/TexturesTextureRegionSpriteBatch

TextureRegion

Spritesheet (single image)

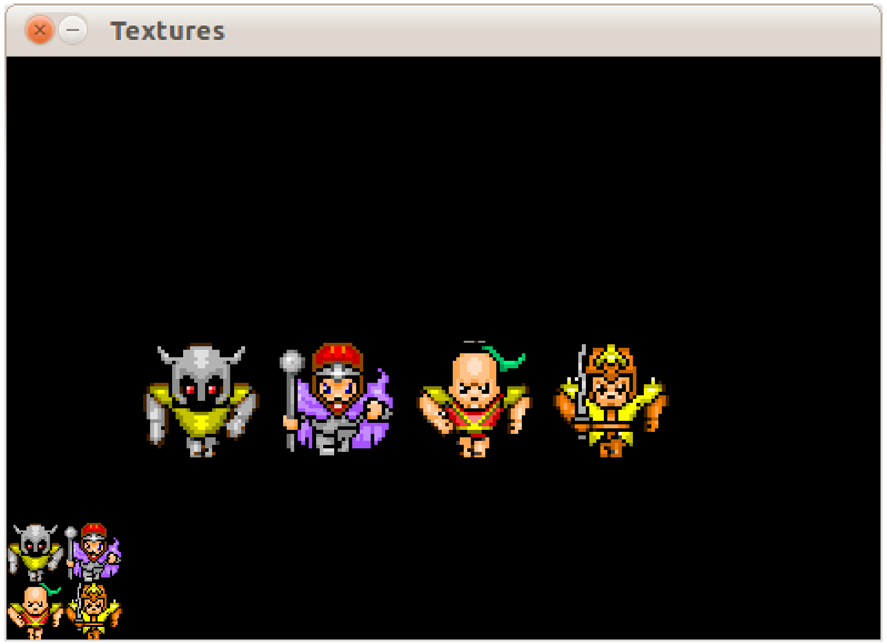


Regions with coordinates

TextureRegion

```
public class TextureFun implements ApplicationListener {
    private Texture texture;
    private SpriteBatch batch;
    private TextureRegion[] regions = new TextureRegion[4];

    public void create() {
        texture = new Texture(Gdx.files.internal("sprite_sheet.png"));
        batch = new SpriteBatch();
        regions[0] = new TextureRegion(texture, 0, 0, 64, 64);
        regions[1] = new TextureRegion(texture, 0.5f, 0f, 1f, 0.5f);
        regions[2] = new TextureRegion(texture, 0, 63, 64, 64);
        regions[3] = new TextureRegion(texture, 0.5f, 0.5f, 1f, 1f);
    }

    public void render() {
        batch.begin();
        batch.draw(texture, 0, 0, 64, 64);
        for (int i = 0; i < regions.length; i++)
            batch.draw(regions[i], 75 * (i + 1), 100);
        batch.end();
    }
```

TextureRegion



```
...
TextureRegion[][] regions = TextureRegion.split(texture, 64, 64)
...
```

Blending & Viewport

- Viewport is determined automatically
  - Mapping pixels to pixels
  - Determined at begin()
- Blending
  - Enabled by default
  - Disable for performance critical things

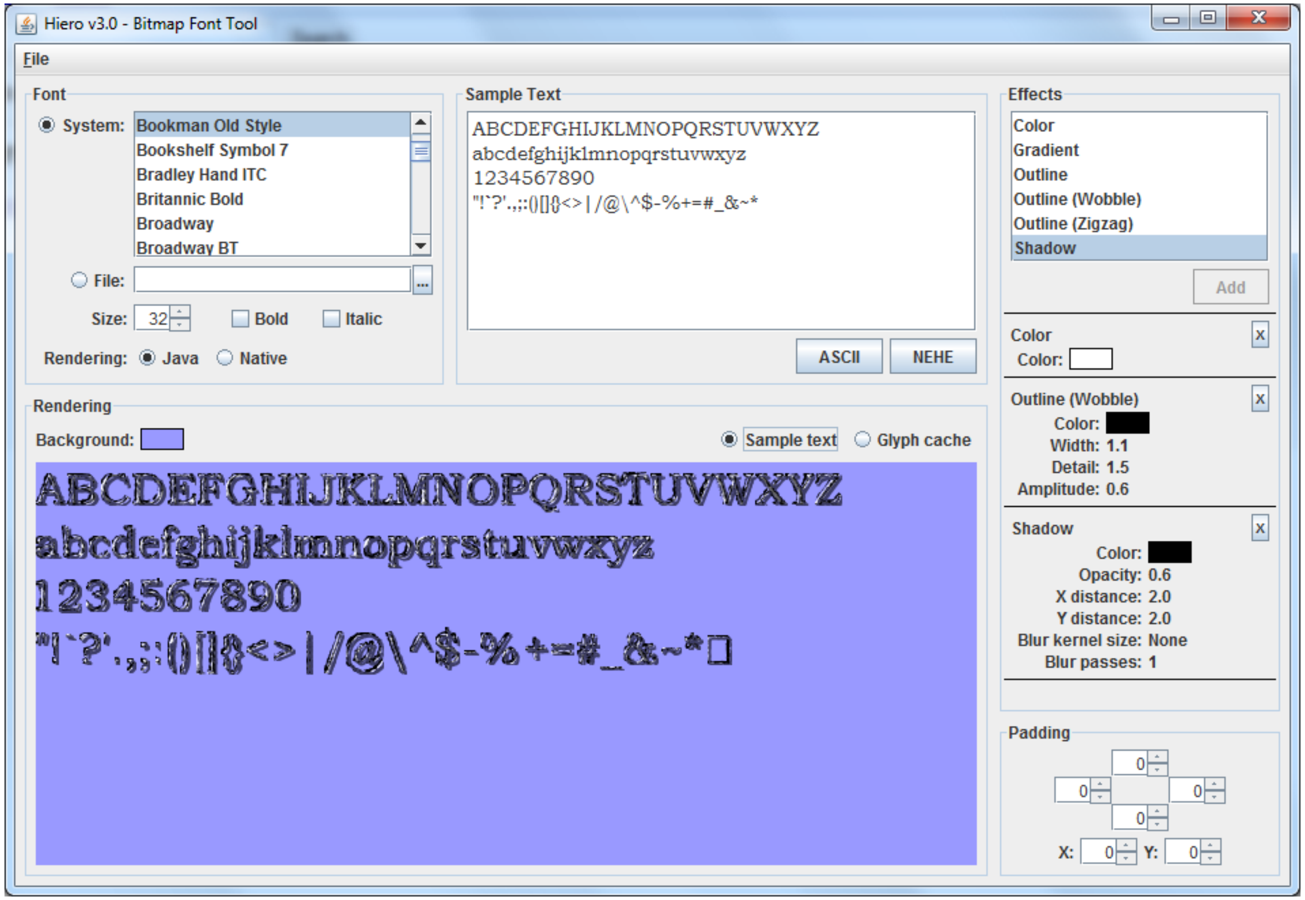
Orthographic Camera

- Parallel projection from 3D to 2D
- Used to map game space to view
  - ie. Game is written for full HD.
  - Rendered on various devices.

Orthographic Camera

```
public void create() {
    camera = new OrthographicCamera(1920, 1080);
    camera.position.set(1920 / 2, 1080 / 2, 0);
    // ...
}

public void render() {
    camera.update();
    batch.setProjectionMatrix(camera.combined);
    batch.begin();
    // ...
}
```

Using Fonts



Hiero v3.0 - Bitmap Font Tool

File

Font

[✓ System:] Bookman Old Style

Bookshelf Symbol 7

Bradley Hand ITC

Britannic Bold

Broadway

Broadway BT

File:

Size: 32 Bold Italic

Rendering: [✓ Java] Native

Sample Text

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890
"!`?'.,;:()[]{}<>|/@\^$-%+=#_&~*

ASCII NEHE

Effects

Color
Gradient
Outline
Outline (Wobble)
Outline (Zigzag)
Shadow

Add

Color
Color:

Outline (Wobble)
Color:
Width: 1.1
Detail: 1.5
Amplitude: 0.6

Shadow
Color:
Opacity: 0.6
X distance: 2.0
Y distance: 2.0
Blur kernel size: None
Blur passes: 1

Rendering

Background: [✓ Sample text] Glyph cache

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890
"!`?'.,;:()[]{}<>|/@\^$-%+=#_&~*

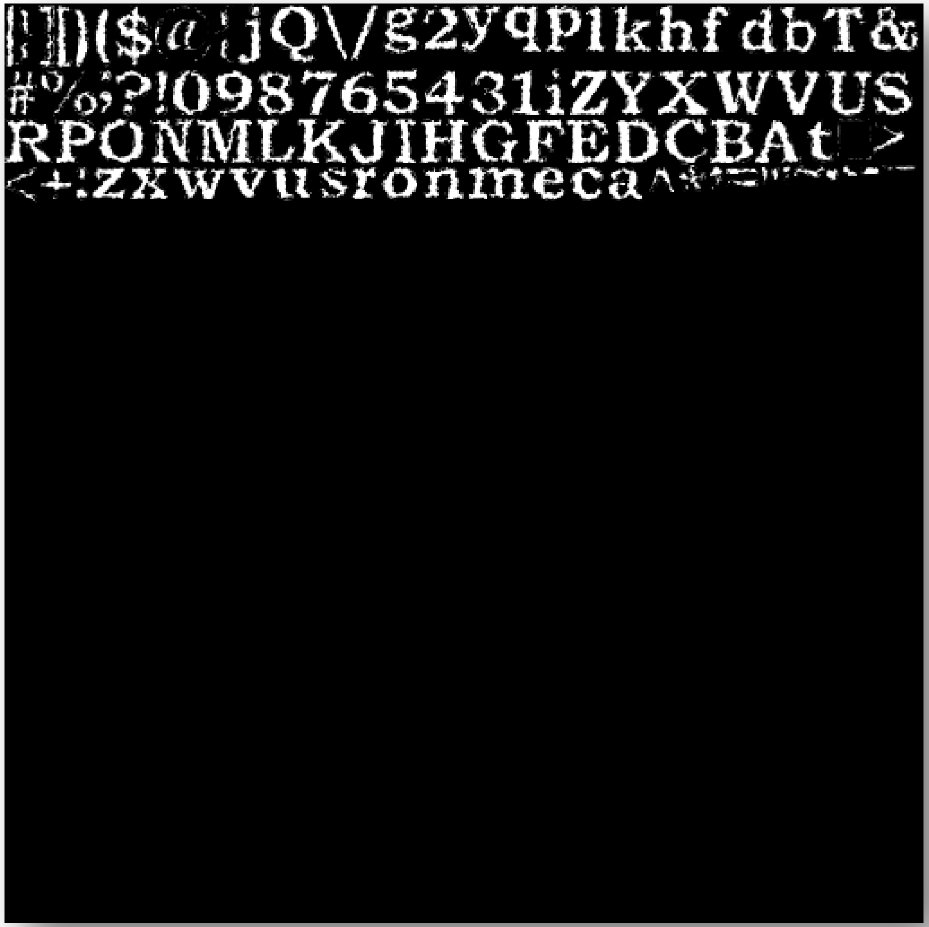
Padding

0 0 0 0

X: 0 Y: 0

Hiero Bitmap Font Tool

- Creates 2 files
  - myFont.fnt
  - myFont.png



Using Bitmap Fonts

```
// Member
BitmapFont font;
// [...]

// init font from files
font = new BitmapFont(new FileHandle(new File("myFont.fnt")),
            new FileHandle(new File("myFont.png")), false);
// [...]

// draw in the main loop / sprite batch
font.draw(spriteBatch, "3765 pts.", 5, Gdx.graphics.getHeight() -5);
```

Input

- Interface to the input facilities
- Allows to poll state of
  - the keyboard
  - touch screen
  - accelerometer
- Event based input available as InputProcessor

Music

- Music interface represents streamed audio
  - Music instance is created with Audio.newMusic(FileHandle)
  - Looping, volume, pause and play

Sound

- Sound interface represents in-memory audio clips
  - Feasible for small files
  - Can be played multiple times at once
  - Looping, volume, play and stop

Walkthrough …

- GdxTestGame …

Vielen Dank ...

... für die Aufmerksamkeit