# Computer Games 2011
# Selected Game Engines

Dr. Mathias Lux
Klagenfurt University

# libGDX features

- High-performance, cross-platform game development framework.
  - Android 2.0+
  - Windows, Linux & MacOS (Java)
- Basis for engines and games.
  - e.g. AndEngine

# libGDX features

- Multiple backends
  - Jogl, LWJGL, Angle (NVIDIA 3D Vision) and the Android APIs
- 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

# libGDX features

- **Audio**
  - Music and SFX from WAV, MP3 and OGG
  - Fast Fourier transforms
  - Audio decoding of OGG and MP3 via JNI
  - Direct interface to audio device
- **File I/O**
  - Abstracting Android assets, classpath and file-system

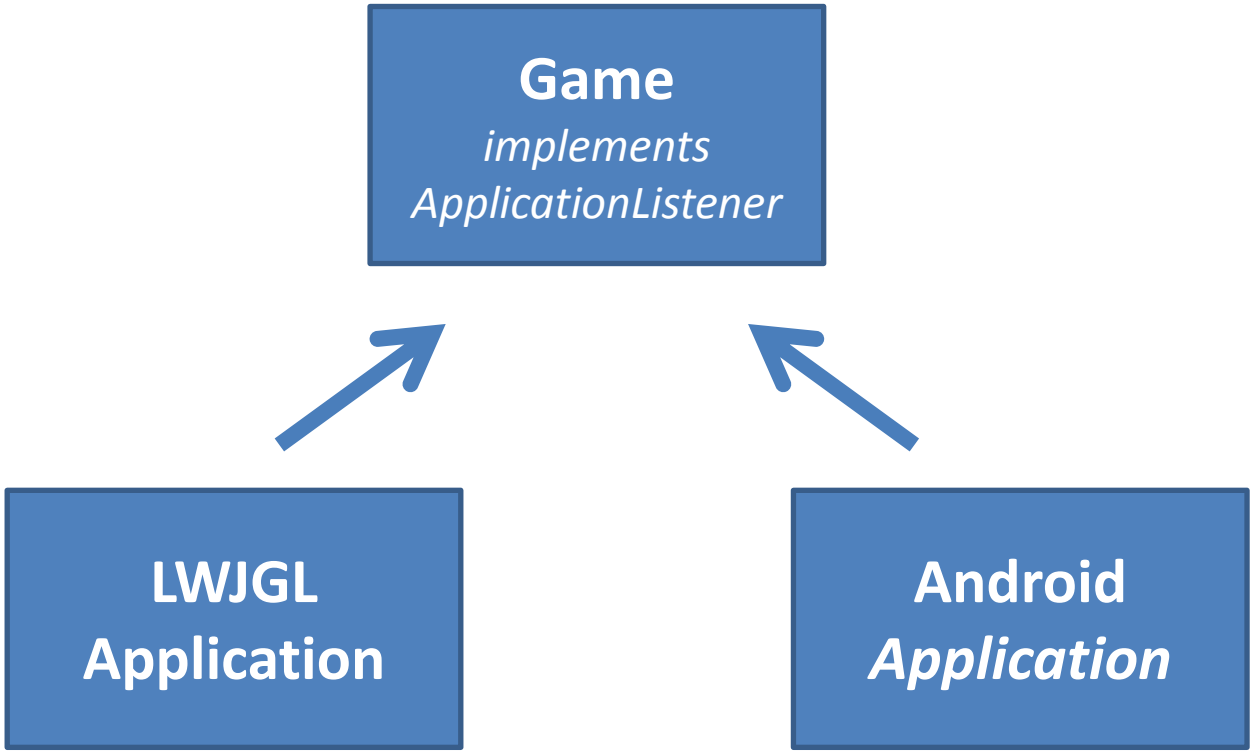
# libGDX features

- **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
  - Hiero bitmap font generator
  - Texture packer
  - Themable Widget Library support

# 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);
    }
}
```



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
  - Always $2^n$ side length (eg. 64x64 or 16x256)
- SpriteBatch
  - Takes care of displaying textures
  - Texture mapping and creation of rectangles

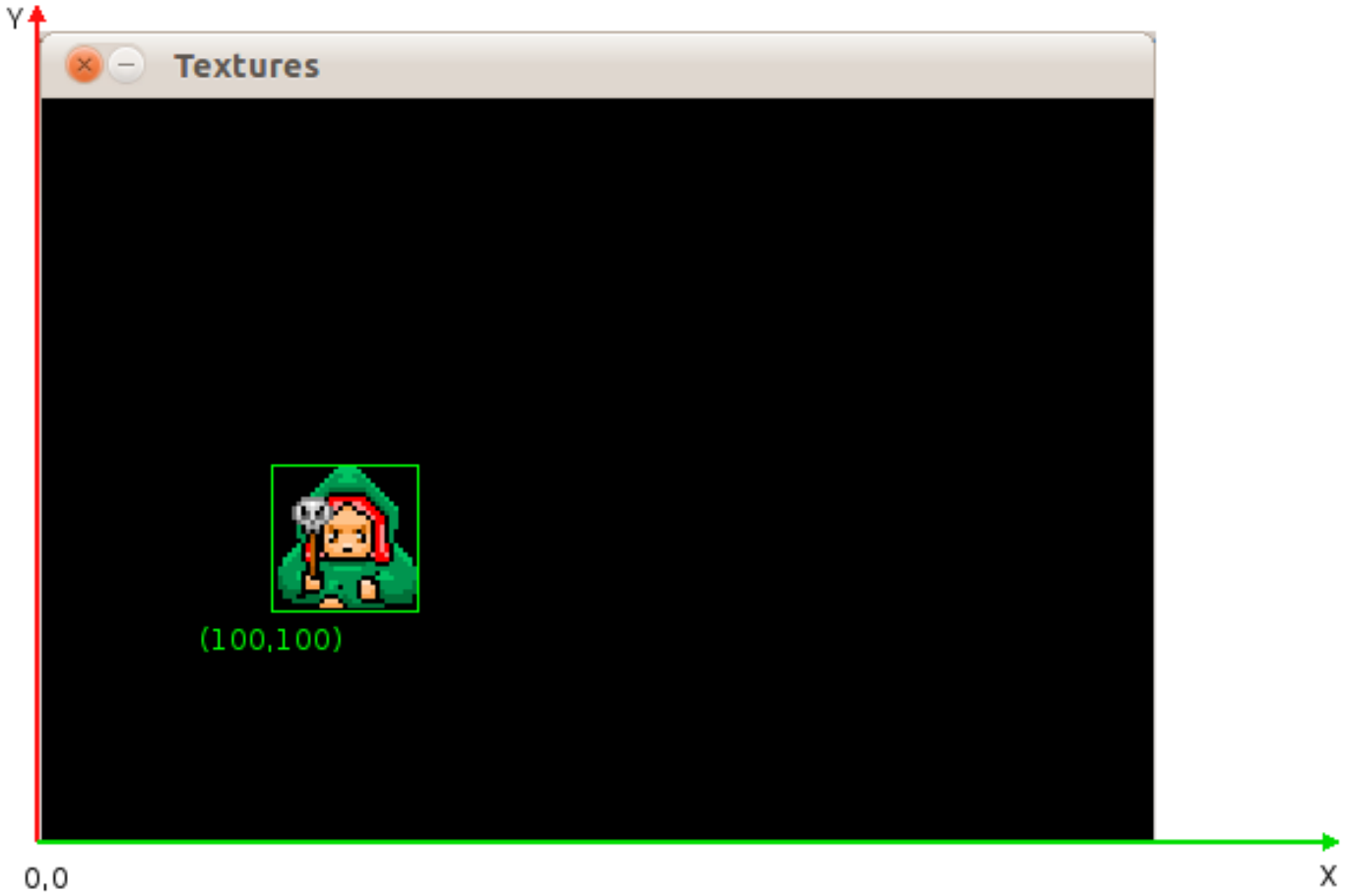
# Drawing a single texture

```
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

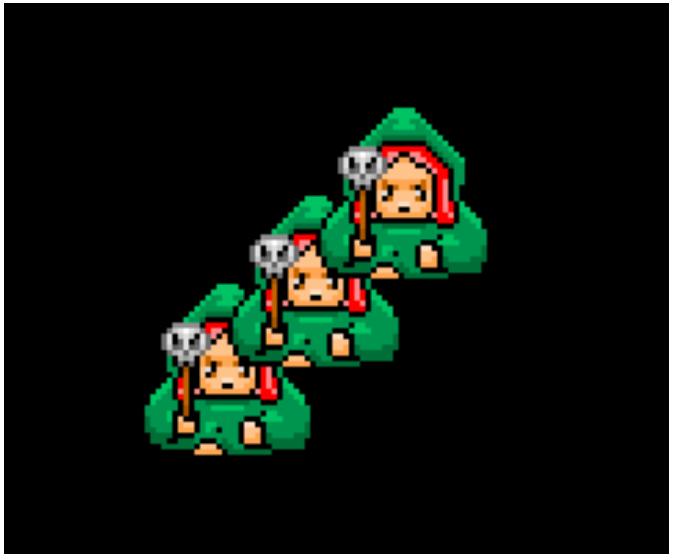
# Multiple Calls

```
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

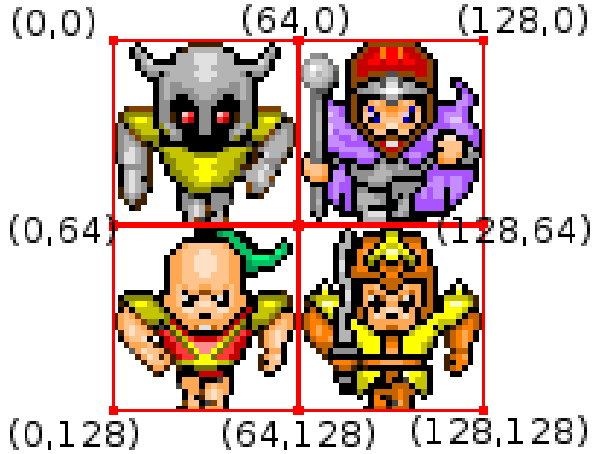
# Rotation & Scaling

```
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)

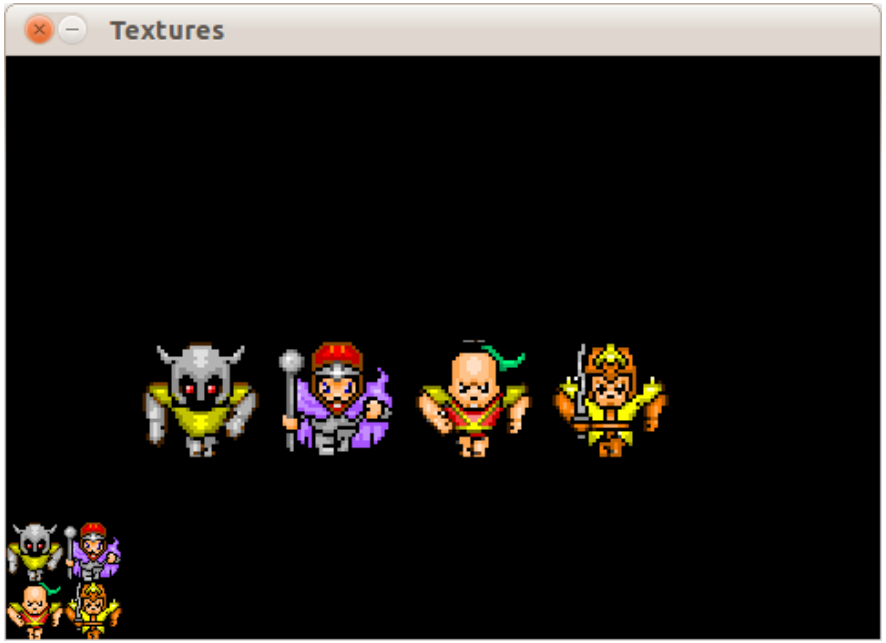



Regions with coordinates

# 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

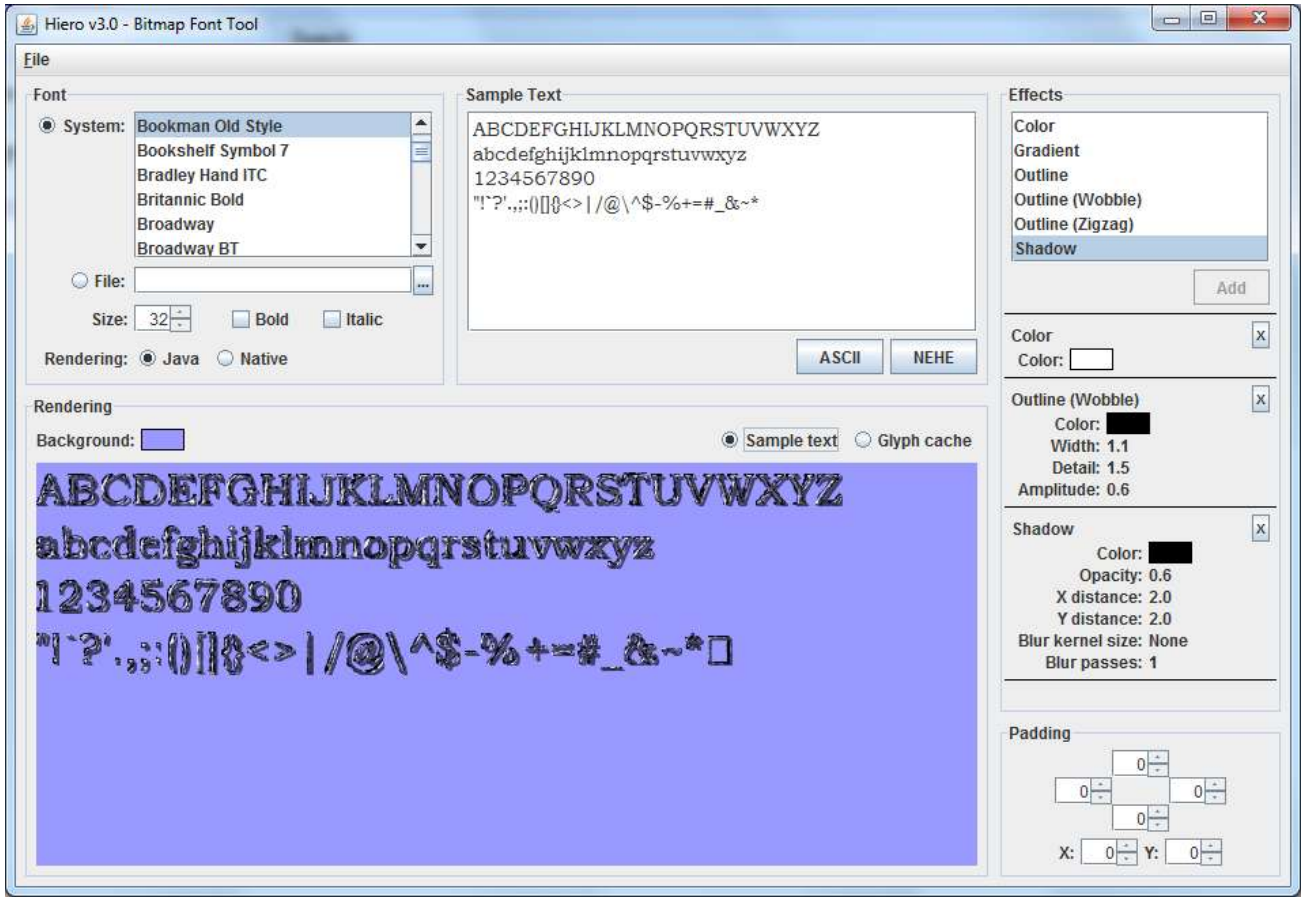



```
...
TextureRegion[][] regions = TextureRegion.split(texture, 64, 64)
...
```

# Blending & Viewport

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

# 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
X:
Y:

# 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 …

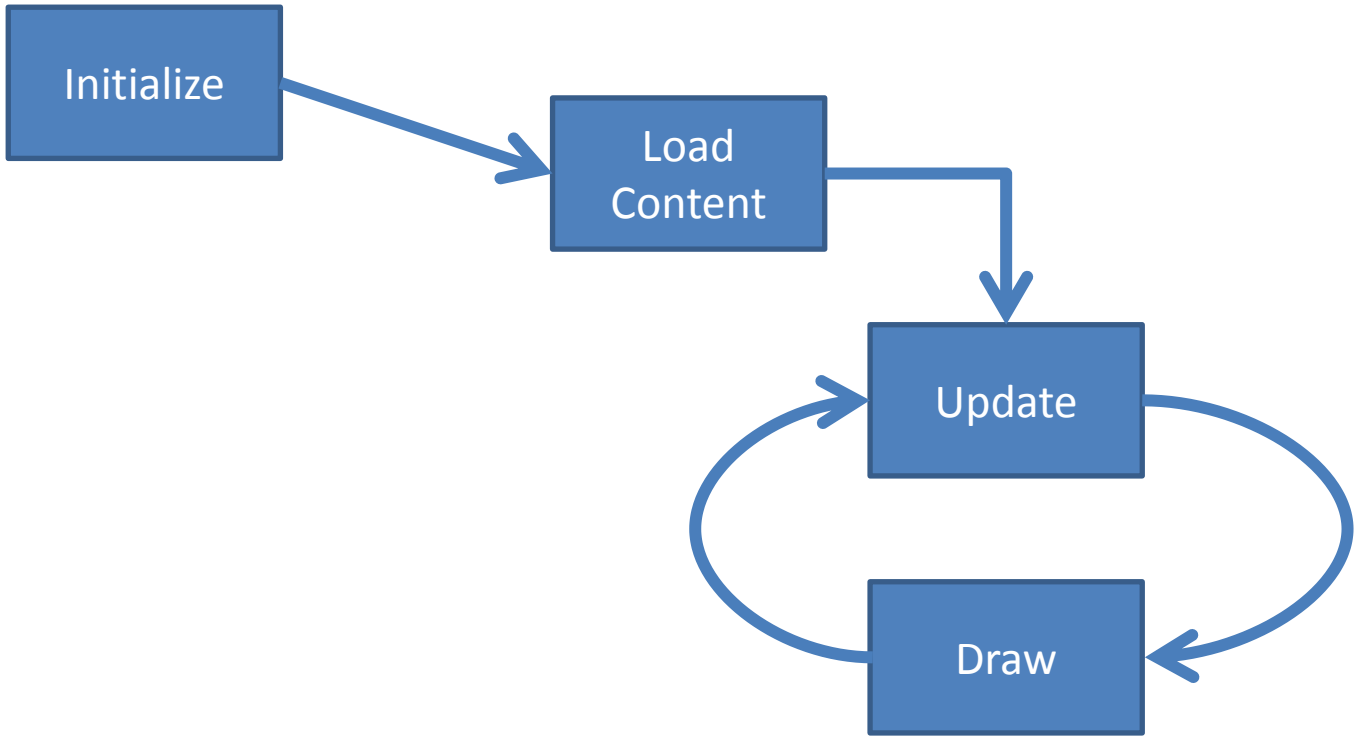
# What is MS XNA?

- A framework & managed runtime for video game development
- XNA stands for "**X**NA is **N**ot an **A**cronym“
- XNA is based on the .NET Compact Framework
- XNA lets you develop for
  - Windows, Windows Phone, Xbox & Zune

# What is MS XNA?

- Integrates with Visual Studio
- C# und VB supported
- Windows Phone Emulator

# XNA Basic Structure



# XNA Basic Structure

```
/// Allows the game to perform any initialization it needs to before starting to run.
protected override void Initialize()

/// LoadContent will be called once per game and is the place to load
/// all of your content.
protected override void LoadContent()

/// UnloadContent will be called once per game and is the place to unload all content.
protected override void UnloadContent()

/// Allows the game to run logic such as updating the world,
/// checking for collisions, gathering input, and playing audio.
protected override void Update(GameTime gameTime)


/// This is called when the game should draw itself.
protected override void Draw(GameTime gameTime)
```

# Sprites

```
// --< in LoadContent() >--
// Create a new SpriteBatch
spriteBatch = new SpriteBatch(GraphicsDevice);
// Sprite and position
sprite = Content.Load<Texture2D>("speaker");
posSprite = new Vector2(graphics.GraphicsDevice.Viewport.Width / 2 -
   sprite.Width/2,
   graphics.GraphicsDevice.Viewport.Height - 60 - sprite.Height);

// --< in Draw(GameTime gameTime) >--
GraphicsDevice.Clear(Color.Black);
spriteBatch.Begin();
spriteBatch.Draw(sprite, posSprite, Color.White);
spriteBatch.End();
```

# Sprites

- Rotation
  - spriteBatch.Draw(sprite, screenpos, null, Color.White, RotationAngle, origin, 1.0f, SpriteEffects.None, 0f)
  - origin.x, origin.y give the center of rotation relative to sprite
- Scaling
- Tinting

# Drawing Text

```
SpriteFont Font1;
Vector2 FontPos;
protected override void LoadContent()
{
    spriteBatch = new SpriteBatch(GraphicsDevice);
    Font1 = Content.Load<SpriteFont>("Courier New");
    FontPos = new Vector2(graphics.GraphicsDevice.Viewport.Width / 2,
        graphics.GraphicsDevice.Viewport.Height / 2);
}
```

- Loading from a true type font.

# Drawing text

```
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);
    spriteBatch.Begin();
    string output = "Hello World";
    Vector2 FontOrigin = Font1.MeasureString(output) / 2;
    spriteBatch.DrawString(Font1, output, FontPos,
        Color.LightGreen, 0, FontOrigin, 1.0f,
        SpriteEffects.None, 0.5f);
    spriteBatch.End();
    base.Draw(gameTime);
}
```

# Getting Input

```
// --< in Update(GameTime gameTime) >--
if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
   ButtonState.Pressed)
   this.Exit(); // exit game
if (GamePad.GetState(PlayerIndex.One).DPad.Left ==
   ButtonState.Pressed)
   posSprite.X -= 1; // move sprite to the left
if (GamePad.GetState(PlayerIndex.One).DPad.Right ==
   ButtonState.Pressed)
   posSprite.X += 1; // move sprite to the right
// analog input
posSprite.X +=
   GamePad.GetState(PlayerIndex.One).ThumbSticks.Left.X * 5;
```

# Sound Effects

```
SoundEffect soundEffect;
// ...
soundEffect = Content.Load<SoundEffect>("kaboom");
// ...
instance.IsLooped = true; // in case of looping
soundEffect.Play();

// streaming sound
soundfile = TitleContainer.OpenStream
   (@"Content\tx0_fire1.wav");
soundEffect = SoundEffect.FromStream(soundfile);
```

# XNA Additional Features

- Xbox Live
  - gamer profiles, presence, avatars
- Networking & multiplayer
  - finding & joining sessions
  - sending & receiving network data
  - network session management
- Packaging & sharing

# Vielen Dank …

… für die Aufmerksamkeit