SFML Code Referenz

Inhalt

Shapes

Shapes erstellen

```
// Deklariere hier Objekte oder Klassenvariablen!

RectangleShape box;
CircleShape circle;

// Setup, wird immer einmal zu Beginn eines Screens aufgerufen
// Hier Startwerte setzen!
public override void setup()
{
    box = new RectangleShape();
    box.Size = new Vector2f(50, 50);
    box.Position = new Vector2f(400, 300);
    box.FillColor = Color.Green;
    box.Origin = box.Size * 0.5f;
    box.Rotation = 45f;

    circle = new CircleShape();
    circle.Radius = 25;
    circle.Position = new Vector2f(400, 300);
    circle.FillColor = Color.Blue;
    circle.Origin = new Vector2f(1, 1) * circle.Radius;
    circle.Rotation = 45f;
}
```

Shapes zeichnen

```
// Render wird immer nach loop aufgerufen
// Hier alle draw Befehle aufrufen
public override void render()
{
    draw(box);
    draw(circle);
}
```

Shapes bewegen/drehen

```
// Loop, wird jeden Frame (60 mal die Sekunde) aufgerufen
// Hier alle Berechnungen für die Veränderung von Objekten einfügen
public override void loop()
{
    // Bewegen: 3 nach rechts und 2 nach oben
    circle.Position += new Vector2f(3, -2);

    // Drehen: 2*60 = 120 Grad pro Sekunde
    box.Rotation += 2f;

}
```

Sprites

Sprite erstellen

```
// Deklariere hier Objekte oder Klassenvariablen!
Sprite sprite;

// Setup, wird immer einmal zu Beginn eines Screens aufgerufen
// Hier Startwerte setzen!
public override void setup()
{
    //Bild sollte im Asset Ordner liegen, nicht vergessen in den Eigenschaften->"kopieren wenn neuer" einzustellen!
    Texture tex = new Texture("assets/myTexture.png");

    sprite = new Sprite(tex);

    sprite.Position = new Vector2f(400,300);
    sprite.Scale = new Vector2f(2, 2);
    sprite.Origin = (Vector2f)tex.Size * 0.5f;

}
```

Sprite zeichnen

```
// Render wird immer nach loop aufgerufen
// Hier alle draw Befehle aufrufen
public override void render()
{
    draw(sprite);
}
```

Text

Text erstellen

```
// Deklariere hier Objekte oder Klassenvariablen!
Text text;

// Setup, wird immer einmal zu Beginn eines Screens aufgerufen
// Hier Startwerte setzen!
public override void setup()
{
    //Font datei muss in assets liegen, arial.ttf ist im Template enthalten
    Font font = new Font("assets/arial.ttf");

    //Höhe eines großen Buchstaben in Pixel
    uint textGroesse = 30;

    text = new Text("This is my Text", font, textGroesse);


}
```

Text zeichnen

```
// Render wird immer nach loop aufgerufen
// Hier alle draw Befehle aufrufen
public override void render()
{
    draw(text);
}
```

Text ändern

```
// Loop, wird jeden Frame (60 mal die Sekunde) aufgerufen
// Hier alle Berechnungen für die Veränderung von Objekten einfügen
public override void loop()
{

    text.DisplayedString = "Ersetzt alten Text.";

    text.DisplayedString += "Alten Text nicht löschen, nur dranhängen.";
}
```

Arrays

Array erstellen

```
// Deklariere hier Objekte oder Klassenvariablen!

RectangleShape[] boxArray;

// Setup, wird immer einmal zu Beginn eines Screens aufgerufen
// Hier Startwerte setzen!
public override void setup()
{
    // Erstelle Array mit 100 Shapes (array ist noch leer)
    boxArray = new RectangleShape[100];

    //Fülle Array
    for (int i = 0; i < boxArray.Length; i++)
    {
        //Erstelle lokales Shape und füge es hinzu
        RectangleShape box = new RectangleShape();
        box.Size = new Vector2f(20,20);
        box.FillColor = Color.Yellow;
        box.Position = new Vector2f(100 * (i/8), 100 * (i%8));
        boxArray[i] = box;
    }

}
```

Einträge im Array verändern

```
// Loop, wird jeden Frame (60 mal die Sekunde) aufgerufen
// Hier alle Berechnungen für die Veränderung von Objekten einfügen
public override void loop()
{
    // Bewege jedes Shape im Array
    for (int i = 0; i < boxArray.Length; i++)
    {
        boxArray[i].Position += new Vector2f(i*0.5f, 0);
    }
}
```

Shapes im Array zeichnen

```
// Render wird immer nach loop aufgerufen
// Hier alle draw Befehle aufrufen
public override void render()
{
    // Zeichne jedes Shape im Array
    for (int i = 0; i < boxArray.Length; i++)
    {
        draw(boxArray[i]);
    }

}
```

Listen

Listen erstellen

```
List<CircleShape> circleList;

// Setup, wird immer einmal zu Beginn eines Screens aufgerufen
// Hier Startwerte setzen!
public override void setup()
{
    //Erstelle leere Liste
    circleList = new List<CircleShape>();


}
```

Listenelemente hinzufügen

```
// Loop, wird jeden Frame (60 mal die Sekunde) aufgerufen
// Hier alle Berechnungen für die Veränderung von Objekten einfügen
public override void loop()
{
    //Erstelle lokales CircleShape
    CircleShape circle = new CircleShape();
    circle.Radius = 30;
    circle.FillColor = Color.Red;

    //Füge circleShape in Liste ein
    circleList.Add(circle);
}
```

Shapes in Liste zeichnen

```
// Render wird immer nach loop aufgerufen
// Hier alle draw Befehle aufrufen
public override void render()
{
    for (int i = 0; i < circleList.Count; i++)
    {
        draw(circleList[i]);
    }

}
```

Listenelemente entfernen

```
// Loop, wird jeden Frame (60 mal die Sekunde) aufgerufen
// Hier alle Berechnungen für die Veränderung von Objekten einfügen
public override void loop()
{
    for (int i = circleList.Count - 1; i >= 0; i--)
    {
        if(circleList[i].Position.X > 800)
        {
            circleList.RemoveAt(i);
        }
    }

}
```

Input

Tastenstatus abfragen

```
// Loop, wird jeden Frame (60 mal die Sekunde) aufgerufen
// Hier alle Berechnungen für die Veränderung von Objekten einfügen
public override void loop()
{

    if(Keyboard.IsKeyPressed(Keyboard.Key.Space))
    {
        //Space wird gedrückt gehalten
    }

    if (Mouse.IsButtonPressed(Mouse.Button.Left))
    {
        //Linke Maustaste wird gedrückt gehalten
    }
}
```

Events abfangen

```
// Setup, wird immer einmal zu Beginn eines Screens aufgerufen
// Hier Startwerte setzen!
public override void setup()
{

    game.gameWindow.KeyPressed += onKeyPress;

    game.gameWindow.MouseWheelMoved += onMouseWheelMove;
}

private void onMouseWheelMove(object sender, MouseWheelEventArgs e)
{
    if(e.Delta > 0)
    {
        //Mausrad wurde nach oben bewegt
    }
    else if (e.Delta < 0)
    {
        //Mausrad wurde nach unten bewegt
    }
}

private void onKeyPress(object sender, KeyEventArgs e)
{
    if(e.Code == Keyboard.Key.Space)
    {
        // Space wurde gedrückt (einmaliger Aufruf)
    }
    else if(e.Code == Keyboard.Key.Return)
    {
        // Enter wurde gedrückt (einmaliger Aufruf)
    }
}
```

Mausposition abfragen

```
// Loop, wird jeden Frame (60 mal die Sekunde) aufgerufen
// Hier alle Berechnungen für die Veränderung von Objekten einfügen
public override void loop()
{
    // Mausposition auslesen und in Vector2f konvertieren
    Vector2f mousePos = (Vector2f)Mouse.GetPosition(game.gameWindow);

}
```

Entities

Entity im Screen einbauen

```
// Deklariere hier Objekte oder Klassenvariablen!
Entity entity;

// Setup, wird immer einmal zu Beginn eines Screens aufgerufen
// Hier Startwerte setzen!
public override void setup()
{
    // Immer this als parentScreen übergeben, gegebenfalls mehr Parameter
    entity = new Entity(this);


}

// Loop, wird jeden Frame (60 mal die Sekunde) aufgerufen
// Hier alle Berechnungen für die Veränderung von Objekten einfügen
public override void loop()
{
    //Entity.loop() von jeder Entity sollte immer einmal pro loop() durchlauf aufgerufen werden
    entity.loop();


}

// Render wird immer nach loop aufgerufen
// Hier alle draw Befehle aufrufen
public override void render()
{
    //Entity.draw() von jeder Entity sollte immer einmal pro loop() durchlauf aufgerufen werden
    entity.draw();
}
```

Entity-Klasse erstellen (in eigener Datei)

```
class Entity
{
    // Graphic Variable sollte in jeder Entity drin sein
    // RectangleShape, CircleShape oder Sprite
    public RectangleShape graphic;

    Screen screen;

    // Konstruktor, eigene Parameter nach parentScreen hinzufügen
    public Entity(Screen parentScreen, Vector2f startPos, Color color)
    {
        screen = parentScreen;
        graphic = new RectangleShape();
        graphic.Size = new Vector2f(50,50);
        graphic.FillColor = color;
        graphic.Position = startPos;
    }

    // Loop Methode
    public void loop()
    {

        graphic.Position += new Vector2f(2,5);

    }

    public void draw()
    {
        // screen.draw statt nur draw in Entities
        screen.draw(this.graphic);
    }


}
```