C# - SFML Klassen für Spielobjekte

Will man in SFML Spielobjekte erstellen, die zu kompliziert sind um sie als einfaches RectangleShape, CircleShape oder Sprite darzustellen, so muss man eine eigene Klasse für diese Objekte erstellen. Diese Art von Spieleobjekte nennt man "Entity".

Im Template gibt es neben der MainScreen Klasse auch andere Klassen. Unter dem Ordner entities/ gibt es die Klasse Entity. Diese sieht wie folgt aus:

using System;

using SFML.Window;

using SFML.Audio;

using SFML.Graphics;

using SFML.System;

using System.Collections.Generic;

using System.Text;

using SFML\_GAME.framework;

using SFML\_GAME.screens;

namespace SFML\_GAME.entities

{

class Entity

{

// Graphic Variable sollte in jeder Entity drin sein

// RectangleShape,CircleShape oder Sprite

public RectangleShape graphic;

MainScreen screen;

// Konstruktor

public Entity(MainScreen parentScreen)

{

screen = parentScreen;

graphic = new RectangleShape();

graphic.Size = new Vector2f(50, 50);

graphic.FillColor = Color.Blue;

graphic.Position = new Vector2f(300, 300);

}

// Loop Methode

public void loop()

{

}

public void draw()

{

screen.draw(this.graphic);

}

}

}

Sie besteht aus den Methoden:

1. Entity: Der Konstruktor der Entity, wird bei der Erstellung der Entity aufgerufen. Kann wie setup() in MainScreen benutzt werden.
2. loop: soll jeden Frame aufgerufen werden. Kann wie loop() in MainScreen benutzt werden.
3. draw: In dieser Methode sollen alle Objekte die zur Entity gehören gedrawt werden.

In der Konstruktor-Methode(Entity) wird ein Screen übergeben. Dieser dient der Entity als Referenz, wo es existiert. Jede Entity sollte diese Variable speichern.

Außerdem wird schon ein RectangleShape erstellt und ihm Werte zugewiesen.   
In der draw Methode wird das RectangleShape auf dem Screen gedrawt.

In unserem MainScreen können wir nun die Entity erstellen, als würden wir andere Objekte(Sprite, RectangleShape, CircleShape) erstellen. Allerdings müssen wir bei der initialisierung (new Entity()) zwischen die Runden Klammern ein "this" schreiben, damit die Entity den Parameter bekommt. In loop() sollten wir die loop() und draw() Methode von ihr aufrufen.

namespace SFML\_GAME.screens

{

class MainScreen : Screen

{

// Deklariere hier Objekte oder Klassenvariablen!

Entity box;

// Setup, wird immer einmal zu Beginn eines Screens aufgerufen

// Hier Startwerte setzen!

public override void setup()

{

box = new Entity(this);

}

// Loop, wird jeden Frame (60 mal die Sekunde) aufgerufen

public override void loop()

{

box.loop();

box.draw();

}

}

}

Jetzt können wir in der Entity in ihrer loop() mit dem Shape arbeiten wie wir es von MainScreen gewohnt sind.