8.2 Sprites & Sounds

Will man in SFML ein Bild einfügen, so muss man Sprites und Texturen dafür nutzen. Die "Sprite" Klasse lässt sich fast genauso benutzen wie ein "RectangleShape". Wir können also in Mainscreen ein Sprite erstellen.

public Sprite graphic;

Wenn wir unsere Sprite initialisieren mit "new", müssen wir allerdings eine Textur angeben. Die Textur stellt hierbei die Bilddatei dar, die wir benutzen. Zum Testen benutzen wir die "duck.png".

Wir müssen also erst unser Bild in eine Textur laden und dann können wir das Sprite mit der Textur erstellen.

Texture duck = new Texture("assets/duck.png");

graphic = new Sprite(duck);

Hierbei sollte man folgendes beachten:

Eine Textur belegt viel Speicher und sie zu erstellen dauert lange. Wenn möglich sollte man **nie mehrere Texturen erstellen mit demselben Bild**, da es Platzverschwendung ist! Sprites dagegen verbrauchen nicht viel Platz oder Leistung.

Jetzt sollte unsere Entity die Ente als Bild haben.

Eigenschaften wie Fillcolor oder Size sind nichtmehr vorhanden, da Sprite automatisch die Größe seiner Textur annimmt. Möchte man das Sprite trotzdem vergrößern/verkleiner, sollte man die Eigentschaft "Scale" benutzen.

## Musik & Sound

Um in SFML Musik oder Sounds abzuspielen, brauch man dementsprechend ein "Music" oder "Sound" Objekt.

Äquivalent zur Textur/Sprite gibt es hier den "Soundbuffer". Wir müssen unseren Sound also erst in den Soundbuffer laden, und diesen dann unseren Sounds zuweisen. Bei dem "Musik" Objekt brauchen wir diesen allerdings nicht. Auch hier gilt, niemals mehrere Music/Soundbuffer Objekte, die dasselbe beinhalten!

Obwohl Musik auch als langer Sound angesehen werden könnte, gibt es in SFML eine Extra Klasse für Musik, sie ist besser darauf ausgelegt lange und in Schleife laufende Audiospuren auszugeben.

Als Beispiel Benutzen wir hier die "happyTune.ogg" als Musik und die "explosion.wav" als Sound.

Das Abspielen von einem Musik und Sound Objekt ist fast identisch.

SoundBuffer explosion = new SoundBuffer("assets/explosion.wav");

Sound boom = new Sound(explosion);

boom.Play();

Music happyTune;

happyTune = new Music("assets/happyTune.ogg");

happyTune.Play();

Bemerkung:

Bei Sounddateien gibt es viele Formate, die leider nicht unterstützt werden (z. b. mp3). Generell ist das "ogg" Format zu empfehlen. Online gibt es viele Converter die alle möglichen Formate umwandeln können.

Von allen Assets nimmt Audio, insbesondere lange Musik, am meisten Speicherplatz ein. Man sollte hierrauf ein Auge haben. Man kann die kbps(Kilobyte per second) von Audiodateien kleiner wählen(beim konvertieren) um bei gleicher Länge diese zu verkleinern,jedoch auf Kosten der Soundqualität.