



Die Engine-Alpha wurde in **Version 4** von Grund auf neu implementiert und dadurch von vielen unschönen Konstruktionen und inkonsistenter Syntax befreit sowie um neue Features erweitert.

Code für die Version 3 ist NICHT kompatibel mit der Version 4 !!!

Neue Grundprinzipien

- Die **Core-Engine** ist durchgehend in **englischer** Sprache. Diese Bezeichner und Methoden-Namen sind von der EDU-Variante abgekoppelt.
- Alle Methoden-Namen der **EDU-Variante** sind **durchgehend in deutsch**.
- Methoden-Namen** der EDU-Variante **beginnen immer mit einem Verb**:
z.B. `makeAktiv()` statt `aktivMachen()`
- Das **Koordinaten-System** hat seinen **Ursprung** nun **in der Mitte** des Fensters und die **y-Achse** zeigt **nach oben**.
- Die **Einheiten des Bildschirms** sind nun nicht mehr Pixel sondern **Meter** (Bildschirm-Meter im Maßstab 30 Pixel = 1m).
Der Grund dafür liegt in der neuen Physik !
- Die **Einheiten für die Zeit** sind nun **Sekunden** und nicht mehr Millisekunden.
z.B. `tickerIntervallSetzen(0.15f)` statt `tickerInterVallSetzen(150)`
Der Grund dafür liegt in der neuen Physik !
- Alle Einheiten der neuen **Physik** sind **SI-Basis-Einheiten**:
Länge in m , Zeit in s , Geschwindigkeit in m/s , Masse in kg
*Beschleunigung (Schwerkraft) in m/s² , Impuls in kg*m/s*
- Die abstrakte **Super-Klasse** für alle grafisch darstellbaren Klassen heißt nun **EduActor**. (früher: Klasse **RAUM**)
- Die Methode `warte()` von SPIEL ist **inkompatibel** mit `tick()`, `tasteReagieren(...)` und anderen Automatismen.

Neue Features

- Die Klasse SPIEL bietet eine statische Methode, um das **Koordinaten-System ein- und aus zu blenden**.
- Es gibt **mehr** vordefinierte **Farbnamen**. Dabei ist die Groß-/Kleinschreibung egal. Auch ä, ö, ü, ß werden akzeptiert. Außerdem wurde eine **neue Methode** `setzeFarbe(rot, guren, blau)` hinzu gefügt.
- Bei **Tastatur** und **Maus** kann nun **auch** auf das **loslassen** reagiert werden. Auch das **Mausrad** kann eingesetzt werden.
- Interfaces** ermöglichen es nun, diese Effekte (**Ticker, Maus, Tastatur**) **auch direkt auf Objekte** anzuwenden, nicht nur auf das SPIEL-Objekt.
- Die Klassen **BILD, WECHSELBILD** und **FIGUR** wurden zu einer **neuen Klasse FIGUR** vereint. Diese kann nun **Einzel-Bilder, Animierte Bilder, Bild-Folgen, Ordner mit Bildern und SpriteSheets** verarbeiten.
- Für die neue **Klasse FIGUR** gibt es nun **Zustands-basierte Animationen** (Zustands-Automat mit automatischen und Event-getriebenen Übergängen). Diese ermöglicht zusammen mit mehreren animierten Bildern bzw. Bildfolgen völlig neuartiges Verhalten von Spielfiguren.
- Für Objekte können **individuelle Collider** (zur Kollisions-Kontrolle) angelegt werden. Damit wird realistischeres Verhalten möglich.
- Die **Grafik arbeitet** nun **Frame-weise**. Es gibt auch einen „FrameUpdateListener“ (Interface `BildAktualisierungReagierbar`), der es ermöglicht, Echtzeit-Anwendungen zu realisieren wie z.B. die **Methode kleiner Schritte**.
- Es gibt **Reibung und Elastizität**. Dies ermöglicht sehr realistische Bewegungen.
- Auf Wunsch kann man für jedes Objekt ein **„Rotation-Lock“ setzen oder lösen** werden, so werden Rotationen zusätzlich zu **Bewegungen** möglich, die sehr **realistisch wirken**.
- Man kann mehrere **Objekte auf verschiedene Art verbinden** und damit z.B. **Gelenke** oder **Ketten** nachbilden.
- Man kann mit der **Methode parallel() der Klasse **SPIEL** nun beliebige Abläufe parallel (nebenläufig) und kompatibel zu anderen Automatismen absetzen.**
- Für **Menüs** oder **Levels** können nun durch **Scenes** realisiert werden. Zwischen Scenes kann beliebig gewechselt werden. Inaktive Scenes werden eingefroren. Bewegungen setzen sich bei Reaktivierung dort fort, wo sie aufgehört haben.