

Ziele:

- HTML- sowie CSS-Kenntnisse auffrischen
- Die Bibliotheken von CreateJS kennenlernen und die Dokumentation nutzen
- Debugging im Browser kennenlernen
- Grundkonzepte beherrschen
- Einfache Animationen beherrschen
- Sounds abspielen
- Bilder und Sounds effizient laden



Aufgabe 1-1: Vorhang auf, Bühne frei...

- Erstellen Sie eine HTML-Datei mit einem *canvas*-Element. Geben Sie dem *canvas*-Element eine Größe von 1000 x 800 Pixeln und eine *id*.
- Binden Sie CreateJS in ihre Datei ein.
- Erweitern Sie nun im `<script></script>`-Bereich ihrer Datei die *init()*-Methode, indem Sie ein Bühnenobjekt erzeugen, das auf das bereits erstellte *canvas*-Element verweist.
- Stellen Sie zunächst über *createjs.Ticker.setFPS()* eine Framerate von 25 fps ein und registrieren Sie einen EventListener, der in jedem Frame die Inhalte der Bühne über die bereits vorhanden Funktion *tick()* neu zeichnet.
- Zeigen Sie das Bild *background.png* als Hintergrundbild an, indem Sie die Methode *drawBackground()* vervollständigen.
- Geben Sie auf der Bühne den Text Hello World! aus.
- Zeichnen Sie einen Kreis, ein Viereck sowie ein Dreieck in beliebiger Farbe.

Aufgabe 1-2: CSS

Verschönern Sie das gesamte HTML-Dokument. Informieren Sie sich auch über die Funktionen von CSS3, die zum Teil schon in vielen Browsern implementiert sind.

- Gestalten Sie den Hintergrund der Webseite.
- Zentrieren Sie das canvas-Element auf der Seite und heben sie das canvas-Element optisch vom Hintergrund ab.
- Fügen Sie eine Überschrift ein. Verwenden Sie z.B. eine Schriftart von Google Webfonts und fügen Sie mit CSS einen Schlagschatten hinzu.

Aufgabe 1-3 Windrad mit Tween-Animation

CreateJS enthält mit TweenJS auch eine Bibliothek, mit der man Tweens erstellen kann. Damit kann man beliebige Attribute von Display-Objekten animieren (Interpolation). Erstellen Sie ein vereinfachtes Windrad, dessen Blätter sich drehen. Vervollständigen Sie hierzu die Methode *drawWindmill()*. Die Methode definiert bereits die Display-Objekte für die Elemente des Windrads, diese müssen jedoch noch ausgestaltet werden.

- Definieren Sie zunächst eine Form, die den Mast des Windrads darstellt.
- Fügen Sie dann dem *blades*-Container Rechtecke für die einzelnen Blätter des Windrads hinzu. Achten Sie darauf, dass sie einen geeigneten Registrierungspunkt für die Rotation der einzelnen Blätter wählen (Attribute *regX* und *regY*).
- Animieren Sie den *blades*-Container mit einem Tween, sodass sich das Windrad fortlaufend dreht.
- Fügen Sie dem *windmill*-Container den Mast sowie den *blades*-Container hinzu und platzieren Sie das Windrad in der Landschaft.
- Optional: Ersetzen sie rechteckigen Blätter des Windrads durch Instanzen der Bilddatei *blade.png*.

Aufgabe 1-4: Bitmap-Animation

Sehen Sie sich die Datei *sprite.png* an. Ziel ist es, die Einzelbilder schnell hintereinander abzuspielen, sodass der Eindruck einer fließenden Bewegung entsteht. *EaselJS* stellt hierfür die Komponenten *SpriteSheet* und *BitmapAnimation* bereit. Vervollständigen sie die Methode *drawWalker()*.

- Erstellen Sie dann ein *SpriteSheet*-Objekt. Lesen Sie hierzu die Dokumentation unter <http://www.createjs.com/Docs/EaselJS/classes/SpriteSheet.html>.
- Erstellen Sie eine *BitmapAnimation* aus dem *SpriteSheet*, starten Sie die Animation mit dem *gotoAndPlay()*-Befehl und fügen Sie die Animation der Bühne hinzu.
- Die Figur soll nun fortlaufend von links nach rechts durchs Bild laufen.

Aufgabe 1-5 Sounds und Bilder laden

Mit PreloadJS kann man Dateien, auf die man in seiner Anwendung zurückgreift, einheitlich laden und organisieren. Dabei kann eine sog. LoadQueue behilflich sein, der man eine Liste mit zu ladenden Dateien übergibt und die während des Ladevorgangs verschiedene Events verschicken kann, z.B. ein *complete*-Event, das signalisiert, dass alle Dateien fertig geladen sind.

Mit SoundJS lassen sich relativ einfach Audiodateien im Webbrowser abspielen. Dabei wird versucht, Klänge zunächst mit der Webaudio-API wiederzugeben, wobei es verschiedene Fallback-Möglichkeiten gibt.

Vervollständigen Sie die *init()*-Methode, indem Sie alle Bilder sowie die Datei *steps.mp3* über die *LoadQueue* vorladen.

- Fügen Sie der Variable *manifest* eine Liste mit den zu ladenden Dateien hinzu.
- Implementieren Sie die Funktion *handleComplete()*, so dass alle Display-Objekte erst gezeichnet werden, wenn alle Bilder geladen sind.
- Fügen Sie der Methode *drawWalker()* einen Aufruf der Methode *playSound(id)* hinzu, damit man die Schritte der Figur hört.