

Übungsblatt 5 – Multimediaprogrammierung

Achtung: Zur Lösung dieser Übungsaufgabe dürfen ausschließlich die Module verwendet werden, die mit der Standardinstallation von Python 2.5.4 und Pygame 1.8.1 kommen. Abgaben, die zusätzliche Module enthalten oder nicht in Python 2.5.4 geschrieben sind erhalten automatisch null Punkte.

Zur Lösung der Programmieraufgaben können Sie Ihre Lösung aus Übungsblatt 4 verwenden. Alternativ können Sie auch die Beispiellösung von der Webseite verwenden.

Aufgabe 1: Eine Klasse für Sprite-Animationen (6 Punkte)

- a) Erstellen Sie ein Modul „spriteanimation.py“. Definieren Sie in diesem eine Klasse „Animation“. Die Klasse soll folgende Eigenschaften erfüllen. Achten Sie auf die korrekte Schreibweise aller Methoden etc.
 - a. Der Konstruktor erwartet als Argument eine beliebig lange Liste bzw. ein beliebig langes Tupel von Surface-Objekten. Jedes Surface steht für ein Frame der Animation (in der definierten Reihenfolge).
 - b. Die Klasse hat folgende Methoden:
 - i. *set_frames(list)* ersetzt die aktuellen Frames durch eine neue Liste (bzw. Tupel).
 - ii. *add_frame(Surface, pos)* Setzt ein neues Frame (Surface) an einer Position (ein Integerwert mit $0 \leq pos \leq$ Länge der Liste) in die Animation ein. Entspricht die Position der Länge der Liste, dann wird ein weiteres Frame an die Animation angehängt. Ist sie kleiner, dann wird das Frame an dieser Position eingefügt und die folgenden Frames um eine Position verschoben. Wird pos nicht angegeben, wird das Frame automatisch hinten an die Liste angehängt.
 - iii. *replace_frame(Surface, pos)* ersetzt das Frame an einer angegebenen Position durch ein neues Surface.
 - iv. *remove_frame(pos)* entfernt das Frame der angegebenen Position.
 - v. *set_framerate(rate)* bestimmt wie viele Frames pro Sekunde angezeigt werden sollen. Wurde diese Methode nicht ausgeführt, dann wird als Standardwert 30 Frames pro Sekunde angenommen.
 - vi. *get_frame(time_passed)* liefert ein Surface zurück, das dem aktuellen Frame entspricht. Das Argument *time_passed* gibt an, wie viele Sekunden seit dem letzten Aufruf der Methode vergangen sind.
 - vii. *reset_animation()* setzt die Animation zurück auf das erste Frame.
- b) Verwenden Sie nun diese Klasse, um die Bewegungs-Animation aus Übungsblatt 4 zu realisieren.

Achten Sie darauf, dass sich alle Bilddateien in Ihrer Abgabe befinden. Um Probleme auf verschiedenen Systemen zu vermeiden sollten sich die Bilddateien im gleichen Verzeichnis wie Ihr Skript befinden.

Erstellen Sie in Ihrer Abgabe einen Ordner „aufgabe1“, der die Lösung zu dieser Aufgabe enthält.

Aufgabe 2: Erweiterte Kollisionsabfrage (4 Punkte) (optional für Bachelor-Studierende)

In Aufgabe zwei aus Übungsblatt 4 wurden mehrere Spielfiguren animiert. In dieser Aufgabe soll eine Kollisionsabfrage in diese Anwendung eingebaut werden.

Berührt eine der Spielfiguren eine andere, werden diese solange markiert (z.B. mit einem bestimmten Symbol auf der Figur), bis die Kollision vorbei ist. Die Kollisionsabfrage soll hierbei pixelgenau funktionieren. D.h. transparente Bereiche der Spielfigur verursachen keine Kollision. Um diese Aufgabe zu Lösen schauen Sie sich das pygame Modul „mask“ an (<http://www.pygame.org/docs/ref/mask.html>). Die Kollision mit Bäumen und Büschen muss nicht abgefragt werden.

Achten Sie darauf, dass sich alle Bilddateien in Ihrer Abgabe befinden. Um Probleme auf verschiedenen Systemen zu vermeiden sollten sich die Bilddateien im gleichen Verzeichnis wie Ihr Skript befinden.

Erstellen Sie in Ihrer Abgabe einen Ordner „aufgabe2“, der die Lösung zu dieser Aufgabe enthält.

Aufgabe 3: Schlüsselbildanimation (4 Punkte) (optional für Diplom-Studierende)

Sie kennen den Begriff Schlüsselbild (keyframe) schon aus der Videokompression. Hierbei werden nur einige der Teilbilder eines Videos vollständig gespeichert. Zwischenbilder werden anhand bestimmter Regeln berechnet.

Eine ähnliche Technik wird auch bei der so genannten Schlüsselbildanimation (keyframe animation) verwendet. Erklären Sie den Begriff Schlüsselbildanimation. Gehen Sie hierbei auch auf den Begriff der Interpolation ein. Bei welcher Art von Animationen bietet sich diese Technik besonders an? Bei welchen Animationen stößt sie schnell an ihre Grenzen?

Skizzieren Sie ein Beispiel, bei dem sich diese Art der Animation besonders gut anbietet. Zeichnen Sie dieses Beispiel (die Schlüsselbilder).

Nennen Sie ein Beispiel aus der Übung, bei dem diese Art der Animation angewendet wurde.

Erstellen Sie in Ihrer Abgabe einen Ordner „aufgabe3“, der die Lösung zu dieser Aufgabe enthält. Die Abgabe der Lösung muss als Textdatei (Endung .txt) bzw. als PNG-Datei erfolgen. Andere Abgaben werden nicht berücksichtigt.

Abgabe

„Optional für Bachelor-Studierende“ bedeutet, dass diese Aufgaben nur von Diplom-Studierenden abgegeben werden müssen. „Optional für Diplom-Studierende“ bedeutet, dass diese Aufgaben nur von Bachelor-Studierenden abgegeben werden müssen.

Bitte geben Sie Ihre Lösung als ZIP-Datei bis zum 01.06.2009 10:00 Uhr im UniWorx Portal (<http://www.pst.ifi.lmu.de/uniworx>) ab. Bitte fügen Sie Ihrer Abgabe eine Datei studiengang.txt hinzu, welche Ihren Studiengang enthält. Dies dient der einfacheren Korrektur.