


# A CHRISTMAS GAME: EIN JUMP & RUN-SPIEL ZU WEIHNACHTEN

## Praktische Einführung in einige grundlegende Techniken für 2D-Spiele

Struktur und Inhalt des Kurses wurden 2012 von Markus Heckner entwickelt. Im Anschluss haben Alexander Bazo und Christian Wolff Änderungen am Material vorgenommen. Die aktuellen Folien wurden von Alexander Bazo erstellt und können unter der **MIT-Lizenz**  verwendet werden.

## AKTUELLER SEMESTERFORTSCHRITT (WOCHE 9)

Kursabschnitt	Themen		
Grundlagen	Einführung	Einfache Programme erstellen	Variablen, Klassen & Objekte
	Kontrollstrukturen & Methoden	Arrays & komplexe Schleife	
Klassenmodellierung	Grundlagen der Klassenmodellierung		Vererbung & Sichtbarkeit
Interaktive Anwendungen	Event-basierten Programmierung	String- & Textverarbeitung	
Datenstrukturen	Listen, Maps & die Collections	Speicherverwaltung	Umgang mit Dateien
Software Engineering	Planhaftes Vorgehen bei der Softwareentwicklung		
	Qualitätsaspekte von Quellcode		

# A CHRISTMAS GAME



**Hinweis:** Wir entwickeln ein 2D-*Jump & Run*-Spiel auf Basis der *GraphicsApp*. Wir versuchen den einfachen Spielablauf so gut wie möglich durch qualitativ hochwertigen Code umzusetzen. Gefundene Lösungsstrategien sollen im besten Fall wiederverwendbar umgesetzt werden.

# ZENTRALE ANFORDERUNGEN

Unser Spiel soll die folgenden Anforderungen erfüllen:

- In regelmäßigen Abständen werden Hindernisse in Form von Schneebällen erzeugt, die sich von rechts nach links bewegen
- Durch Springen kann die Spielfigur diesen Hindernissen ausweichen
- Passiert ein Schneeball die Spielfigur, ohne diesen zu berühren, erhält der/die SpielerIn einen Punkt
- Wird die Spielfigur von einem Schneeball getroffen, wird der Punktezähler zurückgesetzt

# WEITERE ANFORDERUNGEN

Bei der Umsetzung der zentralen Funktionen beachten wir weitere Anforderungen:

- Dem eigentlichen Spiel ist ein Intro mit Einleitungstext vorangestellt
- Das Spiel suggeriert eine kontinuierliche Bewegung der Spielfigur von links nach rechts
- Die Bewegung der Spielfigur und der Schneebälle werden durch Animationen dargestellt
- Das Spiel wird von einer Hintergrundmusik begleitet

# HILFSMITTEL (1/2)

**Hinweis:** Bei der Umsetzung des Spiels werden wir einige fortgeschrittene Techniken verwenden. Sie können, **müssen diese aber nicht in den zukünftigen Aufgaben und Studienleistungen verwenden.**

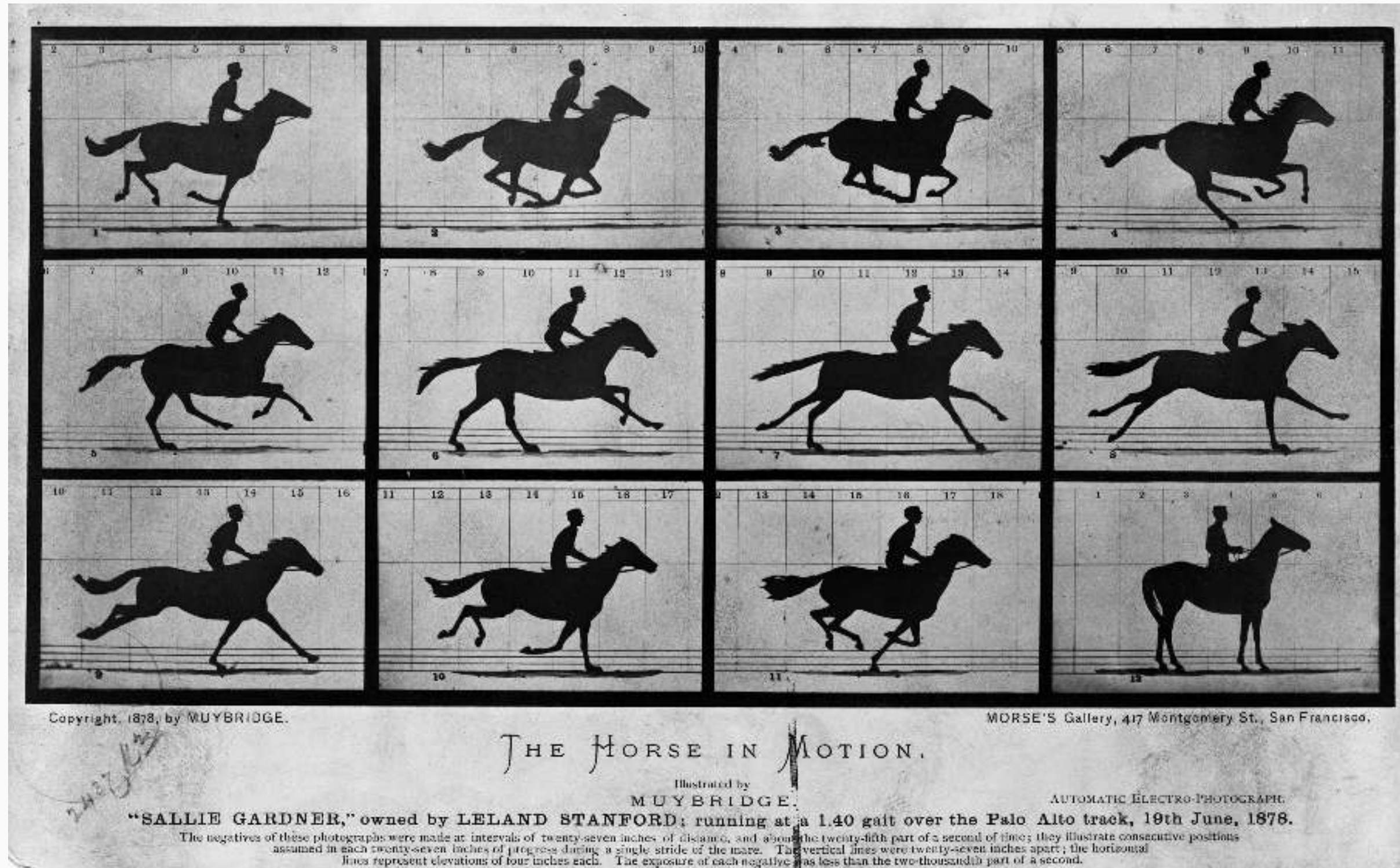
- **Interfaces** für Konstanten: Konstanten, die von vielen Klassen verwendet werden, können in *Interfaces* definiert werden. Alle Klassen, die die Werte benötigen, implementieren die Interfaces.
- **Observer Pattern:** Zustandsveränderungen zwischen Objekten werden - wenn sinnvoll - über das *Observer Pattern* realisiert. Wir schreiben dazu auch eigene *Interfaces*.
- **Audiowiedergabe:** Die verwendete Version der GraphicsApp unterstützt die Wiedergabe von unkomprimierten Audiodateien.

## HILFSMITTEL (2/2)

- **Methoden in Enums:** Enums können Methoden beinhalten. Damit können Sie die Vorteile, z.B. die numerische Repräsentation der Werte, noch besser verwenden!
- **Statische Methoden für die Objekterstellung:** Komplexe Konstruktionsvorgänge werden in statische Methoden der jeweiligen Klasse ausgelagert.
- **Packages:** Umfangreiche Projekte mit vielen Klassen, *Interfaces* und anderen Methoden können sinnvoll durch hierarchisch oder funktional gegliederte *Package*-Strukturen aufgeräumt werden.

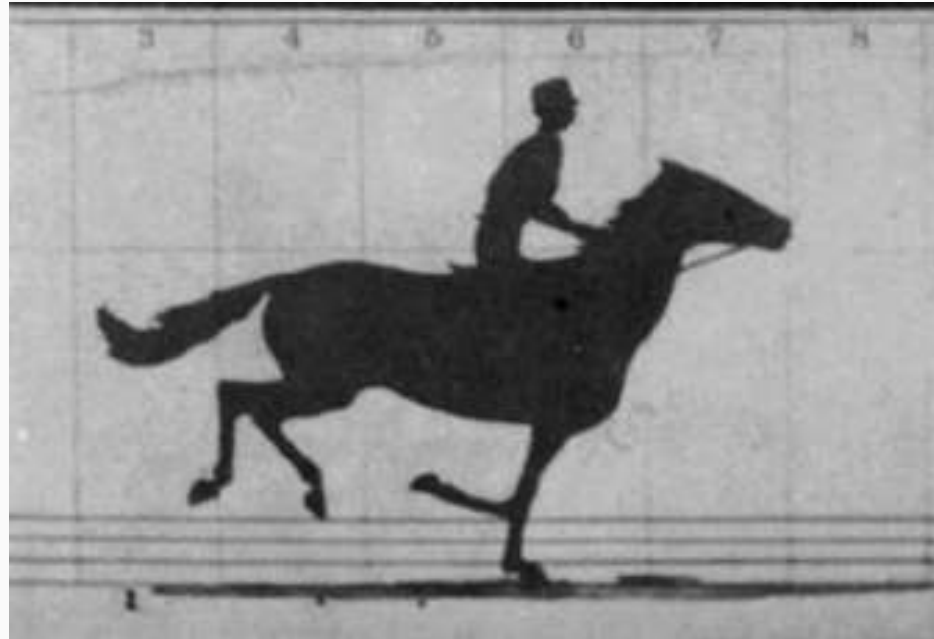


## EADWEARD MUYBRIDGE: THE HORSE IN MOTION (1/2)





## EADWEARD MUYBRIDGE: THE HORSE IN MOTION (2/2)



Wir verwenden eine ähnliche Technik: Wir verfügen über Einzelbilder, die die unterschiedlichen Animationsphasen der Bewegung, von z.B. der Spielfigur, repräsentieren. Während die Figur sich bewegt, wechseln wir im *Draw Loop* regelmäßig das darzustellende Bild und durchlaufen dabei die intendierte Animationsreihenfolge. In der Spielentwicklung wird diese Methode auch als *Spriteanimation* bezeichnet.

# ÜBERSICHT DER BILDMATERIALIEN



Im Ordner data des Starterpakets finden Sie eine Reihe von Grafiken, die für die Darstellung und Animation der Spielobjekte verwendet werden können. Hinweise auf Quellen und Lizenzen finden Sie in dem jeweiligen Unterordner.




# BILD- UND TONMATERIAL

Datei(en)	Verwendungszweck	Szene
music/Background.wav	Hintergrundmusik	-
intro/IntroCard.png	Statischer Hintergrund	Intro
background/Background.png	Statischer Hintergrund	Spiel
background/Objects.png	Beweglicher Teil des Hintergrunds	Spiel
snowball/Snowball*.png	Bilder für die verschiedenen Animationsphasen des Schneeballs	Spiel
santa/santa_running_*.png	Bilder für die verschiedenen Animationsphasen der laufenden Spielfigur	Spiel
santa/jumping_1.png	Bild zur Darstellung der springenden Spielfigur	Spiel

# VORGEHEN

1. **Szenenmodel:** Wir teilen unser Spiel in mehrere Szenen (Intro und Spiel) auf und erlauben den Wechsel zwischen diesen.
2. **Musik:** Wir testen die Integration der Hintergrundmusik.
3. **Animationen:** Wir erarbeiten eine wiederverwendbare Lösung für animierte Spielobjekte.
4. **Spielfigur:** Wir implementieren die Spielfigur (Darstellung und Sprungbewegung).
5. **Schneebälle:** Wir implementieren die Schneebälle, deren Erstellung und Bewegung.
6. **Hit-Test:** Wir verbinden Spielfigur und Schneebälle und prüfen mögliche Kollisionen.
7. **Scoring:** Wir zählen Punkte und implementieren eine einfache Anzeige für den aktuellen *Score*.

## VIELEN DANK FÜR IHRE AUFMERKSAMKEIT. WENN SIE MÖCHTEN, SEHEN WIR UNS IM ANSCHLUSS IN DER ZENTRALÜBUNG!

Fragen oder Probleme? In allgemeinen Angelegenheiten und bei Fragen zur Vorlesung wenden Sie sich bitte an Alexander Bazo ([alexander.bazo@ur.de](mailto:alexander.bazo@ur.de) ). Bei organisatorischen Fragen zu den Studienleistungen und Übungsgruppen schreiben Sie bitte Florin Schwappach ([florin.schwappach@ur.de](mailto:florin.schwappach@ur.de) ). Bei inhaltlichen Fragen zu den Übungen, Beispielen und Studienleistungen schreiben Sie uns unter [mi.oop@mailman.uni-regensburg.de](mailto:mi.oop@mailman.uni-regensburg.de) .