

User-Interfaces



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Visual Computing

Winter Semester 2017-2018

Prof. Dr. A. Kuijper

Mathematical and Applied Visual Computing (MAVC)

Graphisch-Interaktive Systeme (GRIS)

Fraunhofer IGD

Fraunhoferstrasse 5

D - 64283 Darmstadt

E-Mail: office@gris.tu-darmstadt.de

<http://www.gris.tu-darmstadt.de>

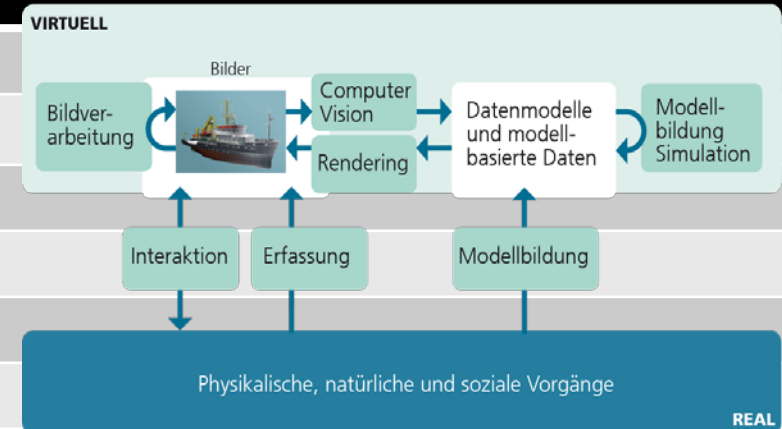
<https://www.mavc.tu-darmstadt.de>

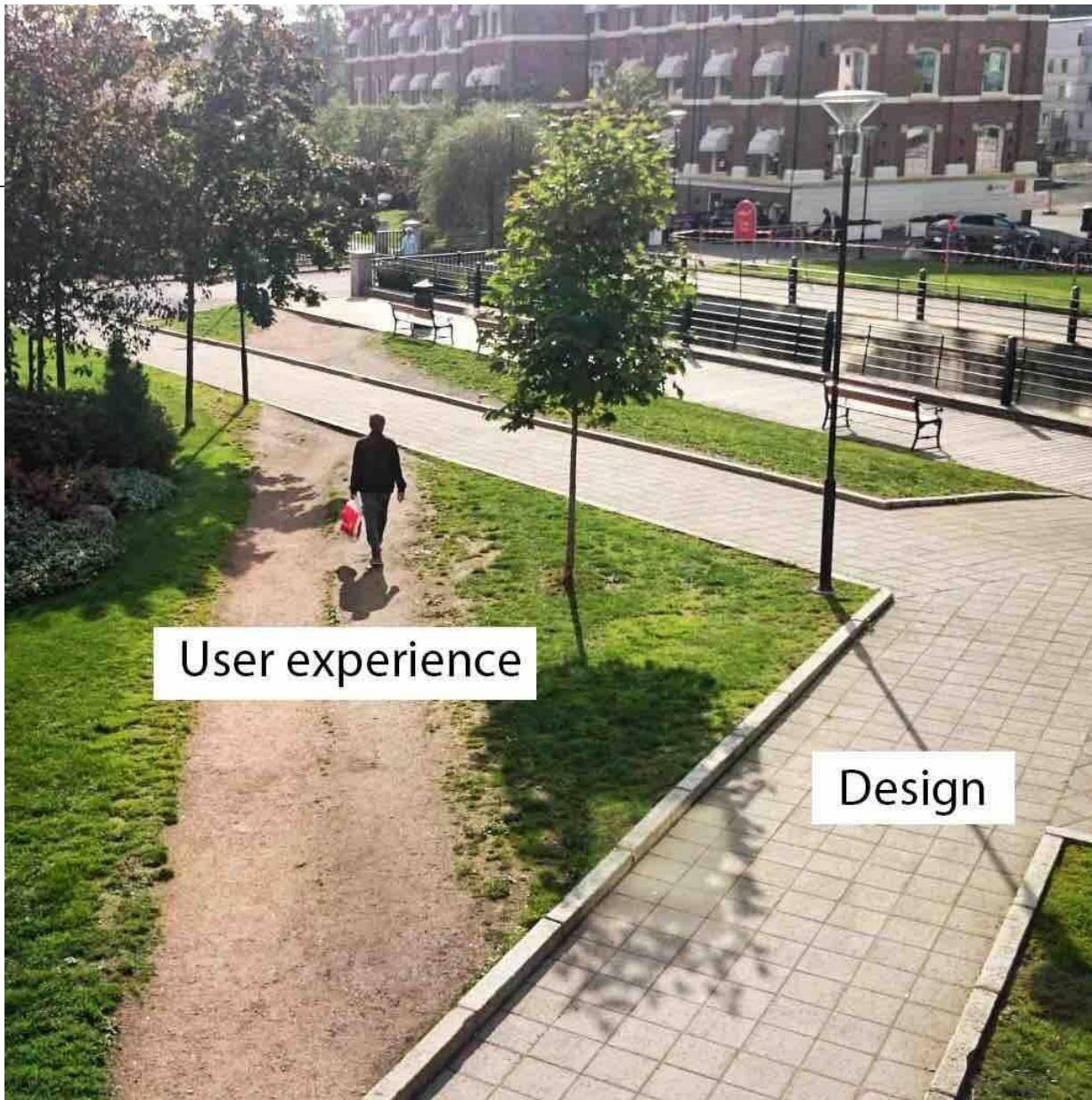
Semesterplan



TECHNISCHE
UNIVERSITÄT
DARMSTADT

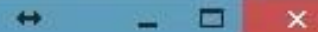
Datum	
20. Okt	Einführung + Visual Computing
27. Okt	Wahrnehmung
03. Nov	Objekterkennung und Bayes
10. Nov	Fourier Theorie
17. Nov	Bilder
24. Nov	Bildverarbeitung
01. Dez	Grafikpipeline & Eingabemodalitäten & VR+AR
08. Dez	Transformationen & 2D/3D Ausgabe
15. Dez	3D-Visualisierung
12. Jan	X3D – 3D in HTML
19. Jan	Informationsvisualisierung
26. Jan	Farbe
02. Feb	User Interfaces + Multimedia Retrieval
01. Mrz	Klausur





User experience

Design


**MATHEMATICAL
AND APPLIED
VISUAL COMPUTING**



Links



Mitte



Rechts



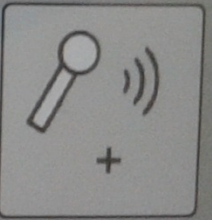
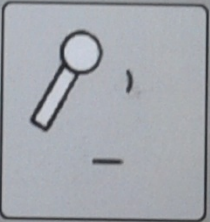
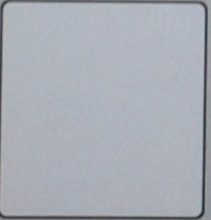
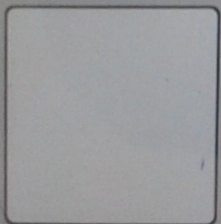
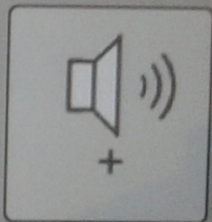
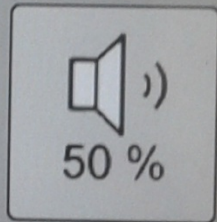
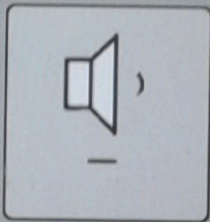
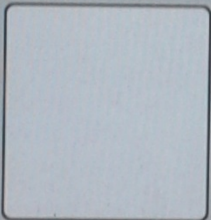
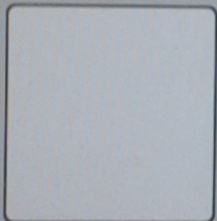
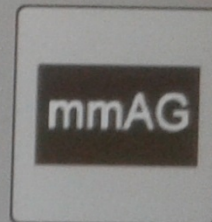
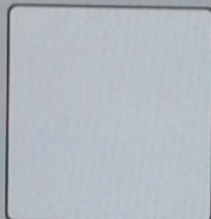
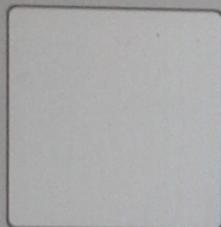
Audio



Licht



Raum



S1|01/A1



Hilfe



ON AIR



Interaktionsparadigmen



TECHNISCHE
UNIVERSITÄT
DARMSTADT





- Einführung
- **UI: Interaktion**
 - **Interaktionsmöglichkeiten**
 - Interaktionsdesignprozess
- GUI: Benutzeroberflächen
 - Komponenten
 - Menübasierte Programme
 - GUI-Anwendungen und eventbasiertes Programmieren
- 3D-Interaktion
- Zusammenfassung

1. Kommandozeile



- Schnell und mächtig
- Viele Befehle sind Abkürzungen
 - Schnell und effizient
- Befehle können auf mehrere Objekte gleichzeitig angewandt werden
 - Schnelle Eingabe
- Einige Befehle haben mehrere Parameter, die präzise und flexibel gesetzt und angepasst werden können

```
Terminal
File Edit View Terminal Tabs Help
hlouey@linux: ~> more /etc/shells
/bin/ash
/bin/bash
/bin/bash1
/bin/csh
/bin/false
/bin/ksh
/bin/sh
/bin/tcsh
/bin/true
/bin/zsh
/usr/bin/csh
/usr/bin/ksh
/usr/bin/passwd
/usr/bin/bash
/usr/bin/rbash
/usr/bin/tcsh
/usr/bin/zsh
hlouey@linux: ~> █
```


2. Menüs

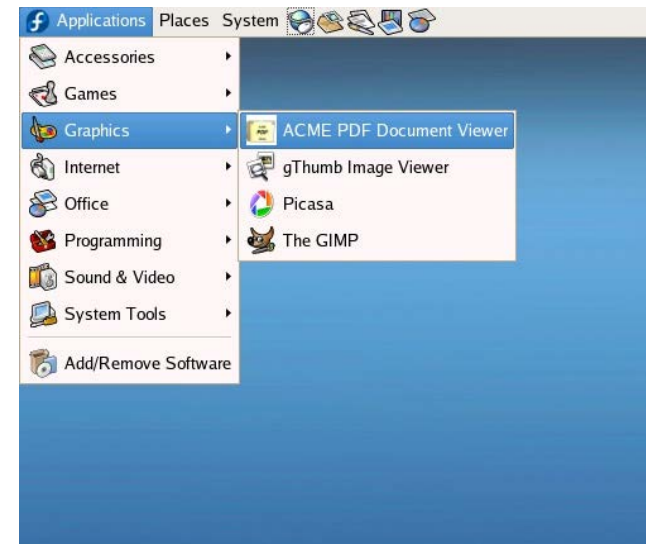
Menübasierte Interfaces konfrontieren Benutzer mit sequentiellen, hierarchischen Menüs, die Listen von Funktionen enthalten.

Bedienung

- Pfeiltasten oder Maus zur Funktionswahl
- Alternativ Nummerntasten und weitere Tastaturkürzel

Vorteile

- Man muss keine Befehle auswendig kennen
- Auswahl aus angezeigten möglichen Optionen
- Die Liste ist beschränkt
- Für kleine Bildschirme geeignet



Menübasierte Interfaces

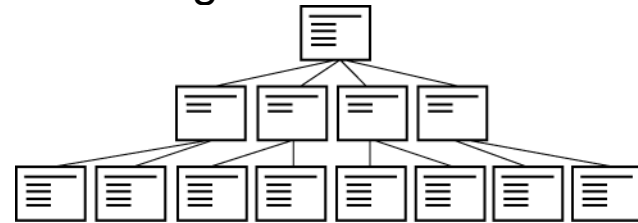
Die meisten Menüs sind Variationen einiger Grundkategorien:



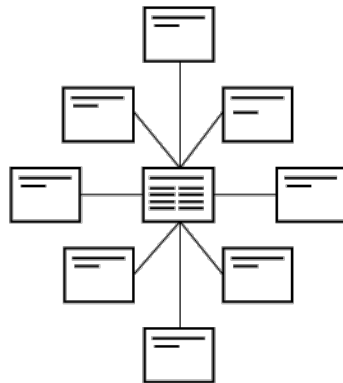
Einfach



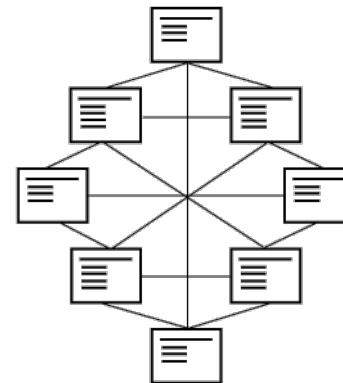
Sequentiell



Hierarchisch



Stern



Netz

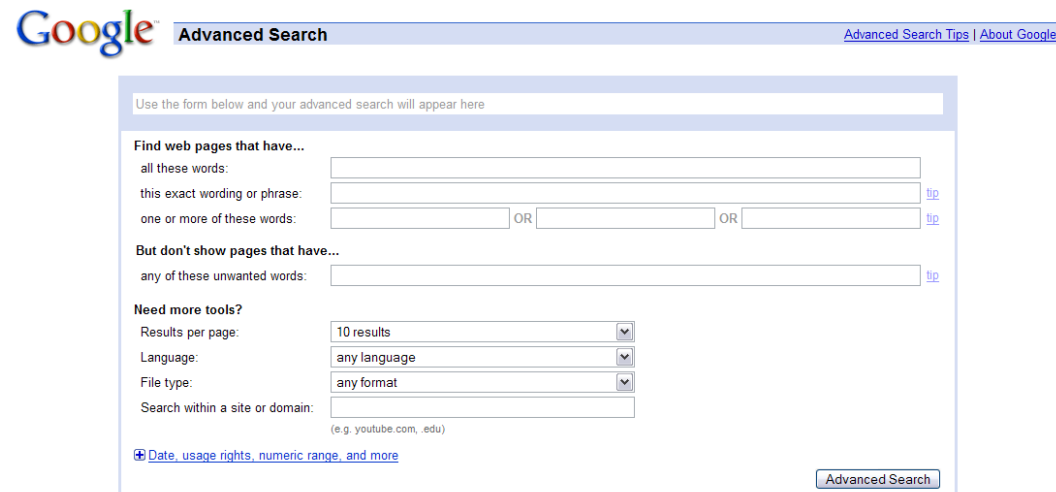
3. Formulare

Ähnlich zu Menüs werden Informationen bildschirmweise angezeigt

Im Gegensatz zu Menüs werden die erfassten Informationen linear verarbeitet, es gibt keine hierarchische Struktur

Benutzer müssen verstehen, welche Daten benötigt werden und in welchem Format diese sein sollen

- Beispiel: Datum
 - 02.01.2013, 01/02/2013, oder 2. Januar 2013?



The screenshot shows the Google Advanced Search interface. At the top, the Google logo is followed by "Advanced Search" and links for "Advanced Search Tips" and "About Google". Below this is a light blue box containing the search form. The form has several sections: "Find web pages that have..." with three input fields for "all these words:", "this exact wording or phrase:", and "one or more of these words:" (separated by "OR"); "But don't show pages that have..." with an input field for "any of these unwanted words:"; and "Need more tools?" with dropdown menus for "Results per page:" (set to 10), "Language:" (set to any language), and "File type:" (set to any format). There is also an input field for "Search within a site or domain:" with a hint "(e.g. youtube.com, edu)". At the bottom left, there is a link for "Date, usage rights, numeric range, and more". At the bottom right, there is an "Advanced Search" button.

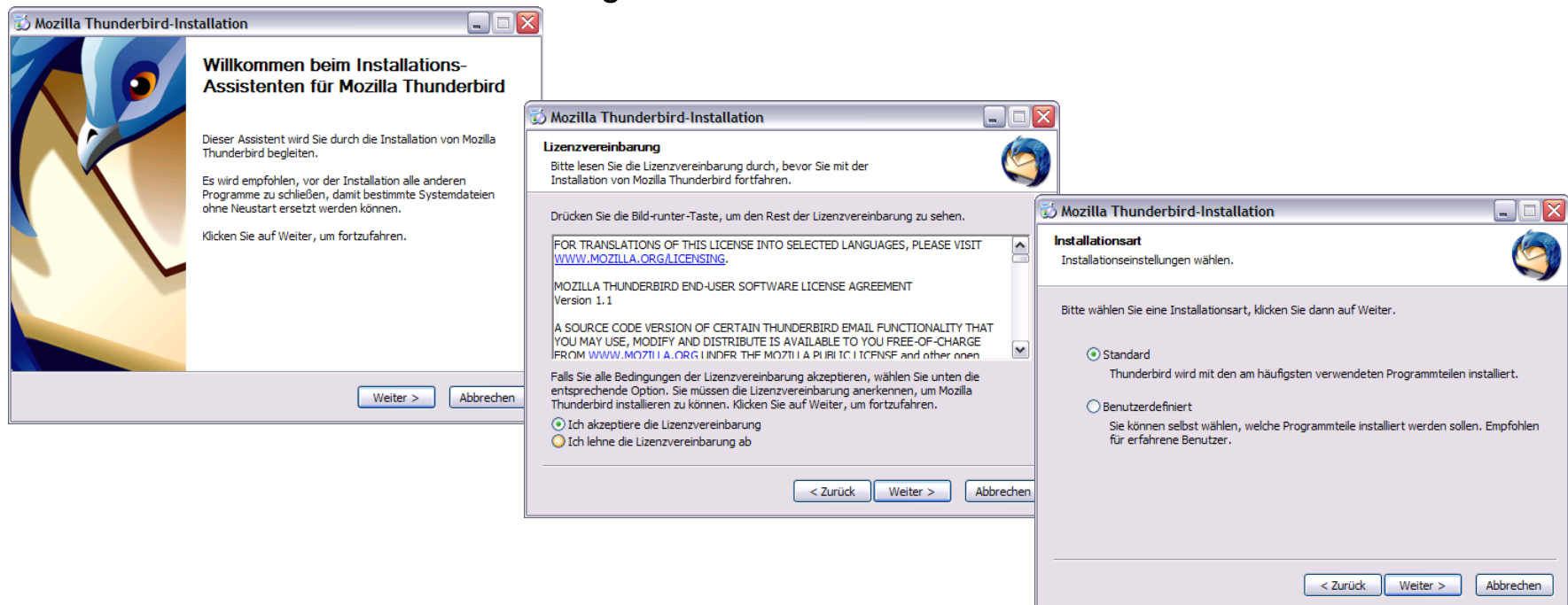
4. Fragen und Antworten

Frage-und-Antwort-Interfaces werden auch „Wizard“ genannt

Sie schränken Experten ein

Sind dafür leicht verständlich für Anfänger

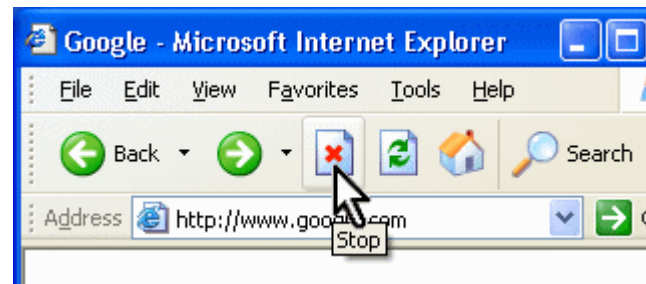
Eventuell bieten sie nicht die benötigten Informationen



5. Direkte Manipulation

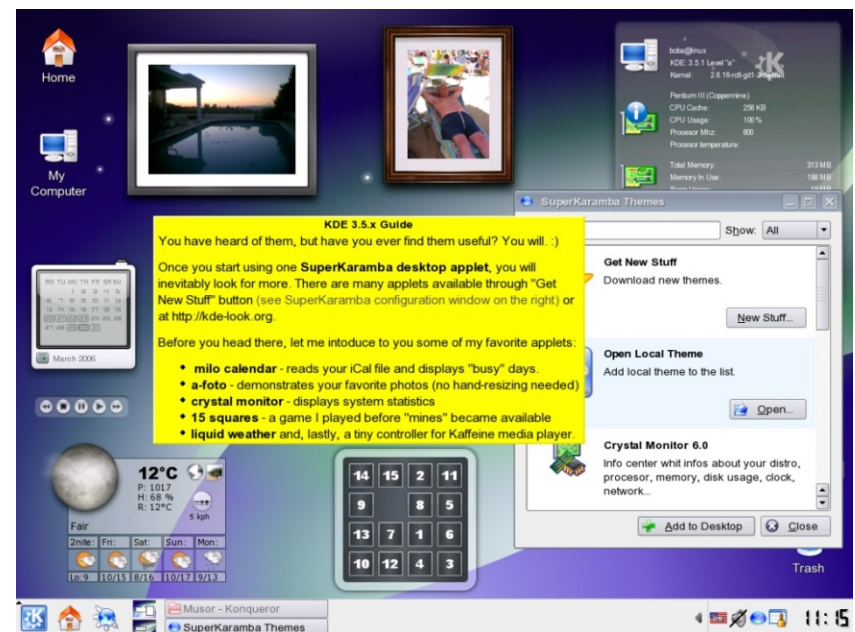
Ben Shneiderman (1982):

- Objekte und Aktionen sollen konsistent mit passenden visuellen Metaphern dargestellt werden
- Physische Aktionen oder Drücken von beschrifteten Buttons statt komplexe Syntax
- Schnelle, inkrementelle und widerrufbare Aktionen, deren Einflüsse auf Objekte sofort sichtbar sind

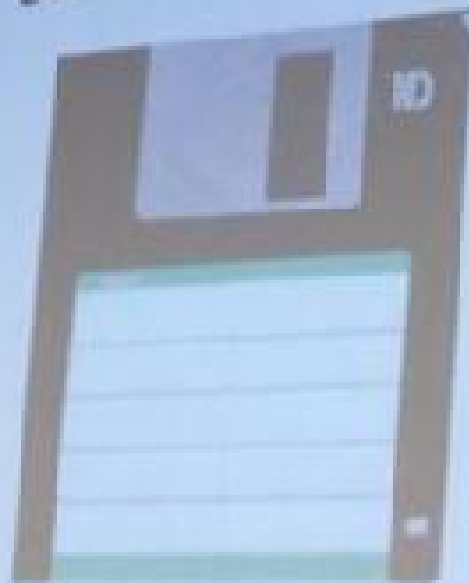


Metaphern

GUIs benutzen visuelle Beziehungen zu Objekten aus der Realität (Metaphern)
Metaphern können Menschen helfen, komplexe Sachverhalte zu verstehen, indem sie ihr Wissen aus der echten Welt miteinbeziehen
Erfahrungen aus der echten Welt gelten auch hier



I showed my 12 year old son an
old floppy disk....



He said "Wow... Cool!
You 3D printed the Save Icon!"

6. 3D-Umgebungen

3D-Interaktion ist in der Realität
natürlich

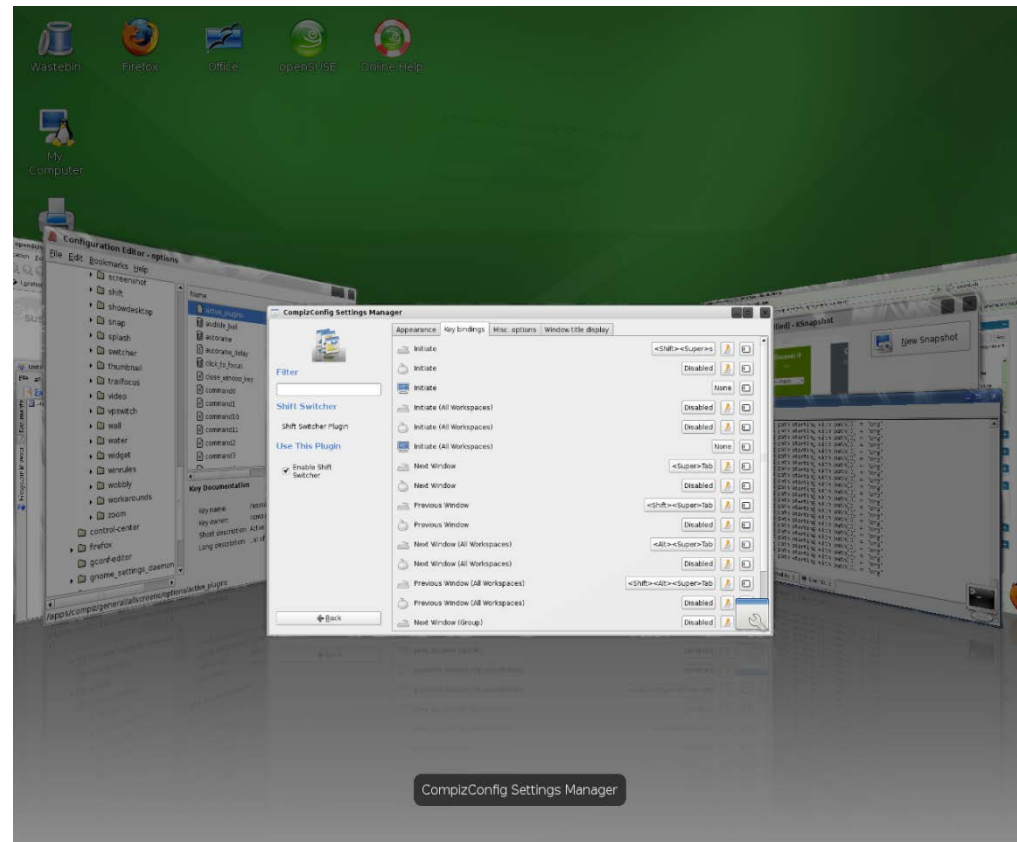
3D-Umgebungen sind in Spielen oft
anzutreffen

Komplexe 3D-Umgebungen benötigen
viel Rechenleistung

Desktop 3D

- Derzeitige GUIs sind hauptsächlich 2D
- 3D-Umgebungen auf 2D-Bildschirmen
sind schwierig zu bedienen

➔ Problem: 3D-Interaktion



7. Natürliche Sprache

Natural Language Interaction (NLI): Interaktion mit Computern mittels
Alltagssprache

Probleme:

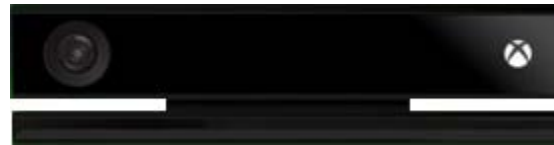
- Mehrdeutigkeit
- Abhängigkeit vom Kontext
- Abhängigkeit von Visuellem

Zwei Forschungsgebiete:

- Spracherkennung
- Semantik

Anwendungsgebiete:

- Spracheingabe
- Sprachausgabe



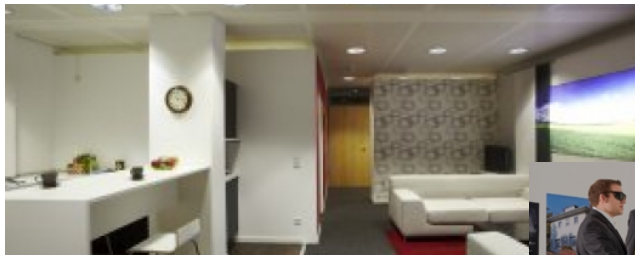
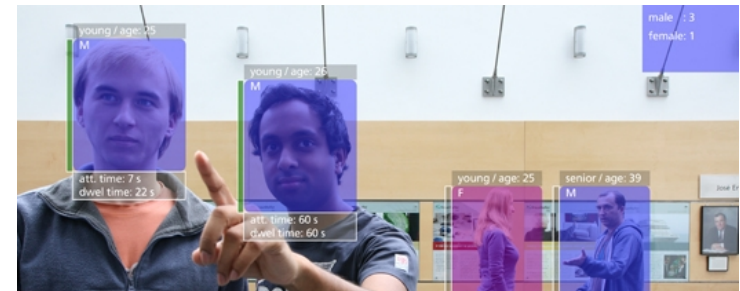
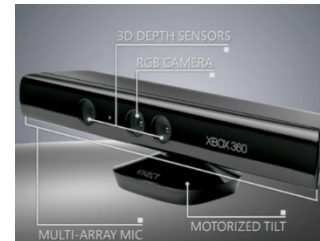
8. Gesten

Bewegungen werden aufgezeichnet

- Finden von Gesten
- Interaktion durch Zeigen
- Interaktion durch Wischen
(ohne Berührung)

Ambient Assisted Living

- Automatische Interaktion mit Menschen,
die sich in der Umgebung befinden





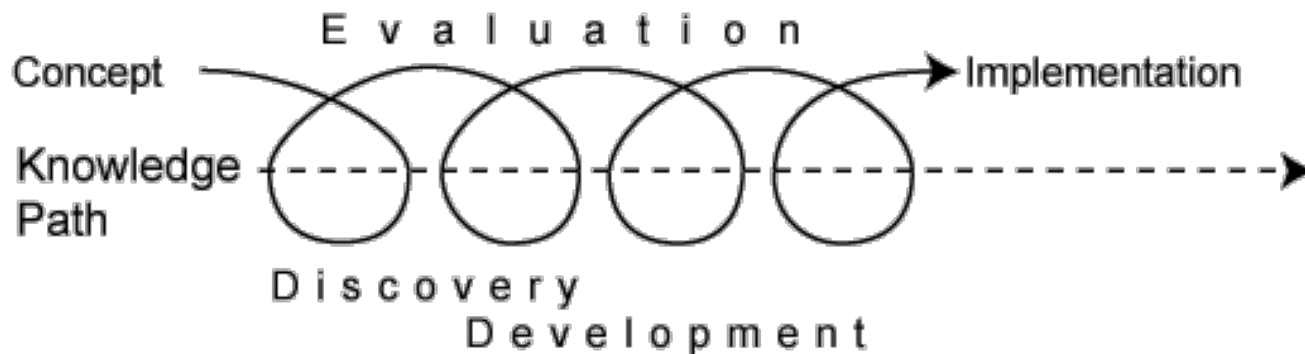
- Einführung
- **UI: Interaktion**
 - Interaktionsmöglichkeiten
 - **Interaktionsdesignprozess**
- GUI: Benutzeroberflächen
 - Komponenten
 - Menübasierte Programme
 - GUI-Anwendungen und eventbasiertes Programmieren
- 3D-Interaktion
- Zusammenfassung

Iteratives Design

Interaktionsdesign ist ein iterativer Prozess

- Ein Schritt vorwärts, zwei Schritte zurück

Der Pfad des Wissens bewegt sich konstant vorwärts



User-Centered Design (UCD)

Das Ziel von UCD ist es, ein Framework zu entwickeln, das es Interaktionsdesignern ermöglicht, besser benutzbare Systeme zu entwerfen.

ISO Standard – [Human Centered Design Processes for Interactive Systems](#)

Fundament des UCD:

- Früher Fokus auf die Benutzer und ihre Aufgaben
- Andauernde Bewertungen bezogen auf leichte Erlernbarkeit und Benutzbarkeit
- Iteratives Design

Continental bringt die Kinotechnik ins Auto



TECHNISCHE
UNIVERSITÄT
DARMSTADT

<http://www.faz.net/aktuell/wirtschaft/unternehmen/continental-entwickelt-augmented-reality-head-up-display-13036774.html>

Als erstes Unternehmen der Autoindustrie hat Continental einen Apparat entwickelt, der dem Fahrer individuelle Wegweiser direkt in das Sichtfeld projiziert. Dahinter steckt ein großes Geschäft.

„Für die **Akzeptanz** automatisierten Fahrens spielt diese Technologie in unseren Augen eine sehr wichtige Rolle. Das Fahrzeug zeigt dem Fahrer, was die Assistenzsysteme sehen. Das macht es dem Menschen am Steuer leichter, **Vertrauen** in die Technik zu fassen“, sagt Continental-Manager Spoelder.



1. Wasserfallmodell

Winston Royce (1970)

- Weltraummissionen
 - Planen
 - Kommandieren
 - Nachträgliches Analysieren

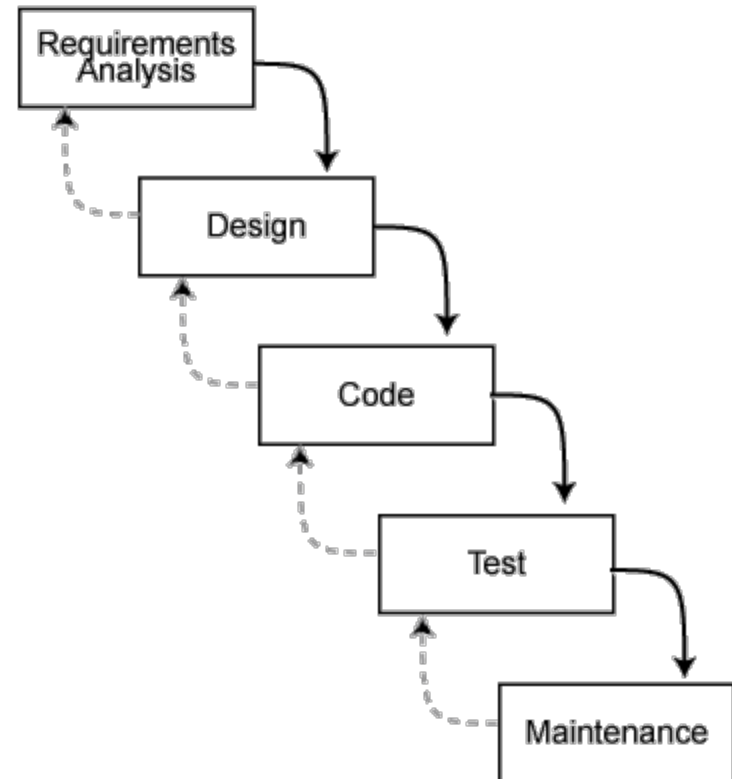
Analysieren und Programmieren

Kleine Projekte

Programmierer sind die Benutzer

Anforderungen stehen fest

Nicht benutzerzentrisch



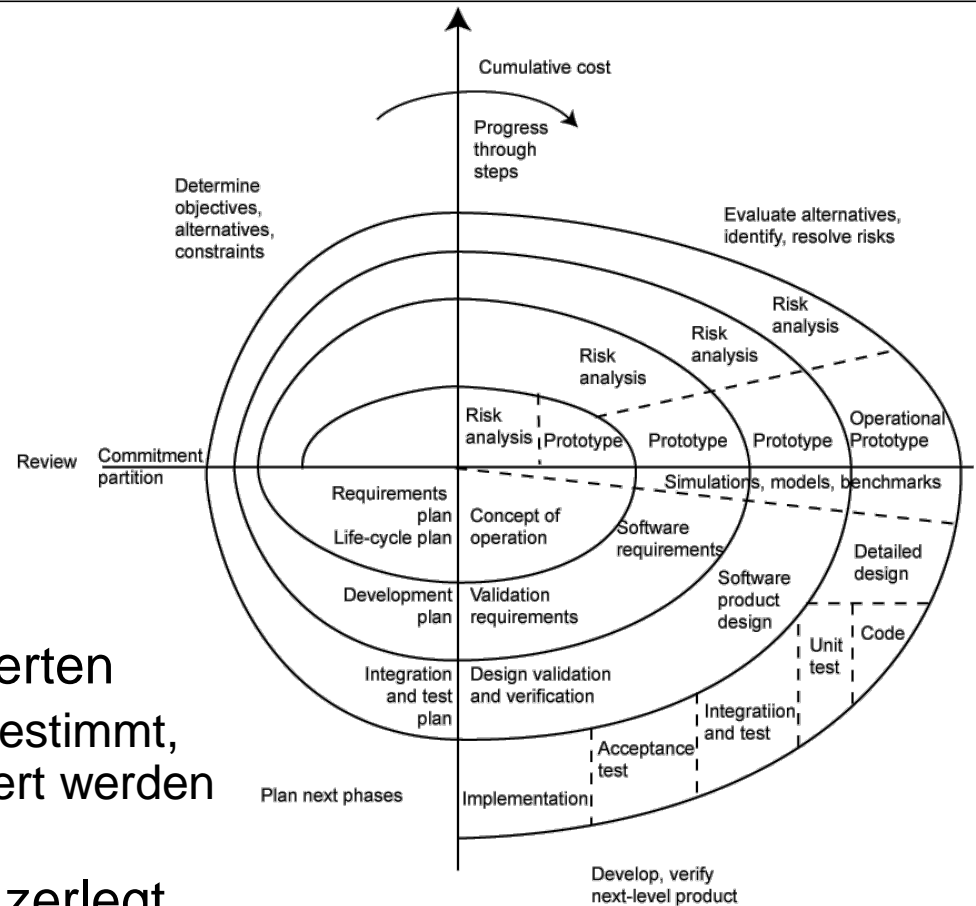
2. Spiralmodell



Barry Boehm (1988)

- Flexibler
- Fokus auf Risikominimierung
- Beinhaltet Prototyp-Entwicklung
- Ermutigt zu Iterationen
- Beginnt mit Vorschlägen von Werten
 - Ein Unternehmensleitbild wird bestimmt, das durch Technologie verbessert werden kann

Das Projekt wird in Unterprojekte zerlegt,
konkrete Risiken werden identifiziert

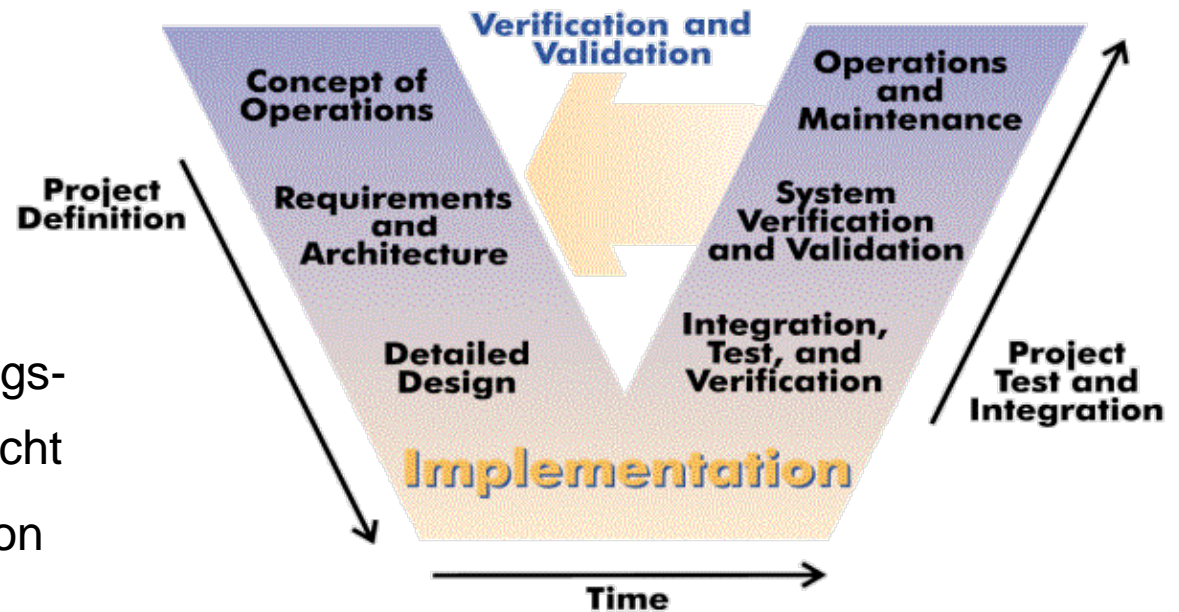


3. V-Modell

Beschreibung des Entwicklungsprozesses aus technischer Sicht
Prozesse werden als Folge von Aktivitäten modelliert, die

- „Produkte“ erstellen oder
- „Produkte“ verarbeiten

„Produkte“: Sämtliche Ergebnisse einer Aktivität (Software, Dokumente, ...)



4. Dynamic Systems Development Method (DSDM)

Rapid Application Development (RAD)

Dynamic Systems Development Method (DSDM)

Zeitfenster und Ressourcen

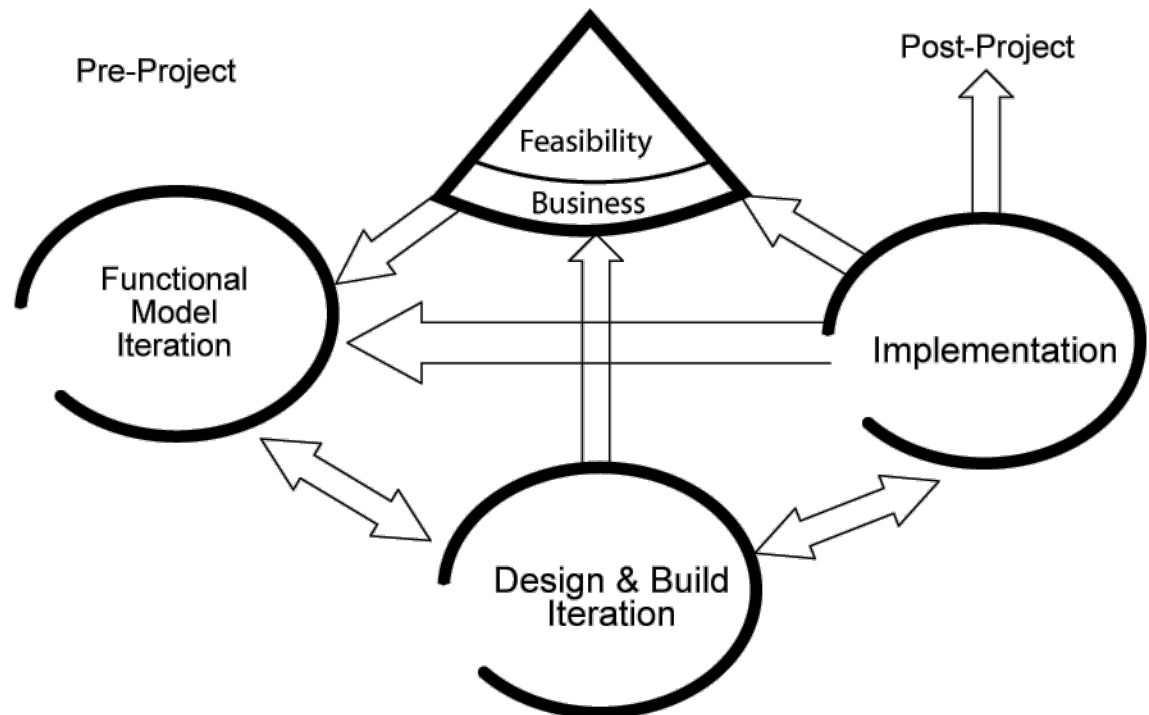
sind fest

Funktionale Anforderungen

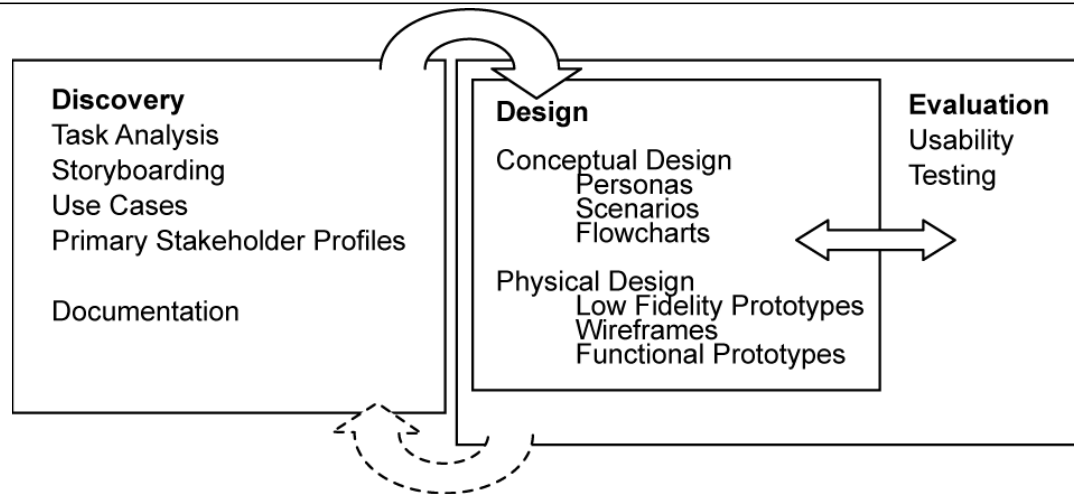
sind flexibel

Drei Schritte

- Pre-Project, Machbarkeitsstudie, und BWL-Phasen
- Iterationen zwischen der funktionalen Modelliteration, Design- und Builditeration und Implementierungsphase
- Post-Project-Phase



5. Design Process Model



Design—Die Designphase besteht aus zwei Teilen:

- **Konzeptuelles Design**—Was sind die Möglichkeiten, mit denen das Design den Problemen begegnen kann?
- **Physisches Design**—Was sind die Möglichkeiten, mit denen das konzeptuelle Design umgesetzt werden kann?

Evaluation—Fragen

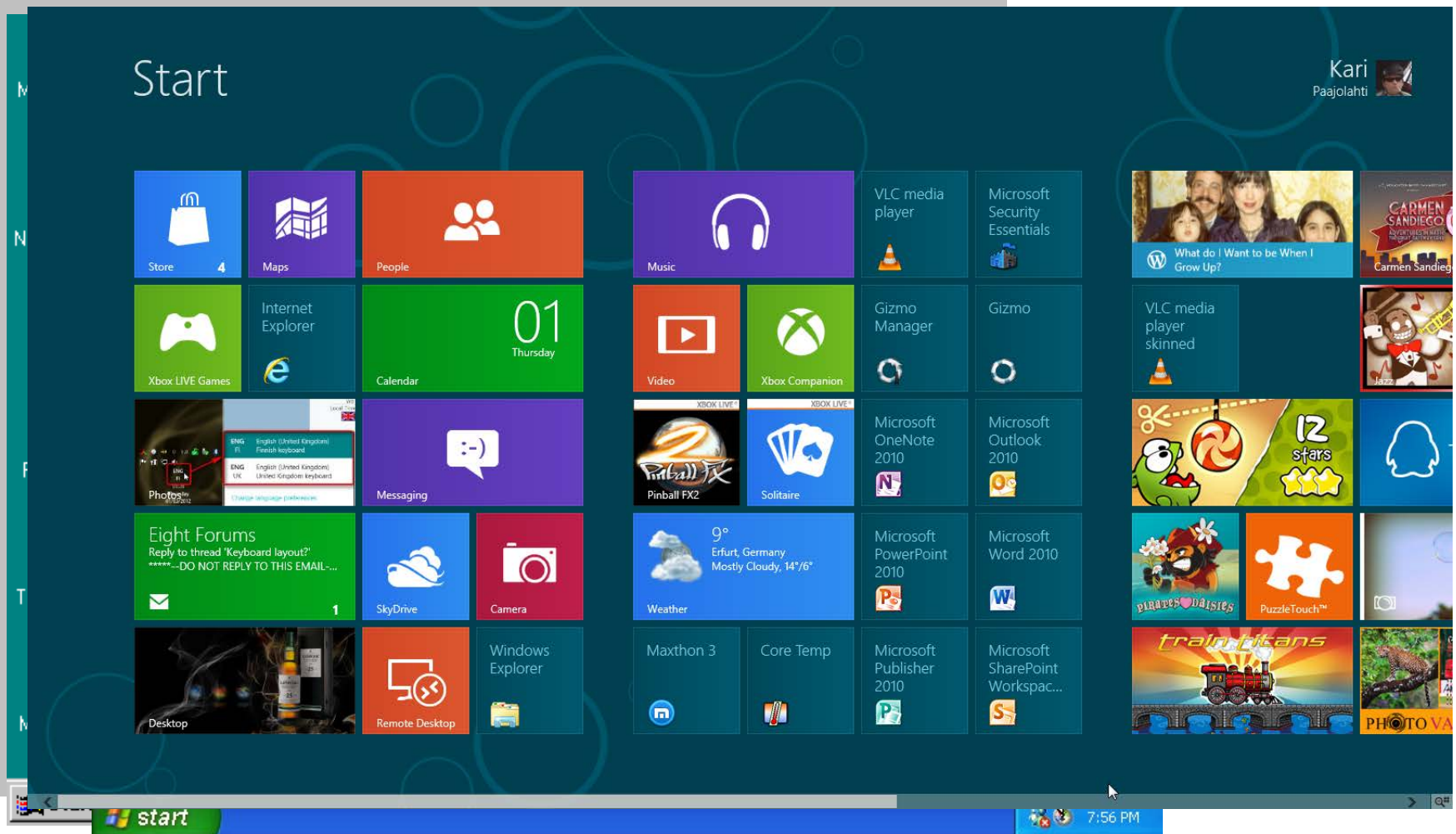
- Wie können die relativen Vorzüge eines Designs über einem anderen bestimmt werden?
- Wie kann der Erfolg eines vorgeschlagenen Designs gemessen werden?
- Wie können echte Nutzer dazu gebracht werden, Feedback zu einem Design zu geben?
- Wie können Usability-Tests im frühen Designprozess untergebracht werden?

Dies wird durch die Ergebnisse von formellem und informellem Usability-Testing dokumentiert.

<http://www.guidebookgallery.org/screenshots>



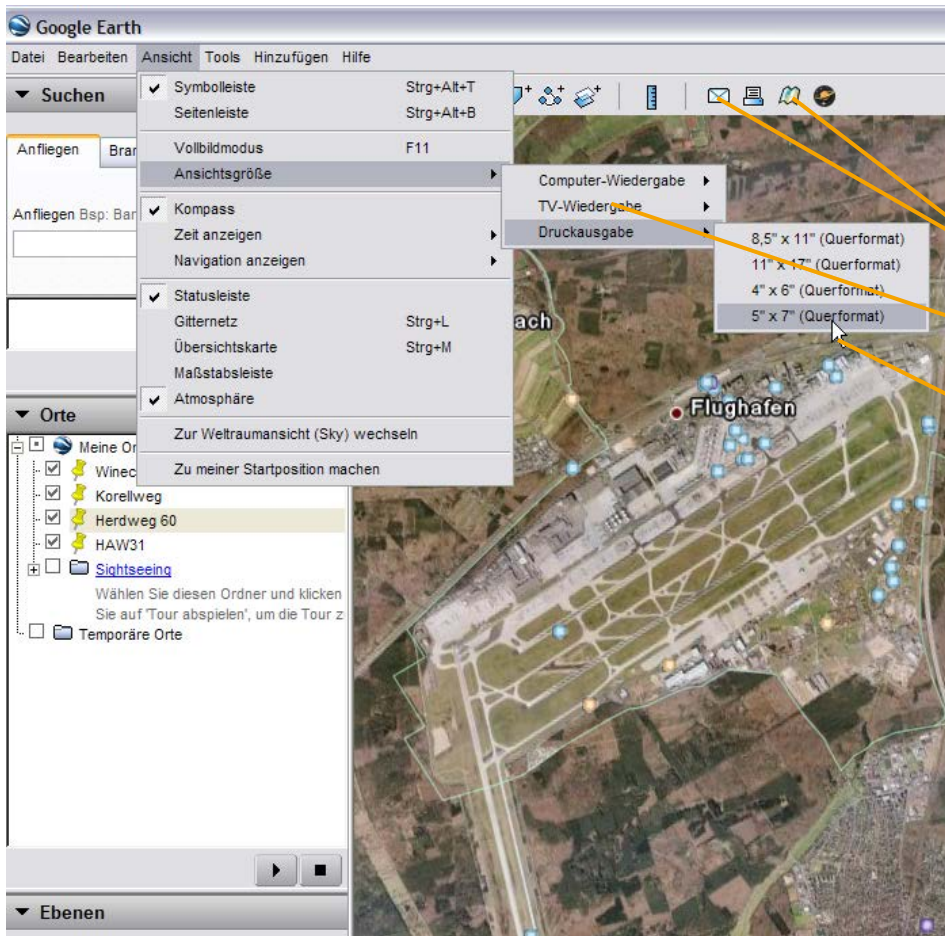
TECHNISCHE
UNIVERSITÄT
DARMSTADT





- Einführung
- UI: Interaktion
 - Interaktionsmöglichkeiten
 - Interaktionsdesignprozess
- **GUI: Benutzeroberflächen**
 - **Komponenten**
 - Menübasierte Programme
 - GUI-Anwendungen und eventbasiertes Programmieren
- 3D-Interaktion
- Zusammenfassung

Das WIMP-Interface



Derzeitige GUIs werden manchmal als WIMP-Interfaces bezeichnet, da sie aus folgenden Komponenten bestehen:

- Windows
- Icons
- Menus
- Pointers

GUIs sind Fenster-Interfaces:

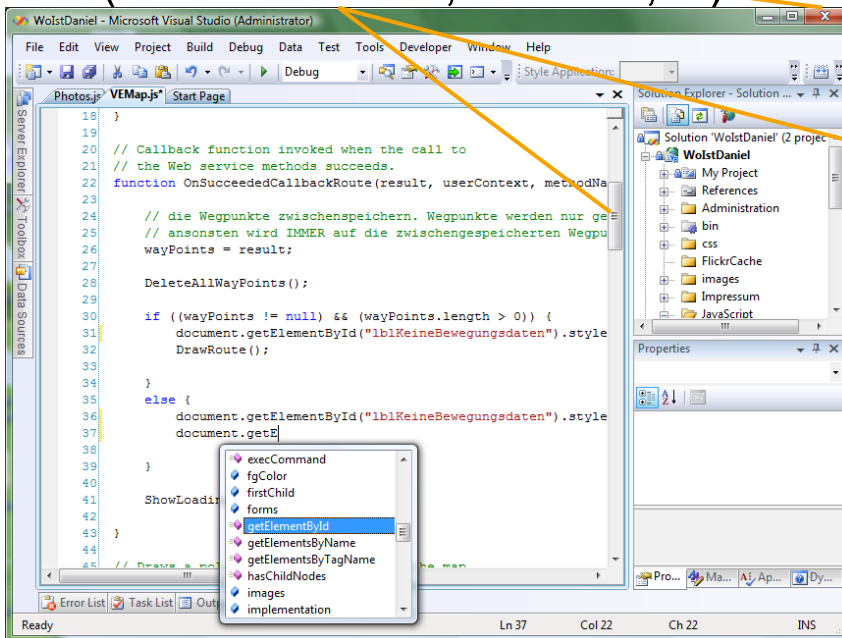
Rechteckige Boxen, „Fenster“, zeigen die Komponenten einer Anwendung an.

Zuerst beim Xerox Alto, später auch in Apple OS und Microsoft Windows

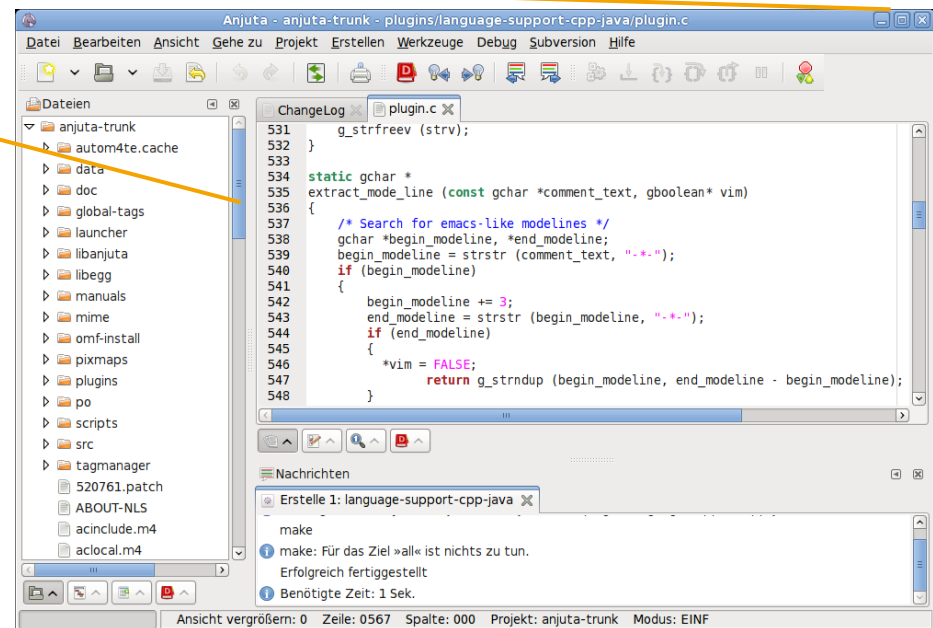
Fenster-Komponenten

Die meisten Fenster-Systeme verwenden Standardfenster, die
ähnlich aussehen und sich ähnlich verhalten

(z.B. Scrollbars, Buttons,...)



IDE: *VisualStudio*, Windows Vista

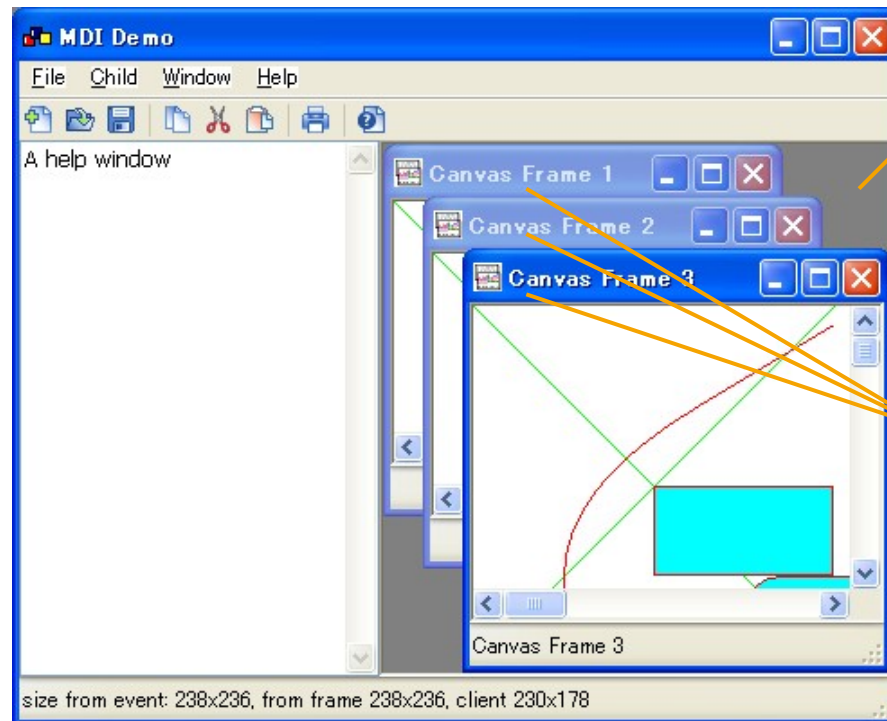


IDE: *Anjuta*, Linux/Gnome

Window Interfaces

Multiple Document Interface (MDI)

- Anwendungszentrisch
- Ein Hauptfenster dient als Arbeitsbereich für mehrere geöffnete Dokumente



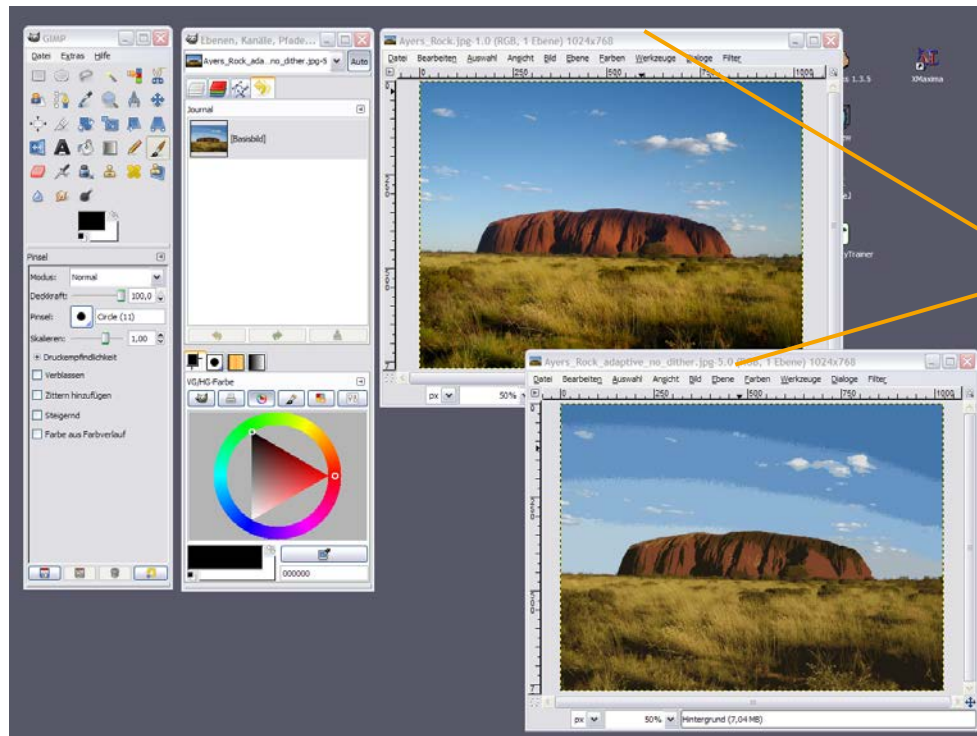
Anwendungsfenster

Mehrere
Dokumente

Window Interfaces

Single Document Interface (SDI)

- Dokumentzentrisch
- Für jedes Dokument wird ein Hauptfenster geöffnet



Multiple primary
windows

Window Interfaces

Tabbed Document Interface (TDI)

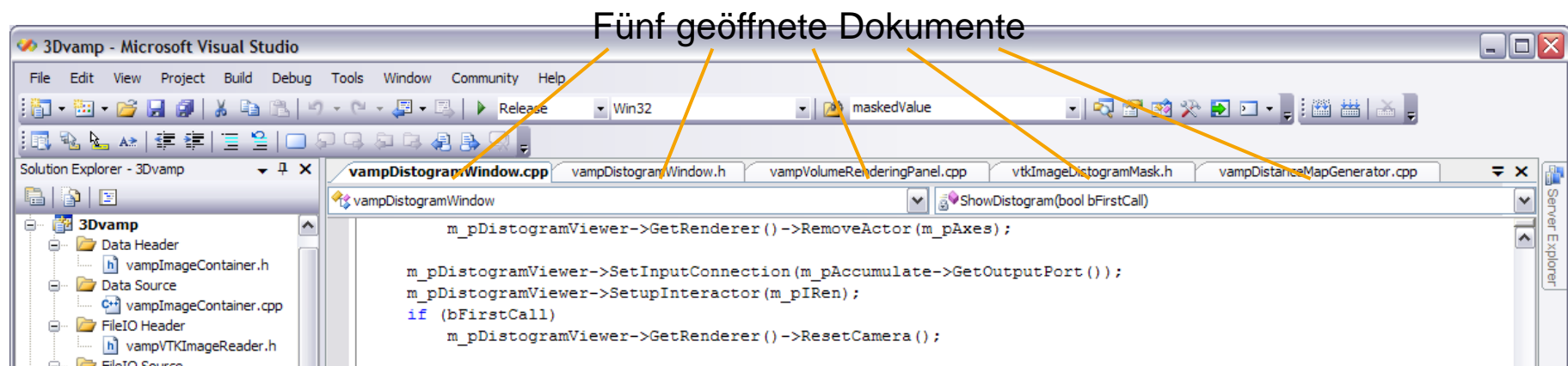
Eine Variante des MDI, auch „Workbook“ genannt

Tabs werden dazu verwendet, zwischen Dokumenten umzuschalten

Manche TDIs fixieren alle Dokumentfenster im maximierten Zustand, wodurch kein Nebeneinanderbetrachten und kein Überlappen möglich ist

Andere erlauben es Dokumenten, in der Größe verändert zu werden.

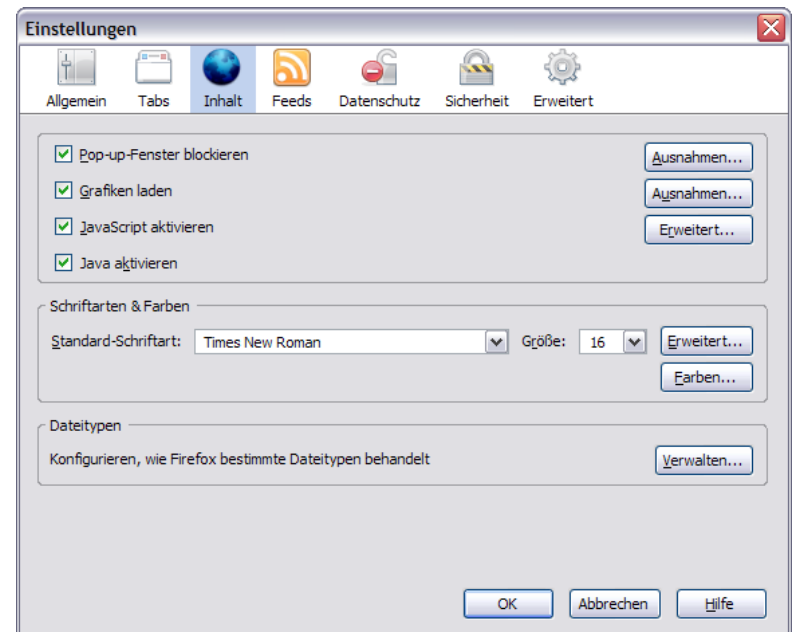
Die Tabs werden dann entfernt, das Interface wird MDI.



Dialogboxen

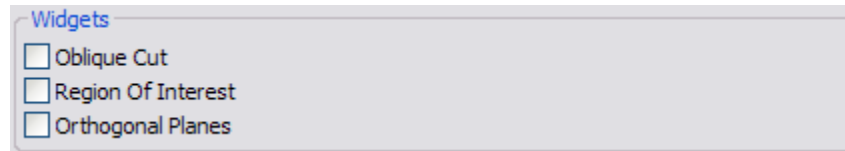
Dialogboxen stellen Platz für untergeordnete Funktionalitäten bereit. Beispielsweise:

- Setzen und Verändern von Objekteigenschaften
- Funktionen ausführen
(z.B. “Speichern” mit Zusatzfunktionen)
- Prozesse ausführen
(z.B. “Kopieren” mit Fortschrittsanzeige)
- Aktionen bestätigen
(z.B. “wirklich beenden?”)
- Den Benutzer informieren
(z.B. über Fehler)



Checkboxes repräsentieren binäre Sachverhalte

- „Ja“ und „Nein“
- „An“ und „Aus“
- ...

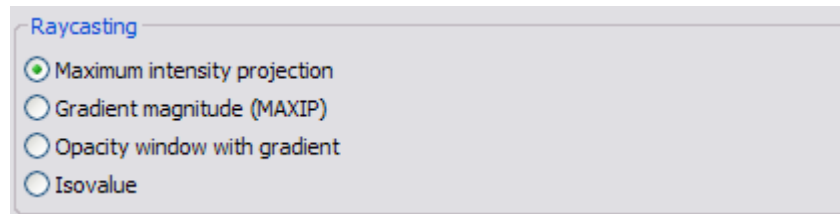


Sie können entsprechend ausgewählt sein oder nicht
(zumeist durch einen Haken bzw. eine leere Box
repräsentiert)

Radio Buttons

Radio buttons verhalten sich ähnlich wie Checkboxes.

- Beliebig viele Radio Buttons sind einer *Gruppe* zugeordnet
- Innerhalb einer Gruppe kann genau ein Radio Button ausgewählt sein.
- Alle anderen sind nicht ausgewählt. Wird ein neuer Radio Button ausgewählt, wird die Auswahl vom vorigen automatisch entfernt.

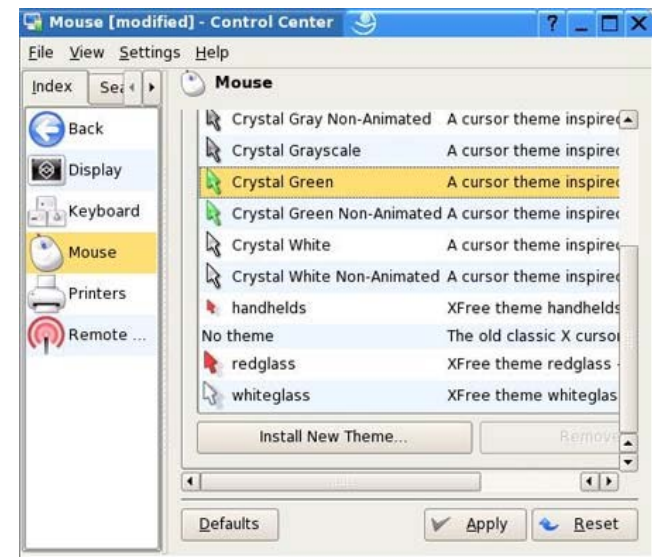


Listboxen

Listboxen repräsentieren eine eindimensionale Datenmenge. Einträge verzweigen nicht weiter. Es kann i.d.R. festgelegt werden, ob nur ein Eintrag oder aber mehrere Einträge auf einmal ausgewählt werden können.



Windows XP

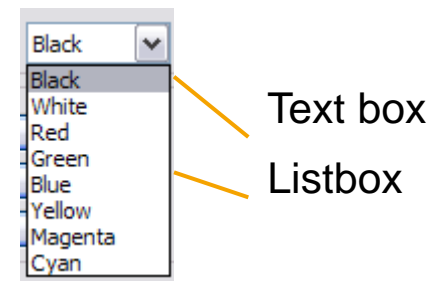


Linux/KDE

Comboboxen

Comboboxen verhalten sich ähnlich wie Listboxen.

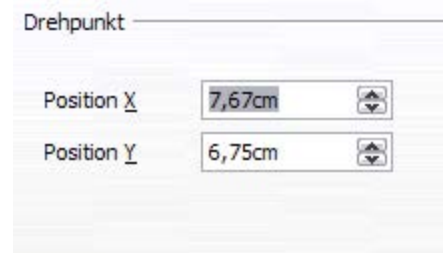
- Liste nur bei Bedarf sichtbar
- Ansonsten „eingeklappt“, sodass nur das ausgewählte Element angezeigt wird
- Gut geeignet für Listen, deren Inhalte nicht immer sichtbar sein müssen
- Platzsparend



Spinner

Spinner enthalten eine beschränkte Liste an Werten.

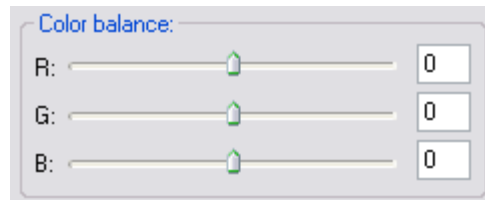
- Inkrementierung oder Dekrementierung mit Pfeilen möglich



Slider

Slider sind Kalibrierungswerkzeuge

- Die Wertemenge ist links und rechts beschränkt
- Dazwischen verlaufen die Werte gleichförmig

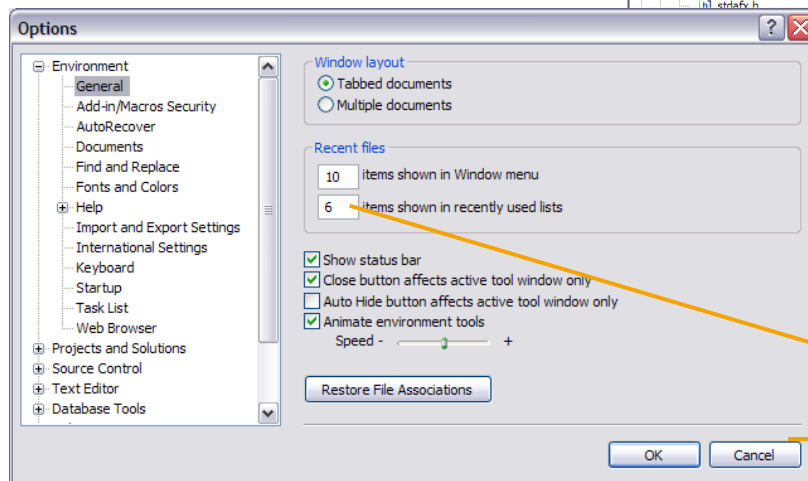
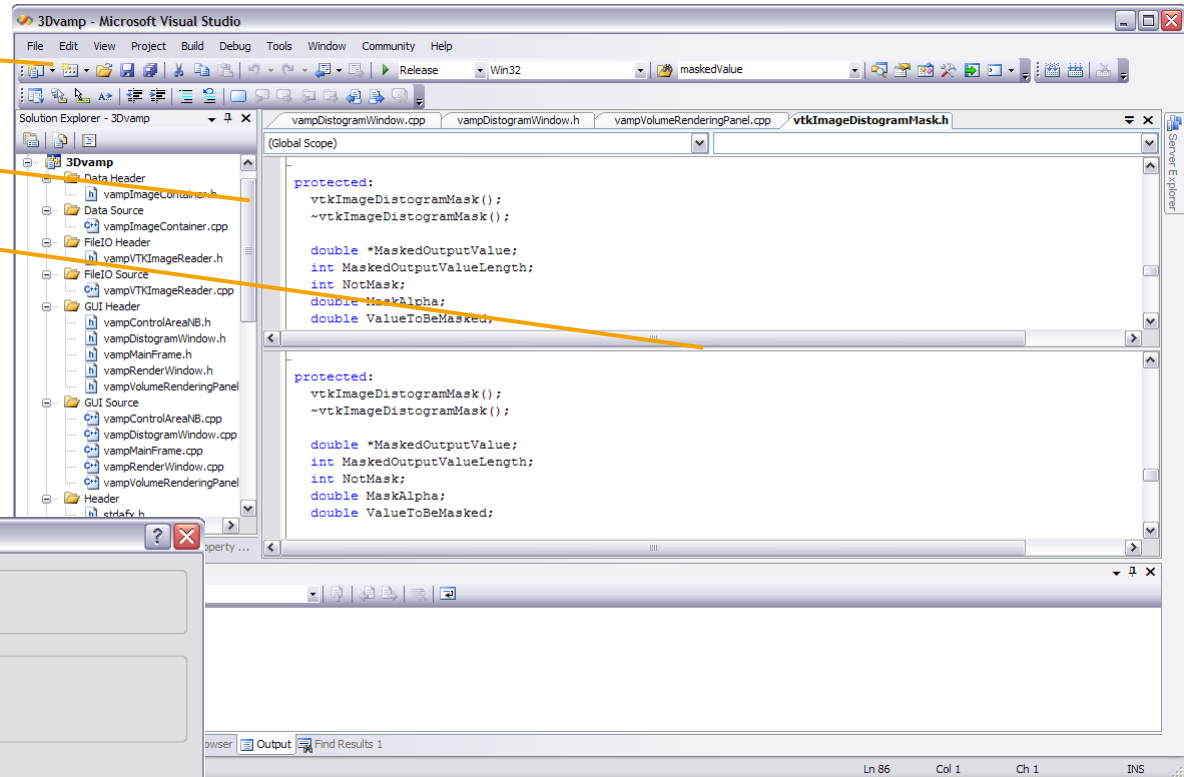


Weitere Komponenten

Toolbars

Scrollbars

Splitter



Textboxen

Buttons



- Einführung
- UI: Interaktion
 - Interaktionsmöglichkeiten
 - Interaktionsdesignprozess
- **GUI: Benutzeroberflächen**
 - Komponenten
 - **Menübasierte Programme**
 - GUI-Anwendungen und eventbasiertes Programmieren
- 3D-Interaktion
- Zusammenfassung

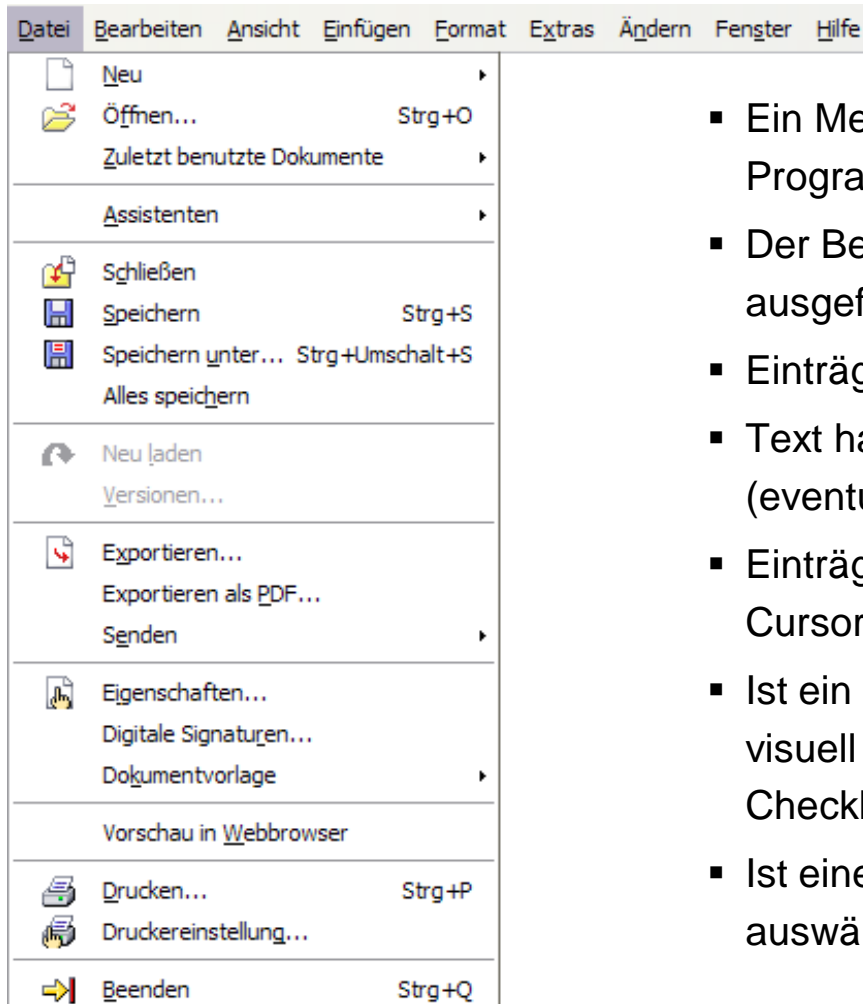
Menübasierte Programme



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Menüs

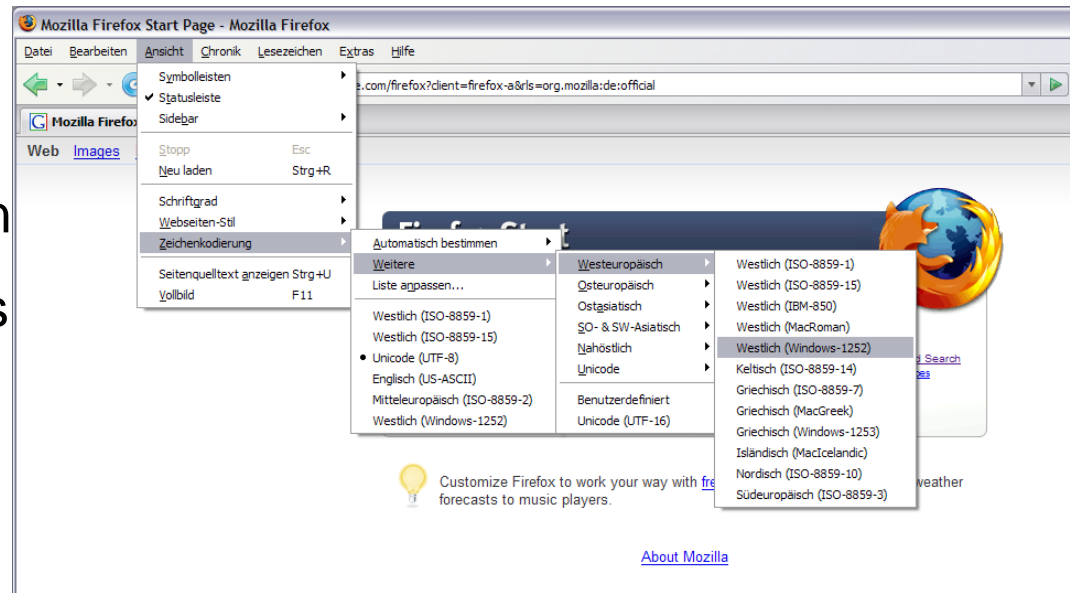


- Ein Menü ist eine Liste von Aktionen, die von einem Programm angezeigt wird
- Der Benutzer kann auswählen, welche Aktion ausgeführt werden soll.
- Einträge können beliebige Daten sein
- Text hat sich aber im Allgemeinen durchgesetzt (eventuell mit Icons visualisiert)
- Einträge sollten hervorgehoben werden, wenn der Cursor darüber liegt
- Ist ein Eintrag auswählbar, so sollte eine Auswahl visuell hervorgehoben werden (z.B. mithilfe einer Checkbox)
- Ist eine Option nicht verfügbar, sollte sie nicht auswählbar und zudem visuell „ausgegraut“ sein

Untermenüs

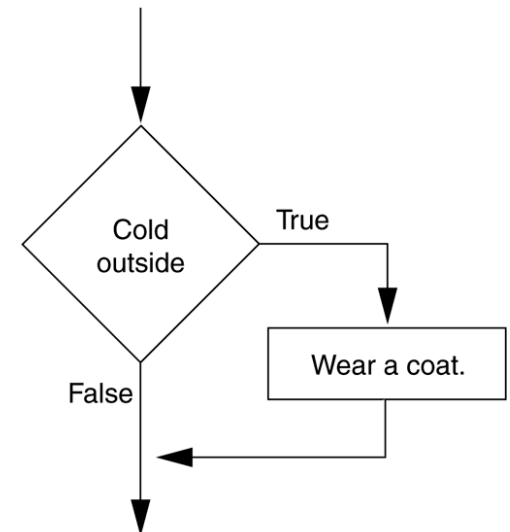
Menüs können Untermenüs enthalten, um zusammengehörige Optionen zu gruppieren.

- Die Entscheidung wird erleichtert, da die Möglichkeiten eingeschränkt werden
- Einträge, die zu Untermenüs führen, sollten dies visuell hervorheben



Eine *Entscheidungsstruktur* kann verwendet werden, um Selektionen im Menü durchzuführen.

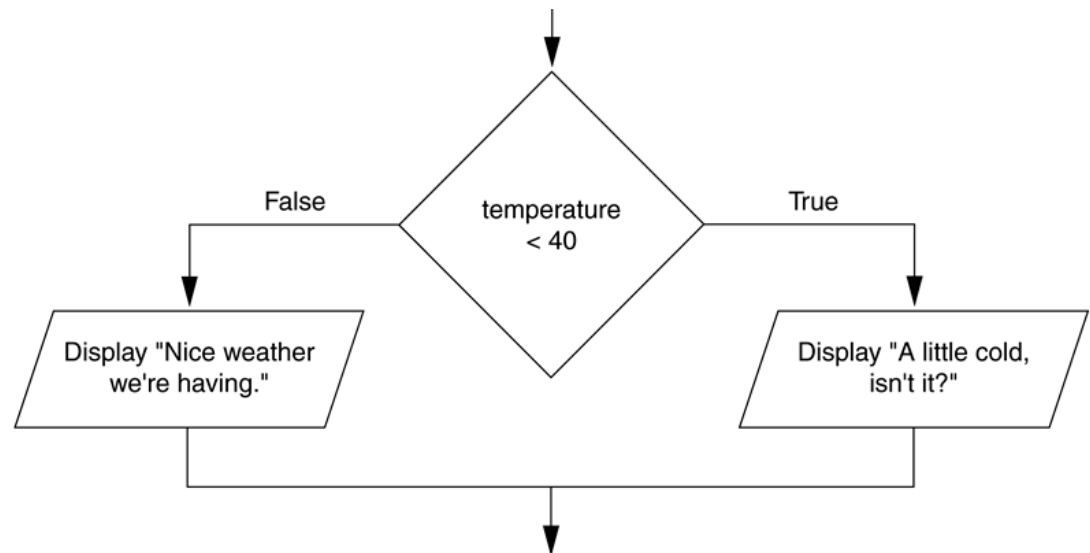
- Es kann entweder eine **case-Struktur** oder eine Verschachtelung von **if-then-else-Ausdrücken** verwendet werden
- Eine case-Struktur ist leichter verständlich
- Eine case-Struktur stellt für jeden Menüeintrag einen Fall bereit und zusätzlich einen Standardfall (default)



Eine einfache Entscheidungsstruktur

if-then-else-Ausdruck

Eine Gruppe von Ausdrücken wird ausgeführt wenn ein boolscher Ausdruck *true* ist, eine andere Gruppe wird ausgeführt, wenn der Ausdruck *false* ist.



Relationale Operatoren

- Bestimmen, ob ein bestimmter Bezug zwischen zwei Werten besteht
 - Werden innerhalb der Bedingung verwendet
- $x > y$ $x < y$ $x \geq y$ $x \leq y$ $x == y$ $x != y$

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

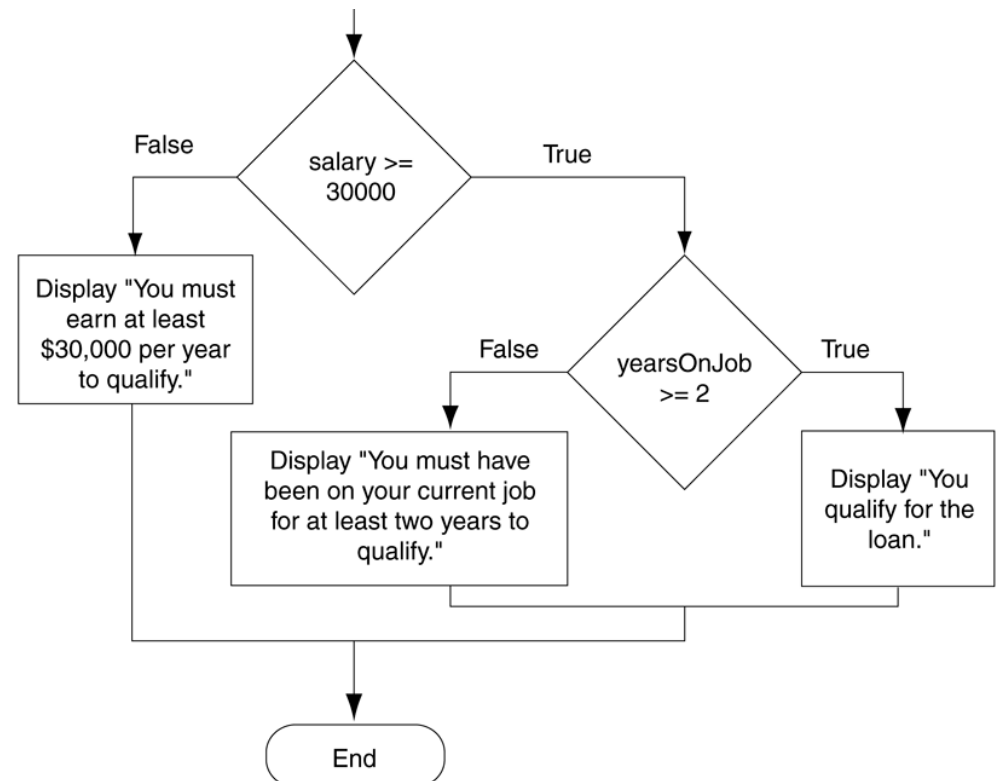
Verschachtelte Entscheidungsstrukturen



Entscheidungen werden geschachtelt, um mehrere Bedingungen zu prüfen

→ **if-then-else-if**

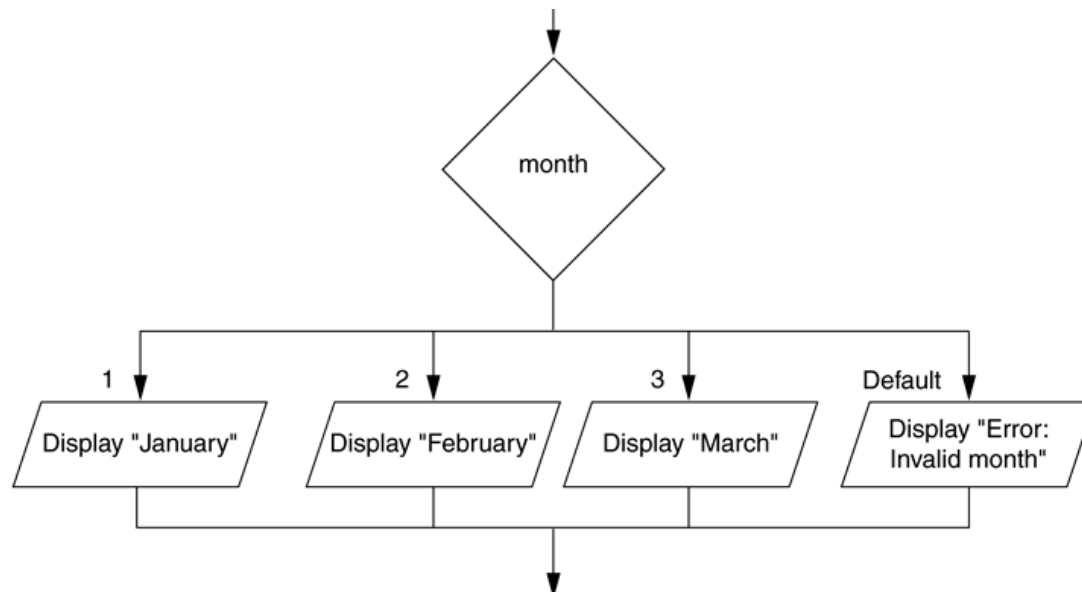
```
If score < 60 Then
    Display "Grade is F."
Else If score < 70 Then
    Display "Grade is D."
Else If score < 80 Then
    Display "Grade is C."
Else If score < 90 Then
    Display "Grade is B."
Else
    Display "Grade is A."
End If
```



Die case-Struktur

Hier legt der Wert einer Variablen oder eines Ausdrucks fest, welchen Pfad das Programm nehmen wird.

- Kann als Alternative zu verschachtelten Bedingungen verwendet werden

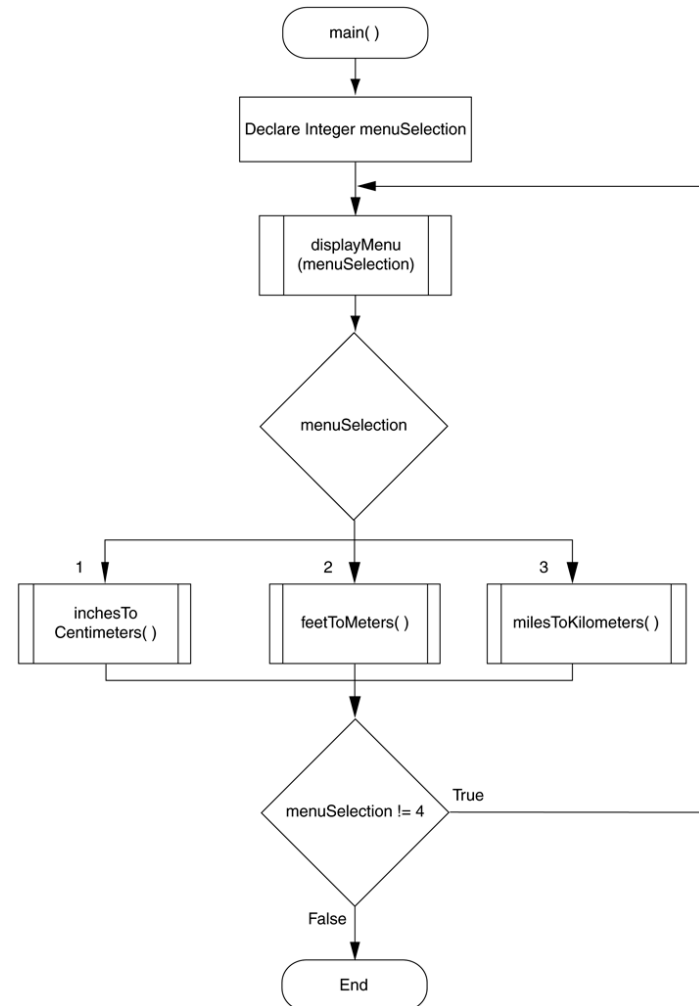


Modularisierung eines menübasierten Programms

Da ein menübasiertes Programm viele Aufgaben ausführen kann, sollte es in Module aufgeteilt werden

- Ein Modul sollte für jeden ausführbaren Fall existieren
- Jeder Menüeintrag würde dann einfach Module aufrufen
- Erlaubt einen klaren Programmfluss

Menübasierte Programme benutzen Schleifen, um wiederholt zu prüfen, welcher Eintrag gewählt wurde





- Einführung
- UI: Interaktion
 - Interaktionsmöglichkeiten
 - Interaktionsdesignprozess
- **GUI: Benutzeroberflächen**
 - Komponenten
 - Menübasierte Programme
 - **GUI-Anwendungen und eventbasiertes Programmieren**
- 3D-Interaktion
- Zusammenfassung

GUI-Anwendungen und eventbasiertes Programmieren



TECHNISCHE
UNIVERSITÄT
DARMSTADT



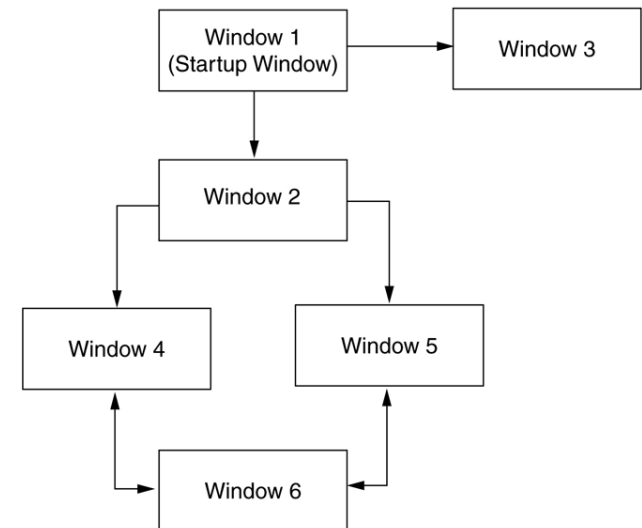
Graphical User Interfaces (GUIs)

Erstellen eines GUI-Programms:

- Die meisten Schritte bleiben gleich
- Allerdings müssen zusätzlich die GUI-Elemente gestaltet werden, die in jedem Fenster enthalten sind
- Dies beinhaltet den Verlauf von Fenster zu Fenster

GUI-Entwicklung beinhaltet die Erstellung aller Fenster und grafischer Komponenten

- Früher war das ein mühsamer Prozess
- Neuere Sprachen erlauben einfachere Entwicklung
 - Mit Visual Studio
 - C#
 - C++
 - Visual Basic
 - Mit NetBeans und JBuilder
 - Java

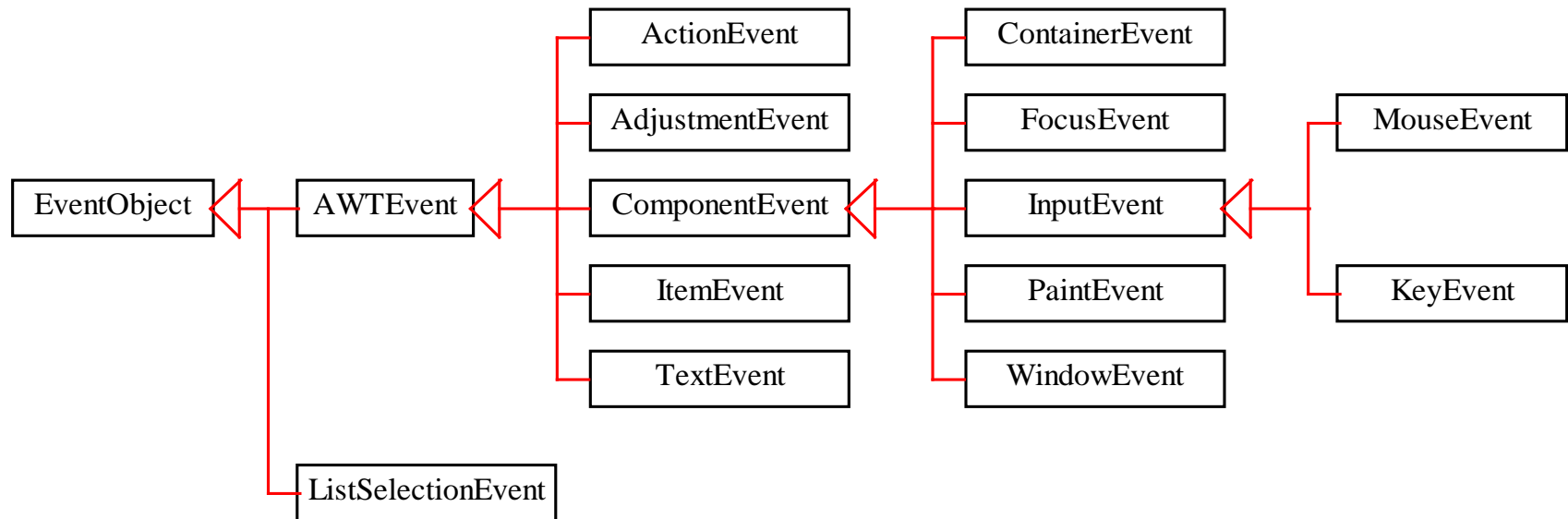


Event-Handler reagieren auf Ereignisse, die durch Aktionen innerhalb der GUI ausgelöst werden

(z.B. durch das Klicken auf einen Button)

- Event (Ereignis): Eine Aktion, die innerhalb eines Programms geschieht
- Event Handler: Ein Modul, das ausgeführt wird, wenn ein bestimmtes Ereignis ausgelöst wird
- In prozeduraler Programmierung wird Code in prozeduraler Reihenfolge ausgeführt
- In eventbasierter Programming wird Code als Antwort auf auftretende Ereignisse ausgeführt

Event-Klassen (Java)



Ausgewählte Nutzeraktionen (Java)



Nutzeraktion	Ereignistyp	Quelle
Klick auf einen Button	ActionEvent	JButton
Doppelklick auf einen Listeneintrag	ActionEvent	JList
Auswählen eines Listeneintrags	ItemEvent	JList
Auswählen eines Comboboxeintrags	ItemEvent	JComboBox
Maustaste über einer Komponente betätigen	MouseEvent	Component

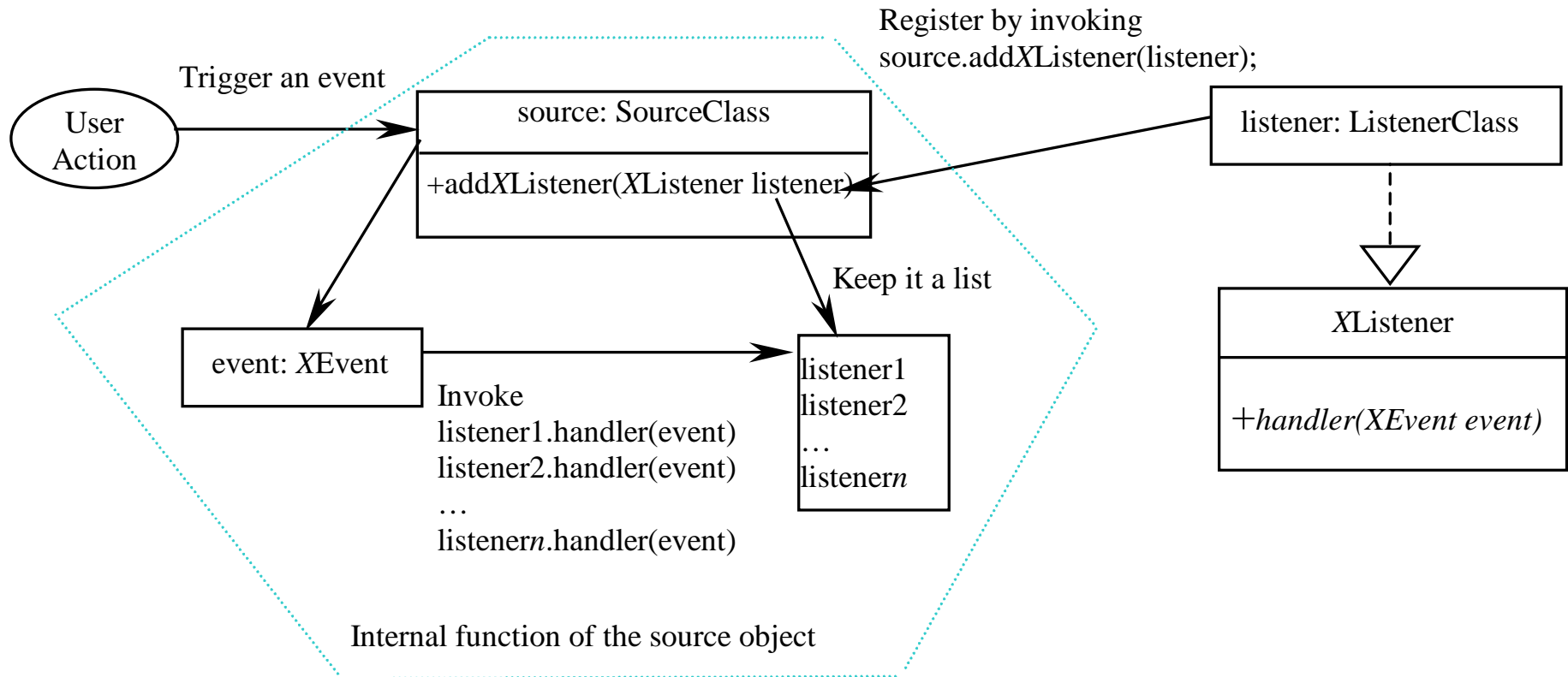
// Beispiel: Ein Event-Handler, der ein Fenster schließt

```
Module exitButton_Click()
```

```
    Close
```

```
End Module
```

Das Delegationsmodell (Java)



Ein Ereignis wird nur von der Quellkomponente weitergereicht, wenn dafür Interessenten (Listener) registriert sind (zB -> <http://www.teialehrbuch.de/Kostenlose-Kurse/JAVA/>)

Beispiel: Mausereignisse verarbeiten (Java)



TECHNISCHE
UNIVERSITÄT
DARMSTADT

In Java stehen zwei Interfaces zur Verfügung, um Mausereignisse zu verarbeiten:

- **MouseListener**
 - Klicken
 - Maustaste drücken und loslassen
 - Mit dem Mauszeiger Komponenten betreten und verlassen
- **MouseMotionListener**
 - Bewegen der Maus
 - Dragging von Objekten



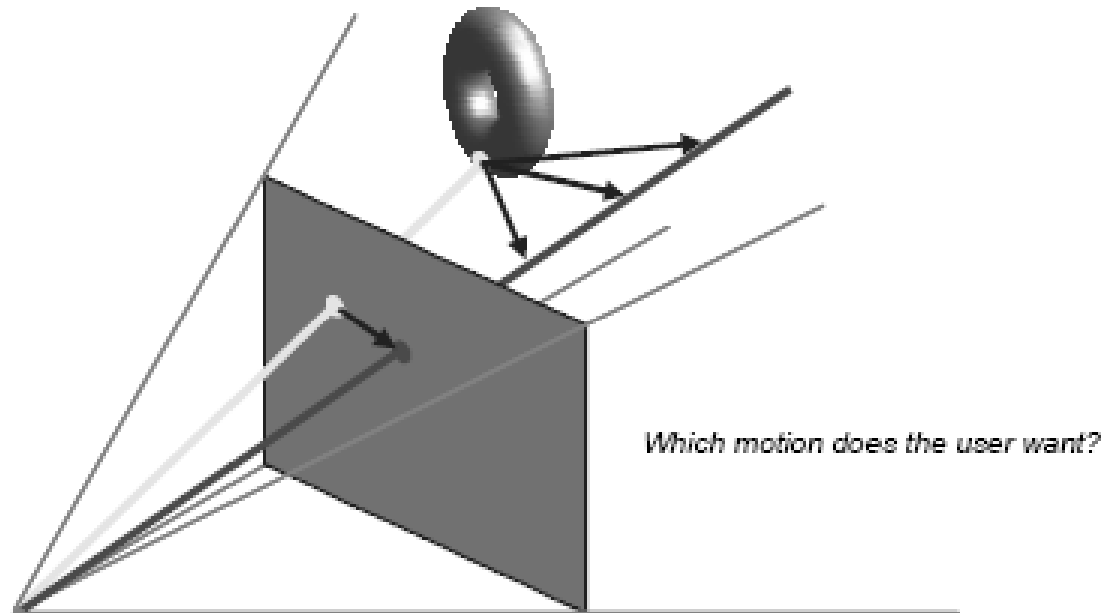
- Einführung
- UI: Interaktion
 - Interaktionsmöglichkeiten
 - Interaktionsdesignprozess
- GUI: Benutzeroberflächen
 - Komponenten
 - Menübasierte Programme
 - GUI-Anwendungen und eventbasiertes Programmieren
- **3D-Interaktion**
- Zusammenfassung

3D-Interaktion mit 2D-Eingabegeräten

Problem:

Welche Art der Bewegung möchte der Benutzer ausgeführt haben?

- Mehrdeutigkeit
- VL Transformationen





- Einführung
- UI: Interaktion
 - Interaktionsmöglichkeiten
 - Interaktionsdesignprozess
- GUI: Benutzeroberflächen
 - Komponenten
 - Menübasierte Programme
 - GUI-Anwendungen und eventbasiertes Programmieren
- 3D-Interaktion
- **Zusammenfassung**

Zusammenfassung UI & GUI

Es gibt viele Interaktionsmöglichkeiten und -methoden

- Es wird dem Benutzer ermöglicht, mit Computern zu interagieren

GUIs erleichtern den Umgang mit Geräten

- Metaphern: Visuelle Entsprechungen realer Objekte

Heutige GUIs: Meist WIMP

- Die Komponenten sind zwischen verschiedenen Betriebssystemen ähnlich

GUI-Anwendungen reagieren durch ereignisbasierte Programmierung auf Benutzereingaben

Interaktion mit einer 3D-Welt mittels eines 2D-Geräts ist schwierig und erfordert zusätzliche Manipulatoren

Die Folien basieren teilweise auf Folien, die von *Pearson Education* als Material zu folgenden Büchern bereitgestellt werden:

- **Steven Heim:** *The Resonant Interface: HCI Foundations for Interaction Design*, 1/e, ©2008, Addison-Wesley
- **Tony Gaddis:** *Starting Out with Programming Logic and Design*, 1/e, ©2008, Addison-Wesley
- **Y. Daniel Liang:** *Introduction to Java Programming with JBuilder*, 3/e, ©2004, Prentice Hall