

Zusammenfassung Interaction

Philipp Jäcks

22. Februar 2016

Inhaltsverzeichnis

1	Interaction Engineering - Fragenkatalog	2
1.1	Introduction	2
1.2	Basics of Graphics and Interaction Programming	5
1.3	Fundamental Interaction Concepts	9
1.4	Advanced Interaction Concepts	13

1 Interaction Engineering - Fragenkatalog

1.1 Introduction

1. Erkläre Stärken und Schwächen des Menschen und des Computers in der Human-Computer Interaction!

Mensch	Computer
<ul style="list-style-type: none">• Kreativität• Abstrahieren und Erzeugen von Modellen• auf Unerwartetes reagieren	<ul style="list-style-type: none">• 'exakte'¹ Berechnungen• stundenlanges Ausführen derselben Aufgabe ohne zu Ermüden• exakter Speicher (Gedächtnis)

Tabelle 1: Stärken des Menschen und Computers in HCI. (Stärken = Schwächen des anderen!)

2. Kernunterschied bei der Entwicklung von computational solutions und interactive solutions

- Interaction: Kernfaktor ist das Human Computer Interface
- Computational: Kern ist effizienter Algorithmus(?)
- Rechenleistung steigt immer weiter während menschliche Aufnahmefähigkeit stagniert/konstant ist
- Kernfaktor bei der Informationsverarbeitung durch den Menschen ist das Interface
- \Rightarrow Interaction soll smooth und effizient; Feedback soll reich an Informationen und instantan sein
- HC-Interaction - Mensch und Computer gehen Hand in Hand, jeder erfüllt die Aufgaben, die er am besten lösen kann (siehe Tabelle 1)

Computation - closed system	Interaction - open system
<ul style="list-style-type: none">• Eingabe• Verarbeitung• Ausgabe• deterministisch, Endzustand	<ul style="list-style-type: none">• Veränderung in der Umwelt• Empfange Events• Reagiere auf Events• endlos, nichtdeterministisch

3. Action Cycle by Norman

- a) Mensch hat Ziel im Kopf (Goal)
- b) Planen der notwendigen Schritte (Plan)
- c) Spezifizieren der Schritte (Specify)

¹Denke an max. Genauigkeit von Fließkommazahlen!

- d) Umsetzen der Schritte in der Welt (Perform)
- e) Feedback in der Welt beobachten (Perceive)
- f) Feedback interpretieren (Interpret)
- g) Ergebnis mit Zielen vergleichen (Compare)
- h) Beginne bei Schritt 1 bzw. 2

ACTION CYCLE

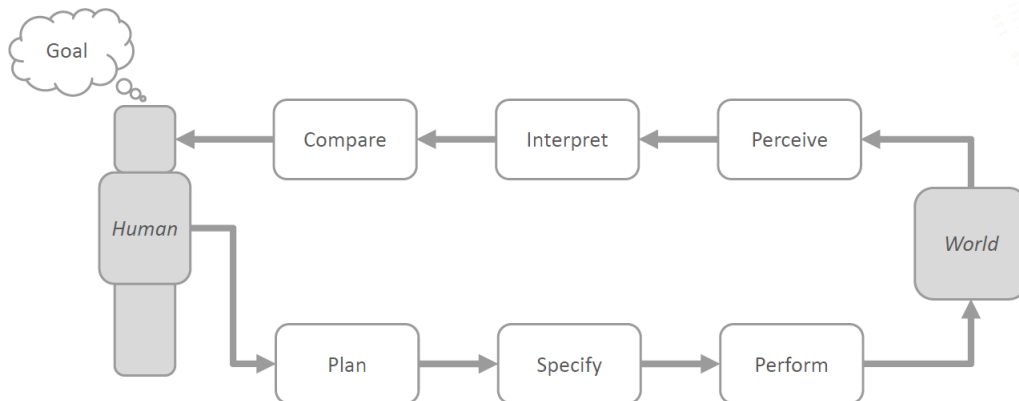


Abbildung 1: Action Cycle nach Norman

4. Iteration/Bsp für den Action Cycle

Am Beispiel: Kaffee holen in der Mensa

- **Goal:** Kaffee in der Mensa holen
- **Plan:** Aus dem Büro gehen
- **Specify:** Operation - Türgriff betätigen um Bürotür zu öffnen
- **Perform:** Türgriff drücken
- **Perceive:** Griff öffnet das Schloss, Tür öffnet sich
- **Interpret:** Tür ist offen
- **Compare:** Schritt erfolgreich, Führe weitere Schritte aus

5. Gulf of Execution and Evaluation

• Execution

beschreibt die Mühe/Aufwand der angestrebten Aufgaben

Kann ich das tun? Wo ist die notwendige Funktionalität? Welches Gerät nutze ich? Wie führe ich das Kommando aus?

- **Evaluation**

Beschreibt die Mühe/Aufwand die Veränderung der Umwelt zu interpretieren
Ist überhaupt etwas passiert? Wo ist etwas passiert? Was ist passiert? Passen Effekt und Absicht zusammen?

- Interaction cost = Summe des physischen und mentalen Aufwandes um ein Ziel zu erreichen
- Beispielhaft am Action Cycle:
 - Cost of Decision (Goal): Fokus muss auf Teilmenge von Informationen und Interfaces gelenkt werden
 - Cost of System Power (Plan): Übersetzen von Zielen im Kopf in Operationsequenzen ist schwer, insbesondere für komplexe Systeme
 - Cost of visual clutter/visuelle Überfüllung/reizung (Perceive): Bsp - Mouse Hover Effekte erzeugen Überreizung und erschweren Zustandswahrnehmung

6. The Three levels of interaction

- **low level:** Selection and Manipulation
- **inter-mediate level:** Exploration and Navigation
- **high level:** Problem-solving

7. The levels of (human) interaction processing

- Instinktiv (Perform and Perceive): vollkommen unterbewusst, ohne Kontrolle, schnell, Basisfähigkeiten - Bsp: Arm bewegen um Türgriff zu fassen
- Behavioral (Specify and Interpret): teilweise unterbewusst, leichte Kontrolle, schnell, gelernte Fähigkeiten - Bsp: Drücken der Klinke öffnet Tür
- Reflective (Plan and Compare): volles Bewusstsein, langsam, komplexe Analyse - Bsp: Tür ist offen, was bedeutet das?

8. at least 5 golden rules or guidelines for interaction

Golden Rules - Norman

- Discoverability: Welche (möglichen) Aktionen können bestimmt werden?
- Feedback: reichhaltiger und kontinuierlicher Fluss an Informationen über den Zustand
- Affordances: angemessene Aufforderungen um die gewünschte Aktion durchzuführen
- Signifiers: effiziente Signalgeber für Discoverability und Feedback
- Mappings: gute Zuordnung zwischen Controls and Actions

Guidelines - Shneiderman

- Konsistenz: ähnliche Situationen sollen ähnliche Aktionen erfordern

- b) Universal Usability: Assistenz anbieten (Hilfe, Shortcuts,...)
- c) Informative Feedback: Feedback für jede Useraktion
- d) Closure: klarer Beginn, Ablauf und Ende einer Aktion; kombiniere mit Punkt 3
- e) Prevent Error: vermeide fehlerhaften input, Recover from user error
- f) Easy reversal of actions: zb undo redo
- g) internal locus of control: User kontrolliert das System
- h) reduce short-term memory load: keep it simple

Sonstige Notizen

- Vor- und Nachteile der Interaction

Vorteile	Nachteile
<ul style="list-style-type: none"> • ist mächtiger als "Algorithmen" • anspruchsvolleres Verhalten 	<ul style="list-style-type: none"> • User muss wissen <i>was</i> er/sie möchte • User muss wissen <i>wie</i> er/sie den Computer bedienen muss um das Ziel zu erreichen <ul style="list-style-type: none"> • Anwendung ist zustandsbehaftet -> User kann sich verlieren/steckenbleiben (stateful things can be broken)

- Bottlenecks - Processing: CPU, RAM, Netzwerk etc.. mittlerweile in vielen Anwendungsgebieten nicht mehr so relevant
- Bottlenecks - Information: enorm wichtig welche Daten auf dem kleinen Bildschirm am Ende angezeigt werden (viele, viele Daten gespeichert; welche Davon sind wichtig und werden angezeigt?)
- Bottlenecks - Aufnahmefähigkeit Mensch

1.2 Basics of Graphics and Interaction Programming

1. Illustrate the interplay of the action cycle and the model-view-controller pattern!
2. Discuss different forms of presentation!
Wahrnehmungen durch die 5 Sinne, geordnet nach Relevanz für HCI:
Visuell, Audio, Fühlbar, Geruch, Geschmack
wichtige Aspekte: Bandbreite, Aufmerksamkeit, Vergänglichkeit/Flüchtigkeit
3. Discuss different forms of expression!
Sprache, Point & Gestures, Physical movement (sich selbst, Objekte wie Maus, Tastatur..)
wichtige Aspekte: Genauigkeit, Geschwindigkeit, Aufwand

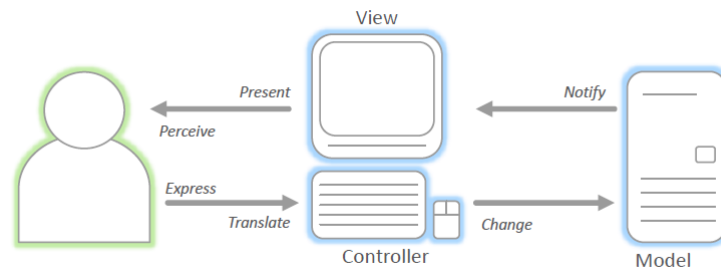


Abbildung 2: Zusammenspiel Action Cycle und MVC

4. Discuss pros and cons of uni-modal and multi-modal interaction!

Uni-Modal: genau eine Form der Presentation und Expression (zb Visuell und Point & Gestures) \Rightarrow ein Kanal

Multi-Modal: mehrere Formen/Kanäle; zb Visuell, Audio und Touch, Sprache, Point..

	Vorteile	Nachteile
Uni-Modal	einfache Implementierung	nur ein Kanal
Multi-Modal	umfangreiche Formen der Interaktion	komplexe Implementierung und schwieriger zu lernen (für den User)

5. What are the mental, implementation and represented model? Why are they important?

- **Mental:** gedankliche Vorstellung des Modells; entspricht menschlicher Natur; beschreibt v.a. Operationen zum Erreichen des Ziels
- **Implementation:** interne Darstellung; technisch limitiert und vom Entwickler vorgegeben; enthält Daten, Parameter, Algorithmen,...
- **Represented:** Darstellung des (Implementation) Models auf dem PC; vom Entwickler vorgegeben;

Important, weil Mental Model nicht 1:1 in Computer dargestellt werden kann; müssen abstrahieren

6. What is a graphics context and what can it be used for?

Wird gestellt vom Window System und ist verknüpft mit dem Zeichenareal. Stellt Funktionalitäten zum Zeichnen bereit.

7. Explain the basic procedure for drawing paths!

begin path; move to; add (line, curve, arc,...); end path

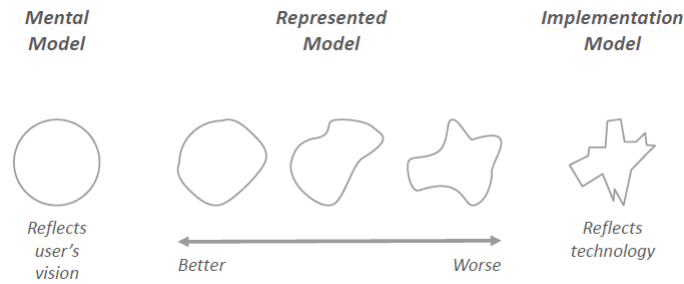


Abbildung 3: Zusammenspiel der 3 Modelle

8. Name three interaction devices and characterize the input they deliver
 Pointing Device (Maus): absolute oder relative Koordinaten
 Triggers (Tasten)
 Value Input (Sensoren)
9. Give examples of atomic and composite inputs!
 atomic: single click, Bewegung des Zeigers, aktiviere Trigger,...
 composite: double click, drag n drop, Gesten,...
10. Illustrate the interconnections between model, view and controller in MVC pattern!

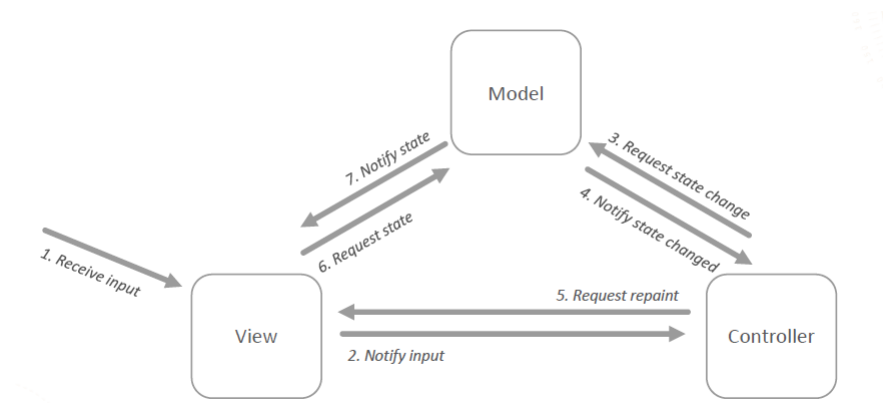


Abbildung 4: Interconnection der drei Bestandteile des MVC Pattern

11. Discuss advantages and disadvantages of the MVC!

Vorteile	Nachteile
klare konzeptuelle Trennung der Bestandteile	• enge Kopplung der Komponenten (durch Interaktion)
• Entwurf für generelle Architektur	• komplex zu implementieren

12. What loop is the fundamental ingredient of interactive systems!
event loop: Endlosschleife von Reaktionen auf Events; verschiedene Eventtypen; verschiedene Reaktionen
13. Name at least five types of events in interaction systems!
- Application level: loaded, finished,...
 - Widget level: Repaint, Resize, Activated, focused,...
 - Input level: key pressed/released/typed,.. Touch start/moved/end, Mouse pressed/...
 - Window System signals external event: repaint of window required; resize of window; Input von Peripherie (zb Maus)
 - Application signals internal events: state change, timer elapsed
14. How are events propagated in interactive systems?
 Werden in EventQueue reingesteckte und nach FIFO verarbeitet;
 Dispatch Events: Traversiere Window Tree und wähle vorderstes Fensters aus
 Receive Events: definiere Verhalten bei Event; ausgeführt durch Listener
15. How can callback functions be used to react to events?
 Callback als Parameter einer Funktion -> Funktion ausgeführt, dann führe Callback aus. Auf Events: Ich sage Programm mach etwas und du sollst mir Bescheid geben(=callback), Ausführung erzeugt Event und callback wird gerufen(omfg.. xD)
 gut für basic (device) events; but limited flexibility
16. What are delegates and listeners? Interface um auf Events zu reagieren; Logik muss vom Entwickler umgesetzt werden
 cover many different events, also user-generated ones; but sometimes complex
17. What are signals and slots and which toolkits make use of them? QT, GTK+,...
 Kommunikation zwischen Objekten; Signal wird emitted wenn Event auftritt; Slot = Funktion die bei bestimmtem Event aufgerufen wird
 ähnlich zu Callbacks, aber mit höherer Flexibilität
18. Express in pseudo code the main event loop and explain its components!
 Block and render -> wartet auf Event und rendert nur bei Bedarf neu

```

while (true || event != QUIT) {
    event = wait_for_event();
    do_repaint = update_model(event)
    if (do_repaint) {
        render_model()
    }
}

```

check and render full-throttle -> render mit jedem Durchlauf und reagiert nur auf Events


```

while (true || event != QUIT) {
    if (check_for_event()) {
        event = get_event()
        update_model(event)
    }
    render_model()
}

```

19. What is a disadvantage of method overriding for reacting to events? sollte in Sub-Class ausgelagert werden um Custom Verhalten umzusetzen; im Gegensatz zum Listener Konzept wesentlich unübersichtlicherer Code

1.3 Fundamental Interaction Concepts

- At which three levels can interaction be considered?
 - Low level: basic picking and manipulation
 Event Notifications: *Etwas* ist passiert;
 Event Type: *Was* ist passiert;
 Interaction abhängig vom räumlichen Kontext: also *wo* etwas passiert ist;
 Interaction Handling = Event Notification + Event Type + räumlicher Kontext
 - Intermediate level: Kombiniere low level Techniken zu navigieren, exploration, modellieren
 - High level: kombiniere intermediate level Techniken zu komplexen Prozessen (Understanding, Kreativität)
- Why is picking necessary for human-computer interaction?
 Picking - Grundelement für Interaktion, Mensch ist es in realer Umgebung gewohnt Dinge anzufassen bevor er mit ihnen interagiert
 Für Picking extra Ebene -> MDPC - model, display view, picking view, controller
 Picking view beschreibt Interaction Geometry (ermöglicht räumlichen Kontext) -> Regionen mit denen interagiert werden kann
 Vorteil der View Trennung: einfacheres Modellieren, Testen; GUI kann ausgetauscht werden ohne Interaktion zu verlieren; verbessertes Verhalten (Bsp: Dropdown Mouse Movement)
- Characterize the picking problem! What is given, what is sought(gesucht)? Given: Interaction Geometrie $G = \{g_1, \dots, g_n\}$ und eine Position auf dem Bildschirm $P = (x, y)$
 Gesucht: Interaction Geometrie $G' \subset G$ am Punkt P , wobei
 $|G| = 0$, wenn nichts matcht
 $|G| = 1$, für einen eindeutigen Match
 $|G| > 1$, für mehrere Matches
Technical Requirements: Speed -> Picking muss innerhalb ms identifiziert werden;

Picking requests können sehr schnell eintreffen

Genauigkeit: muss exakt sein, front-most object

Human R: Fitt's law: betrachte beim Design die Zeit, die Nutzer für Interaktion vor. benötigen wird

4. Explain the role of essential geometry and "MDPC"! MPDC - siehe 2 Fragen vorher
Essential Geometry - Grundlegende Geometrie (Punkte, Linien, Shapes, Objects, Rays):

Controller muss drag region auf model mappen. Regions setzen sich zusammen aus den essential Geometry Bestandteilen -> Essential Geometry dient quasi als Connection zw. View und Controller beim Picking

Vier muss Regions kennen (zwecks rendern)

5. Sketch the basic implementation strategies for picking!

Screen Space Picking

- Render Szene in Picking Buffer, Geometrie hat dort eine ID
- Benutze Pointer Koordinaten als direkter Index für den Picking Buffer
- Javascript: Draw Path -> apply HitRegion -> Event: Abfrage welche HitRegion getroffen wurde
- OpenGL: zusätzliches RenderTarget -> Elementen IDs zuweisen -> Geometrie mit Hilfe der IDs (in Form von Farben) rendern -> look up ID from picking buffer

Object Space Picking

- Projiziere Pointer in die Szene mit Hilfe inverser Transformationen
 - Berechne Containment/Intersection mit der Szenen Geometrie
 - Nearest Neighbor Search:
 - Finde Geometrie am nächsten zu P
 - Optional: beschränke Picking auf Epsilonumgebung
 - machbar für Punkte(0D) und Linien(1D)
 - Punkte: Berechne Distanz von P zu allen Punkten; nimm Punkt mit geringster Distanz
 - Linien: Lot von P auf Linien fällen -> nimm Linie mit kürzestem Lot
-
- Containment Search:
 - Finde Geometrie, die P enthält
 - machbar für Shapes (2D) und Objekte (3D)
 - Shapes: Check if P is contained in geometry

- Objects: Wandle P in Ray/Strahl um und prüfe ob Strahl die Dreiecke des Objekts schneidet
6. Discuss different ideas to accelerate the picking! **Computation Performance**
- Reduziere Komplexität der Geometrieberechnung
 - Sub-linear Search -> benutze hierarchische Strukturen (Bäume,...) um Suchen zu beschleunigen
 - Pointer Kohärenz: gucke zunächst in Nachbarschaft

User Performance

- unterschiedliche Picking Mechanismen mit teils mehr Features
 - Sticky Cursor
 - Bubble Cursor
 - Bubble Lens
7. Selektionsarten:
- Click Selection: Nutzer klickt Elemente einzeln an -> Serie von einzelnen Pickingaktionen; typisch: Simple Click, Strg+ Click, Shift + Click
 - Rectangle Selection: Nutzer zeichnet Rechteck. Selektion entweder *inside*: shape was auch immer muss zu 100% im Rechteck enthalten sein gut für kleine, reguläre Elemente -> Bsp Desktop Icons
Nachteil: bei unregelmäßigen oder großen Strukturen muss ich gut abschätzen können, ob mein Rechteck alle Elemente einkapselt *overlap*: Elemente müssen Rechteck nur schneiden
Vorteil: gut für unregelmäßige und große Elemente (zb Landkarte)
Nachteil: Frage: wie viel muss geschnitten werden?!
 - Lasso Selection: Selektion innerhalb free-form shape -> Nutzer zeichnet mit Maus quasi eine freie Form und innerhalb dieser wird selektiert
8. Discuss pros and cons of inside and overlap selection! siehe Frage vorher
9. Grundlegend: Manipulation
Object der Szene -> Interaktion benutzt um Geometrie und Eigenschaften des Objektes zu ändern
View auf die Szene -> Kameramanipulation -> verändert die Perspektive auf die Szene
- Constrained Manipulation: Manipulation nur in bestimmten Bedingungen mgl, Bspw. Scrollbar nur in bestimmte Richtungen manipulierbar
10. What is the difference between absolute and relative manipulation? Mappen des räumlichen Kontexts auf die Manipulation, entweder
absolute: Punkt im Raum wählen -> dann Ergebnis -> abhängig von der aktuellen

Pointer Position

relativ: Pfad durch den Raum wählen -> dann Ergebnis -> Ergebnis abhängig von Serie von Pointer Positionen

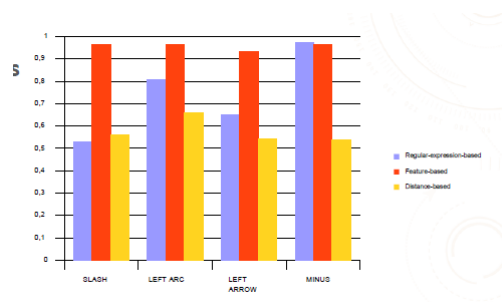
11. Wofür Navigation? -> große Mengen an Daten können nicht alle auf dem Bildschirm dargestellt werden -> Nutzer muss durch diese navigieren können
Wie? Besuche verschiedene Teile der Informationen auf verschiedenen Detaillevel
Requirements: Wo bin ich? Wohin kann ich gehen? Wie komm ich dahin? Was liegt dahinter? Wo ist es sinnvoll hinzugehen? Wo war ich?
12. What is the mental model for actor-centric vs. object-centric manipulation? Discuss consequences for interaction!
Actor: Ich hab das Auge in der Hand und navigiere mich selbst durch die fixe Szene
Object: Ich habe die Welt in der Hand und navigiere indem ich die Szene selbst bewege.
Konsequenzen: Einfluss wie Interaktion mit Hardware umgesetzt wird; Bsp Zoom mittels Mausrad -> roll ich es weg (Object Centered) oder zu mir (Actor Centered)?
13. In which respect is scrolling limited, what is the advantage of scrolling?
lineare Navigation; festgelegte Richtungen; relative Positionierung der View mittels Drag n Drop;
Absolute Positionierung der View mittels Scroll handle;
Feedback der View Position innerhalb der Welt
Pro: Easy to use; Con: Limitierung bei komplexen Problemen durch Linearität
14. Illustrate a space-scale diagram!
 - Grundlegende Idee: durch Skalierung die Einschränkungen des Scrollings umgehen
 - dazu: unterstützt Scaling unterschiedliche Geschwindigkeiten der Navigation -> langsam und präzise, wenn kleine Skalierung; schnell und grob, bei großer Skalierung
 - -> Zoomable UIs = Scaling + Scrolling
 - Space-Scale diagram dient zum Verstehen der ZUIs
 - Zeigt unterschiedlichen Zoomstufen abhängig von Skalierung (Achtung: kann auch unendlich sein)
15. Which questions must a user be able to answer during view navigation?
 - Where can I go?
 - How do I get there?
 - Where am I?
 - What lies beyond?
 - Where can I usefully go?

- Where have I been?
16. Give examples how to support "Where can I go?"
View Space Navigation: Verändere Viewport Position und Größe -> Bsp: smoothes und effizientes Zoomen und Schwenken
Data Space Navigation: exploite Struktur im Datenraum -> Bsp: Kantenbasiertes Reisen
 17. Give examples how to support "How do I get there?"
Schwenken: Drag n Drop (relativ); Pan Wheel (relativ); Scrollbar (absolut)
Zoom: Mouse Wheel rotation (relativ); Zoom Slider (absolute); Elastic Rectangle (Absolute; zoom in only)
 18. Give examples how to support "Where am I?"
Übersichtsfenster (Position); Scrollbar Indikatoren (Position); Zoom Slider (Skalierung); Infinite Grid (Skalierung)
 19. Give examples how to support "What lies beyond?"
Indikatoren, die anzeigen das Elemente außerhalb des Viewport liegens; Bsp: Pfeil, Proxy (Stellvertreter am Rand des Viewport); Halo (Halbkreis mit Objekt als Mittelpunkt); Wedge
 20. Give examples how to support "Where can I usefully go?"
Führung: Fokus = aktuelle View; Context = Nachbarn des Fokus -> bestimmte und bewerte Kandidaten -> Stelle Empfehlungen dar
 21. Give examples how to support "Where have I been?"
History Management wie redo/undo; speichere Zustand des Systems -> Problem: Wann speichern wir diesen? Bei Event? Zeitbasiert? -> Probleme: State change kann sehr schnell passieren (dragging a slider bspw)

1.4 Advanced Interaction Concepts

- Give examples of gestures?
 - **Basic:** Bedeutung des Inputs -> Stroke, Touch, Hand, Body, Facial Gestures
Typen: offline -> Geste performed -> erkannt -> interaction getriggert -> Ende
online: während Geste performed wird -> kontinuierliche Erkennung -> kontinuierliche Interaktion -> Ende
 - **Recognition of Gestures:**
 - Basisprozedur für offline Gesten:
 - Input -> Setze Menge von Referenzgesten und Menge von Interaction Events
 - Processing: Bestimme Referenzgeste passend zum User Input
 - Output: Geste, die User Input matcht

- *Recognition of Stroke gestures:*
 - * Stroke aufnehmen -> Free: ohne Bedingungen; Isochronal: Punkte des Strokes werden in konstanten Intervallen aufgenommen; Equidistant: Punkte des Strokes werden in konstanten Abständen aufgenommen
 - * Distance Based Approach: Menge von Referenzstrokes -> Mappen der Kontrollpunkte -> Zentriere Bounding Boxes des Strokes und Ref-Strokes -> Berechne durchschnittliche Abweichung der Kontrollpunkte
- *Feature Based*
 - * Extract feature Vector aus dem Stroke
 - * Berechne Gemeinsamkeit mit Ref-Stroke
- *RegEx based*
 - * Elemente des Strokes werden auf Chars gemappt
 - * Abgleich des Strings mit RegEx(=Ref-Stroke)
- Are stroke and touch gestures typically online or offline gestures?
stroke offline -> User gibt Input -> Referenz erkennen -> Output
Touch sind online
- Illustrate the procedure of regular-expression-based gesture recognition!
Def. Ref-Stroke als RegEx -> zeichne Stroke als String auf -> RegEx matching
- Which recognition approach you are aware of has the best recognition rate? Why is this so?



Feature Based am genauesten, weil... ich glaub, dort die Referenzstrokes mit lernen, also während der Benutzung an den Nutzer angepasst werden

- Which touch events exist in typical toolkits? What information do they contain?
Touch Start, Move, End, Cancel
Enthalten Liste mit allen Touch Punkten seit die Geste gestartet wurde
- How can a multi-touch pinch gesture be detected on Android devices?
ScaleGestureDetector mit Custom Listener -> füge Detector an der View hinzu -> TouchEvent: verschicke Event an Detector -> Detector ruft listener, einmalig, wenn Geste identifiziert

- Discuss pros and cons of animated visual feedback!
 Änderung startet instantan -> fließender Übergang vom alten zum neuen Zustand durch Animation
 Pros: hilft die mentale Map aktuell zu halten
 Cons: benötigt Zeit, change blindness -> visuelle Veränderung vom Betrachter nicht wahrgenommen
- Sketch the main loop for animated feedback!
 1. Process events
 2. Update Model
 3. render model
 4. Emit repaint Event, wenn mehr Updates notwendig sind
- Are there any studies regarding advantages and disadvantages of animation? ja
- What is the difference between timed animations and converging animations? Animation für eine konstante Zeitperiode; verschiedene Interpolationen (linear, slow-in, slow-out,...); einfacher Weg
 Converging: animiere bis Konvergenz erreicht ist; verschiedene Wege der Zustandsübergangs (spring-mass, smooth and effizient zooming)
- What are easing functions good for?
 vermeiden das unnatürliche Verhalten einer linearen Animation (instantaner Start und Stopp)
- Explain animation via spring-mass systems (Masse Feder System)!
 Konvergierende Animation -> Zustand = Punktmasse mit Pos. p , Beschleunigung a , und Geschwindigkeit v
 Übergang: Unterschied zw. Zuständen erzeugt Kraft; Kraft abhängig von der a der Punkte Masse in Richtung des neuen Zustandes -> a führt zu Bewegung
 Beende Animation, wenn Energie klein genug (und nicht am Ende des Weges); beachte: Feder schwingt ja quasi hin und her
- How can view navigation in ZUIs be enhanced with animation?
 smooth and efficient viewport animation (konvergierendes Verfahren)
 Idee: zoome heraus -> navigiere zum Punkt um zoome währenddessen wieder herein; mit Hilfe von Space Scale Diagram
 effizient = kürzester Pfad; smooth = keine wahrnehmbaren Unterbrechungen
- Contrast interactive lenses against regular interaction, discuss pros and cons?
 Lenses: leichtgewichtig; fokussieren temporäre Effekte
 regulär: schwergewichtig; haben globalen permanenten Effekt
- Give a definition of interactive lenses!
 Lens = Selection + alternative Representation + zusätzliches visuelles Feedback

- Sketch the conceptual model of lenses and the involved components!
 Input data: Selektion von Pixeln, Geometrien oder zugrundeliegenden Daten
 Data processing: Lens Funktion -> verarbeite Input für Linseneffekt; je nach Selektion nur Teilmenge der Verarbeitungsschritte notwendig; kleine Selektion => advanced Operations mgl

 -> 3 Linseneffekte: Veränderung; Ausblendung und Anreicherung Result output:
 Join -> führe Linseneffekt mit Hintergrund zusammen, auf dem Level von Pixeln, Geometrien oder underlying data
- How can lenses be interacted with?
 - Positionierung mittels Maus und Parameterveränderung durch GUI (klassisch)
 - Touch: bsp: zoom der Linse wie aufm Handy halt.. :D
 - fühlbare/greifbare Interaktion: Bsp: Kunststoffring auf Tabletop als Linse -> von Software erkannt und platziert an Rings Position die Linse
 - gaze-based: Interaktion mittels der Augenbewegungen
- What is interaction syntax and why is it important?
 Menge von legalen Eingabesequenzen verbunden mit den Aktionen, die als Ergebnis auf die Eingabe aktiviert werden
 Wichtig für komplexe Interaktion -> erfordert viel Planung wie die Interaktion dem Nutzer gezeigt werden kann
- How can interaction syntax be specified? Discuss benefits and problems!
 - Implizit durch Listener Logik -> führt zu Spaghetti Code
 - Event Diagramme -> Pointer Positionen als Trajektorie + annotierte Events und Zustandsdarstellung entlang der Trajektorie -> sinnvoll für einfache Aufgaben, aber limitiert
 - Interaction State Machines: sollte klar sein.. mit Hilfe von State Charts halt.. siehe SWT VL
- Sketch a basic interaction using a state machine with guards!

