Globetrotter – Dokumentation

Inhaltsverzeichnis

# 1. Thema

## 1.1. Neuausrichtung nach ÖkoSwim

### 1.1.1. Warum nicht mehr ÖkoSwim

- Daten von OÖ teilweise veraltet

- sporadisch erfasst

- nicht vollständig

- Datenformate nicht immer gut maschinenlesbar

### 1.1.2. Analyse anderer Open Data Datenquellen

|  |  |
| --- | --- |
| Land Vorarlberg | - geringe Auswahl an Datensätzen  - Datensätze enthalten nicht viel Informationen  - alle Formate fast nicht maschinenlesbar |
| Stadt Wien | + oft gut maschinenlesbar  + Daten werden aktualisiert  - viele Applikationen bereits umgesetzt |
| Schweiz | + viele Daten  + Aktualität OK  - Datenformat mehrmals nur schwer maschinenlesbar |
| Weltbank | + viele verschiedene Datensätze  + gute Aktualität  + REST Webservice zur Datenabfrage  - manchmal fehlen Länder oder Jahre |

## 1.2. Globetrotter

+ Daten der Weltbank

+ Welt auf dem Globus erkunden können

+ grundlegendes geographisches Wissen aufbauen

+ tiefere Details durch Datensätze der Weltbank

+ Vergleich von Ländern

+ Einsatz im E-Learning denkbar

# 2. Technologien

## 2.1. Applikation

### 2.1.1. WebGL

+ frei verfügbar

- nicht überall unterstützt

- Probleme mit Browserversionen und Grafiktreibern

- noch kein Standard

- zusätzliche Serverkomponente für Steuerung nötig

- JavaScript Client müsste durchgehend Server für Input pollen

- möglicherweise hoher Anspruch an Performance

- kaum Erfahrung im Team

### 2.1.2. Blender

+ frei verfügbar

- müssten zuerst Python erlernen; keine Erfahrung im Team

### 2.1.3. Unity

+ frei verfügbar

+ gute Verbreitung in der Branche

+ Scripting mit C# bei allen Teammitgliedern gut bekannt

- Diagramme wegen fehlender .NET API Teile etwas schwieriger umsetzbar

## 2.2. Steuerung

### 2.2.1. Android Smartphone

+ hohe Verbreitung

+ frei verfügbar

+ breite Kenntnisse im Team zu Android und Java

+ keine Einschränkung auf ein Betriebssystem für die Entwicklung

### 2.2.2. iOS Smartphone

- keine Erfahrung im Team

- Entwicklung nur auf Mac möglich

### 2.2.3. Einsatz von Bewegungssensoren und Kameras

+ im Moment nicht geplant

+ über abstrahierte Schnittstelle leicht später erweiterbar

# 3. Paperprototypes und Interaktionen

## 3.1. Erste Skizzen

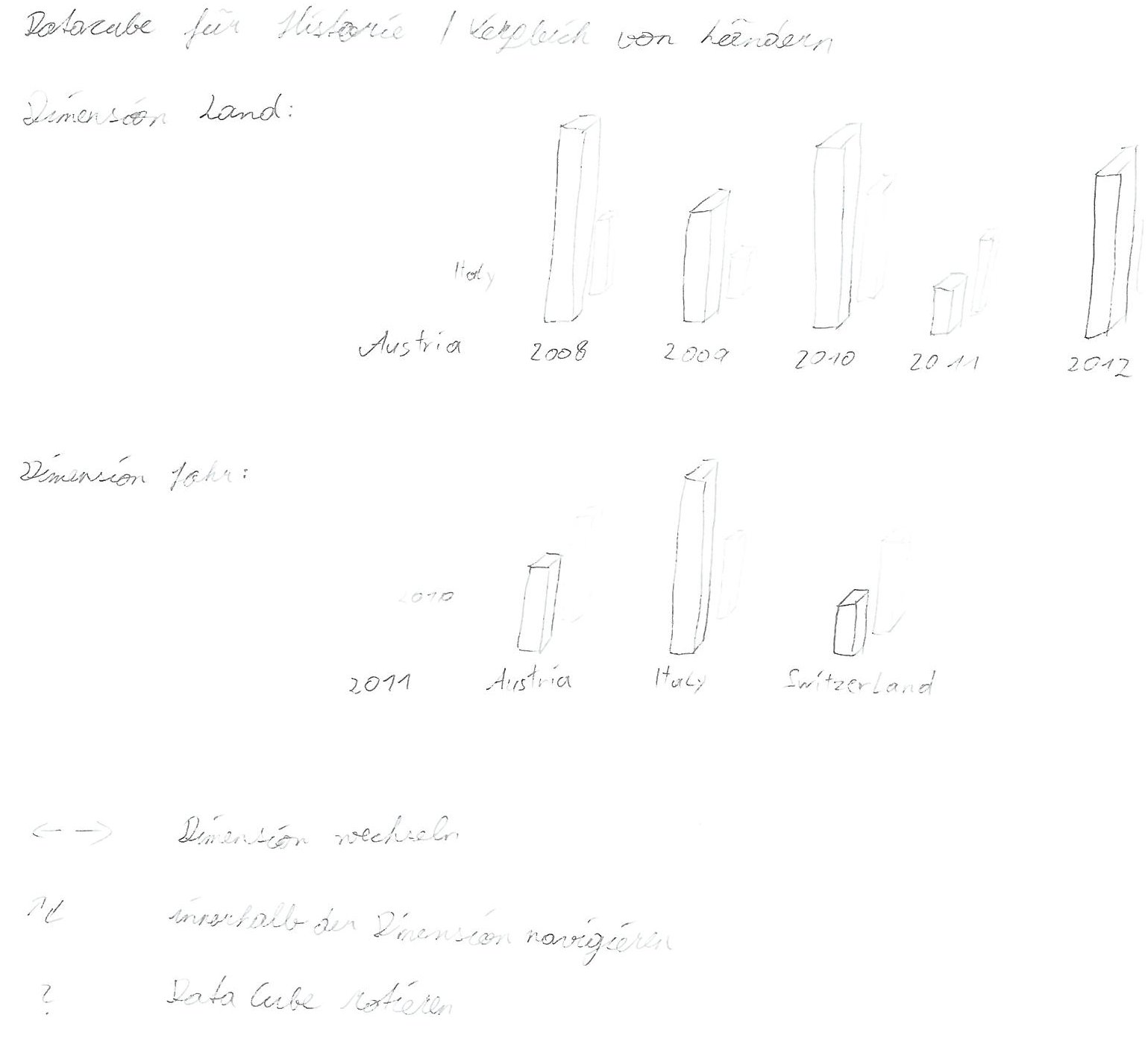
### 3.1.1. Auswahl von Kontinenten, Ländern und Indikatoren



- viele Ebenen in der Auswahl

- lange Listen zum durchklicken

### 3.1.2. Vergleich von Ländern

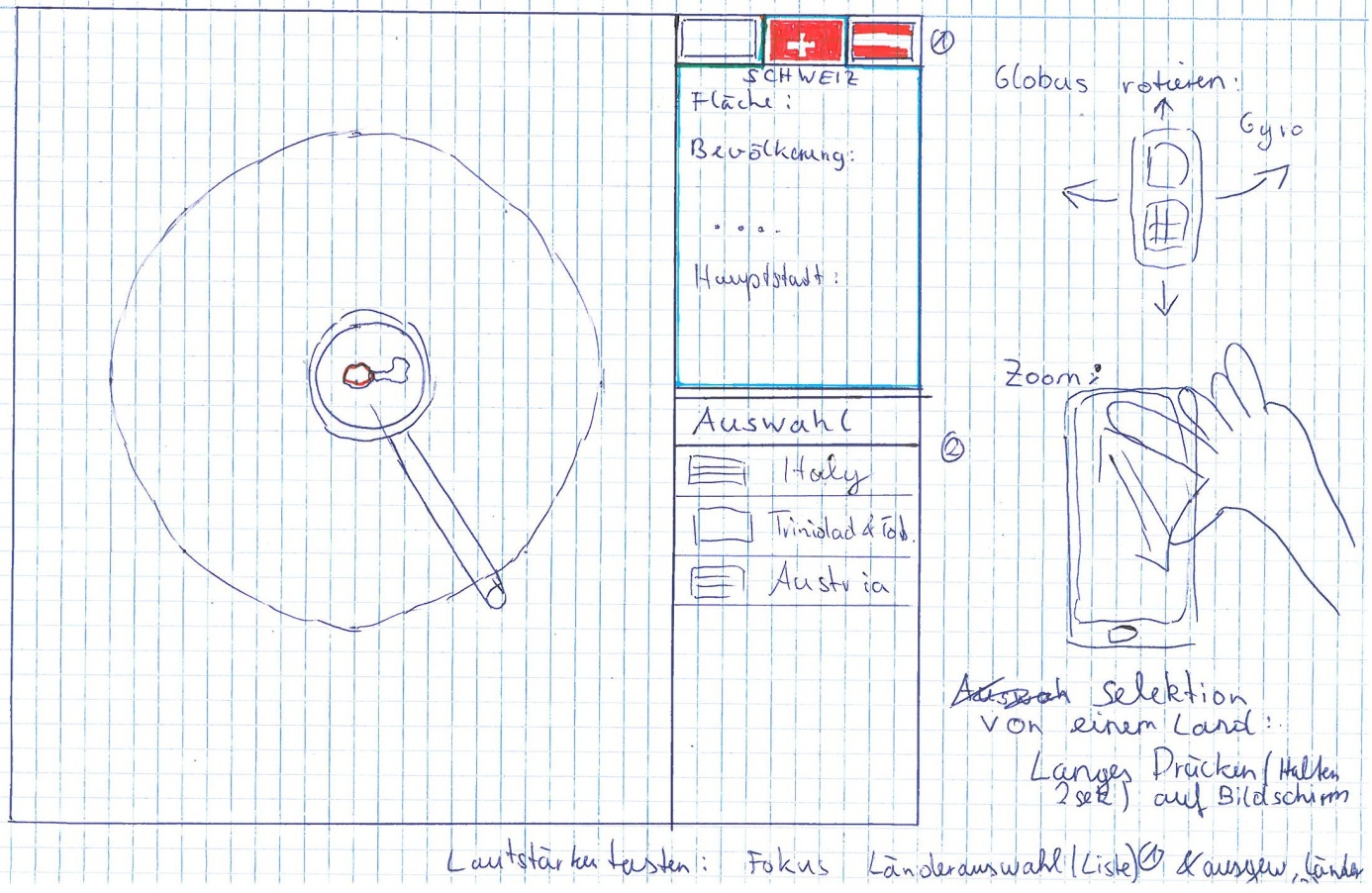


- viele Daten können irritieren

- je nach gewünschter Vergleichsart unpraktisch

## 3.2. Skizzen zum User Interface

## 3.2.1. Länderauswahl



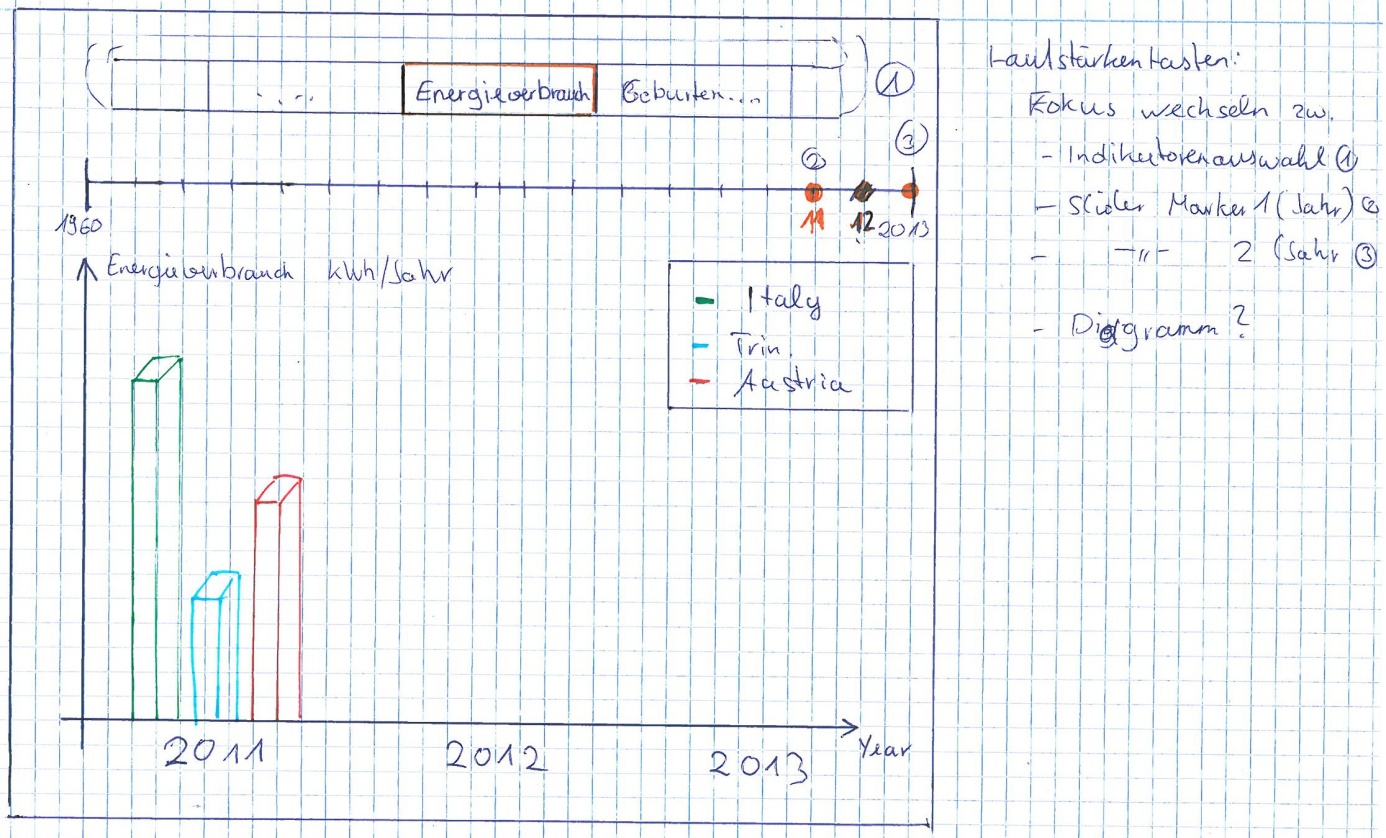
+ primäre Länderauswahl über Globus

+ Listenmodus rechts oben für kleine oder unbekannte Länder

+ grundlegende Info für aktuellen Land

+ Auswahlliste von Ländern für den Vergleich

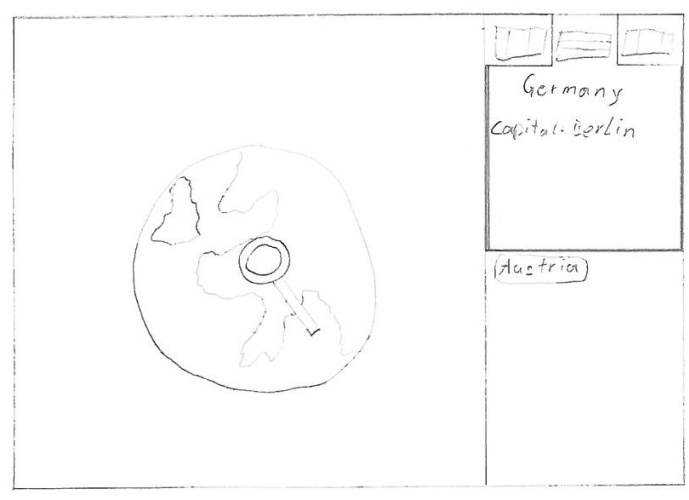
### 3.2.2. Vergleich von Ländern



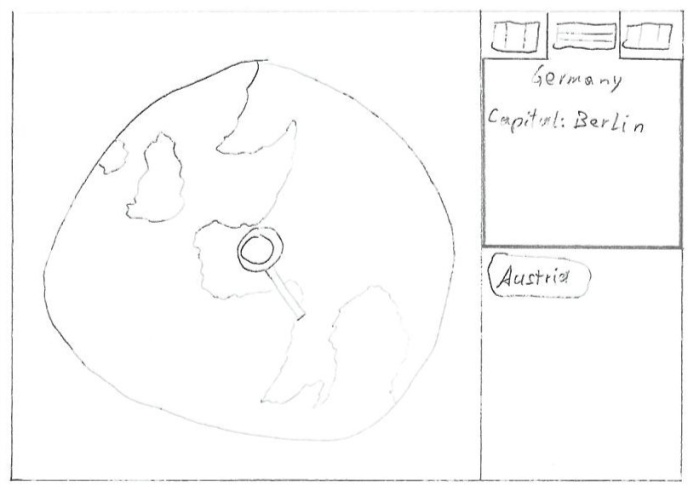
## 3.3. Interaktionen

### 3.3.1. Länderauswahl

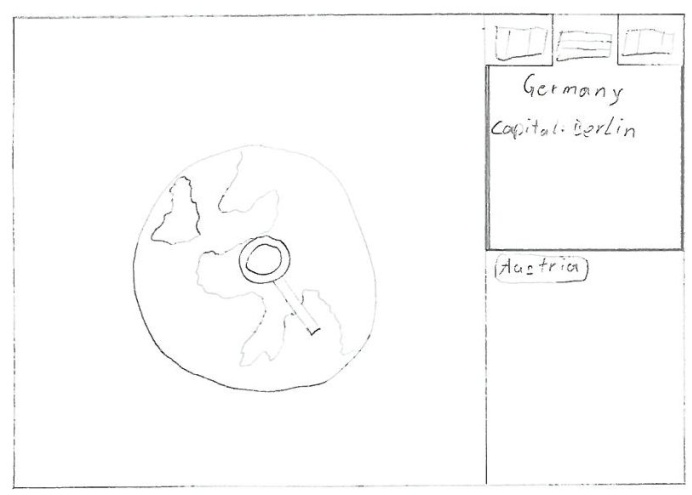
#### 3.3.1.1. Zoom



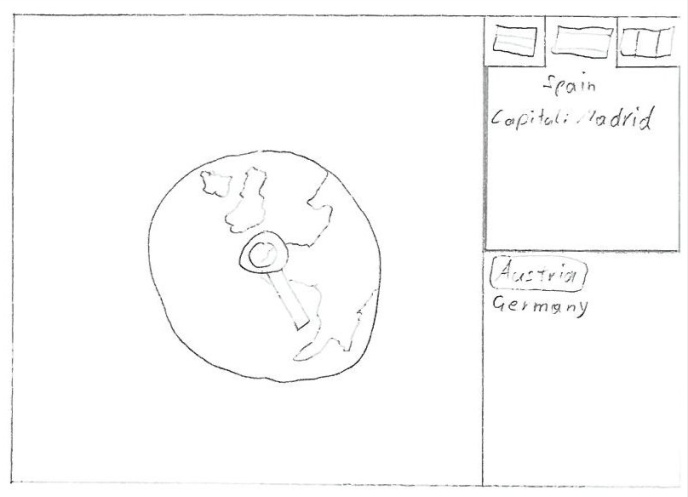
Bekannte Zoomgeste von Smartphones (zwei Finger spreizen oder zusammenziehen)



#### 3.3.1.2. Land auswählen



Smartphone rotieren um Globus zu drehen; Fokus oben gesetzt und horizontal wischen



Weitere gezeichnete „Screenshots“ für verschiedene Aktionen in Scan/pp005.jpg

# 4. Probleme während der Entwicklung und deren Lösung

## 4.1. Anfrage an den Webservice der Weltbank

- Weltbank scheinbar über Weihnachten umgestellt

- Countries OK; Indikatoren nicht mehr? Ohne \r (nur \nHost:) -> \r\n für jede Zeile nötig

- HTTP/1.1 Implementierung blockt weil Connection wohl doch offen bleibt

- HTTP/1.0 funktioniert nun weil Connection nicht offen gehalten wird -> kein Blocken mehr beim read

## 4.2. Textencoding in Unity

- bei XML

- ohne BOM (byte order mark) am Anfang sonst merkwürdige Exception

- System.Xml.XmlException: Text node cannot appear in this state. Line 1, position 1.

- mit MemoryStream read und danach write gelöst

- Fehler in C# Doku (<http://msdn.microsoft.com/en-us/library/system.io.bufferedstream%28v=vs.110%29.aspx>): falsche Offsets