

Praktikum Logikprogrammierung

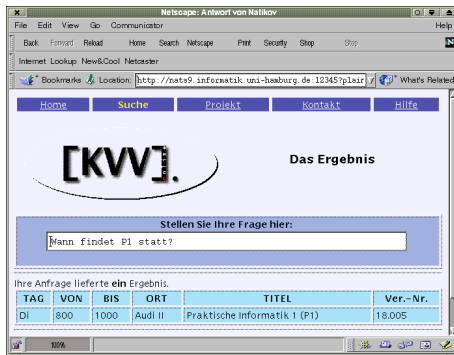
Veranstalter: Timo Baumann
Isabelle Streicher
Wolfgang Menzel

Zeitraum 20.2.2012 - 2.3.2012
Montag - Freitag, 9 - 17 Uhr

Ort Informatik-RZ

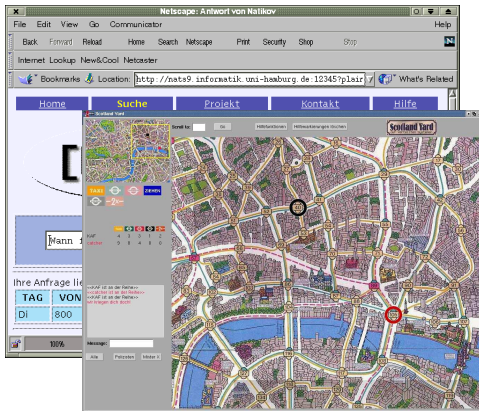
Rückblick

- Natürlichsprachlicher Zugriff zu Datenbanken (3x)



Rückblick

- Natürlichsprachlicher Zugriff zu Datenbanken (3x)
- Scotland Yard (2x)



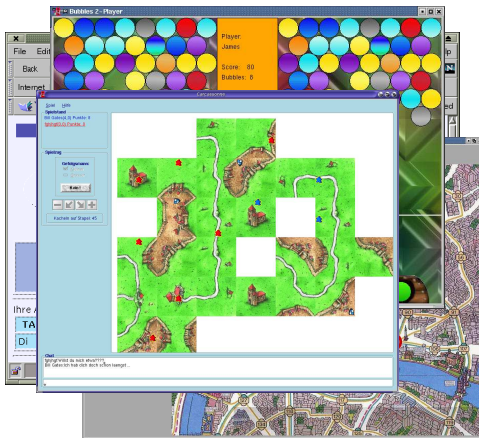
Rückblick

- Natürlichsprachlicher Zugriff zu Datenbanken (3x)
- Scotland Yard (2x)
- Frozen Bubbles



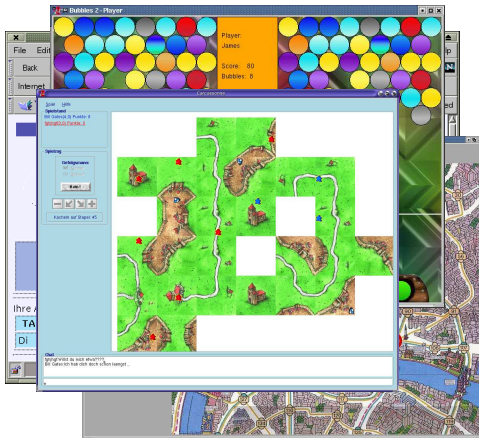
Rückblick

- Natürlichsprachlicher Zugriff zu Datenbanken (3x)
- Scotland Yard (2x)
- Frozen Bubbles
- Schach
- Carcassonne



Rückblick

- Natürlichsprachlicher Zugriff zu Datenbanken (3x)
- Scotland Yard (2x)
- Frozen Bubbles
- Schach
- Carcassonne
- Siedler von Catan



Rückblick

- Natürlichsprachlicher Zugriff zu Datenbanken (3x)
- Scotland Yard (2x)
- Frozen Bubbles
- Schach
- Carcassonne
- Siedler von Catan
- Thurn und Taxis



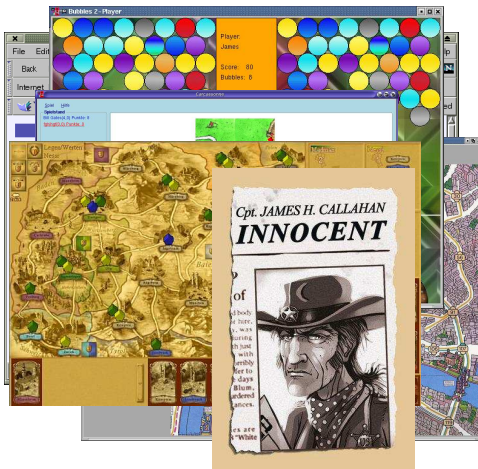
Rückblick

- Natürlichsprachlicher Zugriff zu Datenbanken (3x)
- Scotland Yard (2x)
- Frozen Bubbles
- Schach
- Carcassonne
- Siedler von Catan
- Thurn und Taxis
- Ghost Stories



Rückblick

- Natürlichsprachlicher Zugriff zu Datenbanken (3x)
- Scotland Yard (2x)
- Frozen Bubbles
- Schach
- Carcassonne
- Siedler von Catan
- Thurn und Taxis
- Ghost Stories
- Mr. Jack in New York

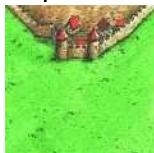


Praktikum Logikprogrammierung



Praktikum Logikprogrammierung

- Repräsentation der Kacheln



```
kacheltyp(e, 5, grundspiel,  
          kanten(stadt(_), gras(G), gras(G), gras(G)),  
          extras([])).
```



```
kacheltyp(g, 1, grundspiel,  
          kanten(stadt(S), gras(_), stadt(S), gras(_)),  
          extras([])).
```

Rückblick

- Kachel auswählen

```
kachelziehen([Kachelname|Pool],  
             kachel(Kachelname, Kanten, Extras),  
             Pool) :-  
    kacheltyp(Kachelname, _, _, Kanten, Extras).
```

- Kante auswählen

```
gib_kante(kachel(_, kanten(N, _, _, _), _), n, N).  
gib_kante(kachel(_, kanten(_, 0, _, _), _), o, 0).  
gib_kante(kachel(_, kanten(_, _, S, _), _), s, S).  
gib_kante(kachel(_, kanten(_, _, _, W), _), w, W).
```

Rückblick

- Kante anlegen

```
anlegen(Vergleich, NeueKachel, NachbarKachel, Richtung) :-  
    gib_kante(NeueKachel, Richtung, NeueKante),  
    gegenrichtung(Richtung, Gegenrichtung),  
    gib_kante(NachbarKachel, Gegenrichtung, NachbarKante),  
    spiegle_kante(NachbarKante, SpiegelNachbarKante),  
    call(Vergleich, NeueKante, SpiegelNachbarKante).
```

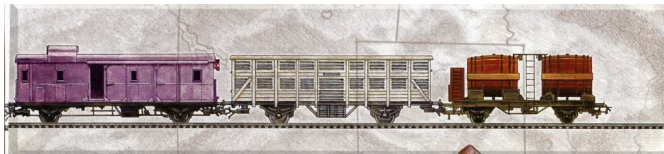
- Aufruf

```
...  
kachel_drehen(Kachel, Drehung, DrehKachel),  
anlegen(Vergleich, DrehKachel, Nachbarn),  
...
```

- Resultat: alle benachbarten Regionen sind durch Koreferenz miteinander verbunden!

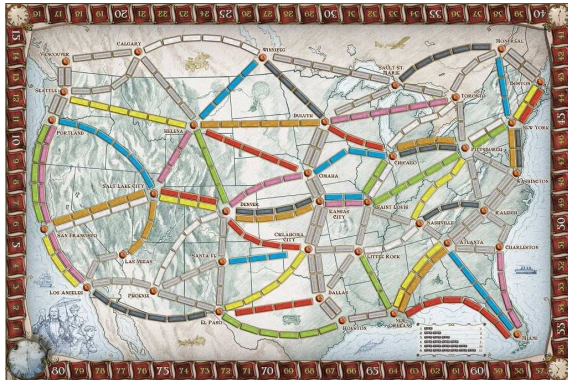
Vorschau

- Zug um Zug
(Ticket to Ride)



Vorschau

- austauschbare Spielfelder



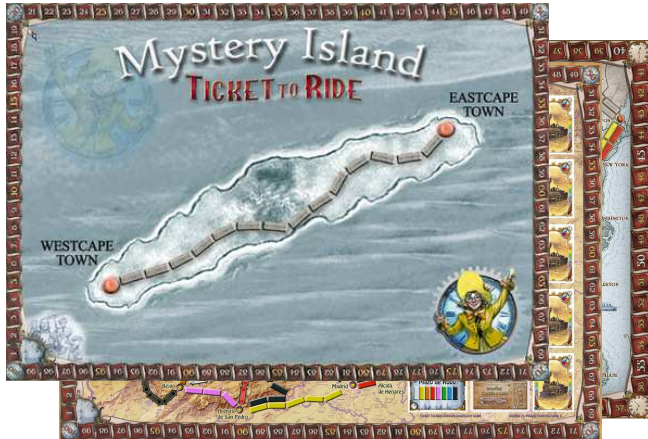
Vorschau

- austauschbare Spielfelder



Vorschau

- austauschbare Spielfelder



Interesse?

Interesse?
Andere Ideen?