Super-Mario

Knowledge Engineering und Lernen in Spielen

TECHNISCHE UNIVERSITÄT DARMSTADT

Übersicht

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Super-Mario (allgemein)
2. Modellierung der Spielerfahrung[1]
   - Lernen, wann ein Level Spaß macht
3. Reinforcement Learning Benchmark[2]
   - Training eines Controllers, der Super-Mario spielt
4. Mario AI Competition 2009[3]
   - Andere Ansätze zur Implementierung eines Controllers

[1] C. Pedersen, J. Togelius and G. N. Yannakakis: Modeling Player Experience in Super Mario Bros

[2] J. Togelius, S. Karakovskiy, J. Koutnik and J. Schmidhuber: Super Mario Evolution

[3] `http://julian.togelius.com/mariocompetition2009/index.php`

Super Mario (allgemein)

Screenshot

TECHNISCHE UNIVERSITÄT DARMSTADT

Super Mario (allgemein)

Beschreibung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Klassisches „Jump & Run“ von Nintendo
  - Spielablauf in Echtzeit
  - Levelende unter Zeitdruck erreichen
  - Abgründe überspringen
  - Gegner meiden
  - Münzen einsammeln
  - Gegenstände verleihen neue Fähigkeiten

Super Mario (allgemein)

Beschreibung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Klassisches „Jump & Run“ von Nintendo
  - Spielablauf in Echtzeit
  - Levelende unter Zeitdruck erreichen
  - Abgründe überspringen
  - Gegner meiden
  - Münzen einsammeln
  - Gegenstände verleihen neue Fähigkeiten
- Hier: „Infinite Mario Bros“
  - Open-Source (Java)
  - Automatisch generierte Levels
  - Im Webbrowser spielbar

Übersicht

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Super-Mario (allgemein)
2. Modellierung der Spielerfahrung
   - Lernen, wann ein Level Spaß macht
3. Reinforcement Learning Benchmark
   - Training eines Controllers, der Super-Mario spielt
4. Mario AI Competition 2009
   - Andere Ansätze zur Implementierung eines Controllers

Modellierung der Spielerfahrung

Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Spieleentwicklung zeitaufwändig
  - Automatische Levelgenerierung
  - Aber: Qualität der Lösung?

Modellierung der Spielerfahrung

Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Spieleentwicklung zeitaufwändig
  - Automatische Levelgenerierung
  - Aber: Qualität der Lösung?
- Daher Bewertungsfunktion nötig
  - Was macht einen Level spaßig?
  - Wie erzeugt man eine Herausforderung...
  - ... ohne den Spieler zu frustrieren?
  - Ansatz: Machinelles Lernen

Modellierung der Spielerfahrung

Datensammlung (Leveldesign)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Modifiziertes „Infinite Mario Bros“
  - Zufälligkeit der Levels begrenzt
    - Feste Anzahl von Gegenständen / Gegnern
  - Systematisch Variation
    - Anzahl Abgründe
    - Durchschnittliche Breite
    - Verteilung im Level (Vorhersagbarkeit)
    - Anzahl „Richtungsänderungen“
  - Merkmale beeinflussen Schwierigkeitsgrad
  - Auch gültig für ähnliche Spiele

Modellierung der Spielerfahrung

Datensammlung (Spielerfahrung)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Teilnehmer spielen über Internet
  - Spiel generiert zwei Levels
  - Levels werden in beiden Reihenfolgen gespielt
  - Fragebogen nach jedem Paar
    - „Level 1 / 2 war mehr *E*“
    - „Beide Levels waren *E*“
    - „Keiner der Levels war *E*“
    - Wobei $E \in$ *fun*, *challenging*, *boring*, *frustrating*, *predictable*, *anxious*
  - Alle Levelkombinationen mindestens einmal gespielt

Modellierung der Spielerfahrung

Datensammlung (Spielerverhalten)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Aufzeichnung des Spielerverhaltens
  - Zeit: Bis Levelende, Dauer des letzten Lebens
  - Gesammelte Gegenstände: Anzahl, Typ
  - Tode: Durch Abgründe, verschiedene Gegnertypen
  - Besiegte Gegner: Springen, Schussfähigkeit, Wurf von Schildkrötenpanzern
  - Sprünge: Anzahl, Schwierigkeit (Heuristik)
  - Sonstiges: Wurde Level beendet?
  - Und weitere...
- Ermöglicht Rückschlüsse auf Fähigkeiten / Spielstil

Modellierung der Spielerfahrung

Aufgabenstellung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Approximiere E = f(Leveldesign, Spielerverhalten)
  - Nur für $E \in$ *fun*, *challenging*, *frustrating*
- Mit möglichst wenigen Features
  - Einfachere Analyse
  - Leichter anwendbar für Levelgenerierung
- Starke Ungenauigkeiten („Noise“) angenommen
  - Subjektivität der Spielerfahrung
  - Nur wenige Durchläufe gespielt
  - Robuste Verfahren zur Verhinderung von „Overfitting“

Modellierung der Spielerfahrung

Statistische Analyse (Vorgehen)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Lineare Zusammenhänge
- Anteil bei dem Reihenfolge relevant
  - Bei Spaß und Frustration egal
  - Jedoch „Noise“ bei Herausforderung
- Anteil mit klarer Präferenz für E
  - 79 % Herausforderung bis 63 % Frustration
- Anteil bei dem Feature bestimmte Präferenz bedingt
  - Signifikantes Subset gewählt

Modellierung der Spielerfahrung

Statistische Analyse (Ergebnis)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Spaß
  - Wenige Features, nur von Spielerverhalten abhängig
  - Konstanter und ungehinderter Fortschritt
  - Komplexe Aktion (Schildkrötenpanzer) wichtigstes Merkmal
- Herausforderung
  - Viele Features, unabhängig von Spaß
  - Ende nicht erreicht, oft gestorben
  - Wenige Gegenstände gesammelt, schwierige Sprünge
- Frustration
  - Stärkste Korrelationen, ähnlich Herausforderung
  - Zusätzlich langer Stillstand

Modellierung der Spielerfahrung

Statistische Analyse (Ergebnis)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Spaß
  - Wenige Features, nur von Spielerverhalten abhängig
  - Konstanter und ungehinderter Fortschritt
  - Komplexe Aktion (Schildkrötenpanzer) wichtigstes Merkmal
- Herausforderung
  - Viele Features, unabhängig von Spaß
  - Ende nicht erreicht, oft gestorben
  - Wenige Gegenstände gesammelt, schwierige Sprünge
- Frustration
  - Stärkste Korrelationen, ähnlich Herausforderung
  - Zusätzlich langer Stillstand
- Herausforderung => Spaß und Frustration, Frustration => kein Spaß
  - Hinweis auf Nichtlinearität

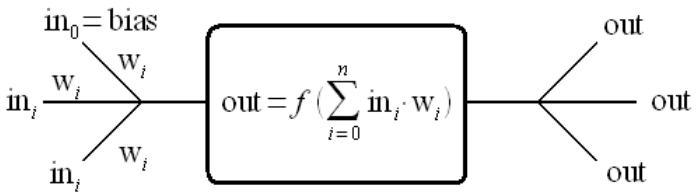
Modellierung der Spielerfahrung

Machinelles Lernen (Vorgehen I)

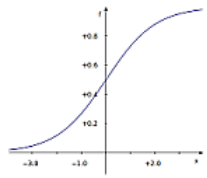
TECHNISCHE UNIVERSITÄT DARMSTADT

- Neuronales Netz
  - Modelliert Gehirnzellen
  - Nichtlineare Zusammenhänge
  - Hier mit nur einem Neuron
    - Einfacher analysierbar
    - Ungenauer als mehrere Schichten

$$in_i \in \{0, 1\} \qquad w_i \in [-5, 5]$$



Sigmoid: $f(x) = \frac{1}{1+e^{-x}}$



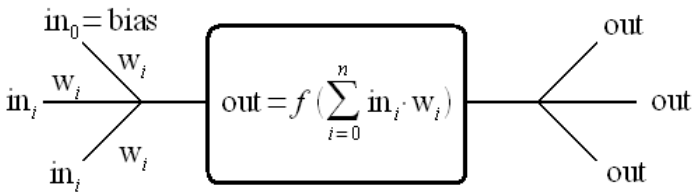
Modellierung der Spielerfahrung

Machinelles Lernen (Vorgehen I)

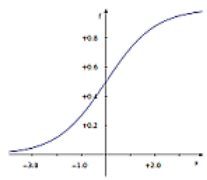
TECHNISCHE UNIVERSITÄT DARMSTADT

- Neuronales Netz
  - Modelliert Gehirnzellen
  - Nichtlineare Zusammenhänge
  - Hier mit nur einem Neuron
    - Einfacher analysierbar
    - Ungenauer als mehrere Schichten
- Genetischer Algorithmus
  - Entspricht natürlicher Evolution / zufälligen Mutationen
  - Abweichung zu E als Fitnessfunktion
  - Nötig da keine differenzierbare Fehlerfunktion

$in_i \in \{0, 1\}$ $w_i \in [-5, 5]$



Sigmoid: $f(x) = \frac{1}{1+e^{-x}}$



Modellierung der Spielerfahrung

Machinelles Lernen (Vorgehen II)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Auswahl eines Subset von Features
  - nBest
    - Features geordnet
    - n beste ausgewählt
  - SFS
    - Hillclimbing
    - Feature hinzufügen das maximalen Wert erzeugt
  - PFS
    - Sukzessives Eliminieren von Features mit geringem Gewicht
  - Nicht notwendigerweise optimal
  - Bester Wert aus 3-Fold-Crossvalidation

Modellierung der Spielerfahrung

Machinelles Lernen (Ergebnis)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Spaß
  - SFS 69 % Genauigkeit bei 3 Features
  - nBest ähnlich mit 10 Features, PFS deutlich schlechter
  - Zeit der Bewegung nach links, auf Gegner gesprungen, Level nicht gespiegelt
- Herausforderung
  - SFS 78 % und 5 Features
  - Großer Abstand zu PFS, nBest noch schlechter
  - Stillstand, Sprungschwierigkeit, wenige Gegenstände und erledigte Gegner
- Frustration
  - SFS 89 % und 4 Features
  - nBest ähnlich, PFS deutlich schlechter
  - Oft in Lücken gefallen und wenig Zeit im letzten Leben
  - Unterschiede zu Herausforderung (wenig Stillstand und leichte Sprünge)

Modellierung der Spielerfahrung

Fazit

TECHNISCHE UNIVERSITÄT DARMSTADT

- Zufriedenstellende Ergebnisse
- Könnten verbessert werden
  - Andere Leveldesign-Features
  - Mehr als ein Neuron verwenden
  - Weitere Trainingsdaten
- Noch kein Praxiseinsatz zur Levelgenerierung sinnvoll

Übersicht

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Super-Mario (allgemein)
2. Modellierung der Spielerfahrung
   - Lernen, wann ein Level Spaß macht
3. Reinforcement Learning Benchmark
   - Training eines Controllers, der Super-Mario spielt
4. Mario AI Competition 2009
   - Andere Ansätze zur Implementierung eines Controllers

Reinforcement Learning Benchmark

Motivation I

TECHNISCHE UNIVERSITÄT DARMSTADT

- Benchmarks erlauben Vergleich verschiedener Algorithmen
  - Liefern reproduzierbare Ergebnisse
  - Stellen ein relevantes Problem dar
  - Existieren schon für Spiele aus anderen Genres
    - „Relevant, da es die menschliche Intelligenz spielen kann“
  - Möglichkeit von Wettbewerben
- Hier für Reinforcement Learning
  - Viele Durchläufe
  - „Verstärke“ Züge, die zum Ziel geführt haben

Reinforcement Learning Benchmark

Motivation II

TECHNISCHE UNIVERSITÄT DARMSTADT

- Super-Mario als Benchmark geeignet
  - Hohe Dimension der Eingabedaten
    - Teilweise kontinuierlicher Zustandsraum
    - Welt enthält statische und dynamische Objekte
    - Zustand partiell beobachtbar
  - 32 mögliche Aktionen
  - Anpassbarer Schwierigkeitsgrad
    - Einfache Levels mit wenig Aufwand schaffbar
    - Später auch komplexe Pläne nötig
    - Bildet eine Lernkurve

Reinforcement Learning Benchmark

Implementierung

TECHNISCHE UNIVERSITÄT DARMSTADT

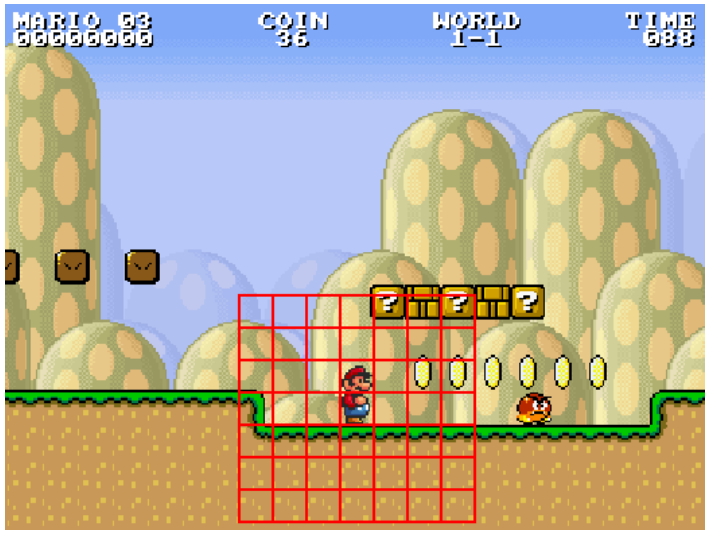
- Ebenfalls modifiziertes „Infinite Mario Bros“
  - Wiederholbare Levelgenerierung
  - Echtzeit-Komponente entfernt
    - Ablauf in Schritten
    - Erlaubt Spielen von 20 Levels / Sekunde
  - Java-Interface für Controller
  - TCP-Interface für andere Sprachen
    - Deutlich langsamer

Reinforcement Learning Benchmark

Anforderungen an den Controller

TECHNISCHE UNIVERSITÄT DARMSTADT

- Controller erhält Beobachtungen
  - Hindernissen und Gegnern in Umgebung
    - Verschiedene Umgebungsgrößen möglich
  - Bias, am Boden, Springen möglich
  - Durch boolsche Werte repräsentiert
  - Erzeugt 21, 53 oder 101 Inputs
- Erzeugt daraus eine Aktion (Tastendruck)
- Kann einen internen Zustand haben

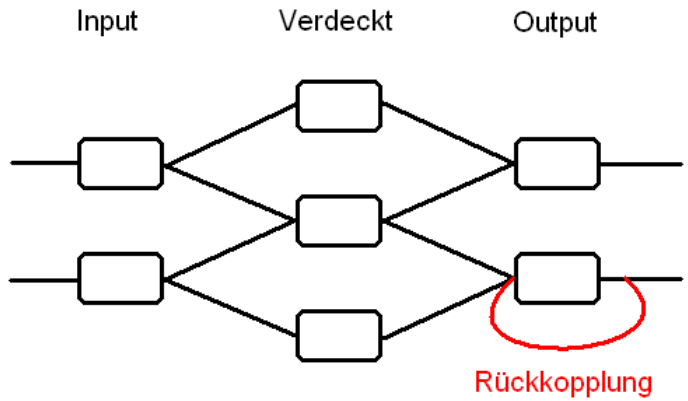


Reinforcement Learning Benchmark

Beispielcontroller

TECHNISCHE UNIVERSITÄT DARMSTADT

- Verwendet Neuronales Netz
  1. Multilayer-Perceptron
     - Eine verdeckte Schicht
  2. Simple Recurrent Network
     - Zusätzlich Rückkopplungen
- Beide mit 10 verdeckten Knoten

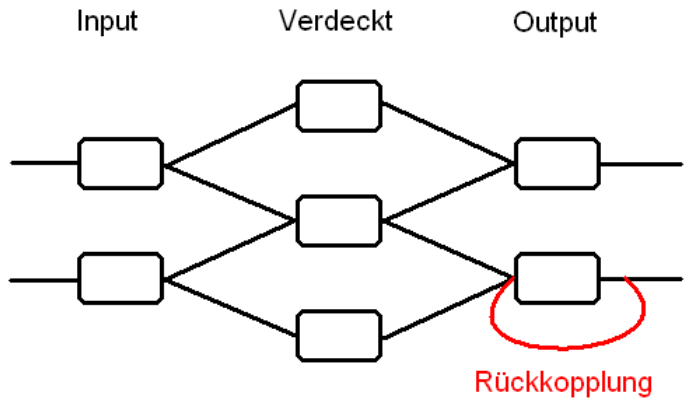


Reinforcement Learning Benchmark

Beispielcontroller

TECHNISCHE UNIVERSITÄT DARMSTADT

- Verwendet Neuronales Netz
  1. Multilayer-Perceptron
     - Eine verdeckte Schicht
  2. Simple Recurrent Network
     - Zusätzlich Rückkopplungen
  - Beide mit 10 verdeckten Knoten
- Gewichte über genetischen Algorithmus
  - 100 Generationen
  - Level beendet -> Schwierigkeit erhöhen
  - Levelfortschritt als Fitnessfunktion

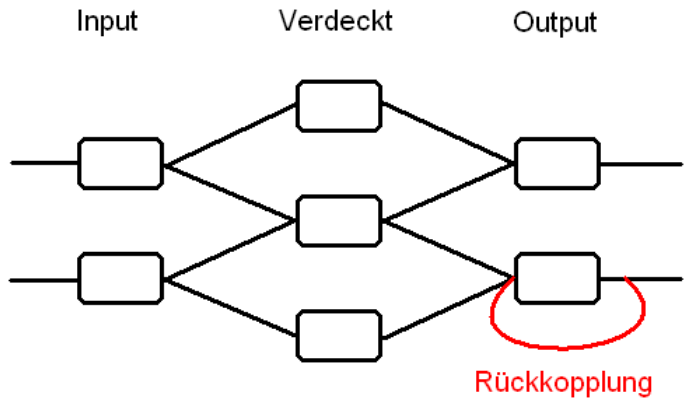


Reinforcement Learning Benchmark

Beispielcontroller

TECHNISCHE UNIVERSITÄT DARMSTADT

- Verwendet Neuronales Netz
  1. Multilayer-Perceptron
     - Eine verdeckte Schicht
  2. Simple Recurrent Network
     - Zusätzlich Rückkopplungen
  - Beide mit 10 verdeckten Knoten
- Gewichte über genetischen Algorithmus
  - 100 Generationen
  - Level beendet -> Schwierigkeit erhöhen
  - Levelfortschritt als Fitnessfunktion
- Verwendung von HyperGP
  - Kombiniert diese Techniken
  - Nutzt Regelmäßigkeiten im Zustandsraum



Reinforcement Learning Benchmark

Ergebnisse I

TECHNISCHE UNIVERSITÄT DARMSTADT

Max. Level			
Methode	**3x3**	**5x5**	**7x7**
MLP	3,0	0,8	0,5
SRN	2,8	1,8	0,3
SRN + HyperGP	1,9	2,1	1,3

- Overfitting bei MLP / SRN und großem Input
- HyperGP erlaubt größere Netze durch Kompression
- Kein Overfitting bei großem Input
- Probleme bei Generalisierung trotz gleichem Schwierigkeitsgrad

Reinforcement Learning Benchmark

Ergebnisse II

TECHNISCHE UNIVERSITÄT DARMSTADT

- Controller springen wenn fester Boden zu Ende...
- ...ohne zu wissen wo sie aufkommen werden
- Sie laufen oft in Gegner
  - Abgründe beenden Spiel sofort
  - Aber bei Gegnern erst nach dem dritten Kontakt
  - Daher als weniger gefährlich eingestuft
- Es werden keine Gegenstände gesehen, dadurch geringe Punktzahlen

Reinforcement Learning Benchmark

Fazit

TECHNISCHE UNIVERSITÄT DARMSTADT

- Generalisierung ist noch ein Problem
- Benchmark ist neue Herausforderung für Reinforcement Learning
- Ansatz: Level auch nach erfolglosem Durchlauf ändern
  - Könnte Generalisierung erleichtern
  - Aber zusätzliches „Noise“
- Erweiterung des Inputs möglich
  - Bisher einmalige Größenordnung
  - Z. B. Art der Gegner

Übersicht

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Super-Mario (allgemein)
2. Modellierung der Spielerfahrung
   - Lernen, wann ein Level Spaß macht
3. Reinforcement Learning Benchmark
   - Training eines Controllers, der Super-Mario spielt
4. Mario AI Competition 2009
   - Andere Ansätze zur Implementierung eines Controllers

Mario AI Competition 2009

Beschreibung

TECHNISCHE UNIVERSITÄT DARMSTADT

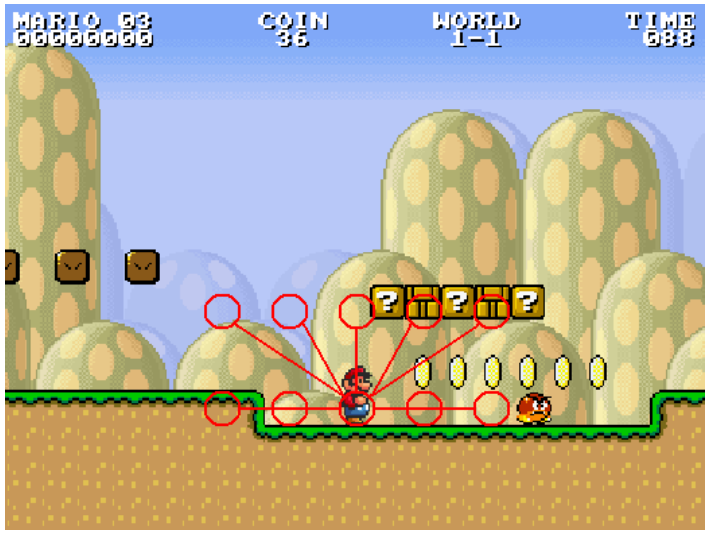
- Basiert auf dem beschriebenen Benchmark
- Controller spielt bis zu 40 festgelegte Levels
- Maximal 40 ms pro Schritt
- Erreichter Level und Punkte gewertet

Mario AI Competition 2009

Sieger

TECHNISCHE UNIVERSITÄT DARMSTADT

- Sieger verwendet A*-Algorithmus zur Bewertung von Knoten
  - Knoten sind mögliche Aktionen
  - Heuristik ist Zeit bis zum Ziel ohne Hindernisse
  - Plant zwei Schritte vorraus
- Hat eigenes Weltbild
  - Sagt Gegnerbewegungen vorher
  - Regelmäßig aktualisiert



Mario AI Competition 2009

Video des Siegers

TECHNISCHE UNIVERSITÄT DARMSTADT

http://www.youtube.com/watch?v=DlkMs4ZHHr8

Mario AI Competition 2009

Andere Ansätze

TECHNISCHE UNIVERSITÄT DARMSTADT

- Neuronales Netz und genetischer Algorithmus
- Durch menschlichen Spieler trainiertes Neuronales Netz
- Regelbasierte Systeme
- State Machines
- Mit genetischem Algorithmus gebildete Hashtabellen (100 MB)

Mario AI Competition 2009

Ergebnisse und Fazit

TECHNISCHE UNIVERSITÄT DARMSTADT

- Benchmark zeigt deutliche Unterschiede
- Implementierungen mit A* sind überlegen
  - Benötigt jedoch die längste Berechnungszeit
  - Funktioniert gut, weil Levels kein Backtracking erfordern
- Wenige Teilnehmer verwendeten (Reinforcement) Learning

Jeweils beste Implementierungen		
Algorithmus	**Level**	**Punkte**
A*	40	47000
A*	40	47000
RB	11	21000
RB	11	18000
EV	8	12000
EV	7	13000
SM	4	12000
SM	3	7000
HT	Absturz	

Ende

TECHNISCHE UNIVERSITÄT DARMSTADT

Danke für die Aufmerksamkeit!