

---

# Fuzzy Movement



**Dirk Gldner**

**Seminar: Knowledge Engineering und Lernen in Spielen**

**TU-Darmstadt, SS 2006, 30.05.2006**

# Übersicht

---

- Begriffsdefinition
- allgemeine Ziele & Ziele in Spieleumgebungen
- Situations- und Informationsbeispiele
- Art der Informationsspeicherung
- Fuzzy-Mengen und deren Operationen
- Art der Informationsinterpretation
- Unterschiede zu Rule-Based-Systems
- Tipps zum Einsatz
- Quellen
- Demo

# Zum Begriff „fuzzy“

---

- ungenau, verschwommen, unscharf
- Beschreibungen können „fuzzy“ sein, wenn...
  - ...sie unklar, zweideutig, mehrdeutig sind
  - ...sie von der Wahrnehmung abhängen
- Beispiel: Waschmaschine
  - „Wäsche ist ziemlich stark verdreckt“  
→ 65g Waschmittel sind nötig.

# Allgemeine Ziele

---

- Entscheidungen durch „Expertenwissen“
  - Abbildung von Daumenregeln in Expertensystemen
- Vereinfachung des Systems, da näher an sprachlichen Definitionen
  - Bleibt die Konversation zwischen dem „Fuzzy-System“ und den „festen Variablen“

# Ziele in Spieleumgebungen

---

- Einhalten zwei wichtiger Kriterien
  - Effektivität
    - Effizientes Verhalten
  - Konsistenz
    - keine unlogischen Abläufe
- Verhalten: optimales != menschliches
- ABER: in „Alltagssituationen“ äußerst oft

# Situationsbeispiele

---

- Aus dem Alltag:
  - Öffnen einer Tür und durch die Tür gehen
  - Eine Leiter hochklettern
- Aus dem Alltag eines Helden:
  - Auf Plattformen springen
  - Jump pads benutzen
  - Raketen ausweichen

# Gewonnene Information

---

- Annahmen
  - z.B.: „Items werden automatisch aufgehoben.“
- Einschränkungen
  - z.B.: „Die Tür muss erst geöffnet werden, bevor man weitergehen kann.“
- Richtlinien
  - z.B.: „Es sollten genug Items vorhanden sein.“

# Art der Informationsspeicherung (1)

---

- Was wird im Speicher gehalten?
  - Linguistische Variablen  
*„Loyalität zu x“, „Wohlbefinden“, „Blickrichtung“, ...*  
meistens auf Mengen definiert
- Was wird außerhalb definiert?
  - „Fuzzy Terms“, evtl. mit „Fuzzy Values“  
*„feindlich“, „super“, „zur Leiter hin“*
  - „Fuzzy Modifiers“  
*„extrem“, „mehr oder weniger“, ...*
  - Regeln  
*„Shot and bleeding badly implies dying“,  
„IF NOT unconfident THEN attack“*



# Art der Informationsspeicherung (2)

- Beispiel Egoshooter:

- Menge „Grad der Feindlichkeit“ muss erstellt werden
- Werte müssen für alle Objekte der Grundgesamtheit definiert werden:

feindlich = {(Tür,0), (Waffe,0), (gegn. Einheit, 0.3), (Monster, 0.9)}

- typisches Beispiel für „Modifiers“:

- gefährlich: {0 0.4 0.8 1}
- *sehr* gefährlich: {0 0.16 0.64 1}

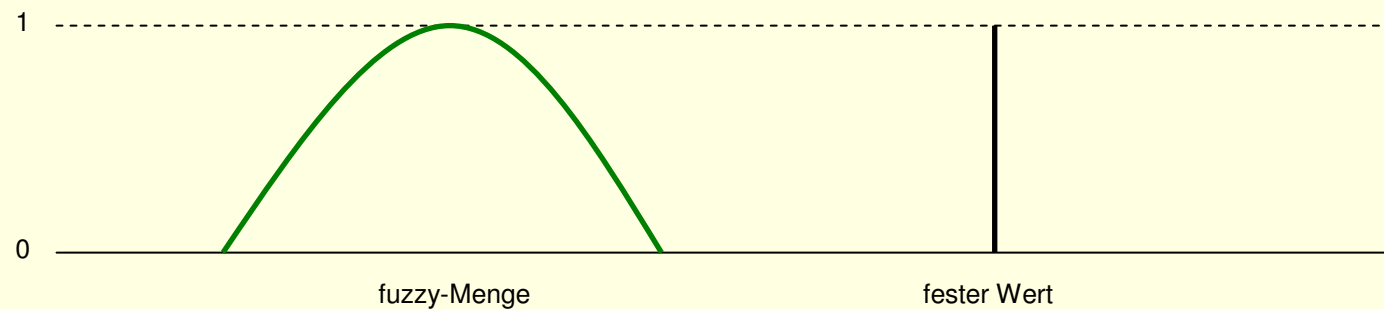
# Art der Informationsspeicherung (3)

---

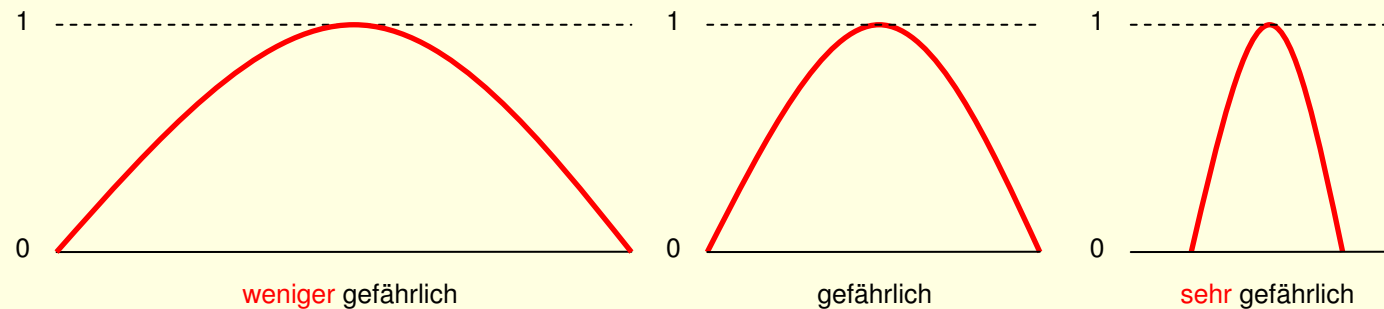
- Zusammenfassung Voraussetzungen:
  - Definition von linguistischen Variablen
  - Definition von Fuzzy Terms
  - Definition von Fuzzy Modifiers
  - Ableitung von Regeln und Schlussfolgerungen
  - ständiges Bewusstsein über den aktuellen physischen Status
  - Definition von Funktionen, die boolesche Variablen in fuzzy-Variablen verwandeln und umgekehrt

# Fuzzy-Mengen (1)

## ■ Vergleich zu festen Werten

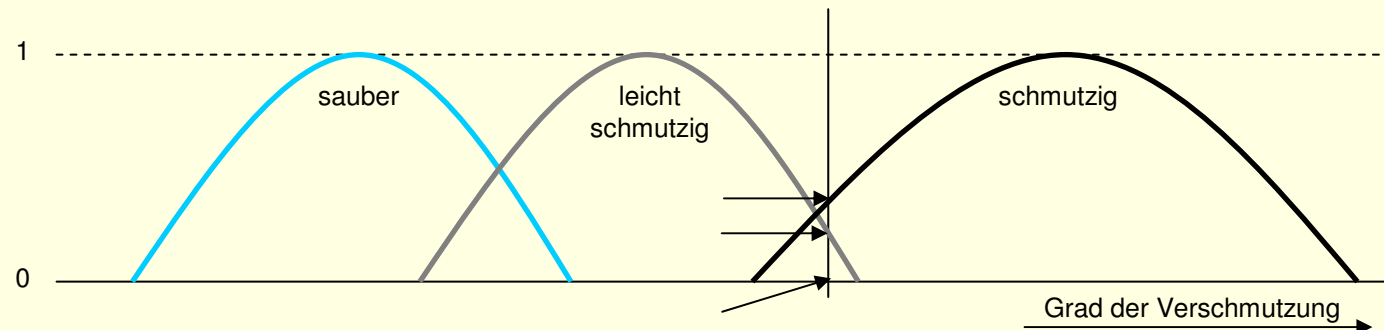


## ■ Betrachtung von „Modifiers“

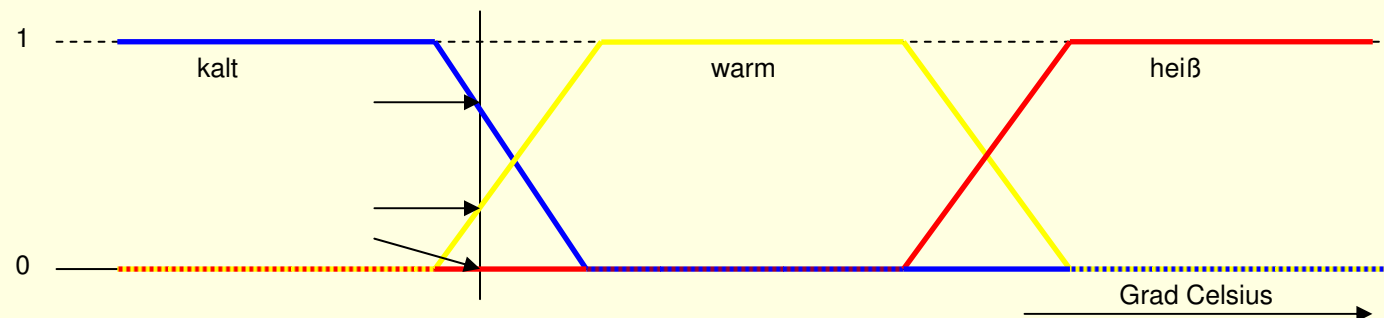


# Fuzzy-Mengen (2)

## ■ Beispiel Waschmaschine



## ■ Beispiel Temperatur



# Operationen auf Mengen

---

- AND

$$C = A \cup B \Leftrightarrow \forall x : C(x) = \min(A(x), B(x))$$

- OR

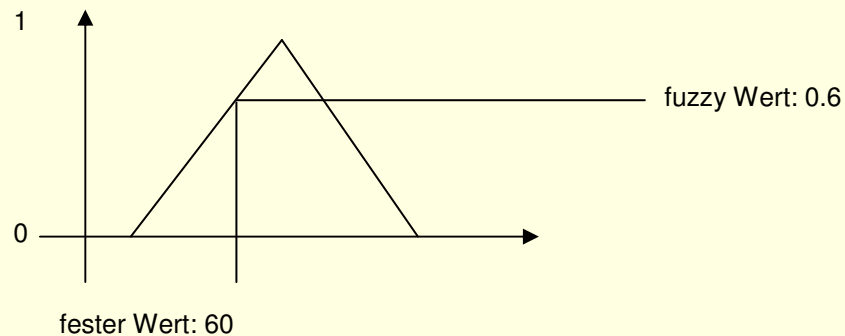
$$C = A \cap B \Leftrightarrow \forall x : C(x) = \max(A(x), B(x))$$

- NOT

$$\overline{A}(x) = 1 - A(x)$$

# Art der Informationsinterpretation

## ■ Fuzzification: Wechsel zu fuzzy-Variablen



## ■ Defuzzification: Wechsel zurück

- Muss nicht eindeutig sein (entspricht Mensch)
- kann Maximum, Mittelwert etc. sein

# Unterschiede zu Rule-Based-Systems

---

- „Matching“ muss über alle Regeln gehen
- Es müssen i.d.R. zwei Arrays im working memory vorhanden sein
  - IF feeling is **good** AND music is loud  
THEN feeling is **very good**
- Die Regel-Bodies müssen per Komposition kombiniert werden

# Was beachtet werden sollte...

---

- Implementierung i.d.R. nicht Gesamtlösung
  - sollte als Verfeinerungsstufe dienen
- Innerhalb des Entscheidungswegs keinen Wechsel zur booleschen Logik durchführen
  - Input / Output / Operationen
- zu viele fuzzy Variablen vermeiden
  - i.d.R.: ein Loop → 1x alle Variablen berechnen
  - Geschwindigkeitseinbußen
  - evtl. Lookup-Tables nötig
- zu viele Regeln vermeiden



# Quellen

---

- Alex J. Champanand: *AI Game Development*, Kapitel 29-31, New Riders Publishing, 2003
- David M. Bourg, Glenn Seemann: *AI for Game Developers*, Kapitel 10 & 13, O'Reilly, 2004
- Wikipedia-Artikel „fuzzy logic“, [http://en.wikipedia.org/wiki/Fuzzy\\_logic](http://en.wikipedia.org/wiki/Fuzzy_logic)
- Earl Cox: *The Seven Truths Of Fuzzy Logic*, <http://www.bytecraft.com/fuzzylogictruths.html>
- The Game AI Page: <http://www.gameai.com>  
→ Rubrik *Software Solutions*
- Edward Sazanov: *Prevention of Load Sway by a Fuzzy Controller*, [http://people.clarkson.edu/~esazonov/neural\\_fuzzy/loadsway/LoadSway.htm](http://people.clarkson.edu/~esazonov/neural_fuzzy/loadsway/LoadSway.htm)

# Demonstration

---

## ■ Quake 2:

- Animat mit der Fähigkeit, Plattformen zu benutzen und durch Türen zu gehen
- Knowledge Base interessant, Ausführung eher nicht



## ■ Verladekran:

- soll Container vom Dock auf Schiff transportieren
- Regeln hierfür werden durch die Anwendung von Fuzzy-Sets aufgestellt
- „Fuzzy Engine“ übernimmt Interpretation im Hintergrund





Vielen Dank  
für Ihre Aufmerksamkeit!