

Einführung in die Spieleprogrammierung

Künstliche Intelligenz (KI)



1996: Ein für heutige Verhältnisse extrem pixeliger Ernter sucht im Strategie-Primus Command & Conquer nach dem Rohstoff Tiberium, rollt dabei jedoch einfältig mitten in das Hauptquartier des Feindes und zerplatzt in einer hässlichen Mini-Explosion.

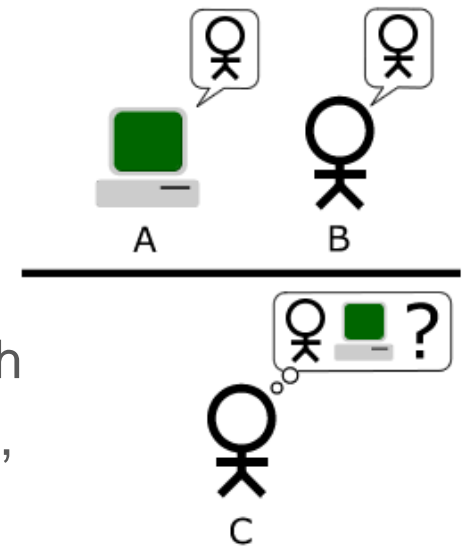
GameStar 2/2008

2007: Ein Ernter auf höchstem grafischen Niveau sucht im Strategie-Hit Command & Conquer 3 nach dem Rohstoff Tiberium, rollt dabei jedoch einfältig mitten in das Hauptquartier des Feindes und wird von gleißenden Laserstrahlen eindrucksvoll in seine Einzelteile zerrissen, die physikalisch korrekt in alle Himmelsrichtungen fliegen.

GameStar 2/2008

Der Turing-Test

- 1950 von Alan Turing auf die Frage „Can Computers Think?“ vorgeschlagen
- Person C („Befrager“) kommuniziert indirekt (Nachrichten, Chat etc.) mit A und B, von denen einer eine echte Person und die andere ein Computerprogramm ist
- A und B geben sich jeweils als denkender Mensch
- Das Computerprogramm besteht den Turing-Test, wenn C nach „intensiver“ Befragung nicht in der Lage ist, zu bestimmen, wer der Mensch ist
- Es existieren zahlreiche Variationen des ursprünglichen Tests



Der Turing-Test

- Seit 1991 ist der Loebner-Preis ausgeschrieben, der das erste Programm, das den Turing-Test besteht honoriert:
 - Goldmedaille: Musik, Sprache, Bilder und Videos (Preisgeld 100.000 \$)
 - Silbermedaille: Schriftlich (25.000 \$)
 - Bronzemedaille: jährlich, bestes Programm unter den eingereichten (2.000 \$)
- Saygin et al.: "Turing Test: 50 Years Later" (2001)

Spieltheorie und rationale Agenten

- Spieltheorie untersucht Systeme mit mehreren Akteuren (menschliche und künstliche Spieler)
- Ziel: Wie sieht rationales Entscheidungsverhalten in sozialen Konfliktsituationen aus?
- Rationale Agenten: Computerprogramme, die in solchen sozialen Konfliktsituationen gewinnen, d.h. die besten Entscheidungen treffen
- Beispiele:
 - Gefangenendilemma
 - Ultimatumspiel
 - Piratenspiel
 - Guess Two Thirds of the Average

Wie einfach ist es, sich rationale zu verhalten?

Beispiel: Das Monty Hall Problem

- Benannt nach einem Moderator in den 70ern
- Kandidat einer Gameshow darf zwischen drei Türen wählen
- Hinter einer ist ein Auto, hinter den beiden anderen je eine Ziege
- Nach seiner Wahl, öffnet der Moderator eine der anderen Türen und zeigt ihm eine Ziege
- Danach hat der Kandidat die Möglichkeit, seine Entscheidung zu überdenken, und auf die andere geschlossene Tür zu wechseln
- Fragestellung: Macht es Sinn, die Tür zu wechseln?

Gefangenendilemma

- Zwei Verbrecher sind eines Verbrechens angeklagt, die **Höchststrafe** beträgt **10** Jahre
- **Gestehen** beide, werden sie zu je **5** Jahren verurteilt
- **Schweigen** beide, werden sie nur zu je **0.5** Jahren verurteilt
- **Verpfeift** einer den anderen, indem er gesteht, wird er **freigesprochen**. Der andere erhält die **Höchststrafe**
- **Nutzenmatrix**

	B schweigt	B gesteht
A schweigt	-0.5 / -0.5	-10 / 0
A gesteht	0 / -10	-5 / -5

- Dilemma: Individuelle vs. Kollektive Entscheidung

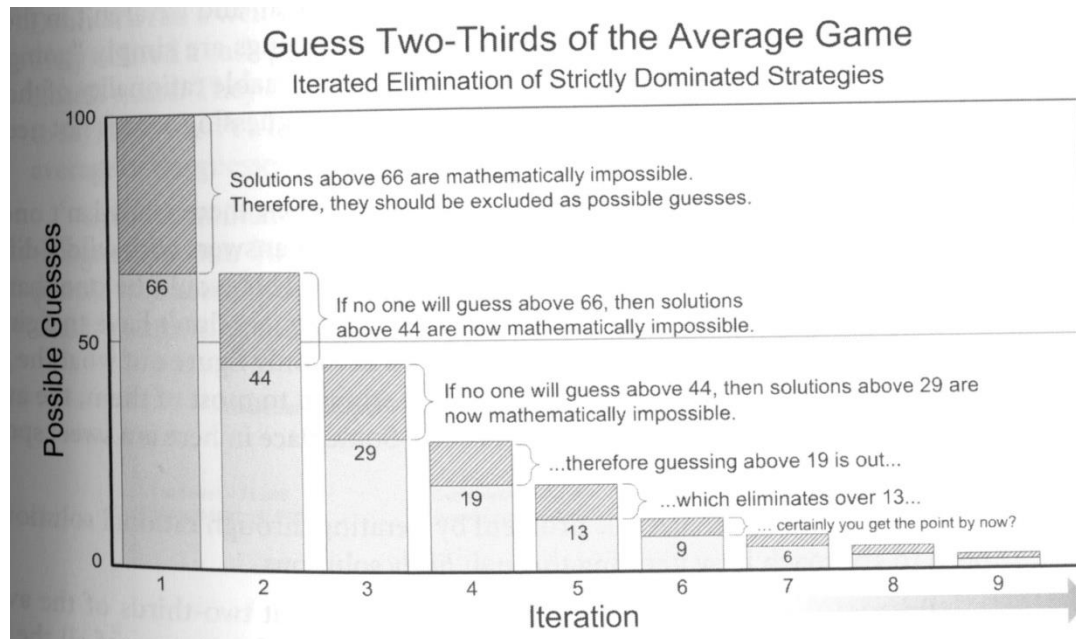
Kann erwartet werden, dass sich jeder rational verhält?

Guess Two Thirds of the Average

- Jeder Teilnehmer rät eine Zahl von 0 bis 100
- Aus allen Zahlen wird der Durchschnitt gebildet, dann mit $\frac{2}{3}$ multipliziert
- Es gewinnt derjenige, dessen Tipp am nächsten an diesem Ergebnis liegt
- Rationalität aller Teilnehmer ist nicht explizit gegeben!

Guess Two Thirds of the Average

- Wenn sich alle Teilnehmer rational verhalten....

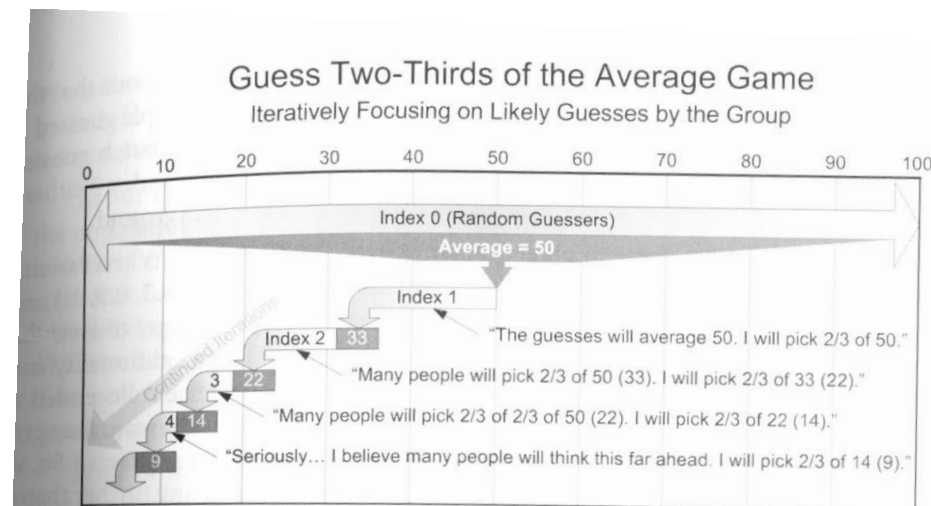


Quelle: Dave Mark - Behavioral Mathematics for Game AI

- ...ist die beste Strategie, 0 zu tippen!

Guess Two Thirds of the Average

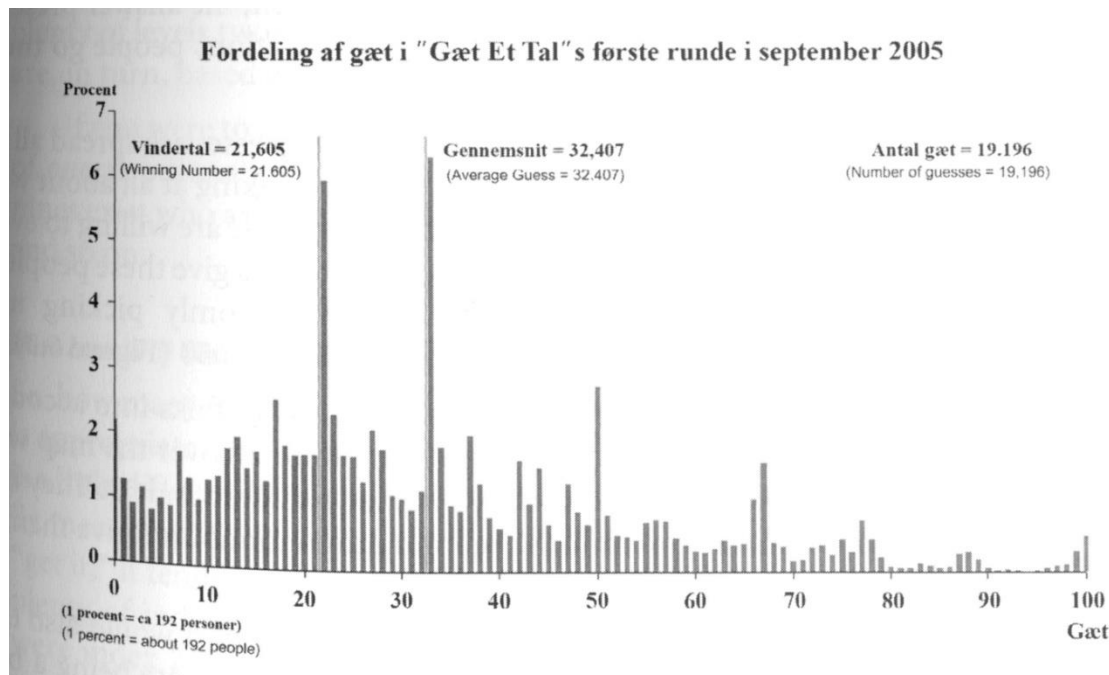
- Wenn sich nicht alle Teilnehmer rational verhalten, dann ist der Durchschnitt der Tipps 50 \Rightarrow 33 tippen
- Aber: Wenn genug Leute so denken, sinkt der Durchschnitt und 33 ist zu hoch...
- Wenn alle auf die Idee kommen, aus diesem Grund 33 zu tippen (sich also gewissermaßen wieder rational verhalten), sollte man wohl 22 tippen
-



Quelle: Dave Mark - Behavioral Mathematics for Game AI

Guess Two Thirds of the Average

- Studie der Uni Kopenhagen
Tyran, Ovlisen: „Make an Educated Guess“ (2005)



Quelle: Dave Mark - Behavioral Mathematics for Game AI

Abgrenzung KI in Spielen zu KI in der Forschung

- KI in der Forschung
 - *strong AI*: Menschliche Gedankenprozesse imitieren
 - *weak AI*: KI-Techniken auf konkrete Probleme anwenden
 - In beiden Fällen geht es um **optimale** Lösungen, Zeit und Ressourcen sind sekundär
- KI in Spielen
 - Limitierte Ressourcen
 - KI muss nicht zwangsläufig intelligent sein, sie muss nur so aussehen
 - KI darf in der Regel nicht optimal sein

- **Zweck:** Zielgerichtetes, realistisches Verhalten:
 - Robustheit in neuen Situationen
 - Erlernen neuer Verhaltensweisen
 - Anpassen an Strategien des Spielers
 - Persönlichkeit und Emotionen
 - Soziales Verhalten
- **Ziel:** Besseres Gameplay:
 - Herausfordernde Gegner
 - Interessante, abwechslungsreiche Story
 - Langzeit-Spielspaß
 - Neue Spielkonzepte

- **Anspruch:**
 - Zielorientiert: KI entscheidet was und wie es zu tun ist und setzt es dann um
 - Reaktiv: KI reagiert unverzüglich auf Veränderungen ihrer Umgebung und adaptiert Wissen
 - Wissend: KI weiß Bescheid über ihre Umgebung und ihr eigenes Verhalten
 - Charakteristisch: KI verkörpert glaubhafte, konsistente Charaktere
 - Schnelle und einfache Entwicklung
 - Geringer Bedarf an Systemressourcen
- **Aber:** Alle Ziele lassen sich (momentan) nicht gleichzeitig verwirklichen (insbes. nicht in Echtzeit)

Komplexitätsmaße einer KI

- Komplexität der Ausführung:
 - Wie wirkt sich das Hinzufügen von Wissen auf die Laufzeit aus?
 - Wie viel mehr Speicher braucht man beim Hinzufügen von Wissen?
- Komplexität in der Spezifikation:
 - Wie schwer ist es zu programmieren?
 - Wie viel Implementierungsaufwand ist mit dem Einpflegen von neuem Wissen verbunden?
 - Wie aufwändig ist es zu warten?
 - Was für Entwicklungskosten und Risiken gibt es?

Charakterisierung von Spielwelten

- Zugänglich / Nicht zugänglich:
 - Umgebung kann vollständig wahrgenommen werden
 - Nicht alle Zustände der Umgebung sind bekannt
- Statisch / Dynamisch:
 - Umgebung wird allein durch simulierte Aktionen verändert
 - Zusätzliche unbekannte Einflussgrößen
- Deterministisch / Nicht deterministisch:
 - Effekte von Aktionen sind bekannt
 - Effekte von Aktionen z.B. durch Wahrscheinlichkeitsverteilungen definiert
- Diskret / Kontinuierlich:
 - Begrenzte Menge an Zuständen
 - Zustände werden durch kontinuierliche Größen repräsentiert

- Polling:
 - Aktives Beobachten der Welt durch die Objekte in festen Intervallen
 - Feststellen, was sich geändert hat
 - Reaktion auf Änderungen
 - i.A. ineffizient
- Event-Driven:
 - Objekte werden durch Nachrichten der Welt auf Änderungen hingewiesen
 - Aber: Wann (und wie oft) sollen Nachrichten versendet werden?
- Smart Objects (Beispiel: The Sims)
 - Agent schickt Nachricht „Habe Hunger“
 - Kühlschrank: „Habe Essen“
 - Agent: „Wie komm ich da ran?“
 - Kühlschrank: „Öffne Mich“, „Nimm dir das Essen“
 - Agent an Essen: „Wie ess ich dich?“
 - Essen: „Du musst mich kochen“
 - Agent: „Wer kann hier kochen?“ etc.



- Paradoxon der Spiele KI: KI soll letztendlich verlieren, aber nur sehr knapp
- Denn: Ziel des Spiels ist es zu unterhalten, nicht zu frustrieren
- Ungewöhnlich intelligente Gegner führen nicht unbedingt zu mehr Spielspaß:
 - Mangelnde Fairness gegenüber dem Spieler
 - Verlust an Glaubwürdigkeit (und damit an Immersion)
 - Falsche (aber nachvollziehbare) Aktionen erhöhen Spielspaß
- L. Lidén: Artificial Stupidity: The Art of Intentional Mistakes (2003)

(am Beispiel von Ego-Shootern)

- Move Before Firing
- Be Visible
- Have Horrible Aim
- Miss the First Time
- Warn the Player
- Attack „Kung-Fu“ Style
- Tell the Player What You Are Doing
- Pull Back at the Last Minute
- Be Vulnerable

Total War



KI-Innovationen:

- Tausende KI-gesteuerte Soldaten können dargestellt werden
- Modellierung von Emotionen und Moral der Soldaten
- Engine wird für Dokumentationen des History Channels verwendet

Façade

(<http://www.interactivestory.net>)



KI-Innovationen:

- Spieler interagiert über Textparser
- Dem Verhalten der Charaktere liegt eine mächtige Beschreibungssprache zu Grunde

Black & White

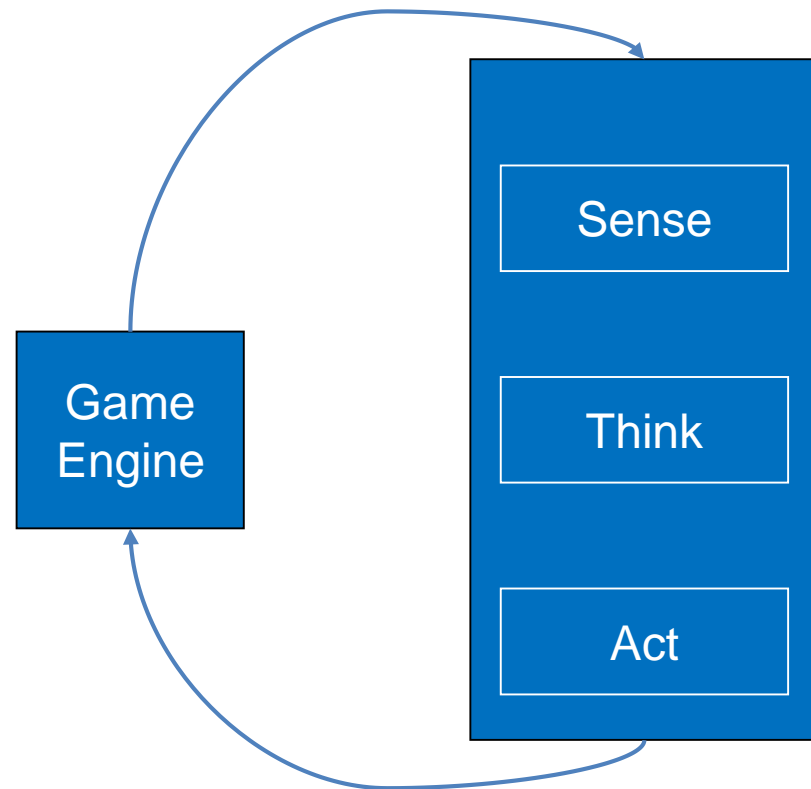
KI-Innovationen:

- Spieler interagiert mit KI-Kreatur, die anhand von Beispielen und Feedback (Belohnung bzw. Bestrafung) des Spielers lernt
- Verbindet verschiedene KI-Techniken wie Entscheidungsbäume und Neuronale Netze



KI Aufgaben

- Basisfähigkeiten:
 - Wegplanung
 - Bewegung
- Verhaltenssteuerung:
 - **Entscheidungs- und Lernverfahren**
 - Animationssteuerung
- Team-Steuerung:
 - Koordiniertes Gruppenverhalten



Grundproblem: Gegeben ein Weltzustand, was soll ich tun?

Entscheidungsverfahren:

- Skripte
- Endliche Automaten
- Regelbasierte Systeme*
- Fuzzy Logic*
- Entscheidungsbäume*
- Planen*
- Behavior Trees*

Lernverfahren:

- Entscheidungsbäume*
- Neuronale Netze
- Genetische Algorithmen*
- Bayes'sche Netze*
- *Reinforcement Learning**

* Nächstes mal

Wichtig: Es gibt nicht **die** ultimative Spiele-KI. Je nach Genre und Aufgabe der KI (Bot, Gegner, NPC, etc.) ist die eine oder andere Technik besser geeignet.



Entscheidungsverfahren

Beispiel: Skript aus Neverwinter Nights

```
// Gesprächsskript: tm_guard_c1
// Nimmt den Ring des SCs
// und belohnt jeden SC mit 50 EP.
void main()
{
    object oPC=GetFirstPC();
    DestroyObject(GetObjectByTag("PASSRING"));

    while (oPC != OBJECT_INVALID)
    {
        GiveXPToCreature(oPC, 50);
        oPC=GetNextPC();
    }
}
```

Beispiel: Skriptsequenz „Lost Brother Challenge“ aus Black & White

```
begin cinema
```

```
    //Create a moody storm
```

```
    start music MOODY_PIANO
```

```
    Storm = create WEATHER DRIZZLE at {Brother}
```

```
    //Move the sister out with a new walking animation
```

```
    move Sister position to {SoapBox}
```

```
    set Sister anim DESPAIR_WALK
```

```
    //Set the camera to dynamically follow the sister
```

```
    set camera focus follow Sister
```

```
    move camera position to {SoapBox}+{-3,4,-3} time 10
```

```
    //when the sister nears the box, make her pray
```

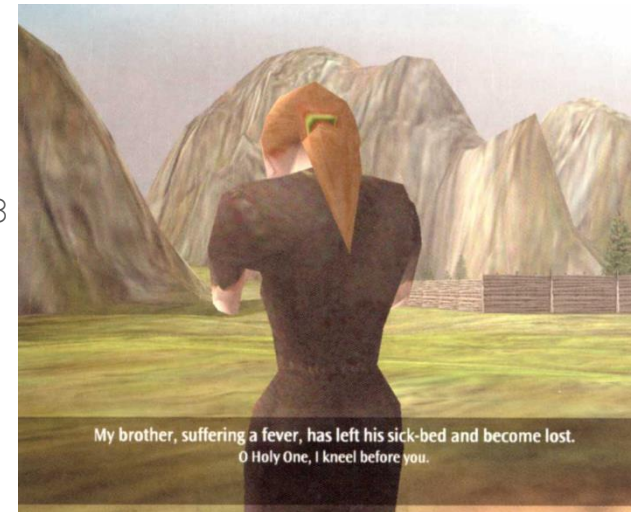
```
    wait until {Sister} near {SoapBox} radius 1
```

```
    Sister play PRAY loop 1
```

```
    wait 4 seconds
```

```
//Sister: „Oh holy one, I kneel before you“  
say LOST_BROTHER_LINE1  
wait until camera ready  
  
//Zoom the camera in on the sister  
move camera position to {Sister}+{-2,1,-2} time 8  
move camera focus to {Sister}+{0,0.7,0} time 8  
  
//Wait until she's said her line and  
//stopped praying  
wait until read and Sister played  
  
//Turn the sister to face her brother and point at him  
set Sister focus to {SickBed}  
Sister play POINTING loop 1  
  
//Sister: „My brother ... [has] ... become lost.“  
say LOST_BROTHER_LINE2
```

...



- Volle Kontrolle über Verhaltensmodell
- Oft für Zwischensequenzen verwendet
- Ideal zur Umsetzung kurzfristiger Ziele
- Durch Ereignisse ausgelöst
- Komplexes Verhalten möglich
- Unterstützung durch Spiele-Editoren
- Deterministisch
- Meist schnell vorhersehbar („geskriptet“)
- Schwierig, auf unvorhergesehenes zu reagieren

Endlicher Automat (EA)

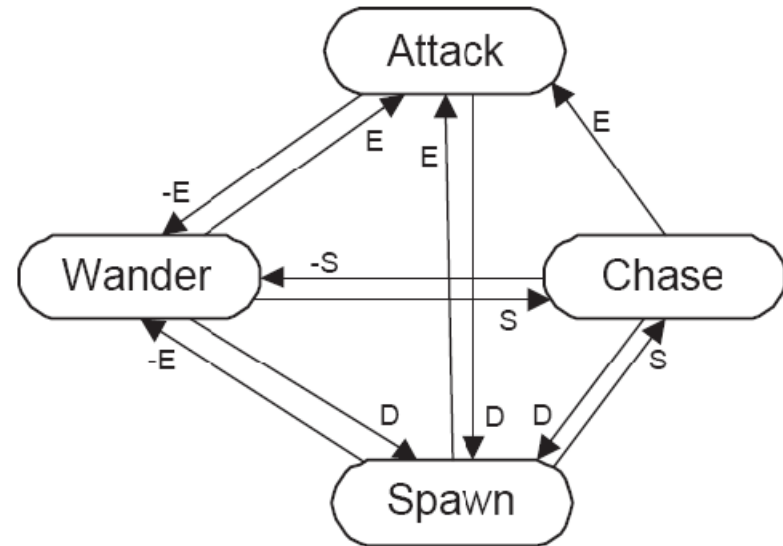
- Eingabe-Alphabet: Σ
 - Ereignisse, Handlungen, Zeit
 - Interne Zustände / Attribute
- Zustände: Q
 - Verhaltensweisen, Ziele, Bedürfnisse, Motivationen etc.
- Ausgezeichnete Zustände:
 - Startzustand q_0
 - Endzustände q_E
 - Aktueller Zustand
- Übergangsfunktion: $\delta_T : Q \times \Sigma \rightarrow Q'$
- Aktionen oder Animationen:
 - Mit Zuständen und / oder Transitionen verbunden

Wie für KI einsetzen?

- Zustände
 - Verhalten der KI -> gefällte Entscheidungen
 - Startzustand: Verhalten bei Spielstart
 - Endzustand: Terminierung der KI
- Übergangsfunktion
 - Ändern des aktuellen Verhaltens -> Entscheidungen der KI
- Eingabealphabet
 - Ereignisse in der Spielwelt
 - Handlungen anderer Figuren
 - Zeit

EA für einen Quake Bot

- Zustände:
 - Feind angreifen (Attack)
 - Welt erkunden (Wander)
 - Feind verfolgen (Chase)
 - Welt betreten (Spawn)
- Eingaben bzw. Ereignisse:
 - E, -E: (kein) Feindkontakt (**E**nemy)
 - S, -S: Geräusch (nicht mehr) wahrgenommen (**S**ound)
 - D: Sterben (**D**ie)



Bewertung von endlichen Automaten

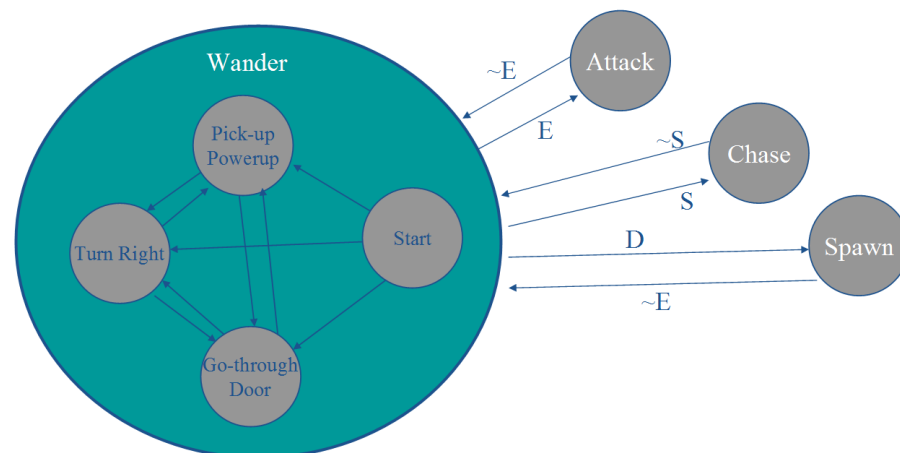
Vorteile:

- Intuitiv
- Ausdrucksstark
- Effizient
- Leicht zu implementieren
- Leicht zu debuggen
- Visuelle Editoren

Nachteile:

- Nicht modular
- Nicht lernfähig
- Beschränktes Repertoire an Verhaltensweisen
- Vorhersehbares Verhalten
- Anzahl der Zustände und Transitionen wächst sehr schnell (exponentiell)
- Vergleichende Propositionen schwierig, z.B. „nimm die bessere Waffe“

- Komplexere Zustände können wiederum durch einen EA spezifiziert werden
- Eingaben sorgen entweder für Zustandsübergänge innerhalb derselben Hierarchieebene oder führen auf die darüber liegende Ebene zurück
- Bei Aktivierung eines Zustands muss ein Startzustand für dessen EA (falls vorhanden) gewählt bzw. festgelegt werden
- Beispiel: Bot in Quake II



- Nichtdeterministische Endliche Automaten
- Wahrscheinlichkeitsverteilungen in Übergangsfunktion δ_T
- Zufällige Auswahl zur Laufzeit
- Markov-Modelle:
Kontext (vorangegangene Transitionen) wird berücksichtigt
- Anwendungen:
 - Spezifisches Verhalten (Gruppierung, Kultur, Persönlichkeit)
 - Beispiel: Quake II Bot mit unterschiedlichen Angriffsmustern

