

# → Game Development

Spring Semester 2009 (MTG)

## AI in Games (Advanced)

→ Georgios N. Yannakakis



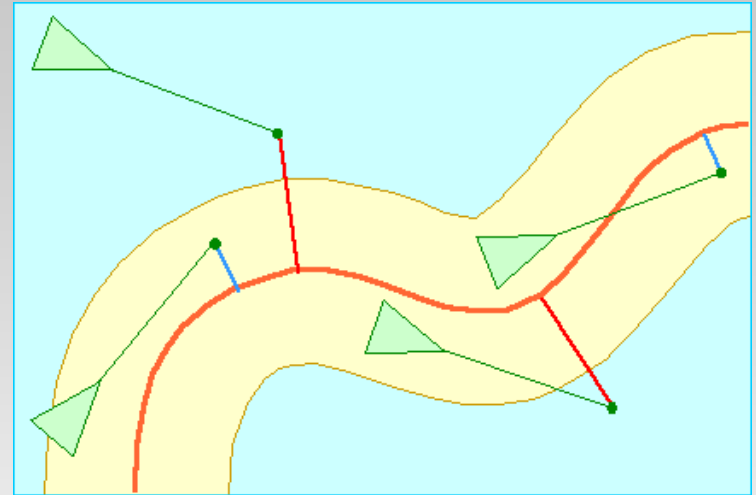
## → Lecture Structure

- Believable AI
  - Perception, controller, action,...
- Finite State Machines
  - In General
  - In Unreal
  - How to expand/improve?
- Fuzzy Logic (briefly)



# → But Before all that... ...a student request

- Steering Behaviors  
[Reynolds, GDC, 1999]



A steering force is applied

*# make sure the desired steering force isn't too big*

```
steering_force = truncate( steering_force, max_force)
```

*# compute acceleration*

```
acceleration = steering_force / mass
```

*# acceleration is added to the old velocity to produce a new velocity*

```
velocity = truncate (velocity + acceleration, max_speed)
```

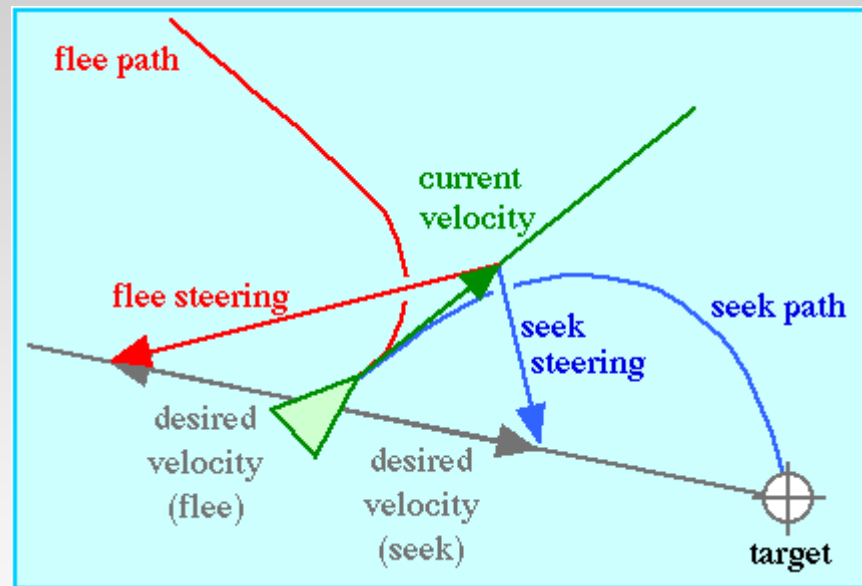
*# velocity is added to the old position*

```
position = position + velocity
```



# → Steering Behavior

Example – Seek and Flee



Demo

*# Steering Vector*

```
desired_velocity = normalize(position - target)*max_speed
```

```
steering = desired_velocity - velocity
```





# Believable AI Agent



# → AI Agent

Environment

- What does it look like?
- How can you model it?  
.....Ideas?



# → AI Agent

## Environment - Perception



Perception (Input)

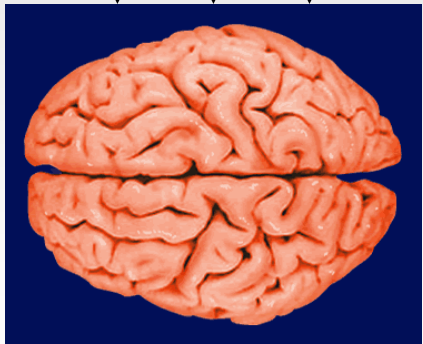
- How can you perceive it?
  - See
  - Touch
  - Hear
  - Smell
  - Taste
- Yes but how NPCs perceive it realistically? Any ideas?
  - See (infra-red, radar-type, line traces, camera)
  - Touch (pressure force sensors)
  - Hear (microphone)
  - Smell (chemical sensors – pheromone)
  - Taste? Hmmm.. Ideas?
  - Sixth sense? Superpowers? Nah...

# → AI Agent

Environment – Perception - Control



Perception (Input)



- Now you perceive the game... so what?



- You (NPCs) have to...  
 ....USE your Brain (AI)  
 Brain (AI) types? **Ideas?**

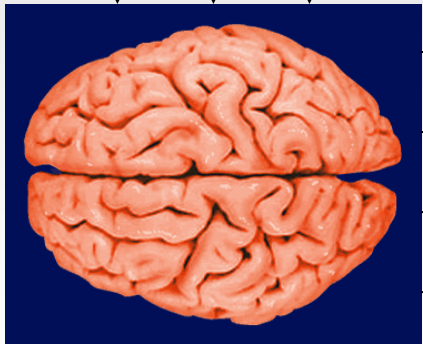
N.B. How we built artificial “brains”? See next...

# → AI Agent

Environment – Perception – Control - Action



Perception (Input)



Action (Output)

- Use your brain to do what? Ideas?
  - Move muscles
  - Alter emotional state
  - Make a decision
  - Plan (sequence of decisions)
  - Remember
  - ...

In other words, ACT

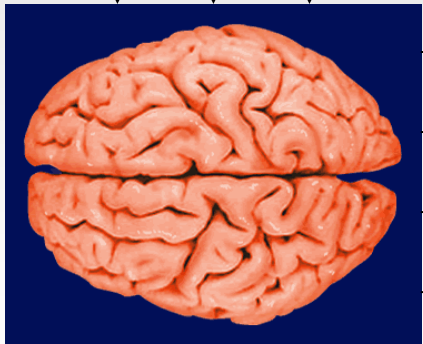
- Realistically
- Reasonably (look believable and not nuts/stupid!)
- This is what NPCs should do too!

# → AI Agent

Environment – Perception – Control – Action – Environment (New)



Perception (Input)

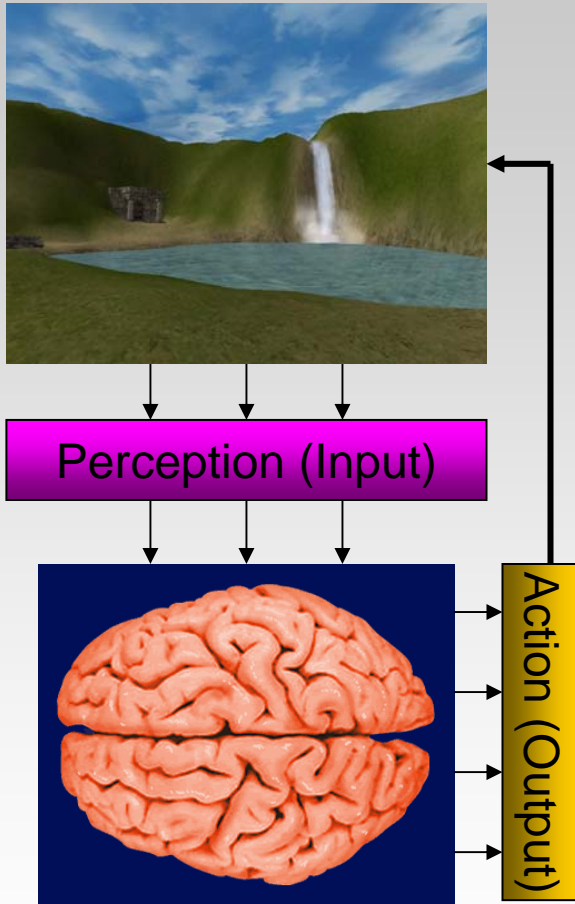


Action (Output)

- Your actions affect your world!!

# → AI Controller

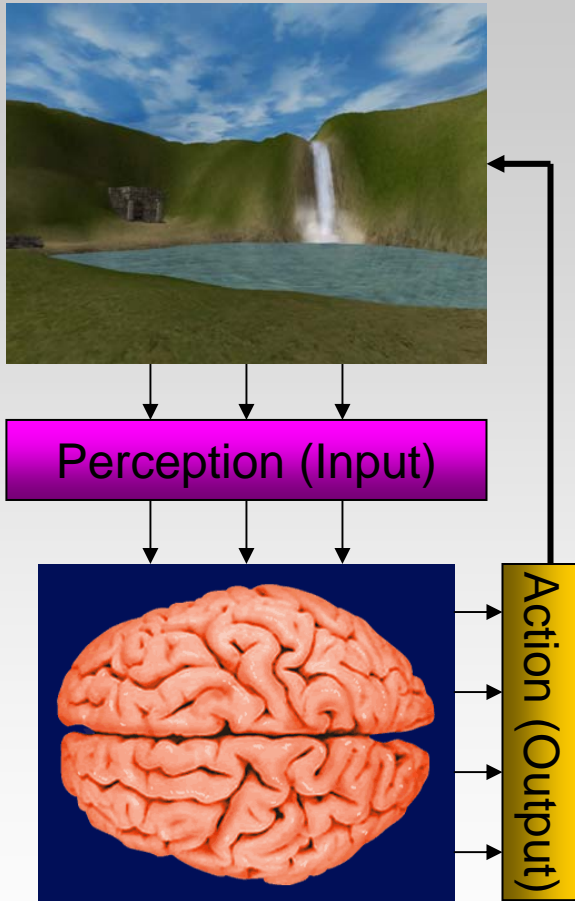
What is it?



- What is a “brain” (AI) like?
- How do I built an artificial brain? Ideas?
- **Basic**
  - I built it from scratch based on my brain!
- **Advanced (next semester...)**
  - I let it learn based on samples of good behavior
  - I let it learn by rewards/penalties
  - I let it learn by itself! (?)

# → AI Control

Expert Knowledge/Ad hoc AI



- Building brains from scratch requires expert knowledge.
  - Are you expert enough???
- Techniques:
  - Finite State Machines
  - Rule-Based Systems
  - Fuzzy Logic
  - ...

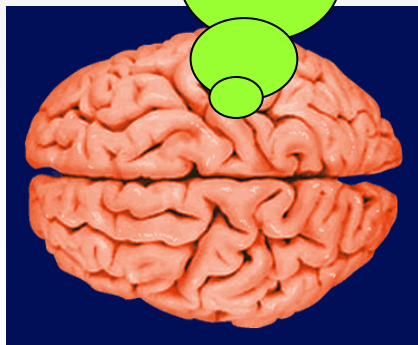
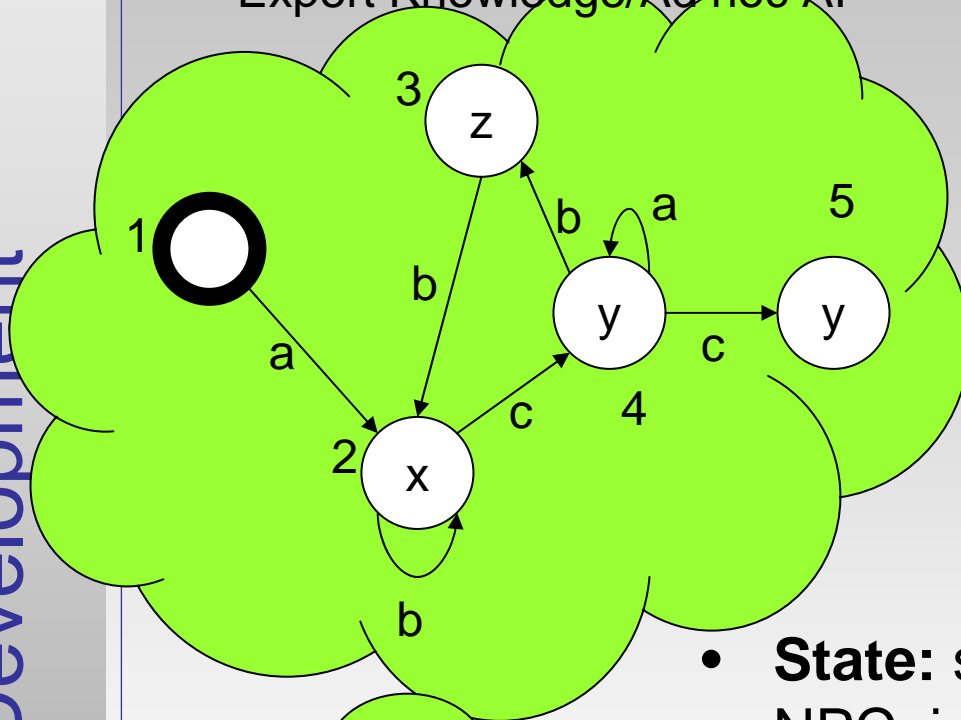


# NPC Control: I.e. a Finite State Machine (FSM)



## → FSM

Expert Knowledge/Ad hoc AI

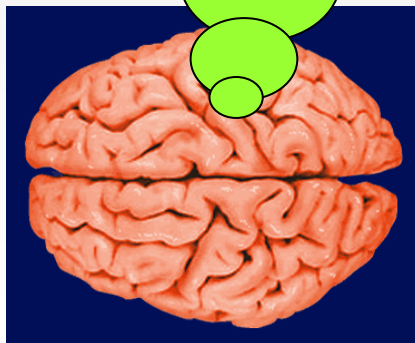
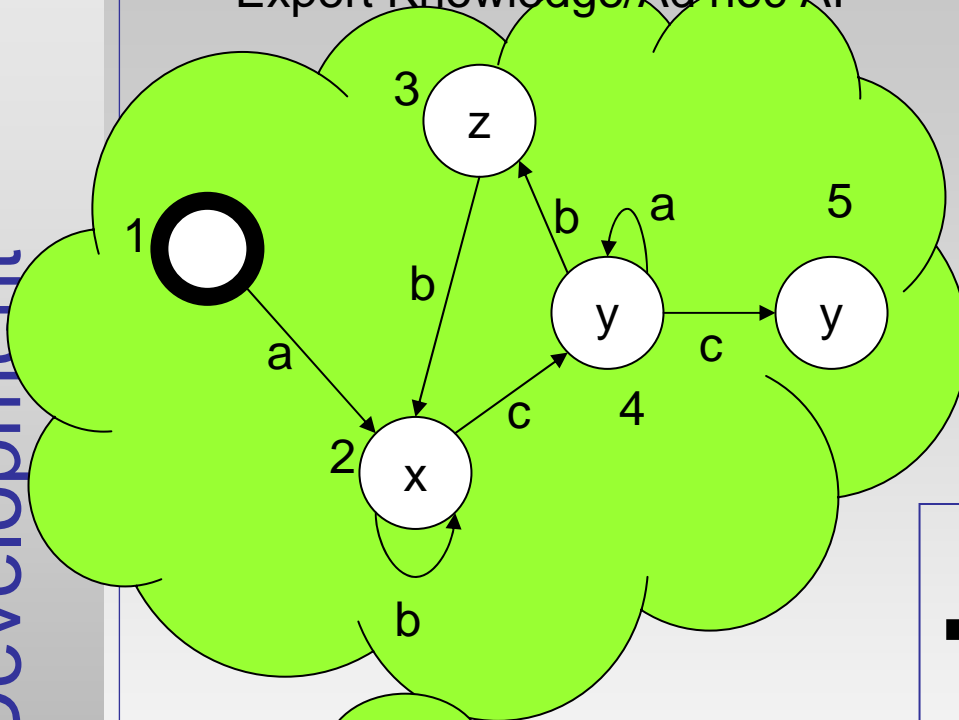


FSMs in a nutshell:

- **State:** stores information about an NPC, i.e. *explore*
- **Transition:** indicates a state change and is described by a **condition** that needs to be fulfilled.
- **Action:** description of an activity that is to be performed.

# → FSM

Expert Knowledge/Ad hoc AI

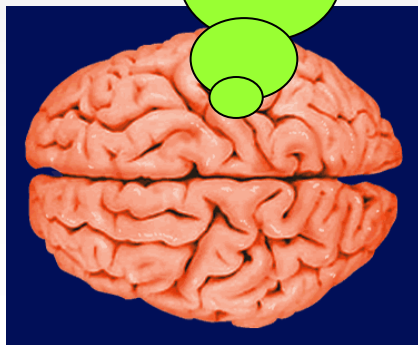
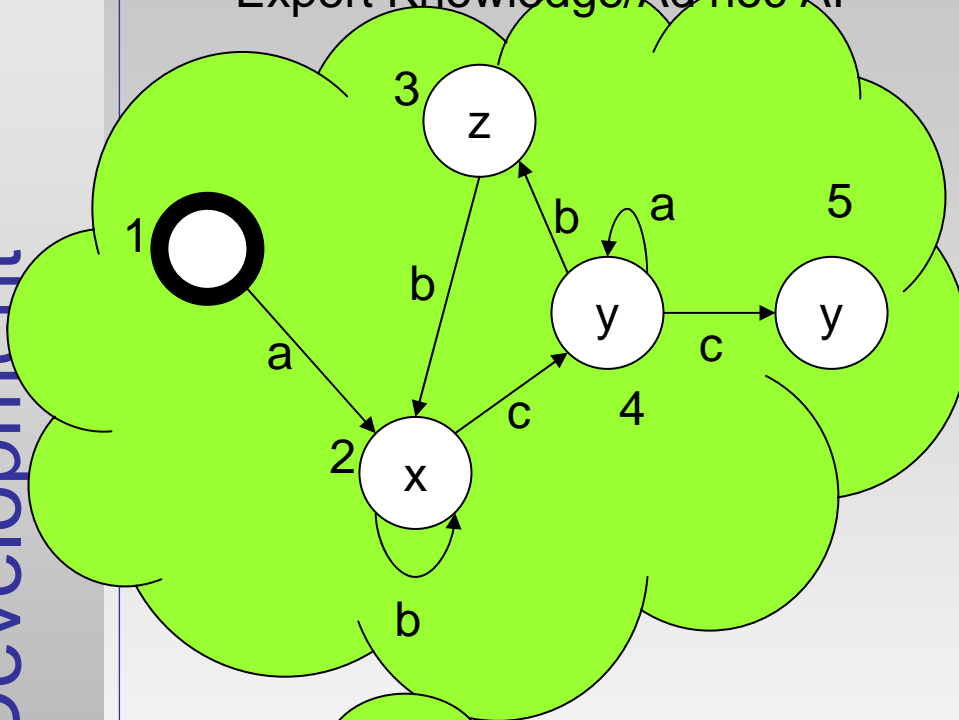


- **Incredibly simple!** (design, implement, visualize, debug)
- **Proof they work well with computer games** (the most popular AI technique!)
- **Used mainly for control** (Every game developer can use them)



# → FSM

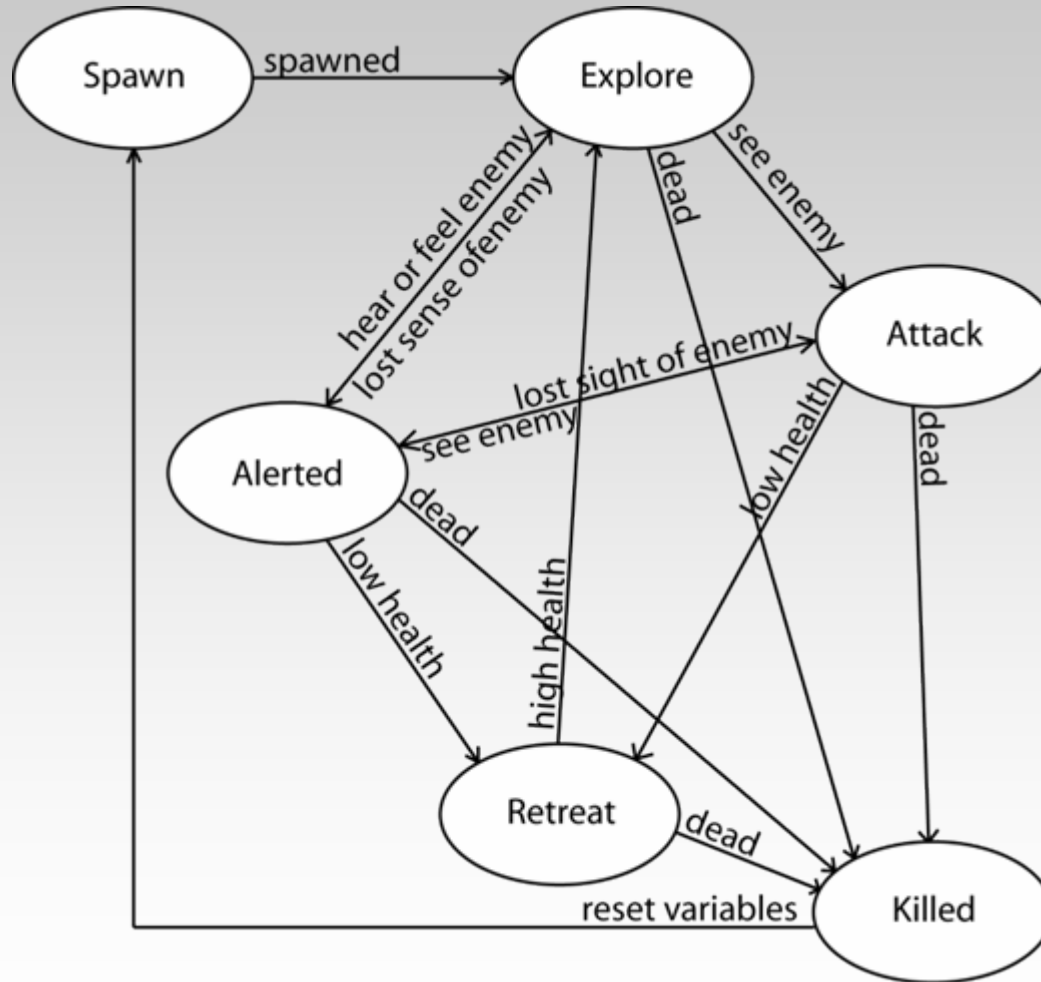
Expert Knowledge/Ad hoc AI



- 
- Complex on large scale
- Computationally limited to certain problems
- Not suited for simple tasks: e.g. counting
- No flexibility in real-time (**no adaptivity**)
- Very Predictable output – **very predictable behaviors!**

# → FSM

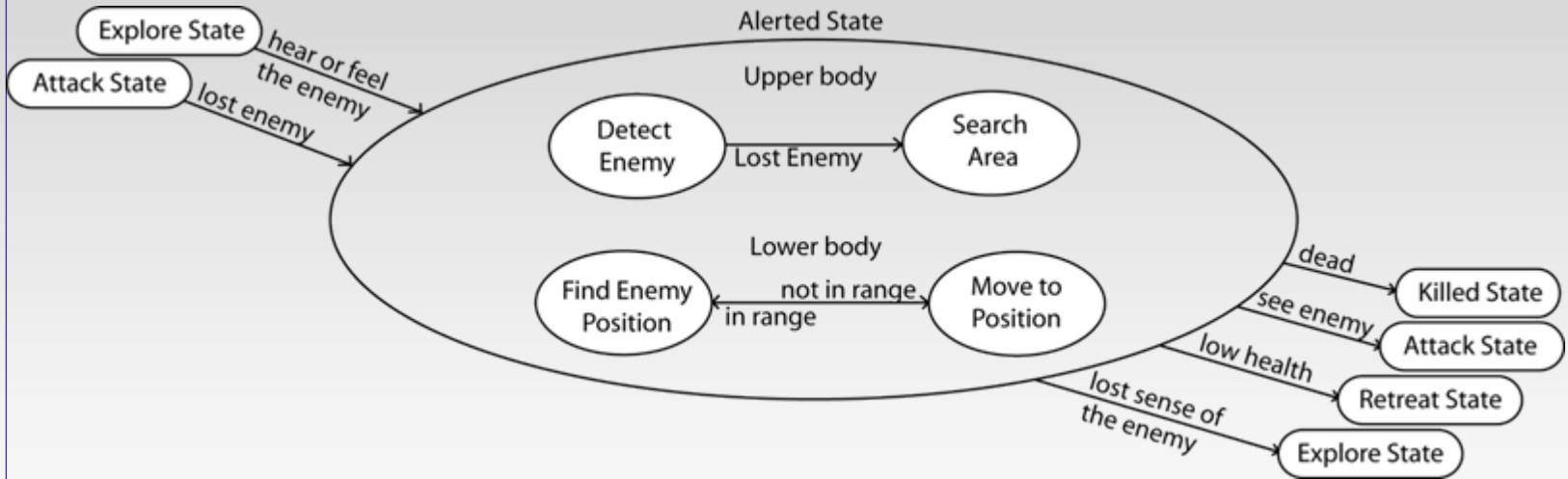
An Example in Quake 4 / Hierarchical FSM – Top Level



Anders Rosendal, "Team Based AI in Quake 4," MSc Thesis, ITU, 2008

# → FSM

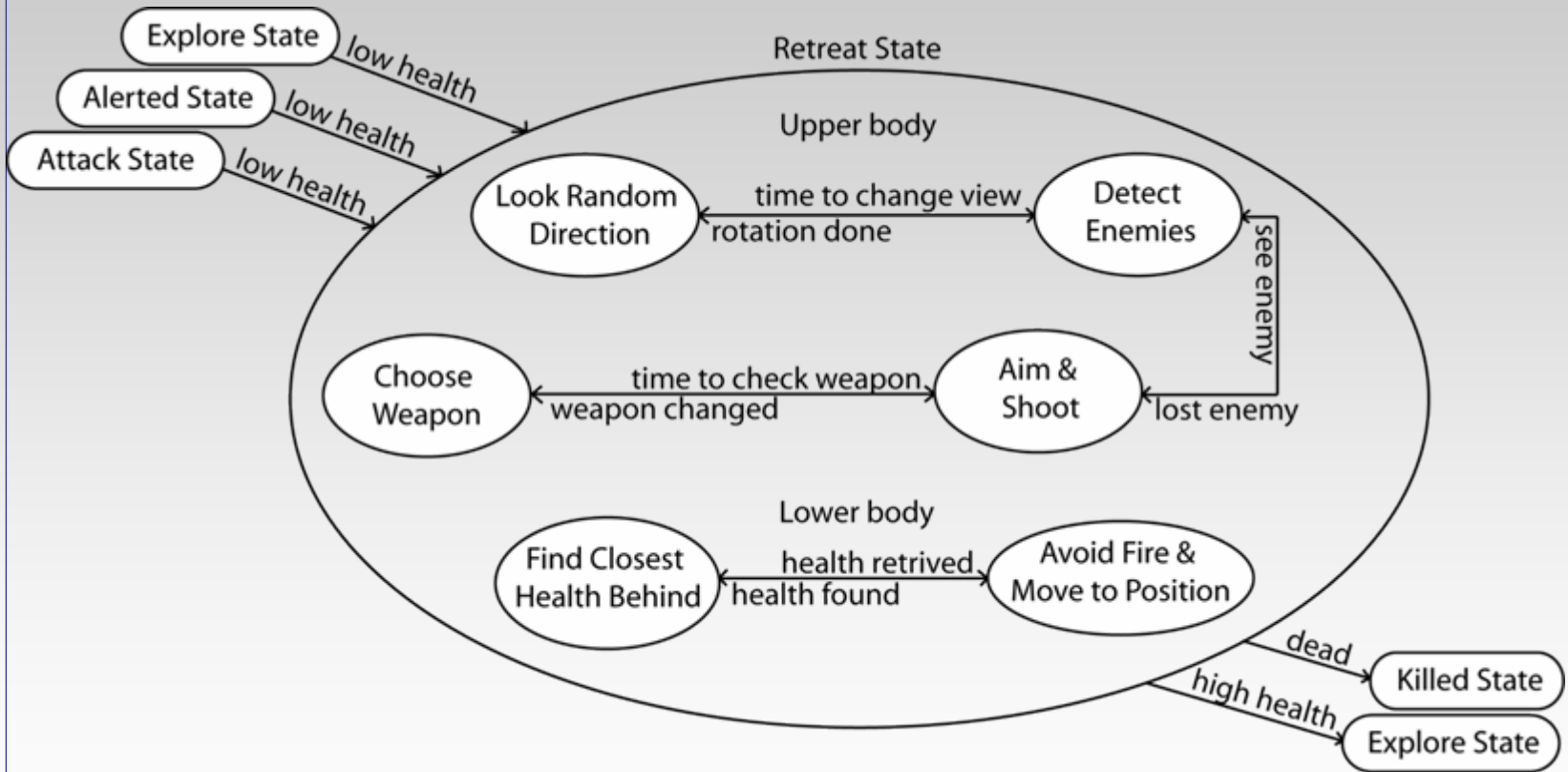
An Example in Quake 4 / Hierarchical FSM – Alerted State



Anders Rosendal, "Team Based AI in Quake 4," MSc Thesis, ITU, 2008

# → FSM

An Example in Quake 4 / Hierarchical FSM – Retreat State



Anders Rosendal, "Team Based AI in Quake 4," MSc Thesis, ITU, 2008



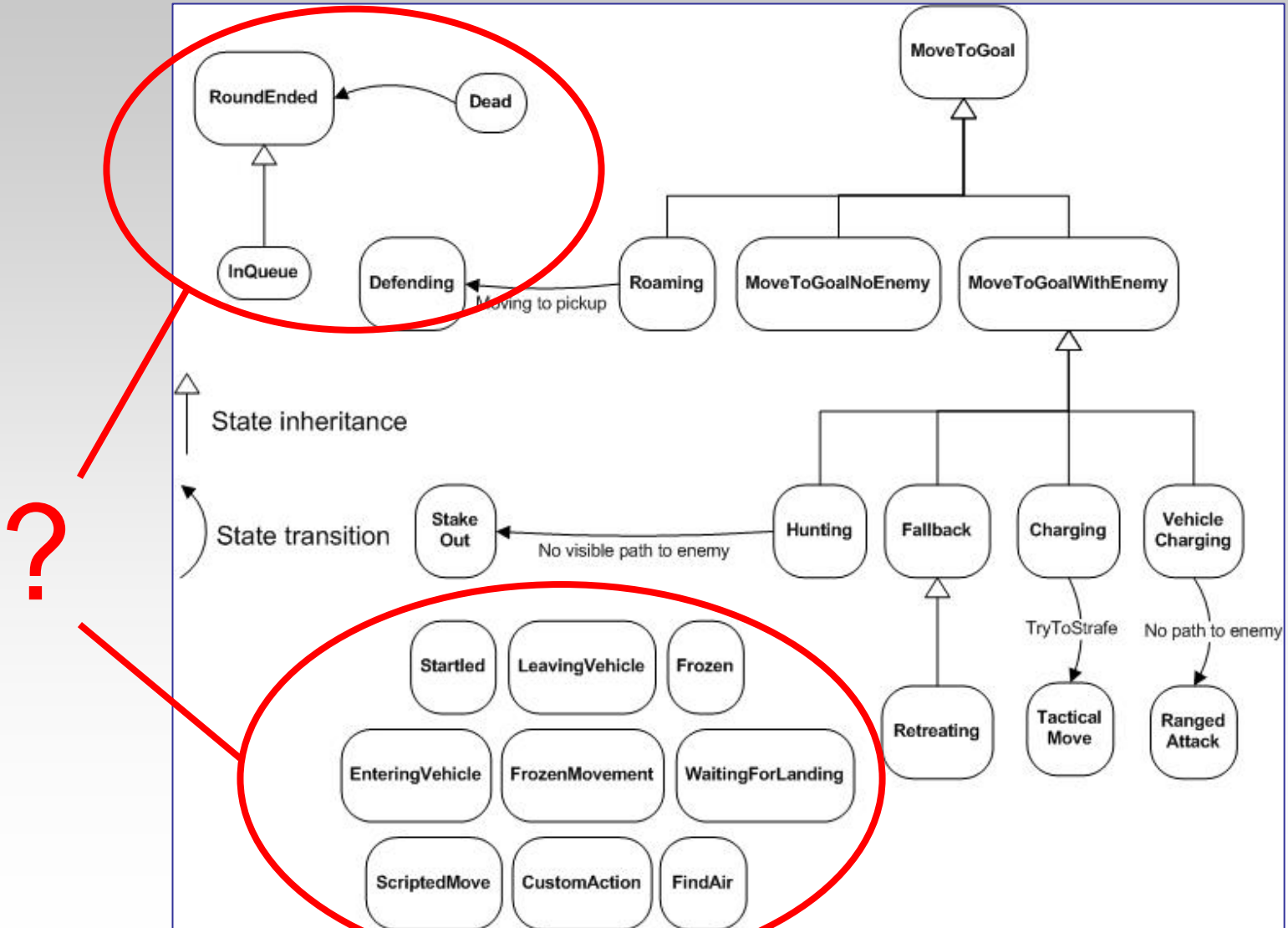
# FSM

An Example / Demo



# → FSMs in Unreal

Expert Knowledge/Ad hoc AI

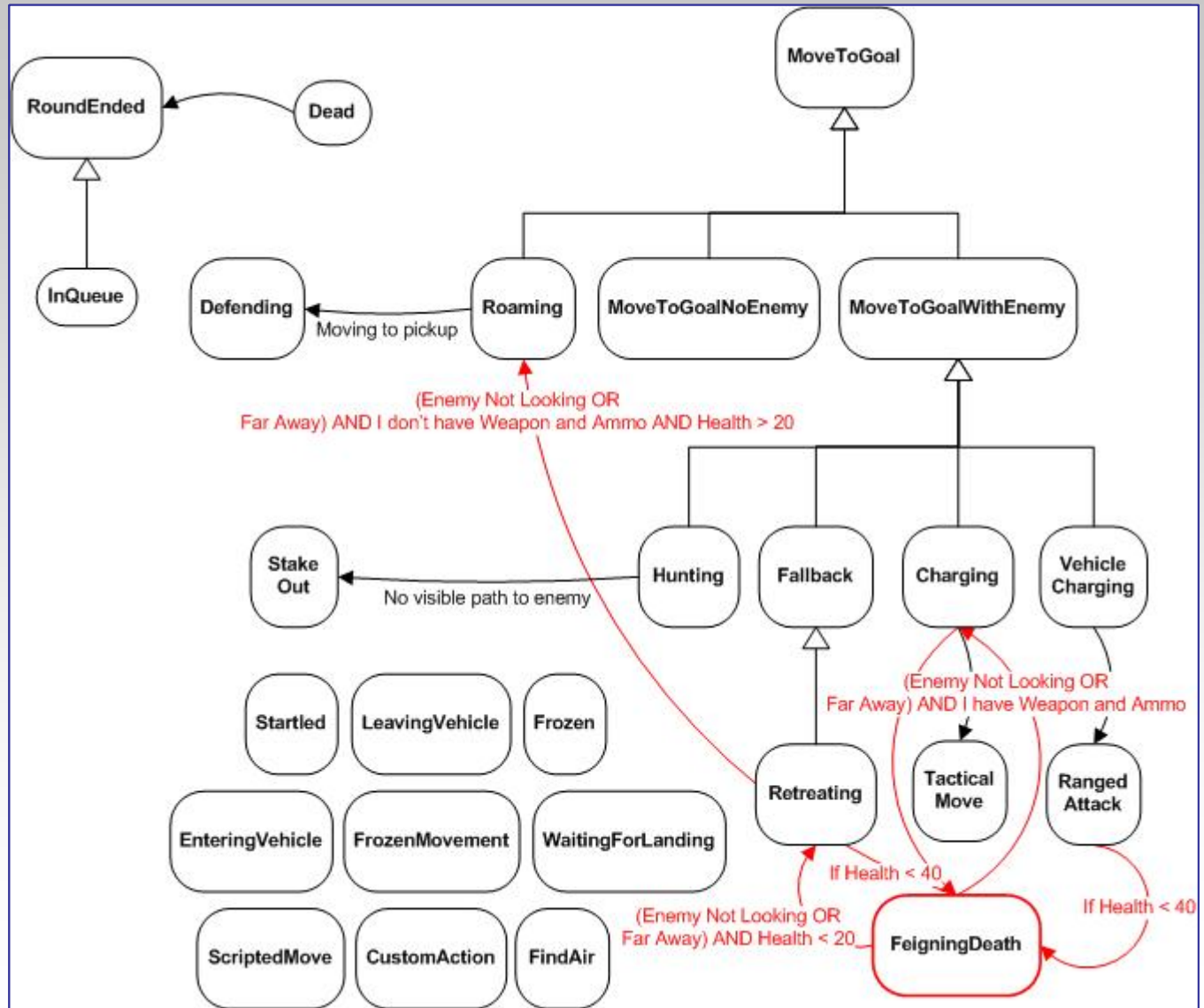


?



# FSMs in Unreal

How to extend – An example



# → FSMs in Unreal

Further Info on this example?

- Source Code on Peder's Unreal Homepage



# → FSMs in Unreal

How to extend – alter the Unreal FSM?

- Two strategies:
  - Extending UTBot
    - Good idea if the behavior of bots in Unreal is describable
  - Extending AIController
    - More of a fresh start approach
- Example: Dancing (!) bots
- More on this by Lawrence at 13:00...





# Fuzzy by Example



# → Fuzzy Logic: Beyond FSMs

- Previous Example: A Rule Based System in *Feigning Death State*:
  - If Enemy turned his back on me OR enemy distance > 1000
    - If Weapon and Ammo available → Charging
    - If Weapon and Ammo NOT available AND Health > 20 → Roaming
    - If Health < 20 → Retreat
  - Else → Feigning Death
- Replace that with a Fuzzy rule system... make it more believable...
  - Other techniques are possible... see next semester...

# → Fuzzy Rules

## Fuzzy Rule Set - Example

- Linguistic expression #1:
  - If *enemy IS far* ( $F(x)>0$ ) AND *ammo IS full* ( $Fu(x)>0$ ) THEN *attack* ( $A$ )
- Fuzzy expression (rule #1):  
$$F \wedge Fu \Rightarrow A \Leftrightarrow \min(F(x), Fu(x)) = A(x)$$
- Linguistic expression #2:
  - If *enemy IS far* ( $F(x)>0$ ) AND *ammo IS moderate* ( $M(x)>0$ ) AND *my health IS OK* ( $OK(x)>0$ ) THEN *stay neutral* ( $N$ )
- Fuzzy expression (rule #2):  
$$F \wedge M \wedge OK \Rightarrow N \Leftrightarrow \min(F(x), M(x), OK(x)) = N(x)$$
- Linguistic expression #3:
  - If *enemy IS far* ( $F(x)>0$ ) AND *my health IS Low* ( $L(x)>0$ ) THEN *retreat* ( $R$ )
- Fuzzy expression (rule #3):  
$$F \wedge L \Rightarrow R \Leftrightarrow \min(F(x), L(x)) = R(x)$$

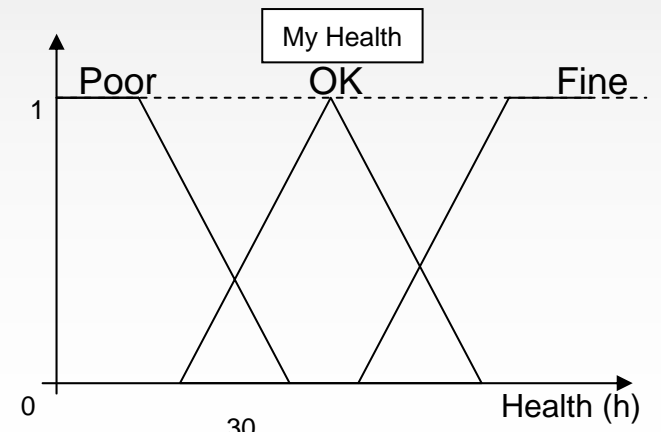
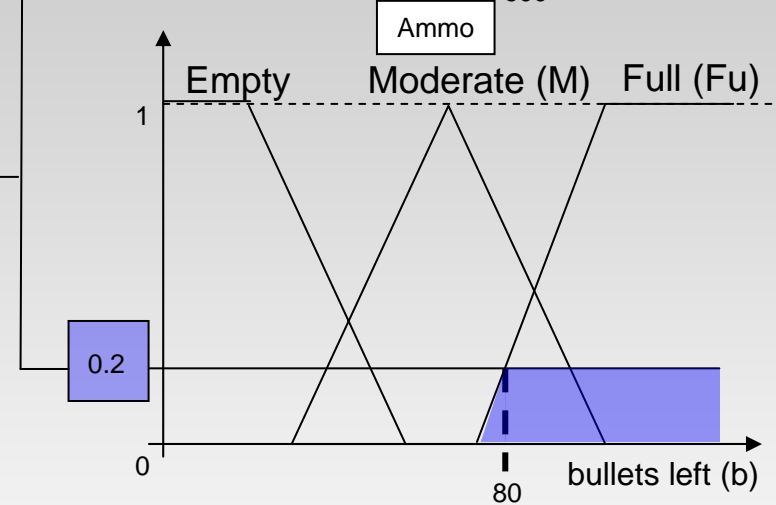
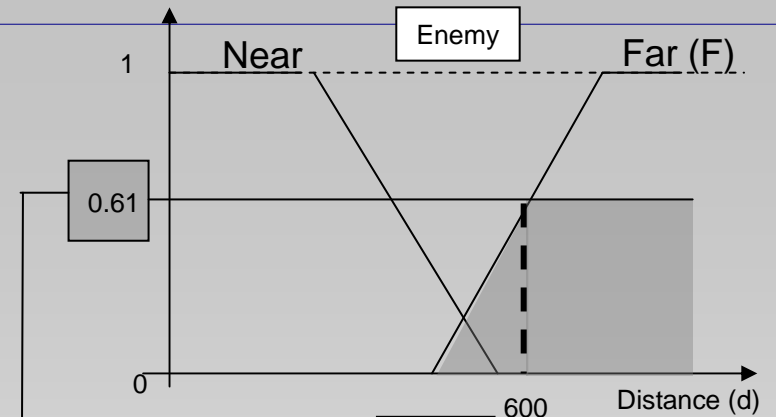
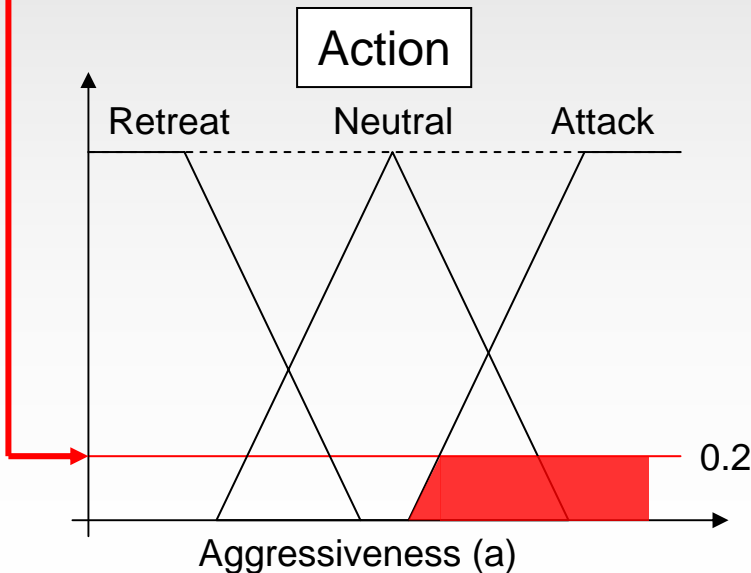


# → Fuzzy Rules

## Fuzzy Rule Set – RULE #1

RULE #1:  
 If *enemy* IS *far* ( $F(x) > 0$ ) AND  
*ammo* IS *full* ( $Fu(x) > 0$ ) THEN  
*attack* (A)

$F \text{ AND } Fu = \min(F(d), Fu(b)) = 0.2$

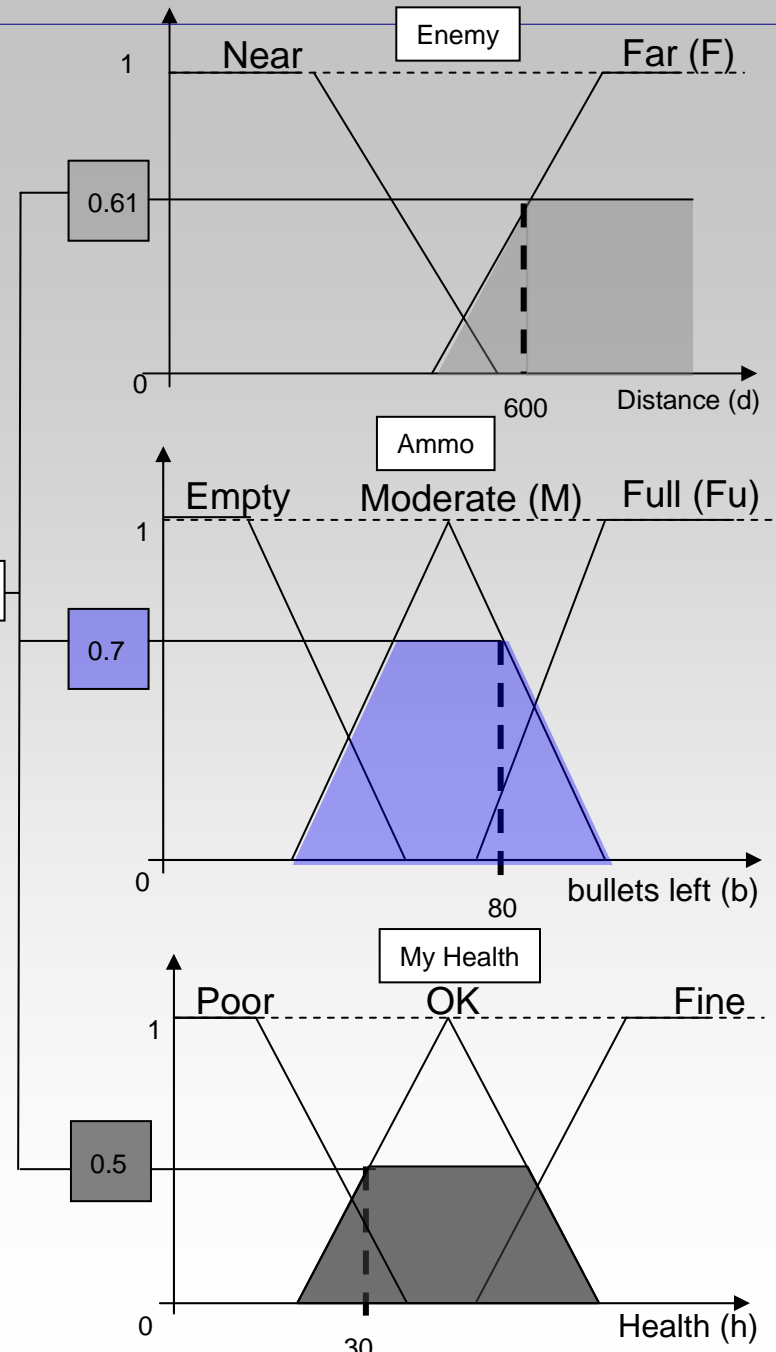
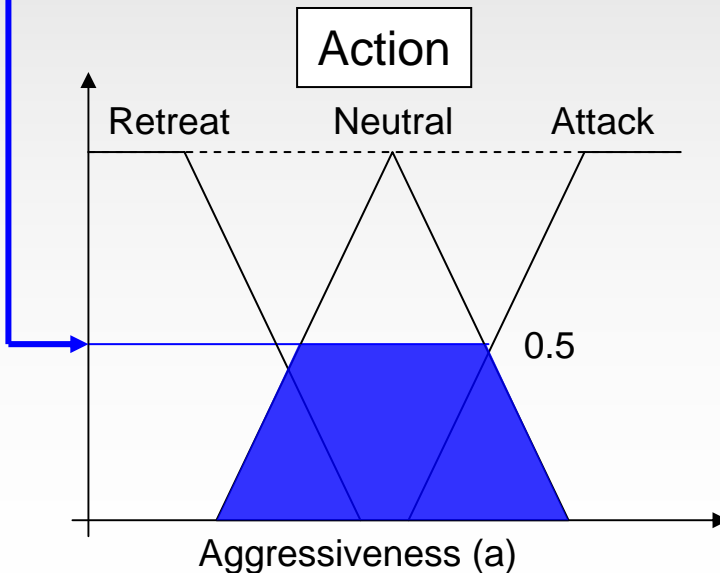


# → Fuzzy Rules

## Fuzzy Rule Set – RULE #2

**RULE #2:**  
 If *enemy* IS *far* ( $F(x) > 0$ ) AND  
*ammo* IS *moderate* ( $M(x) > 0$ )  
 AND *my health* IS *OK* ( $OK(x) > 0$ )  
 THEN stay *neutral* (N)

$F \text{ AND } M \text{ AND } OK = \min(F(d), M(b), OK(h)) = 0.5$

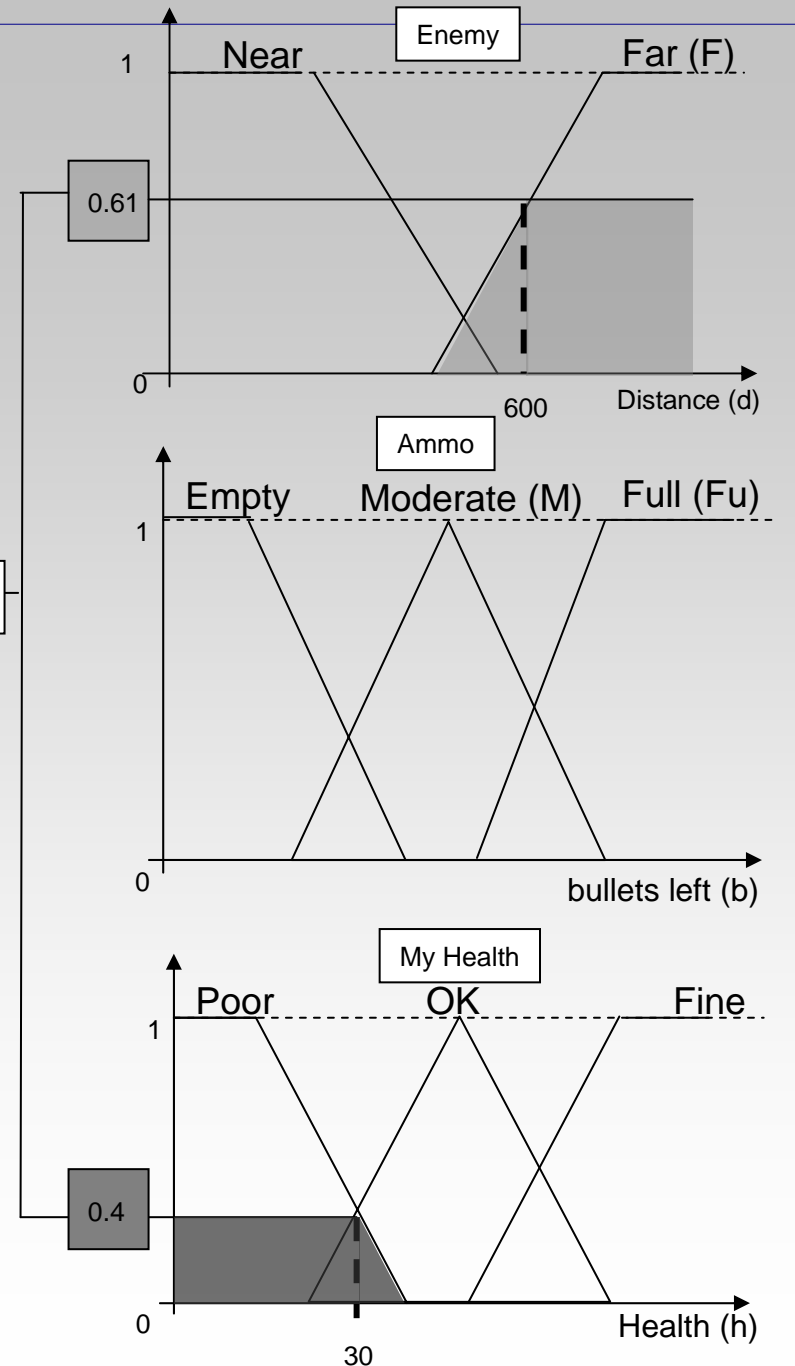
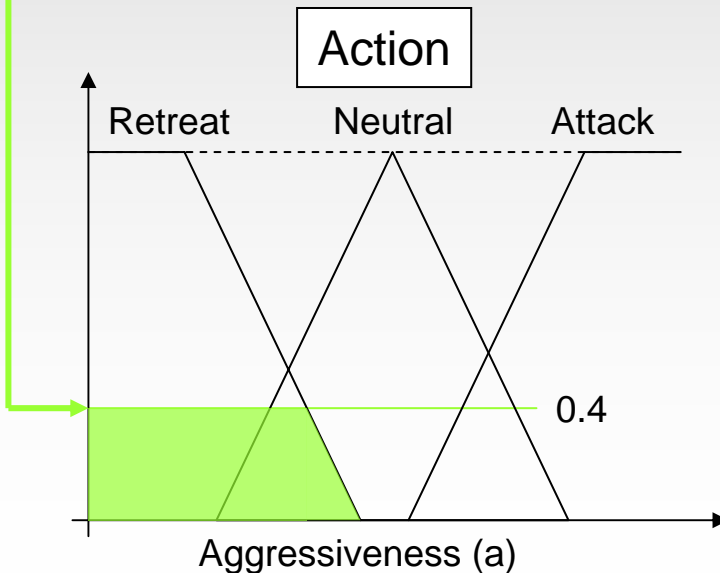


# → Fuzzy Rules

## Fuzzy Rule Set – RULE #3

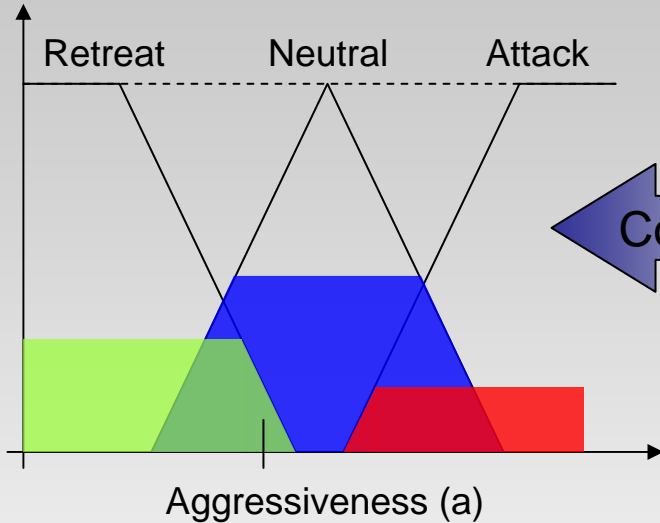
RULE #3:  
 If *enemy IS far* ( $F(x) > 0$ ) AND  
*my health IS Poor* ( $P(x) > 0$ )  
 THEN *retreat* (R)

$F \text{ AND } P = \min(F(d), P(h)) = 0.4$



# → Fuzzy Rules

Fuzzy Rule Set – Output (Action)

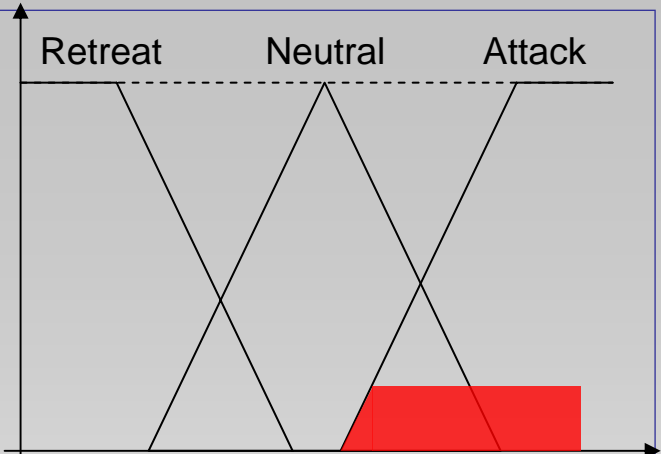


$a = 40$

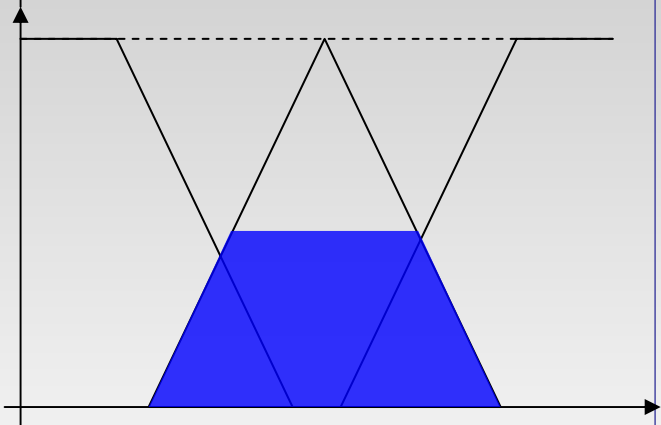
**Defuzzification**  
(Center of gravity of entire fuzzy Set)

Combine Rules

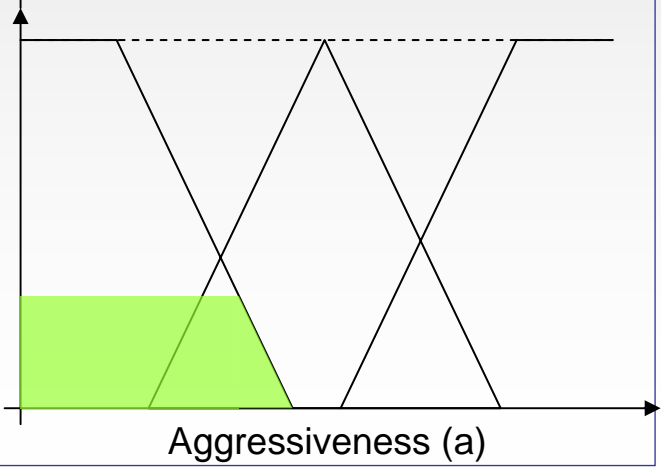
Rule 1



Rule 2

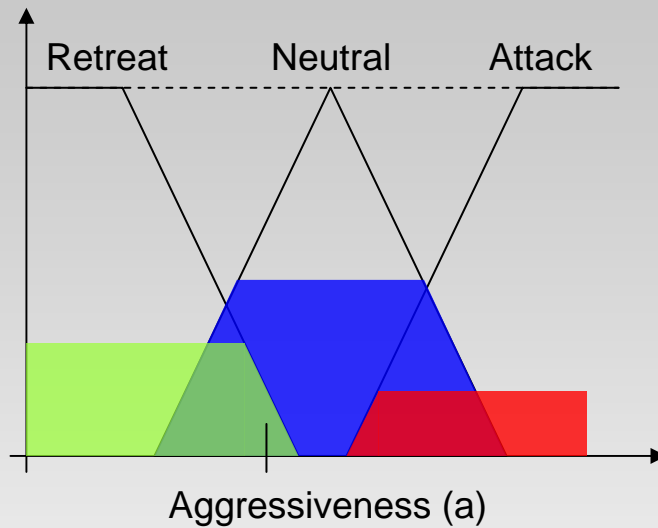


Rule 3



# → Fuzzy Rules

What to do with Output?



- Two main options there...
  - Shift to the corresponding state (Retreat, Roaming, Charging) when its membership degree is higher than a threshold
  - Use defuzzified value (e.g. 40) for an RBS to decide which state to visit next.
    - 0 – 20 → Retreat
    - 45 – 55 → Neutral (Roaming)
    - 80 – 100 → Attack (Charging)

# → Fuzzy Logic

Going further: Creating Emotion

- Fuzzy logic can be used for generating emotional characters
- Characters that respond emotionally to changes in the environment and affect:
  - Their senses (perception)
  - Their abilities (action)
  - Their facial expressions
  - Their physiology

# → Fuzzy Logic

Pros n Cons



- **Quite simple!** (design, implement, visualize, debug)
- **They work well with computer games** (fuzzy-FSMs are quite popular)
- **Used mainly for decision-making**
- **Smooth output**
- **Believable Behaviors!**



- **Not very well-suited for NPC control in games**
- **Expert knowledge required**
- **Behaviors need to be hand-crafted**
- **No flexibility in real-time (no adaptivity)**



# → Conclusion

## Game AI

- AI is here to make games more fun to play
- Use it for generating meaningful intelligence and believability – not for just the sake of it!
- AI boosts game development process
- Try risky/challenging things out!
- Here is your (only?) chance!



# → What's Next?

- Next Tuesday (March 17): **Project Management & Agile Methods** - Lecturer: Søren Lundgaard
- Production Tasks for Next Week
  - Preproduction should be complete
  - Plan production. Reconsider schedule.
  - **Email Aki & Georgios an updated plan, design document and critical stage analysis by Friday, March 20th.**



→ Questions?

Thank You!