# AI

The IT University
of Copenhagen

# Today

1 Production assignment

2 Last part of the course
   9. Jesper
   10. Physics, component, projects
   11. Network games, exam

3 A* path finding

4 Potential functions, Scripted AI, Finite state machines

5 Fuzzy logic

6 Rule-based AI

7 Probabilities and inference



The IT University of Copenhagen

# From Global to local coordinates, done right

Local vector : u = x + y

Global Vector : V = X + Y

Transformation:

$$x = X \cos a + Y \sin a$$

$$y = -X \sin a + Y \cos a$$

(Blackboard drawing)

# A* Path Finding

Add start to open list

While open list is not empty

cur = lowest cost node from open

if cur = goal the done

else

move cur to closed list

for each adjacent node

not on the open list and

not on the closed list and

not an obstacle

move adjacent to open list and calculate cost.

# Is A* optimal?

Go from S to X.

(White=1,Grey=6, Red=13,Black=No go,Heuristic estimate=4)

(White=1+0,Grey=1+5, Red=1+12,Black=No go,Estimated=4)

| | | |
|---|---|---|
| 2 | 3 | 4 |
| S,1 | | X, 5 |
| | | |

| | | |
|---|---|---|
| 6+3*4 | 12+2*4 | 18+1*4 |
| S | | 24 |
| 13+3*4 | | |

Detected path length : 24

Optimal path length : 13+1+1+6=21

# Is A* optimal?

Go from S to X.

(White=1,Grey=6, Red=13,Black=No go,Estimated=1)

(White=1+0,Grey=1+5, Red=1+12,Black=No go,Estimated=4)

| | | |
|---|---|---|
| 2 | 3 | |
| S,1 | | X, 7 |
| 4 | 5 | 6 |

| | | |
|---|---|---|
| 6+3*1 | 12+2*1 | 18+1*1 |
| S | | 21 |
| 13+3*1 | 14+2*1 | 15+1*1 |

Detected path length : 24

Optimal path length : 13+1+1+6=21

# More detailed calculations on A*

Condition for the "red" path being shortest :

4+R+G < 4 + 4 G --> R < 3 G

Conditions for not selecting the minimal path :

1+R+3E>=1+G+3E --> R>=G

1+R+3E>=2+2G+2E --> R+E >=1+2G --> R>=G + (1+G-E)

1+R+3E>=3+3G+E --> R+2E>=2+3G --> R>=G + 2(1+G-E)

1+R+3E>=4+4G --> R>=3+4G-3E --> R>=G + 3(1+G–E)

If G+1 > E then the last condition is most strict : 3+4G-3E

Complete condition for a none optimal path is then :

3G > R >= 3+4G-3E = 3G + (3+G-3E)

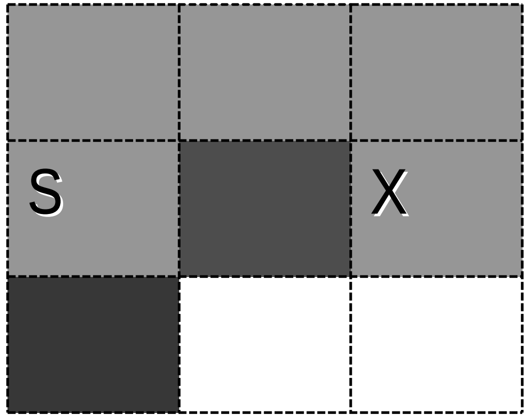
Ex. G=5,E=4,R=12 : Check condition : 15 > 12 >= 11 ;

result : Detected = 24, Optimal = 21

If G+1 <= E then the first condition is the most strict : G

Complete condition for a none optimal path : 3G > R >= G

Ex. G=3, E=5,R=4 : 9 > 4 >=3 Detected =16, Optimal = 11




| | | |
|---|---|---|
| 1+G+ 3E | 2+2G +2E | 3+3G +E |
| | | 4+4G |
| 1+R+ 3E | | |

# A* is optimal if the heuristic is normal distance.

Assume we have two paths P and Q.

The path Q is the optimal path (Cost_Q < Cost_P)

We want to show that we pick the right path (Score_A < Score_C).

| | | |
|---|---|---|
| | B | C |
| | | X |
| A | | |

Cost_to_C = Terrain_P + Dist_P – 1 – Terrain_X

Score_C = Cost_to_C+1 = Terrain_P+Dist_P–Terrain_X

Score_C >= Cost_P > Cost_Q =

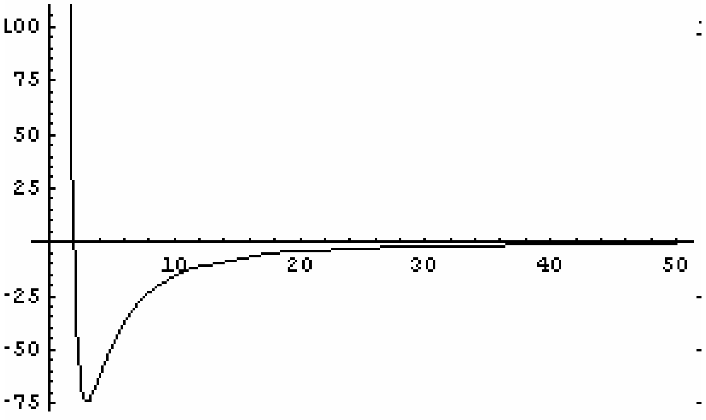
Dist_to_A + Dist_from_A + Terrain_to_A + Terrain_from_A>=

Dist_to_A + Terrain_to_A + Dist_from_A =

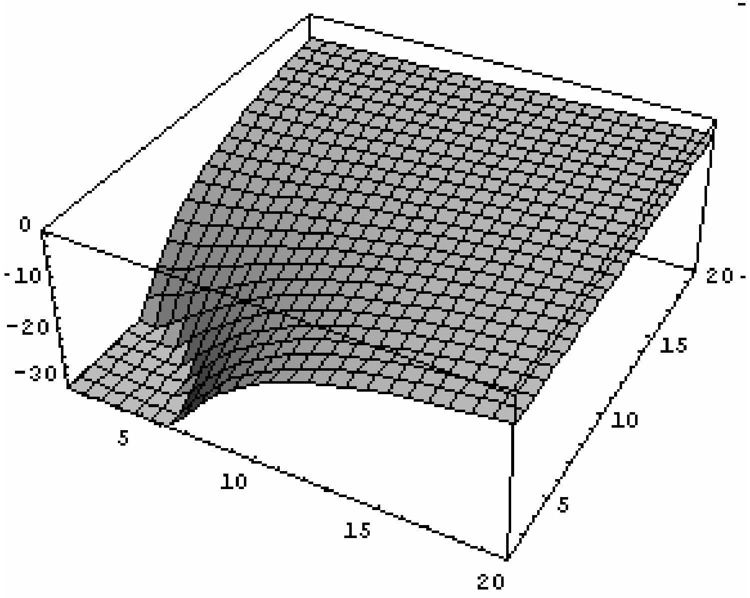
Cost_to_A + Dist_from_A >=

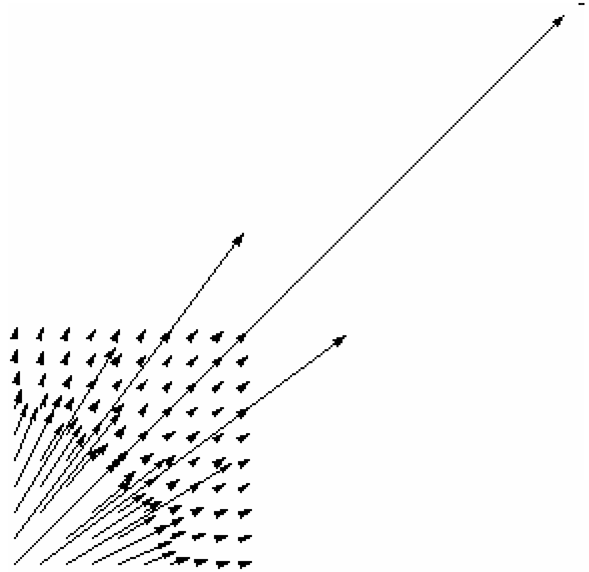
Cost_to_A + Heuristic_dist_from_A = Score_A QED

# Lenard-Jones potential function

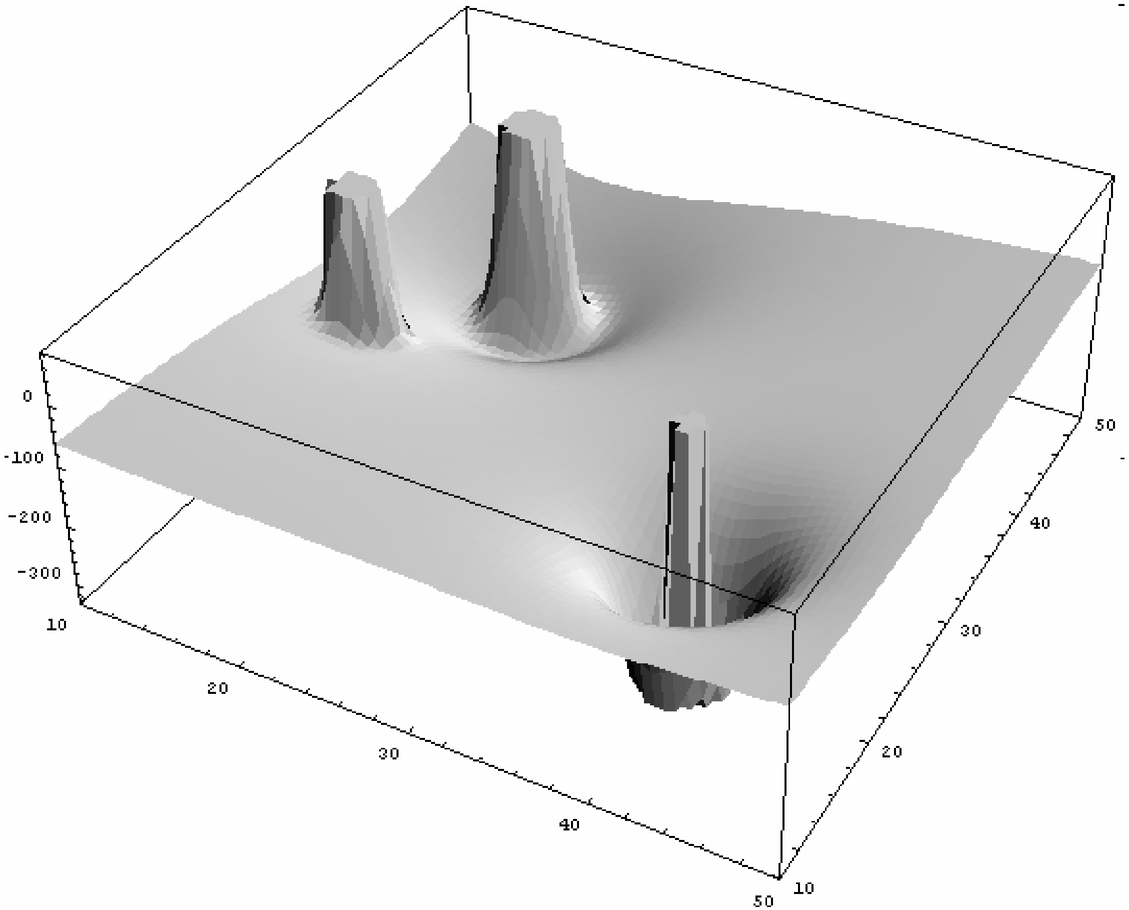



$$\frac{4000}{r^3} - \frac{2000}{r^2}$$



# Combitions of potentials into one




All in one function. Attraction and Repulsion control it all.

Chasing, obstacle avoidence, swarming

# Scripted AI

Scripting attributes

Ex. Intelligence = 80

Scripting behavior (attack/flee, movement pattern)

If (Not(Feeling Lucky) & Punk) then Flee

Verbal interaction

Use state of player to generate dialog.

Use dialog of player to respond appropiate

Scripting Events

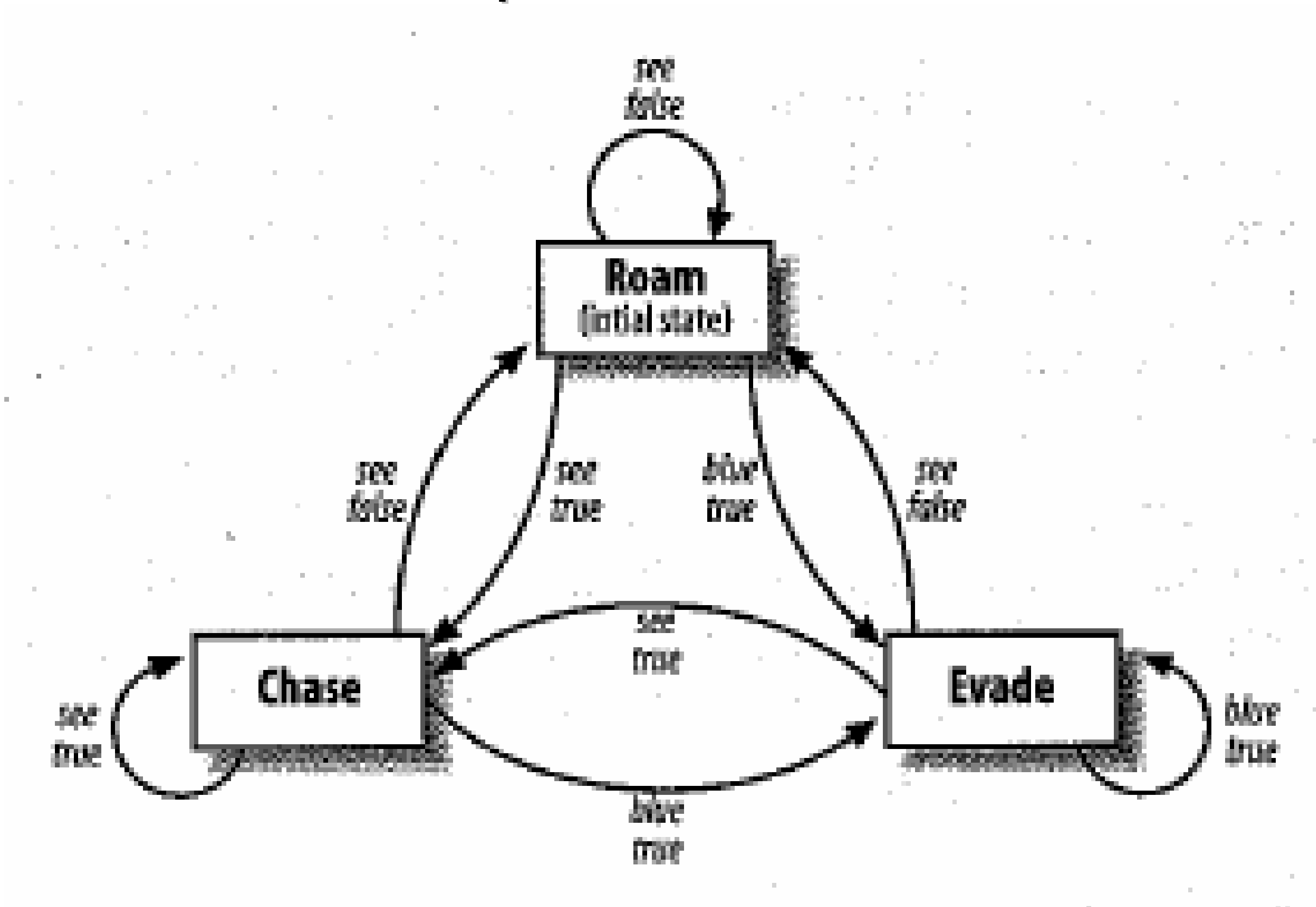
If Position=Sun then Burn

What is a Parser

Script language, spoken language

# Finite State Machines for AI

Ex. Monsters in pacman








The IT University of Copenhagen

# Fuzzy logic, A matter of degree

Smooth parameters instead of booleans.

Allow smooth transition changes.

Allow easy use of catagories (wimpy, easy, moderate, tough)

Fuzzification

Membership/characteristic functions maps variables to a degree of membership in a fuzzy set.

Neighbouring membership function should overlap in their support to allow smooth transitions.

# Fuzzy logic axioms

Disjunction : truth(A or B) = max(truth(A), truth(B))

Conjunction : truth(A and B) = min(truth(A), truth(B))

Negation : truth(not A) = 1 – truth(A)

Attack= ((InRange and Uninjuired) and not HardOpponent) or TreasureIsGood

= max( min(  min(0.8, 0.7),  1-0.4),  0.95)

= 0.95

Attack= ((InRange and Uninjuired) and not HardOpponent) or TreasureIsGood

= max( min(  min(0.7, 0.8),  1-0.5),  0.05)

= 0.5

NB. Alternative definitions for ”not”, ”and” & ”or” exist.

# Fuzzy rule evaluation

Rules are evaluated in parallel (in contrary to serial)

Attack = InRange and Uninjuired or Aggresive

DoNothing = not InRange

Flee = not OutOfRange and not Uninjuired and not Aggresive

Ex.1. Aggresive = 0.85

A = max(min(0.8, 0.3),0.85) = 0.85

D = 1 – 0.8 = 0.2

F = min(min(1 – 0, 1 – 0.3), 1 – 0.85) = 0.15

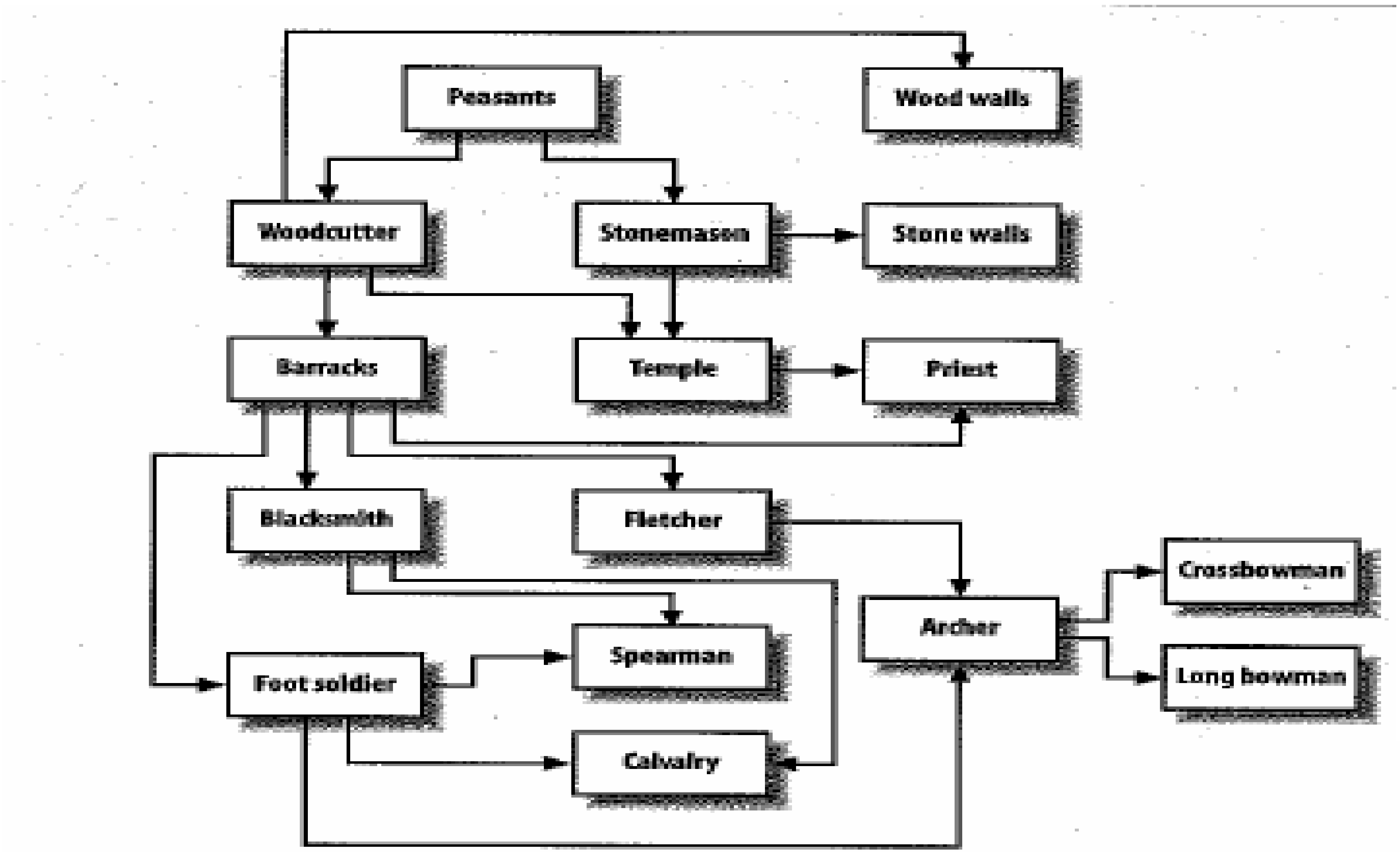
Ex.2. Aggresive = 0.3

A = max(min(0.8, 0.3),0.3) = 0.3

D = 1 – 0.8 = 0.2

F = min(min(1 – 0, 1 – 0.3), 1 – 0.3) = 0.7

# Ruled-based AI, Ex. Tech Tree



# Rule-based AI, Examples

Ex. 1 Technology tree :

Rule 1: If (Woodcutter == True && Stonemason == True && Temple==Unknown)

then Temple=Possible.

Facts : Woodcutter == True, Stonemason == True, Temple==Unknown

Inference : Temple = Possible

Ex. 2 Technology tree :

Rule 1: If (Woodcutter == True && Stonemason == True && Temple==Unknown)

then Temple=Possible.

Facts : Temple = True

In case of only one possible rule

Inference : Woodcutter = True , Stonemason = True

# Ruled-Based AI, Examples

Several applicable rules.

Ex. 2 Technology tree :

Rule 1: If (Woodcutter == True && Stonemason == True && Temple==Unknown)

then Temple=Possible.

Rule 2: If(Magic == True) then Temple =Possible.

Facts : Temple = True

Inference : Woodcutter = Possible , Stonemason = Possible, Magic=Possible

Alternatively maintain several versions:

Woodcutter = True, Stonemason = True or Magic=True





The IT University of Copenhagen

# Rule-Based AI (Expert systems)

Working Memory & Rules Memory

are used to increase content in working memory by inference

Inference is done in three phase loop.

1. Matching. Which rules can be considered based on the facts in working memory.
2. Conflict resolution phase.
   Pick one rule to apply. (First, random, weighted)
3. Firing. Execute the ”then” part

Inference can be done in two ways: Forward and Backward Chaining

# Rule-Based AI, Strike prediction

In the example we accumulate knowledge of the play pattern of the human opponent to improve the computer opponent.

Predict next strike from the two previous ones.

Rules:

1. Weight1
   If PrevPrevStrike == Kick & PrevStrik == Punch then PredictedStrike = Kick
2. Weight2
   If PrevPrevStrike == Kick & PrevStrik == Punch then PredictedStrike = Punch
3. ...

A. Find matching rules and select the rule with the largest weight.

B. Increase weight of the winning rule.

C. If ActualStrike != PredictedStrike then reduced weight of predicting rule

# Probality and randomness

Randomness makes the game harder to predict but expected events should be more likely. Hence things should happen with different probality.

Ex. Moves, Hit spread, Abilities,State transitions

Ex.Adaptability

Randomness using computers is generated in the form of *pseudo-random number generators.*

Pseudo-random numbers have the characteristic that they are *predictable,* meaning they can be predicted if you know where in the sequence the first number is taken from.

# Probality

## Classical probality.

An event E can occur in n ways out of N possible outcomes with a probility $p = P(E) = n/N$ .

Probalities has value between 0 and 1.

The sum of probalities over all possible events is 1.

## Frequency interpretation

Experiments are conducted N times and some event E occurs n times.
The probability of E occuring is $P(E) = n/N$ as N goes to infinity.

Ex. A coin is tossed 1000 times with heads coming up 510 times

P(heads) = 510/1000

# Expectation Value

The expected value of some discrete random variable X, that can take values X0, X1, ... Xn with probabilities p0, p1, ..., pn.

E(X) = X0 p0 + X1 p1 + ... + Xn pn

Ex. Expected number of players at the inn.

| Nr | Probability |
|---|---|
| 6 | 0.18 |
| 5 | 0.32 |
| 4 | 0.27 |
| 3 | 0.09 |
| 2 | 0.09 |
| 1 | 0.05 |

Expected value is

4,27

# Probability Rules

1. A probability is between 0 and 1
2. The entire set S of events occurs with probability 1
3. If E occur with P(E) then the probability of not E is 1-P(E)
4. Mutually exclusive events can only occur one at a time (dead or alive)
   $P(A \cup B) = P(A) + P(B)$
5. Non mutually exclusive may or may not occur simultaneously.
   $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
6. Events are independent if one event do not depend on the occurence of the other.
   $P(A \cap B) = P(A)\,P(B)$
7. Conditional probability . Interdependency of events
   $P(A \cap B) = P(A)\,P(B|A)$

# Bayes' rule

We know A happened what is the probability that B will happen?

$P(B|A) = P(A \cap B)/P(A) = P(B)\ P(A|B)/P(A)$

Because the conditional probability is

$P(A \cap B) = P(A)\ P(B|A) = P(B)\ P(A|B)$

The probability of an event happens equals the sum of all possible conditional probabilities times their corresponding prior.

$P(A) = P(A|B)\ P(B) + P(A|\sim B)\ P(\sim B)$

Ex. P(Snow) = P(Snow|zero or below) P(zero or below) + P(Snow|above zero) P(above zero)

P(Snow) = 0.6 0.1 + 0.01 (1-0.1) = 0.54 + 0.09 = 0.63

# Treasure hunting

A chest can:

be locked or not
be trapped or not
have money or not

We can see if it is locked but not if it is trapped.

Experience :

Out of 100 opened chests 50 contained money. (P(M) = 0.5)

Out of these 50, 40 were trapped. (P(T|M) = 0.8)

Out of these 40, 28 were locked.(P(L|T) = 0.7)

Of the 10 untrapped chest, 3 were locked. (P(L|~T) = 0.3)

Of the 50 chests without money, 20 were trapped. (P(T|~M) = 0.4)

# The treasure hunt continues.

| | | | |
|---|---|---|---|
| P(M) = 0.5 | P(~M) = 0.5 | P(T\|M) = 0.8 | P(~T\|M) = 0.2 |
| P(T\|~M)=0.4 | P(~T\|~M)=0.6 | P(L\|T) = 0.7 | P(~L\|T) = 0.3 |
| P(L\|~T) = 0.3 | P(~L\|~T) = 0.7 | | |

What is the probability of a trapped chest?

P(T) = P(T|M) P(M) + P(T|~M) P(~M) = 0.8 0.5 + 0.4 0.5 = 0.6

Is it trapped if it is locked?

P(L) = P(L|T) P(T) + P(L|~T) P(~T) = 0.7 0.6 + 0.3 0.4 = 0.54

P(T|L) = P(L|T) P(T)/P(L) = 0.7 0.6/P(L) = 0.42/0.54 = 0.78

Is it trapped if it is not locked?

P(T|~L) = P(~L|T) P(T)/P(~L) = 0.3 0.6/0.46 =  0.39

What is the probability of Money in a locked chest?

P(L|M) = P(L|T) P(T|M) + P(L|~T) P(~T|M) = 0.7 0.8 + 0.3 0.2 = 0.62

P(M|L) = P(L|M) P(M)/P(L) = 0.62 0.5/0.54 = 0.57

What is the probality of Money in a unlocked chest?

P(~L|M) = P(~L|T) P(T|M) + P(~L|~T) P(~T|M) = 0.3 0.8 + 0.7 0.2 = 0.38

P(M|~L) = P(~L|M) P(M)/P(~L) =  0.38 0.5/0.46 = 0.41