
+12
Income:
£124.66 Bn
Expenditure:
-£129.55 Bn
Deficit:
£4.90 Bn
Debt:
£450.47 Bn
Socialist
Capitalist
Retired
Commuter
Patriot
Motorist
Liberal
Religious
Trade Unionist
Self Employed
Environmentalist
Wealthy
Poor
Middle Income
Parents
Farmers
State Employees
Conservatives
Youth
Ethnic Minorities
Everyone
Popularity:
GDP
Games AI
Lecture 3.1
Utility-based AI

- Utility-based AI
  - Utility theory
  - Decision factors
  - Designing response curves
  - Implementing utility-based AI

- Utility theory
  - General model of **decision making**
  - **Every possible action** can be scored on a single scale of "goodness" (utility, weight, score, priority, ...)
  - Agents act rationally to **maximise utility**

    "Irrationality" / cognitive biases / personality traits can be added into the model as factors

- Utility != value
  - Value is a measurable quantity, eg. price
  - Utility is how much you desire something
  - Money has fixed value but variable utility depending on context
    - eg. £20 on a desert island is worthles
  - Utility is abstract, but it can be based on values

- Principle of Maximum Expected Utility
  - Multiply utility of each outcome (D) by probability of it occouring (P)
  - e.g. attack in D&D
    - Critical hit: 5%, utility: 0.9
      = 0.045
    - Regular hit: 90%, utility: 0.6
      = 0.56
    - Fumble: 5%, utility:-1
      = -0.05

    .: expected utility of attack = 0.555

$$EU = \sum_{i=1}^{n} D_i P_i$$

- Utility-based AI basics
  - Every action is scored at once using a mathematical model of the value (utility) of each action
  - Top scoring actions change as scores fluctuate dynamically
  - One of top scoring actions is chosen

- Pros
  - More nuanced behaviours than using booleans
  - Hand authored → Tight authorial control
  - Simple to implement
  - Considerations are reusable between agents
  - Considerations reflect decision making process - can be exposed to the player
- Cons
  - Designing behaviours can be a “dark art”
  - No planning or look-ahead


Making Decisions
$580.40
941 Guests
March, Year 3
59F

- Decision factors
  - Multiple pieces of data weigh into any given decision
    - Health, Ammo, Distance to enemy
    - Wealth, hunger, enjoyment of rollercoasters
    - Conversation choices, loyalty, distance to allies
  - Almost anything can be a decision factor
    - Cooldown, repeat penalty, aggressiveness, conscientiousness, ...

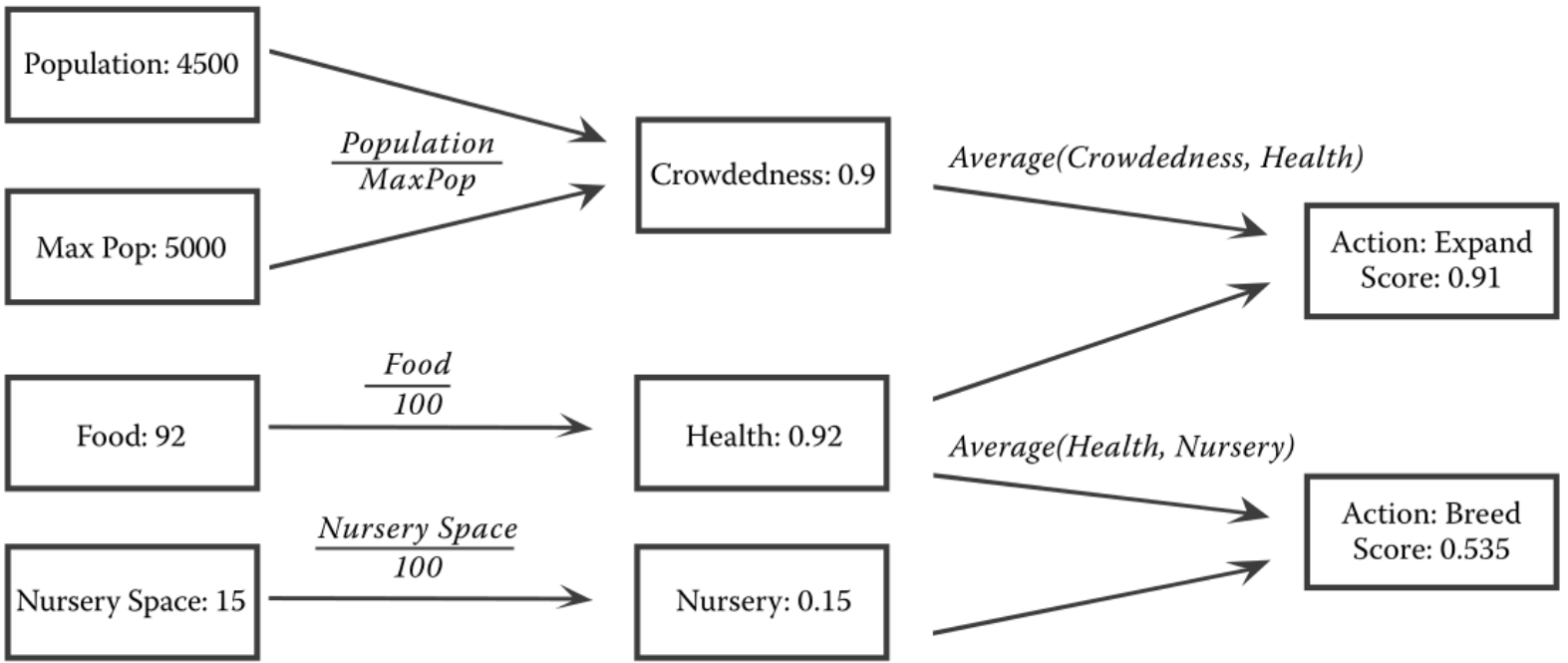
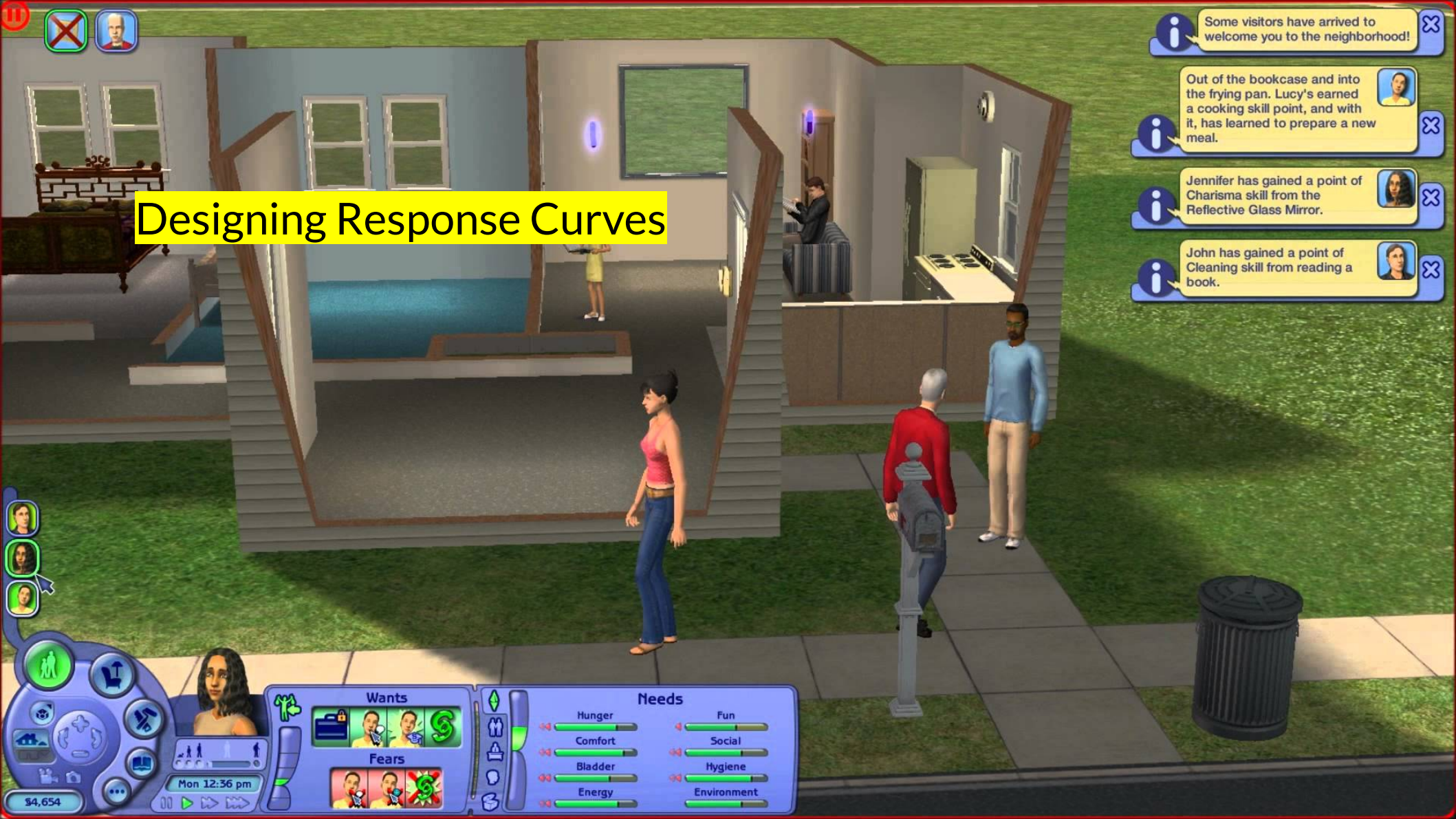


Figure 9.1

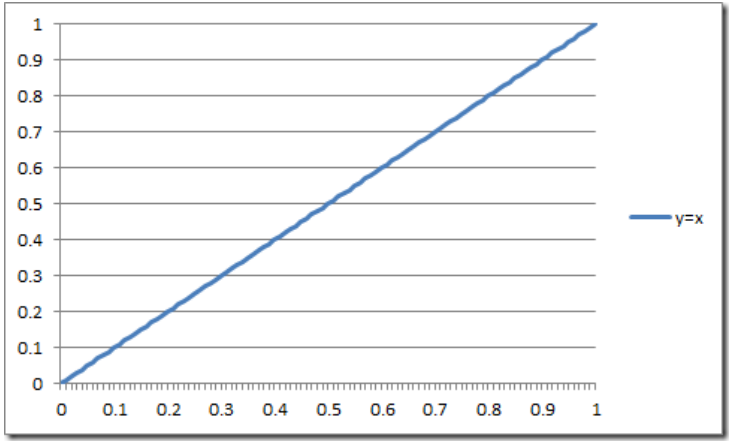
An example of combining utility scores from different decision factors to arrive at a final score.

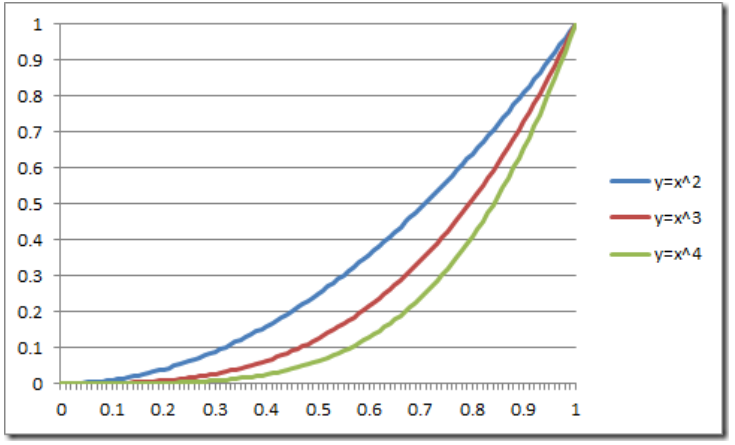
- Take with highest score (absolute utility)
  - Take “best” decision
  - Rigid, predictable
- Select an option at random, but weight probabilities (relative utility)
  - Variation, interesting
  - Periodically will do something that makes your AI look stupid
- Combine the two for the best results

- Decision factors
  - Art of design is mostly selecting considerations to weight actions and pairing with appropriate response curves
  - By averaging normalised scores can build endless chain of combinarions


Designing Response Curves
Some visitors have arrived to welcome you to the neighborhood!
Out of the bookcase and into the frying pan. Lucy's earned a cooking skill point, and with it, has learned to prepare a new meal.
Jennifer has gained a point of Charisma skill from the Reflective Glass Mirror.
John has gained a point of Cleaning skill from reading a book.
Wants
Fears
Needs
Hunger
Fun
Comfort
Social
Bladder
Hygiene
Energy
Environment
Mon 12:36 pm
$4,654

- What is a response curve?
  - A mathematical representation of a relationship modelling how a change in the input affects the output



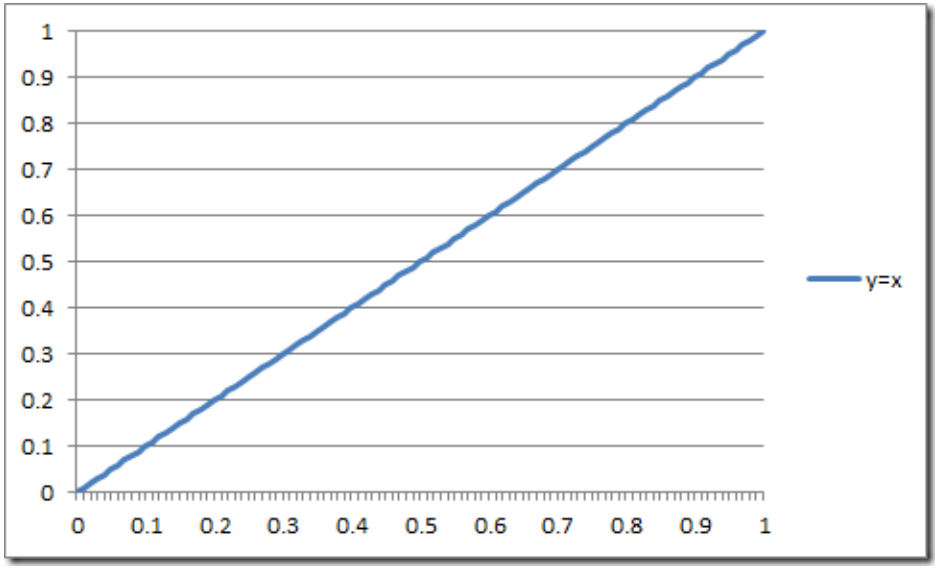


- A response curve is a function with domain and range in [0-1]*
  - It maps some normalised input value (eg. health %) to a consideration for our AI
    - eg. health% → desireToHeal
    - eg. ammo% → desireToReload
  - Useful properties:
    - Multiplying curves always gives result in [0-1]
    - Multiplying by 0 always gives 0
    - Can combine any number of curves
    - Outputs of one curve can be inputs to another

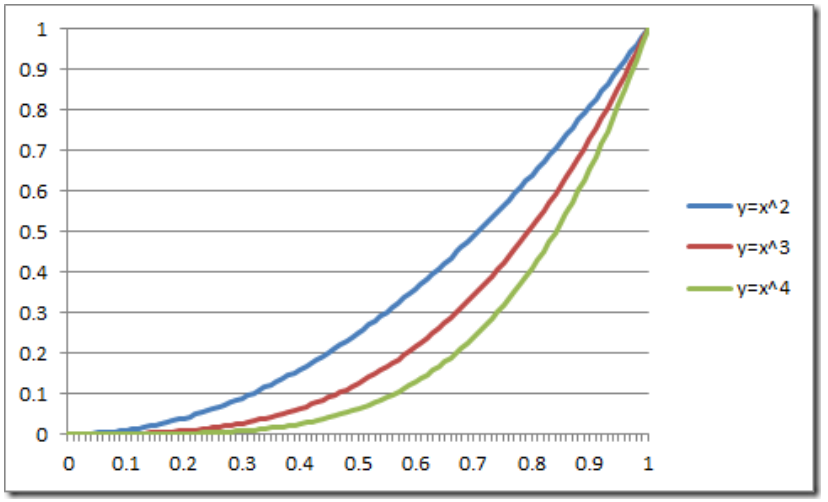
*or another consistent range, but mathematical properties may change

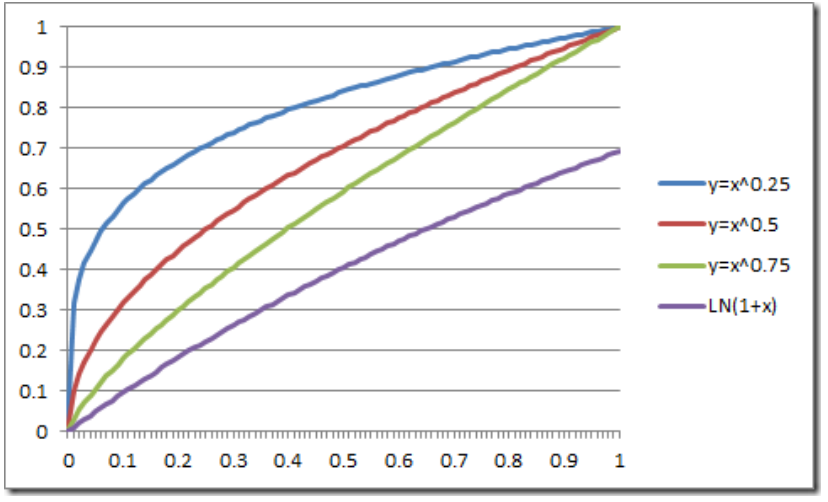
- Four types of curve
  - Linear/Quadratic
  - Exponential
  - Logistic
  - Logit

- Linear
  - **y=mx + c**
  - Change in *x* always causes the same change in *y*

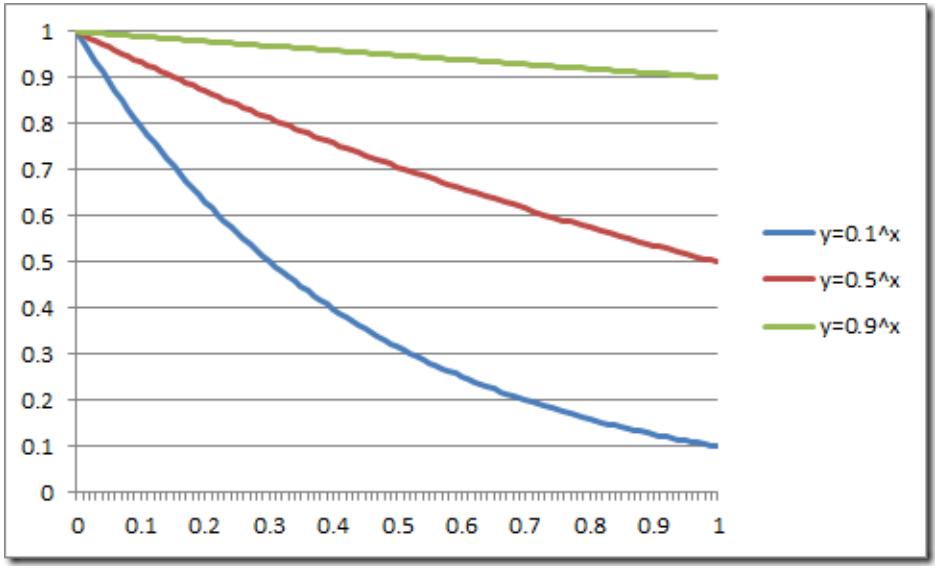


- Quadratic
  - **$y=x^a$**
  - a > 1 increasing gradient
  - 0 < a < 1 decreasing gradient
  - Changes in *x* lead to bigger/smaller changes in *y*

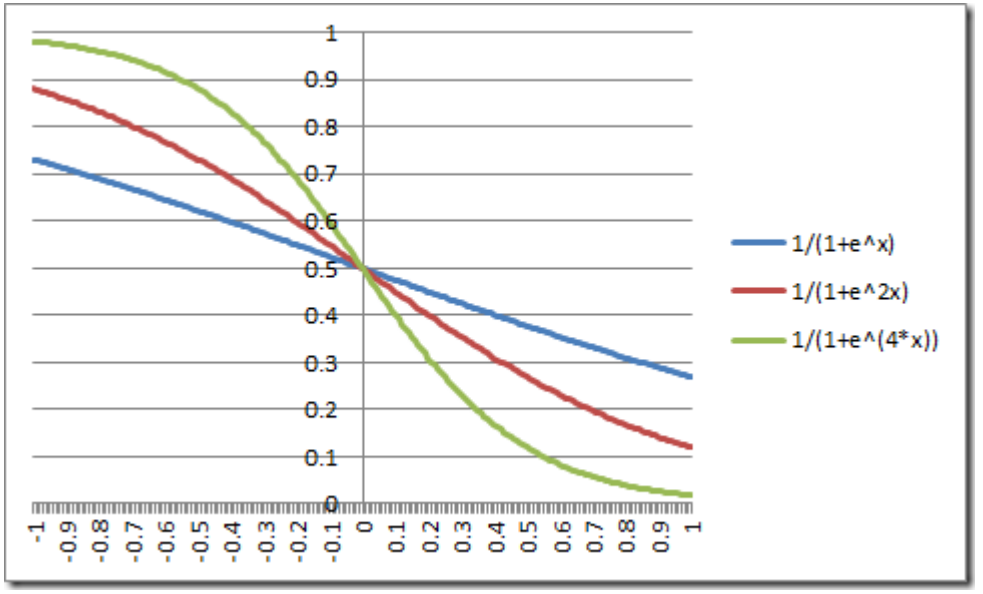




- Exponential
  - **$y=a^x$ -1**: exponential growth
  - **$y=a^x$**, where $0 < a < 1$: exponential decay

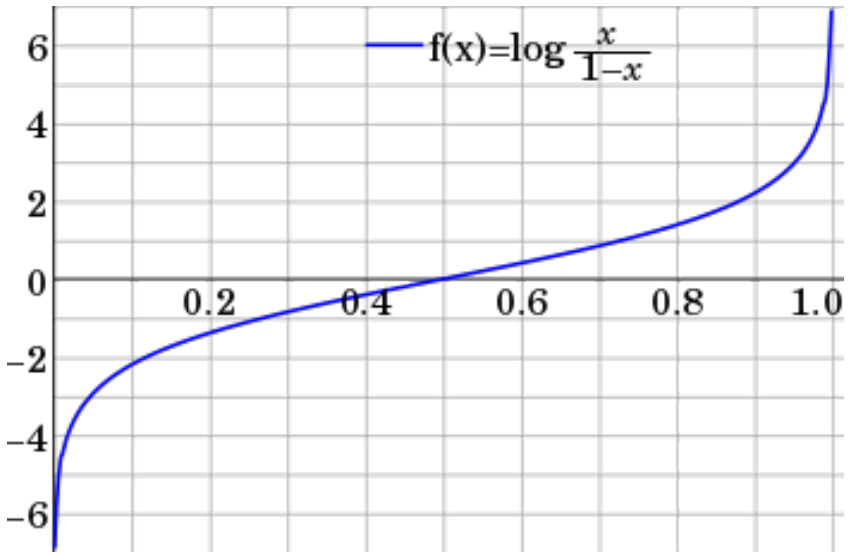


- Logistic
  - **$y = 1/(1+e^{x})$**
  - **$y = 1/(1+e^{-x})$** for reverse
  - “Soft threshold” giving a more or less sudden change between low and high



https://www.desmos.com/calculator/j7ucn0rkfc

- Logit curve
  - **y = log (x / 1-x)**
  - Differences at the extremes are more significant than those in the middle



https://www.desmos.com/calculator/q9hkfpqc2s

- Can customise your own functions using these mathematical components
  - Need to adjust to fit within your scale (e.g. 0-1)
  - Clamp if necessary
- Much easier:
  - Paramatise these four basic function types

- These four curve type can all be paramatised: b,c,k,m
  - Linear/Quadratic
    - $f(x) = m(x-c)^k + b$
  - Exponential
    - f(x)
  - Sigmoid/Logistic
    - $f(x) = (k/(1+1000em^{-1x+c})) + b$
  - Logit
    - $f(x) = 0.5 \log_{100}m ((x/k - c)/1-(x/k - c)) + b + ...?$

- Linear/Quadratic
    - $f(x) = m(x-c)^k + b$
  - m = slope
  - k = how much bend
  - b = vertical shift
  - c = horizontal shift

- Logistic
  - $f(x) = (k/(1+1000em^{-1x+c})) + b$
  - m = slope at inflection point
  - k = vertical size of curve
  - b = vertical shift
  - c = horizontal shift


Implementing Utility-based AI (1)
Infinite Axis Utility System Architecture

- Each response curve can be defined by an enum and four floats:
  - [{Quadratic, Exponential, Logistic, Logit}, m, k ,b ,c]

Curve type	m	k	b	c
Quadriatic	0.5	0.1	0.25	0.0
Logistic	50,	-0.95	1.0	0.6

- Processing a reponse curve
  - Function takes paramaised curve and in input *x*
    - (type, m, k ,b ,c, x)
  - Clamp input to [0-1]
  - Apply function determined by curve type
  - Clamp output to [0-1]
  - Return *y*

- Processing a consideration (decision factor)
  - Each consideration has an array of inputs
    - Inputs can come from other considerations or from a “Clearing House” function that returns normalised input values (eg. health, ammo)
  - Each input is assigned a paramatised utility function
  - The outputs of these functions are multiplied together
  - The result is the utility of the consideration

- Processing an action
  - Each action has an array of considerations
  - The utility of each consideration is calculated and multiplied together
  - The result is then multiplied by a weight, e.g.
    - Idle actions: x1
    - Attack actions: x3
    - Scripted actions: x5
    - Combat responses: x10

- Calculate the score of each action
  - Score actions with targets on a per-target basis
- Push scored actions onto a list of possible actions
- Pick (one of the) top scoring actions from the list

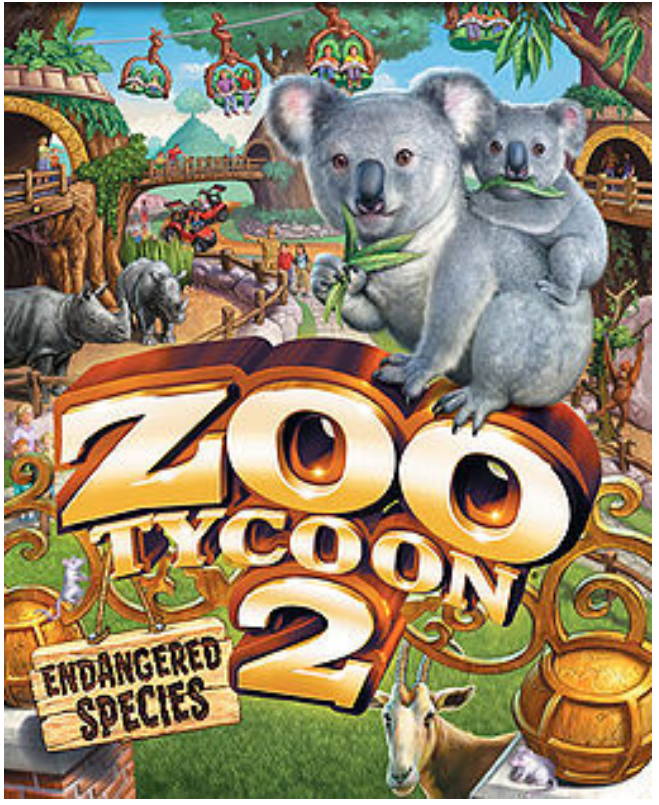
Implementing Utility-based AI (2)

Dual-utility Reasoning

- Each consideration has
  - A utility
  - A rank
- Rank of action is maximum rank of any consideration
  - i.e. considerations can force overall action to be of high rank
- Weight of action determined by the product of utility of considerations

- Dual-utility reasoning
  - 1. Eliminate the options with a weight of zero.
  - 2. Find the highest rank category, and eliminate options that don't belong to it.
  - 3. Find the best remaining option (i.e., the one with the highest weight), and eliminate options that are much worse (i.e., much lower weight) than it.
  - 4. Use weight-based random to select from the options that remain
- Kevin Dill. *Dual-Utility Reasoning* in Game AI Pro 2, ch 3

- Dual-utility reasoning eg.
  - Normal needs rank 0
    - Hunger, entertainment, bathroom
  - Context-specific behaviours higher ranks
    - e.g. only tree climbing behaviours when climbing trees
  - Death rank of 1,000,000



- David “Rez” Graham. *An Introduction to Utility Theory* in Game AI Pro. Ch.9
  - http://www.gameaipro.com/
  - Also several other chapters in the Game AI Pro books about utility systems
- Dave Mark’s Infinite Axis Utility System Architecture at GDC
  - http://www.gdcvault.com/play/1018040/Architecture-Tricks-Managing-Behaviors-in (third speaker)
- ...and more of his GDC talks about Utility Systems:
  - http://intrinsicalgorithm.com/IAonAI/2013/02/both-my-gdc-lectures-on-utility-theory-free-on-gdc-vault/
- Kevin Dill *Dual-Utility Reasoning* in *Game AI Pro 2*. Ch.3
- www.desmos.com Free online graphing tool