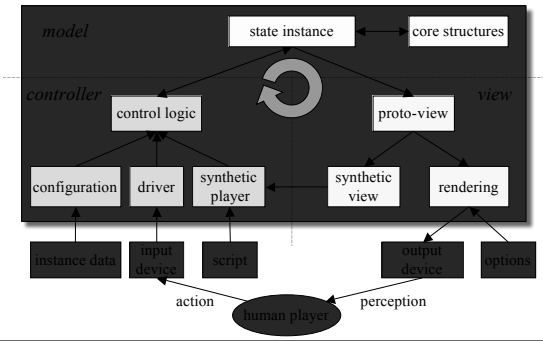
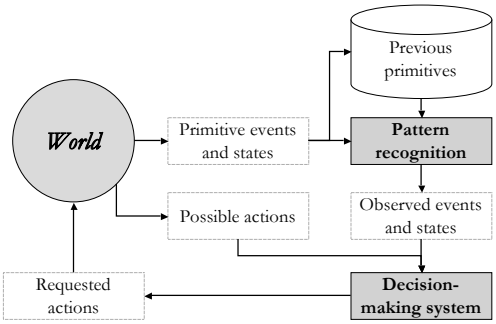
§6 Decision-Making

- decision-making and games
  - levels of decision-making
  - modelled knowledge
  - method
- example methods
  - finite state machines
  - flocking algorithms
  - influence maps
- this will not be a comprehensive guide into decision-making!

MVC (revisited)



Decision-making system

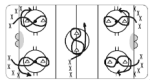


Three perspectives for decision-making in computer games

- level of decision-making
  - strategic, tactical, operational
- use of the modelled knowledge
  - prediction, production
- methods
  - optimization, adaptation

Level of decision-making

- strategic
  - what should be done
- tactical
  - how to actuate it
- operational
  - how to carry it out



Strategic level

- long-term decisions
  - infrequent → can be computed offline or in the background
- large amount of data, which is filtered to bring forth the essentials
  - quantization problem?
- speculative (what-if scenarios)
- the cost of a wrong decision is high

Tactical level

- medium-term decisions
- intermediary between strategic and operational levels
  - follow the plan made on the strategic level
  - convey the feedback from the operational level
- considers a group of entities
  - a selected set of data to be scrutinized
  - co-operation within the group

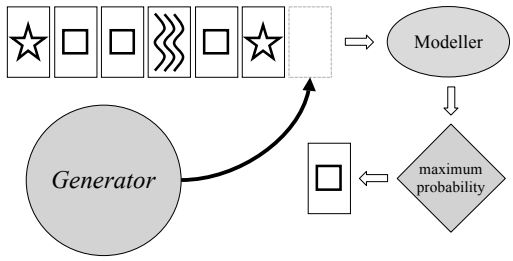
Operational level

- short-term decisions
  - reactive, real-time response
- concrete and closely connected to the game world
- considers individual entities
- the cost of a wrong decision is relatively low
  - of course not to the entity itself

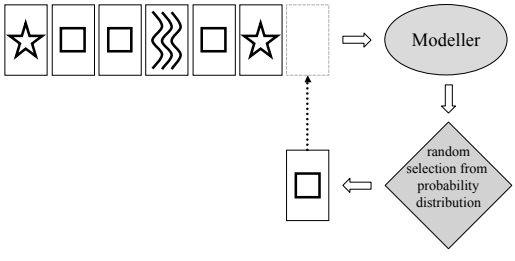
Use of the modelled knowledge

- time series data
- world = a generator of events and states, which can be labelled with symbols
- prediction
  - what the generator will produce next?
- production
  - simulating the output of the generator
- how to cope with uncertainty?

Prediction



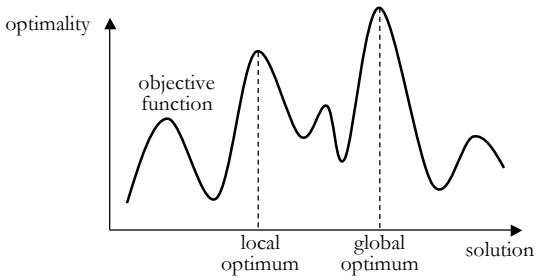
Production



Decision-making methods

- optimization
  - find an optimal solution for a given objective function
  - affecting factors can be modelled
- adaption
  - find a function behind the given solutions
  - affecting factors are unknown or dynamic

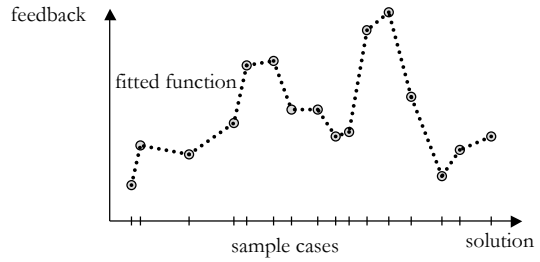
Optimization



Optimization methods

- hill-climbing
  - how to escape local optima?
- tabu search
- simulated annealing
- genetic algorithms
  - multiple search traces
- swarm algorithms

Adaptation



Adaptation methods

- neural networks
  - training
    - supervised learning
    - unsupervised learning (e.g., self-organizing maps)
  - execution
- hidden Markov model
  - recurring structures

Finite state machine (FSM)

- components:
  - states
  - transitions
  - events
  - actions
- state chart: fully connected directed graph
  - vertices = states
  - edges = transitions

Properties of FSM

1. acceptor
   - does the input sequence fulfil given criteria?
2. transducer
   - what is the corresponding output sequence for a given input sequence?
3. computator
   - what is the sequence of actions for a given input sequence?

- these properties are independent!

Mealy and Moore machines

- theoretical cathegories for FSMs
- Mealy machine
  - actions are in transitions
  - the next action is determined by the current state and the occurring event
  - more compact but harder to comprehend
- Moore machine
  - actions are in states
  - the next action is determined by the next state
- helps to understand and use state machines in UML

Implementation

- design by contract
  - two parties: the supplier and the client
  - formal agreement using interfaces
- FSM software components
  - environment: view to the FSM (client)
  - context: handles the dynamic aspects of the FSM (supplier)
  - structure: maintains the representation of the FSM (supplier)

Noteworthy

- structure is static
  - hard to modify
- reactivity
  - memoryless representation of all possible walks from the initial state
- states are mutually exclusive: one state at a time
  - not for continuous or multivalued values
- combinatorial explosion
  - if the states and events are independent
- risk of total rewriting
  - high cohesion of actions