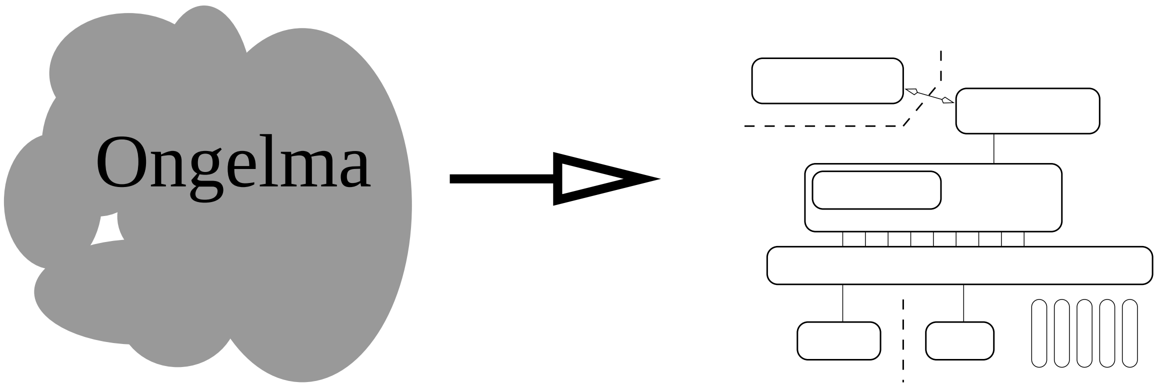
Suunnittelu



- Moduli- ja alihakemistoluetteloita
- Kutsukaavioita, tietovuokaavioita
- Tietorakenteiden kuvauksia
- Tilakoneita, protokollia
- Oliohierarkioita
- Ongelman kuvausta / määrittelyä

Hyvän suunnitelman tunnusmerkkejä

- Yksinkertaisuus
- Helppo muistaa

- Osien geneerisyys, uudelleenkäytettävyys
- Redundantin työn välttäminen

- Jokainen osa sopivan kokoinen ja toteutettavissa (“No loose ends”).

- Myöhempi laajennettavuus ja muuteltavuus

- Toteutuksen ja testauksen rinnakkaisuus

Suunnittelun keinoja

- Top-down
- Bottom-up
- Protoilu
- Suunnittelutavoitekohtaisia menetelmiä.

CASE Minpantu: Suunnittelun suunnittelu, mikä on epäselvää

- Teknisesti haastavia ongelmia:
  - Kuinka tietää miten pallot liikkuvat?
  - Kuinka päätellä miten tietokoneen tulee tehdä siirtonsa?
- $\Longrightarrow$ alan kirjallisuuden tutkimista,
- $\Longrightarrow$ löytyy joitakin algoritmeja, kehitellään itse puuttuvia osia.

- Teknisesti helppoja, mutta helposti työläitä ongelmia?
  - Kuinka tehdään käyttöliittymä, millä työkalulla?
  - Mitä toimintoja käyttöliittymässä, mitä ne tekevät ohjelman sisällä?

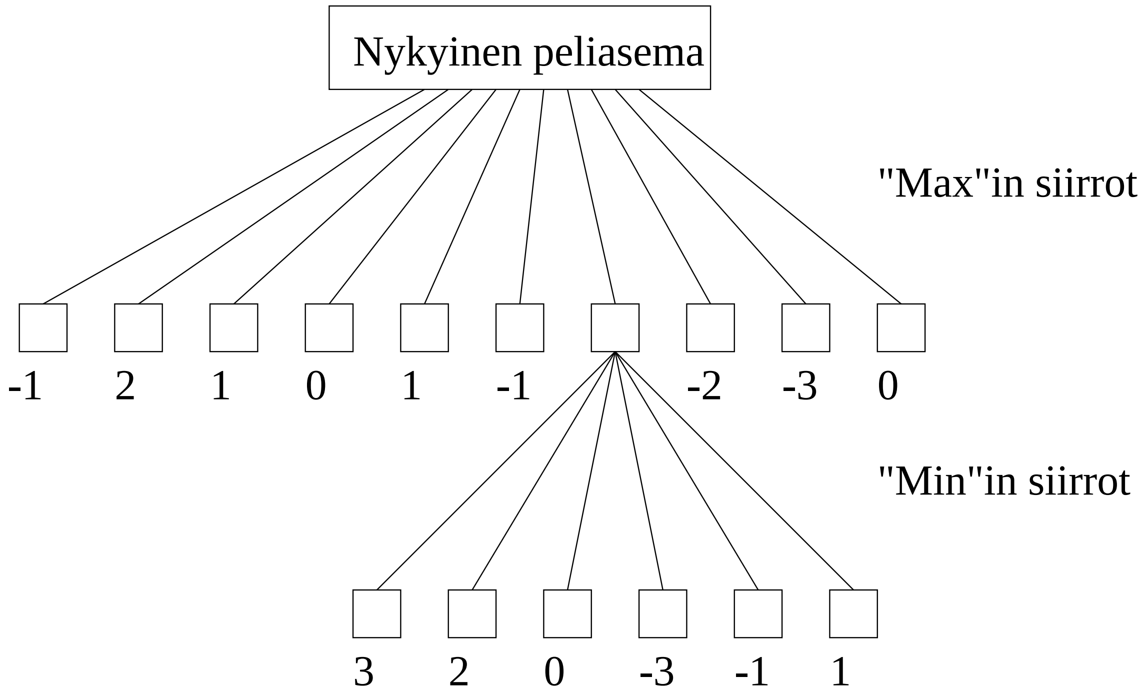
CASE Minpantu: Pallojen liike

- Pallon liikkeitä kuvaa hankala differentiaaliyhtälöryhmä, ei matemaattisesti suljettua ratkaisua.
- $\Longrightarrow$ Miten lasketaan numeerisesti differentiaaliyhtälöiden likiarvoratkaisuja?
- $\Longrightarrow$ Kirjallisuushakua
- $\Longrightarrow$ Löytyy tukku menetelmiä, mm. Runge-Kutta, Bulirsch-Stoer, Adams, PECE, jne. Valitaan jokin.

- Protoillaan:
  - Määritellään miten pelialueen asemaa kuvataan
  - Tehdään funktio, joka laskee aseman derivaatan.
  - Toteutetaan yksinkertainen Runge-Kutta.
  - Pieni kokeilullinen käyttöliittymä, joka osaa vain näyttää pallojen paikat.

CASE Minpantu: Tietokoneen siirto

- Lähtökohta:
  - Tietokone osaa laskea pallojen liikkeitä
  - Tietokone osaa kokeilla mielessään erilaisia vaihtoehtoja.
  - Vastustajakin siirtää vuorollaan $\Longrightarrow$ min-max–algoritmi tekoälytutkimuksesta

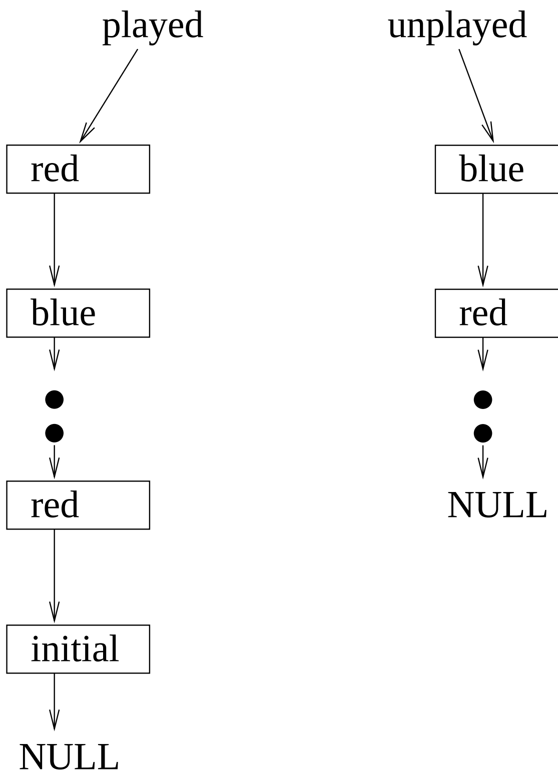


- Protoillaan:
  - Kokeillaan erilaisia asemallisia heuristiikkoja
  - Haku on hidas $\Longrightarrow$
    * Lisätään alfa-beta–algoritmi minimaxin päälle.
    * Lisätään killer-heuristiikka.
    * Profilointi, muutama “käsin”optimointi.
    * Nopeutetaan pallojen ratojen laskentaa jättämällä joitakin voimia pois laskennan myöhäisistä vaiheista.

CASE Minpantu: Käyttöliittymä

- Käyttöliittymä kirjoitettu TclTk:lla.

- Kuinka järjestetään pelatun pelin katselu, siirtyminen aikaisemmaksi ja myöhemmäksi?



- Tarkkaa pohdiskelua siitä, mitä mikäkin nappi tekee, milloin aktiivinen jne.

CASE Minpantu: Sorsat

- `http://www.cs.hut.fi/~cessu/minpantu.html`
- Alphat ja HP:t

  ```
  use tcl
  use tk
  ~cessu/minpantu-0.1-{hp,alpha}/minpantu
  ```

- Noin 1600 riviä C-koodia, 600-riviä Tcl-koodia.