please add a print sized

COVER IMAGE

Diplomski rad

Uenje strategije igranja raunalnih igara

Joel Mislav Kunst

Contents

Tema

Raunalne igre predstavljaju zanimljiv poligon za ispitivanje niza algoritama umjetne inteligencije. Raunalnih igara postoji mnotvo vrsta (kartake, na ploi, strategije u stvarnom vremenu, arkadne, . . .). Igra pri tome moe imati potpunu informaciju (poput aha) ili nepotpunu informaciju (poput pokera).

U okviru ovog diplomskog rada potrebno je za odabranu uu porodicu raunalnih igara definirati unificirani strojno itljiv nain opisa pravila tih igara te na temelju tog opisa razviti programsku implementaciju sustava koji samostalno ui nove strategije igranja takvih igara. Radu je potrebno priloiti algoritme, izvorne kodove i rezultate uz potrebna objanjenja i dokumentaciju. Citirati koritenu literaturu i navesti dobivenu pomo.

Chapter 1

Uvod

Igre su bile dosta popularne ve godinama, a popularne su danas sve vie i vie. Osobe raznih dobi, vjerovanja, stajalista, karakteristika igraju igre. Takoer postoje razne vrste igara, a veina njih zahtjeva nekakvo strateko razmiljanje. Sa stajalita umjetne inteligencije zanimljivo je pogledati kako raunalo moe sudjelovati u tom svijetu igara. Kako moe uiti te igrati te igre.

Postoje brojni primjeri igranja igara pomou umjetne inteligencije. Popularni i poznati primjeri su igra ah u kojoj je raunalo pobjedilo velemajstora iste igre (Poznat je Deep blue kao prvo raunalo koje je to postiglo). Nedavno je Googlova UI pobjedila svjetskog majstora u igri GO.

Koliko god neke igre bile sline, razlika meu igrama zahtjeva dosta individualan pristup kod razvoja umjetne inteligencije za neku od igara. U ovom radu pokuao sam pronai nekakav patern, neto zajedniko izmeu igara te iskoristi to za razvoj openite umjetne inteligencije koja moe igrati vie igara na temelju opisa pravila igre. Nakon mnogo istraivanja o trenutnim zanimacijama za temu, te trenutnim postignuima, nisam pronaao niti jedan primjer sustava koji omoguava jednostavan opis raznih vrsta igara. ak i naizgled male razlike meu srodnim igrama bitno kompliciraju opis pravila igara do te razine, da je puno jednostavnije iskoristiti postojee alate, programske jezike direktno kako bi se opisala neka igra. Tu ve uoavamo veliku kompleksnost a nismo niti poeli razmiljati o tome kako e sustav uiti iz tih opisa, te uspjeno igrati opisane igre.

Radi navedenih razloga odluio sam smanjiti opseg, na uu skupinu igara. No ak i nakon par suavanja opseg je bio pre velik, te sam suavao dok nisam

ostao na jako uskom i specificiranom skupu igara, koje su meusobno vrlo sline, tek imaju minimalne jedinstvene koncepte koji ne postoje meusobno u ostalim igrama. Skupina za koju sam se odluio su online preglednike strateske igre u stvarnom vremenu. Radi se o igrama koje se igraju u internet preglednicima izmeu tisue i vie igraa. Igre su strateke, te se odvijaju u stvarnom vremenu, to znai da ne postoje potezi/redovi kojima igrai igraju. Svatko igra vremenski neovisno o ostalim igraima, ali svi postoje u istom "*svemiru*" (serveru), te meusobno interaktiraju jedin s drugima. Postoje brojne takve igre, a u ovom radu ja sam za primjer uzeo dvije da pokaem kako sustav nije specijaliziran za samo jednu igru, nego za grupu igara.

Igre koje sam odabrao su:

- Ogame

- Travian

U nastavku e biti kratki i osnovni opisi navedenih igara, a detalji e biti opisani kasnije uz njihovu definiciju iz koje sustav ui pravila.

1.1 Ogame

Ogame je MMO (Masive Multiplayer Online) brwser (en: internet preglednik) igra. Prva verzija igre izdana je 2000 godine i do sada ima preko 2 000 000 korisnikih rauna. Korisnik se smjeta u ulogu voe svog planeta. Na svojoj planeti moe razvijati rudnike metala, kristala, te deuterijiuma, te pritom mora paziti da ima dovoljno energije za operiranje rudnicima. Osim razvoja rudnika koji daju resurse koji se koriste za sve ostalo u igri, igra moe razvijati i druge zgrade koje daju razna poboljanja, za ubrzanu granju, za mogunost istraivanja novih istraivanja, granju svemirskih brodova, robotskih topova za obranu, spremista resursa, itd. Svaki planet ima odreeni broj mjesta, a svaki level izgradnje neke graevine troi jedno mjesto. Kao to je spomenuto igra moe obavljati istraivanja koja daju dodatne pogodnosti kao, bolje titove brodovima, veu brzinu, mogunost novih istraivanja, itd. Osim tehnolokog napretka, igra moe graditi flotu brodova, te robotske topove za obranu. U kasnijem stadiju igre igra takoer moe kolonizirati nove planete, te tako proiriti svoje carstvo. Igra na temelju svog napretka i akcija koje poduzima dobiva bodove, te se tako rangira u odnosu na ostale igrae.

Pod “istrauje”, “gradi”, itd u ovoj igri misli se na stisne gumb i eka odreeno

vrijeme da se akcija obavi. Igra vidi sliku i brojeve, nema grafike koja demonstrira izgradnju, napadanje, ili bilo koju od akcija. Time je lako dobiti podatke o trenutnom stanju igre, te automatski upravljati igrom pomou parsiranja HTML stranice.

1.2 Travian

Travian je takoer MMO (Masive Multiplayer Online) brwser (internet preglednik) igra. Prva verzija igre izdana je 2004 godine. Slino kao kod Ogame-a korisnik je voa no ovdje sela umijesto planeta. U travijanu igra moe birati izmeu tri plemena (Gali, Rimljani, Teutonci) i svaki od njih ima neku svoju posebnost. Resursi u ovoj igri su drvo, glina, kamen i hrana. Igra razvija svoje selo tako da gradi graevine te ih razvija dalje na vie razine/nivoe razvoja. Za razliku od Ogame-a selo ima 20 mjesta na koja se mogu postaviti graevine (postoji vie od 20 razliitih graevina), te se jednom postavljene mogu razvijati do razine do koje igra to moe prijutiti resursima. Takoer u selu se mogu vriti istraivanja koja unaprijeuju razne stvari u selu. Naravno postoji i vojska, igra

gradi pjesake, konjanike i razne ratnike za svoju vojsku. Razlika od Ogame-a je ta to u Ogame-u igra moe graditi brodove koliko god ima resursa, u Travijanu postoji pojam populacije, te igra moe izgraditi vojske ovisno o hrani koju proizvodi kako bi mogao uzdravati vojsku. U kasnijim stadijima igre mogue je naseliti dodatna sela te time iriti svoje carstvo.

Sve nadogradnji i izgradnje koje igra napravi donose odreeni broj bodova, a po tim bodovima se igrai rangiraju.

Chapter 2

Sustav

2.1 Komponente sustava

Sustav se sastoji od:

- **definicije igre:** konfiguracijska yaml datoteka koja sadri opis igre
- **jezgre:** uitava konfiguraciju, te podesava strukturu sustava prema konfiguraciji, te komunicira sa sustavom za donoenje odluka
- **sustava za donoenje odluka:** sustav na temelju stanja igre i moguih akcija odabire koju akciju napraviti
- **sustava za automatkso upravljanje:** podsustav koji vee cijeli sustav sa stvarnom igrom na internetu

2.2 Definicija igara

Igre se definiraju YAML strukturom. YAML je format datoteke dizajniran za jednostavno parsiranje, serijalizaciju te deserijalizaciju podataka, te ljudsku itljivost. Nije bilo potrebno osmisljavati posebnu sintaksu za opis igre, jer je YAML dovoljan, a za parsiranje ve postoje gotove biblioteke. YAML-om se mogu opisati strukture rjenika i polja, te numeriki i tekstovni tipovi podataka.

```
YAML datoteke zapoinju s `"-----"` (tri uzastopna znaka '-')
i zavravaju s `"..."` (tri uzastopna znaka '.').

Svi lanovi neke liste su jednako indentirani
te zapoinju sa `"- "` (znak '-', te razmak).

Rjenik je opisan jednostavnim parom klju, vrijednost
`"<klju>: <vrijednost>"` (nakon toke-zarez mora slijediti razmak).

Postoji jo pravila, no navedena su dovoljna za nau definiciju,
te zainteresirani mogu sami posjetiti slubenu YAML web stranicu.
```

YAML strukturom imamo rjeeno deserijalizaciju strukture u memoriju, varijable, strukture unutar programa, no za smisao opisane igre dodajemo dodatna pravila na sadraj definicijske datoteke, te odreenu semantiku pojedinim djelovima strukture.

Za nae definicije opisna struktura na vrnoj razini sastoji se od sljedeih *kljueva*:

- game_variables (en: varijable igre)
- property_variables (en: varijable posjeda)
- resources (en: resursi)
- constructions (en: konstrukcije)
- advancements (en: napredovanja)

2.2.1 Varijable igre

Varijable igre su varijable sustava u kojima se prate stanja igre, ili bilo koje druge informacije vezane za igru koje su bitne te se mogu koristiti u daljnjoj definiciji igre. U sustavu postoje i podrazumjevane variable koje se ne moraju definirati, niti se njihova semantika moe mjenjati.

2.2.2 Varijable posjeda

Varijable posjeda su varijable sustava u kojima se prate pojedine posebnosti posjeda, posjedi su npr. graevine. Te varijable su bitne za daljnju definiciju. U sustavu postoje i podrazumjevane variable koje se ne moraju definirati, niti se njihova semantika moe mjenjati.

2.2.3 Resursi

Resursi su resursi igre, valute koje se mogu dobaviti u igri te se koriste za daljnji razvoj. Za ovaj klju konfiguriraju se samo imena resurasa.

2.2.4 Konstrukcije

Konstrukcije/zgrade su zgrade koje postoje u igri. Za njih se definiraju razliita svojstva.

- **name** (en: ime): - jedinstveno ime konstrukcije
- **requirements** (en: zahtjevi):
  - lista zahtjeva potrebnih da se konstrukcija moe graditi
  - zahtjev je rjenik s kljuevima:
    - type (en: tip) –> koji mogu biti construction (en: konstrukcija) ili advancement (en: napredovanje)
    - name (en: ime) –> ime konstrukcije ili napredovanja koje je zahtjev
    - level (en: nivo) –> nivo do kojeg minimalno mora biti razvijena konstrukcija/napredovanje sa zadanim imenom
  - tip: `construction | advancement`
- **effects** (en: efekti): lista efekata koje konstrukcija daje

  - efekti su rjenik koji ima dosta kljueva, te e oni biti objanjeni uz opis primjera konfiguracijske datoteke

- **costs** (en: trokovi): lista trokova za izgradnju konstrukcije po levelu

  - element liste je rjenik s kljuevima koji odgovaraju imenima resursa, a vrijednosti koliina odgovarajueg resursa potrebnog za izgradnju

- **build_duration** (en: trajanje gradnje) lista vremena u sekundama potrebnih da se izgradi konstrukcija

Napomena: za `costs` i `build_duration` indeks liste odgovara nivou izgradnje. (primjer. u listi costs podatak s indeksom 1 odgovara trokovima izgradnje na 1. nivo)

2.2.5 Napredovanja

Napredovanja su istraivanja/poboljanja koja se mogu provesti kako bi se poboljao neki parametar igre.
Napredovanja imaju jednaku sintaksu kao i konstrukcije.

2.3 Primjer definicije (Ogame)

```
---
:game_variables: []
:property_variables: []
:resources:
- :metal
- :crystal
- :deuterium
- :energy
:constructions:
- :name: Metal Mine
  :requirements: []
  :effects:
    :produce:
      :metal: [150, 165, 360, 595, 875, 1205, 1590, 2045, 2570, 3180, 3890, 4705,
               5645, 6730, 7970, 9395]
```

```
      :energy: [0, -11, -25, -40, -59, -81, -107, -137, -172, -213, -260, -314,
                -377, -449, -532, -627]
  :costs:
  - :metal: 135
    :crystal: 33
  :build_durations: [5, ]
- :name: Crystal Mine
  :requirements: []
  :effects:
    :produce:
      :crystal: [75, 110, 240, 395, 585, 805, 1060, 1360, 1710, 2120, 2590, 3135,
                 3765, 4485, 5313, 6265]
      :energy: [0, -11, -25, -40, -59, -81, -107, -137, -172, -213, -260, -314,
                -377, -449, -532, -627]
  :costs:
  - :metal: 48
    :crystal: 24
  :build_durations: [5, ]
- :name: Deuterium Synthesizer
  :requirements: []
  :effects:
    :produce:
      :deuterium: [0, 65, 140, 235, 345, 475, 625, 805, 1015, 1255, 1535, 1855,
                   2225, 2655, 3145, 3705]
      :energy: [0, -22, -49, -80, -118, -162, -213, -273, -343, -425, -519, -628,
                -754, -898, -1064, -1254]
  :costs:
  - :metal: 225
    :crystal: 75
  :build_durations: [5, ]
- :name: Solar Plant
  :requirements: []
  :effects:
    :produce:
      :energy: [0, 22, 48, 79, 117, 161, 212, 272, 342, 424, 518, 627, 753, 897,
                1063, 1253]
  :costs:
  - :metal: 75
    :crystal: 30
  :build_durations: [5, ]
- :name: Fusion Reactor
  :requirements:
  - :type: construction
    :name: Deuterium Synthesizer
    :level: 5
  - :type: advancement
    :name: Energy Technology
    :level: 3
  :effects:
    :produce:
      :energy: [31, 66, 104, 145, 191, 241, 295, 354, 418, 488, 564, 646, 735,
```

```
                831, 935]
      :deuterium: [-55, -125, -200, -295, -405, -535, -685, -860, -1065, -1300,
                   -1570, -1885, -2242, -2660, -3135]
  :costs:
  - :metal: 900
    :crystal: 360
    :deuterium: 180
  :build_durations: [5, ]
- :name: Metal Storage
  :requirements: []
  :effects:
    :inc_storage:
      :metal: [10000, 20000, 40000, 75000, 140000, 255000, 470000, 865000, 1590000,
               2920000, 5355000, 9820000, 18005000, 33005000, 60510000, 110925000]
  :costs:
  - :metal: 1000
  :build_durations: [5, ]
- :name: Crystal Storage
  :requirements: []
  :effects:
    :inc_storage:
      :crystal: [10000, 20000, 40000, 75000, 140000, 255000, 470000, 865000,
                 1590000, 2920000, 5355000, 9820000, 18005000, 33005000, 60510000,
                 110925000]
  :costs:
  - :metal: 1000
    :crystal: 500
  :build_durations: [5, ]
- :name: Deuterium Tank
  :requirements: []
  :effects:
    :inc_storage:
      :deuterium: [10000, 20000, 40000, 75000, 140000, 255000, 470000, 865000,
                   1590000, 2920000, 5355000, 9820000, 18005000, 33005000, 60510000,
                   110925000]
  :costs:
  - :metal: 1000
    :crystal: 1000
  :build_durations: [5, ]
- :name: Robotics Factory
  :requirements: []
  :effects:
    :faster_build:
      :constructions: [100, 50, 33.3, 25, 20, 16.6, 14.3, 12.5, 11.1, 10, 9.1, 8.4,
                       7.7, 7.1, 6.7, 6.2, 5.9]
  :costs:
  - :metal: 400
    :crystal: 120
    :deuterium: 200
  - :metal: 800
    :crystal: 240
```

```
    :deuterium: 400
  - :metal: 1600
    :crystal: 480
    :deuterium: 800
  - :metal: 3200
    :crystal: 960
    :deuterium: 1600
  - :metal: 6400
    :crystal: 1920
    :deuterium: 3200
  - :metal: 12800
    :crystal: 3840
    :deuterium: 6400
  - :metal: 25600
    :crystal: 7680
    :deuterium: 12800
  - :metal: 51200
    :crystal: 15360
    :deuterium: 25600
  - :metal: 102400
    :crystal: 30720
    :deuterium: 51200
  - :metal: 204800
    :crystal: 61440
    :deuterium: 102400
  :build_durations: [42, ]
- :name: Shipyard
  :requirements:
  - :type: construction
    :name: Robotics_Factory
    :level: 2
  :effects: []
  :costs:
  - :metal: 400
    :crystal: 200
    :deuterium: 100
  :build_durations: [49, ]
- :name: Research Lab
  :requirements: []
  :effects: []
  :costs:
  - :metal: 200
    :crystal: 400
    :deuterium: 200
  :build_durations: [49, ]
- :name: Alliance Depot
  :requirements: []
  :effects: []
  :costs:
  - :metal: 20000
    :crystal: 40000
```

```
  :build_durations: [4937, ]
- :name: Missile Silo
  :requirements:
  - :type: construction
    :name: Shipyard
    :level: 2
  :effects: []
  :costs:
  - :metal: 20000
    :crystal: 20000
    :deuterium: 1000
  :build_durations: [3291, ]
- :name: Nanite Factory
  :requirements:
  - :type: construction
    :name: Robotics Factory
    :level: 10
  - :type: advancement
    :name: Computer Technology
    :level: 10
  :effects:
    :faster_build:
      :constructions: [100, 50, 25, 12.5, 6.25, 3.125, 1.5625, 0.78125]
      :advancements: [100, 50, 25, 12.5, 6.25, 3.125, 1.5625, 0.78125]
      :units: [100, 50, 25, 12.5, 6.25, 3.125, 1.5625, 0.78125]
  :costs:
  - :metal: 1000000
    :crystal: 500000
    :deuterium: 100000
  - :metal: 2000000
    :crystal: 1000000
    :deuterium: 200000
  - :metal: 4000000
    :crystal: 2000000
    :deuterium: 400000
  - :metal: 8000000
    :crystal: 4000000
    :deuterium: 800000
  - :metal: 16000000
    :crystal: 8000000
    :deuterium: 1600000
  - :metal: 32000000
    :crystal: 16000000
    :deuterium: 3200000
  - :metal: 64000000
    :crystal: 32000000
    :deuterium: 6400000
  - :metal: 128000000
    :crystal: 64000000
    :deuterium: 12800000
  :build_durations: [432000, 432000, 432000, 432000, 432000, 432000, 432000, 432000,
```

```
                    432000, 432000]
- :name: Terraformer
  :requirements:
  - :type: construction
    :name: Nanite Factory
    :level: 1
  - :type: advancement
    :name: Energy Technology
    :level: 12
  :effects:
    :construction_space: [3, 6, 9, 12, 15, 18, 21, 24, 27, 30]
  :costs:
  - :crystal: 50000
    :deuterium: 100000
    :metal: 1000
  :build_durations: [4114,]
:advancements:
- :name: Energy Technology
  :requirements:
  - :type: construction
    :name: Research Lab
    :level: 1
  :effects: []
  :costs:
  - :crystal: 800
    :deuterium: 400
  :build_durations: [576, ]
- :name: Laser Technology
  :requirements:
  - :type: advancement
    :name: Energy Technology
    :level: 2
  :effects: []
  :costs:
  - :metal: 200
    :crystal: 100
  :build_durations: [216, ]
- :name: Ion Technology
  :requirements:
  - :type: advancement
    :name: Laser Technology
    :level: 5
  - :type: advancement
    :name: Energy Technology
    :level: 4
  :effects: []
  :costs:
  - :metal: 1000
    :crystal: 300
    :deuterium: 100
  :build_durations: [936, ]
```

```
- :name: Hyperspace Technology
  :requirements:
  - :type: advancement
    :name: Energy Technology
    :level: 5
  - :type: advancement
    :name: Shielding Technology
    :level: 5
  - :type: construction
    :name: Research Lab
    :level: 7
  :effects: []
  :costs:
  - :crystal: 4000
    :deuterium: 2000
  :build_durations: [2880, ]
- :name: Plasma Technology
  :requirements:
  - :type: advancement
    :name: Laser Technology
    :level: 10
  - :type: advancement
    :name: Ion Technology
    :level: 5
  - :type: advancement
    :name: Energy Technology
    :level: 8
  :effects:
    :inc_produce:
      :metal: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
      :crystal: [0, 0.66, 1.32, 1.98, 2.64, 3.3, 3.96, 4.62, 5.28, 5.94, 6.6]
      :deuterium: [0, 0.33, 0.66, 0.99, 1.32, 1.65, 1.98, 2.31, 2.64, 2.97, 3.3]
  :costs:
  - :metal: 2000
    :crystal: 4000
    :deuterium: 1000
  :build_durations: [4320, ]
- :name: Combustion Drive
  :requirements:
  - :type: advancement
    :name: Energy Technology
    :level: 1
  :effects: []
  :costs:
  - :metal: 400
    :deuterium: 600
  :build_durations: [288, ]
- :name: Impulse Drive
  :requirements:
  - :type: advancement
    :name: Energy Technology
```

```
    :level: 1
  - :type: construction
    :name: Research Lab
    :level: 2
  :effects: []
  :costs:
  - :metal: 2000
    :crystal: 4000
    :deuterium: 600
  :build_durations: [4320, ]
- :name: Hyperspace Drive
  :requirements:
  - :type: advancement
    :name: Hyperspace Technology
    :level: 3
  :effects: []
  :costs:
  - :metal: 10000
    :crystal: 20000
    :deuterium: 6000
  :build_durations: [216000, ]
- :name: Espionage Technology
  :requirements:
  - :type: construction
    :name: Research Lab
    :level: 3
  :effects: []
  :costs:
  - :metal: 200
    :crystal: 1000
    :deuterium: 200
  :build_durations: [864, ]
- :name: Computer Technology
  :requirements:
  - :type: construction
    :name: Research Lab
    :level: 1
  :effects:
    :units_send: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
  :costs:
  - :crystal: 400
    :deuterium: 600
  :build_durations: [288, ]
- :name: Astrophysics
  :requirements:
  - :type: advancement
    :name: Espionage Technology
    :level: 4
  - :type: advancement
    :name: Impulse Drive
    :level: 3
```

```
  :effects:
    :max_number_of_properties: [1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8]
    :max_number_of_expeditions: [0, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 4]
  :costs:
  - :metal: 4000
    :crystal: 8000
    :deuterium: 4000
  :build_durations: [8640, ]
- :name: Intergalactic Research Network
  :requirements:
  - :type: advancement
    :name: Computer Technology
    :level: 8
  - :type: advancement
    :name: Hyperspace Technology
    :level: 8
  - :type: construction
    :name: Research Lab
    :level: 10
  :effects: []
  :costs:
  - :metal: 240000
    :crystal: 400000
    :deuterium: 160000
  :build_durations: [460800, ]
- :name: Graviton Technology
  :requirements:
  - :type: construction
    :name: Research Lab
    :level: 12
  :effects: []
  :costs:
  - :metal: 300000
  :build_durations: [1, ]
- :name: Weapons Technology
  :requirements:
  - :type: construction
    :name: Research Lab
    :level: 4
  :effects:
    :units_bonus:
      :weapons: [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110]
  :costs:
  - :metal: 800
    :crystal: 200
  :build_durations: [720, ]
- :name: Shielding Technology
  :requirements:
  - :type: advancement
    :name: Energy Technology
    :level: 3
```

```
  - :type: construction
    :name: Research Lab
    :level: 6
  :effects:
    :units_bonus:
      :shields: [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110]
  :costs:
  - :metal: 200
    :crystal: 600
  :build_durations: [576, ]
- :name: Armour Technology
  :requirements:
  - :type: construction
    :name: Research Lab
    :level: 2
  :effects:
    :units_bonus:
      :hull_plating: [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110]
  :costs:
  - :metal: 1000
  :build_durations: [720, ]
```

2.4 Jezgra

Sama jezgra je u biti veoma jednostavna i koristi ostale sustave, te je centralni dio koji ih po potrebi poziva i integrira u jednu cijelinu. Sustav uitava konfiguracijsku datoteku, te inicijalizira sustav za automatsko upravljanje.

U inicijalizaciji se pomou sustava za automatsko upravljanja dohvate podaci o trenutnom stanju igre. Uz to se u za to predviene klase podeavaju uitane konfiguracije.

Chapter 3

Sustav za donoenje odluka

Sustav za donoenje odluka kao to mu i ime kae slui za donoenje odluka.

Sustav odabire "najbolji" potez, te informaciju o istome vraa jezgri koja onda izvodi taj potez.

Sad to je to "najbolji" potez i kako se on odabire. Kod igara u stvarnom vremenu problem je to to su u stvarnom vremenu, akcije se u "svijetu" igre dogaaju konstantno i neovisno o tome reagiramo li mi ili ne. Ako sustav eli uzeti u obzir sve "poteze"/akcije koje moe odigrati te na temelju toga odabrati "najbolji" potez/akciju, mora uzeti u obzir i vrijeme potrebno da donese odluku, jer tokom njegovog "razmiljanja" ostali igrai igraju, to mogu promijeniti stanje igre bitno za igraa koji donosi odluku.

Ovdje sustav i koristi pristup razmatranja moguih opcija, te deterministiko donoenje odluka. Problemi su kao to smo spomenuli odvijanje igre u stvarnom vremenu, te mogunost suradnje izmeu vlastitih posjeda. Kako igra tokom igre moe posjedovati vie od jednog posjeda, sustav koji igra treba odluiti isplati li se vie razvijati neki od posjeda neovisno o ostalima ili prilagoditi razvitak kako bi posjedi pomogli jedan drugome i time prije doli do eljenog rezultata.

3.1 Grubi opis rada sustava

Sustav radi na nain da trai skup akcija (za svaki posjed po jednu ili dvije, na jednom posjetu se u jedno vrijeme moe u jednom trenutku graditi samo jedna

konstrukcija, te napredovati u jednom od napredovanja). Traeni skup akcija bi trebao biti onaj kojim e se najprije doi do odreenog cilja.

Kao cilj mogu se uzeti razni faktori, ove igre imaju i svoje sustave bodovanja po kojima se radi rang lista lista igraa (svaka igra ima vie razliitih sustava bodovanja, za razne kriterije). Za demonstraciju smo za dvije navedene igre uzeli kao cilj cilj igraa zvanog "farmer". To je igra koji za cilj ima razviti napredovanja i graevine kako bi imao to veu proizvodnju resursa.

3.2 Detaljniji opis rada sustava

Kada sustav treba donjeti odluku, stvori inicijalni skup stanja posjeda (trenutne situacije svih posjeda), dodjeli mu broj koji oznaava njegovu dobrotu, a kao akcije potrebne da se doe do tih stanja postavlja `nil` jer se ve nalazi u tim stanjima te objekt sa svim tim podacima doda u red stanja.
Nakon toga sljedi petlja koja se vrti dok se ne zadovolji jedan od uvjeta:

- u redu ne postoje ne obraena stanja
- postoji stanje koje ima ostvaren eljeni cilj
- obraen je maksimalan broj transakcija koje smo spremni obraditi prije donoenja odluke

3.2.1 Raunanje dobrote

Dobrotu raunamo kao zbroj trenutog stanja resursa i resursa koji mogu biti proizvedeni u sljedeih n sekundi. N je dupli broj sekundi potreban da se izgradi/unaprijedi konstrukcija/napredovanje koji se razvija.

Treba obratiti panju da svi od resursa nisu jednako vrijedni. Takoer akcija kojom se dolo do tog stanja traje neko vrijeme da se izvri, te ako ta akcija mijenja trenutnu proizvodnju, treba se uzeti u obzir da se prije nego to je zbog izvrene akcije dolo do promjene u proizvodnji, proizvodilo po staroj stopi proizvodnje. I to je samo jedan od primjera koji se moe desiti i promjeniti jednostavnu kalkulaciju.

Evaluacija se provodi za svaku od akcija i zbrajaju u ukupnu dobrotu, to znai da se nakon svakog novog stanja dodang u red moe iz tog novog stanja na njemu napraviti akcija koja e igru dovesti u opet novo stanje.

Za svako od tih stanja u nizu se provodi evaluacija. Ovdje je jedna od potekoa bila to to se te korake treba spojiti kod razmatranja, a ne ih gledati neovisno.

Objanjenje

Ako u prvom koraku gradimo zgradu kojoj treba 3 sata da se izgradi, a u drugom koraku (za drugu akciju), gradimo zgradu koja se gradi 4 sata.

Poto za dobrotu u ovom sluaju uzimamo trenutne resurse + resurse koji e se proizvesti u sljedeih 6 sati za prvu akciju, te sljedeih 8 za drugu (duplo od vremena izgradnje), ako oba koraka uzimamo zasebno, prvo emo izraunati trenutnu proizvodnju za 3 sata kojih se gradi konstrukcija, te 3 po novoj stopi proizvodnje. Nakon toga bi tome dodali 4 sata tadanje produkcije, te 4 sati produkcije po produkcijskoj stopi nakon to se izgradila konstrukcija.

Takvo raunanje nije uvijek tono. Ako smo imali dovoljno resursa, mogli smo izgraditi prvu konstrukciju u prvih 3 sata, te drugu u sljedea 4, te preostalih (3 + 4) sati stvarati po stopi nakon izgradnje obju konstrukcija, to e u sluaju da druga izgradnja mjenja stopu produkcije utjecati na rezultat.

Sustav za mogua budua stanja radi simulacije, te pomou njih postavlja objekte koje sadre podatke o stanjima na vrijednosti kao da su se akcije izvrile.

3.2.2 Cilj

Kao cilj smo postavili dublu vrijednost dobrote od one iz poetnog/inicijalnog stanja. Poto se do cilja moe doi na vie naina (beskonana izgradnja konstrukcija), “pobjednik”/”najbolji” potez je onaj koji dovodi do ciljanog stanja dolazi u najmanje vremena.

Elementi se iz reda uzimaju po trenutnom potrebnom vremenu da se doe do stanja u redu, prvi koji se uzme a da je zadovoljio cilj je i “najbolji”.

Iako se do “ciljanog” stanja moe doi obavljanjem vie od jedne akcije, jezgra obavlja samo prvu od akcija, te prije zavretka svake od akcija koje se izvravaju ponovo rauna “najbolji” potez.

Chapter 4

Sustav za automatsko upravljanje

Sustav za automatsko upravljanje je dodatna komponenta koja omoguava igranje igre na stvarnim serverima sa stvarnim igraima.

Sam po sebi sustav “igra” “sam na svijetu” i lokalno. No igre postoje “vani” na serverima i igraju ih brojni igrai diljem svijeta. Mogue je implementirati GameHook adapter klasu koja e kada se akcija treba izvriti, akciju odigrati u stvarnoj postojeoj igri.

Igre se igraju u internet preglednicima, te je umijesto prouavanja zahtjeva koji se salju serverima za obavljanje akcija bila jednostavnija druga opcija. Za automatsko igranje koristi se in-memory (en: u memoriji) preglednik. Preglednik koji se “ne vidi”, nego postoji u memoriji raunala, te se njime moe upravljati programski.

HTML se lako bez muke samostalnog parsiranja moe pretraivati pomou standardnih css selektora, na bilo koji element se moe kliknuti, ili iz njega proitati vrijednosti.

Chapter 5

Saetak

Chapter 6

Zakljuak