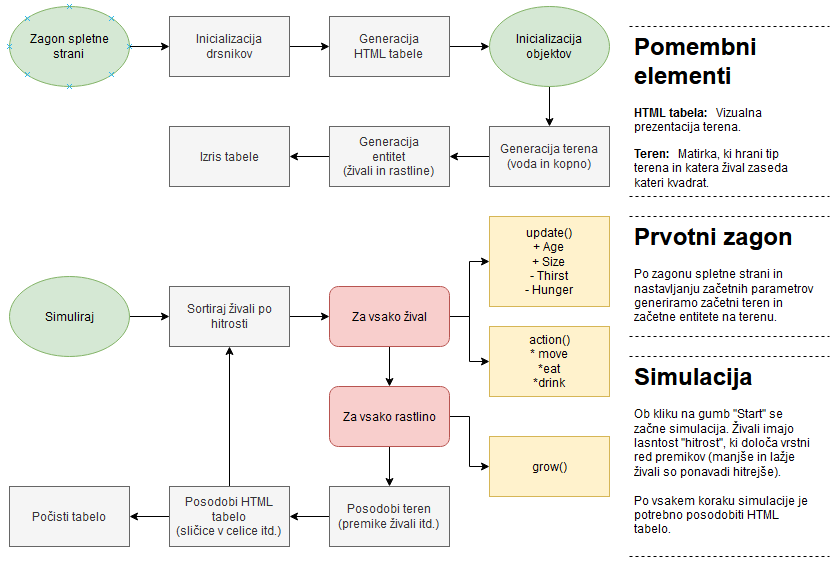
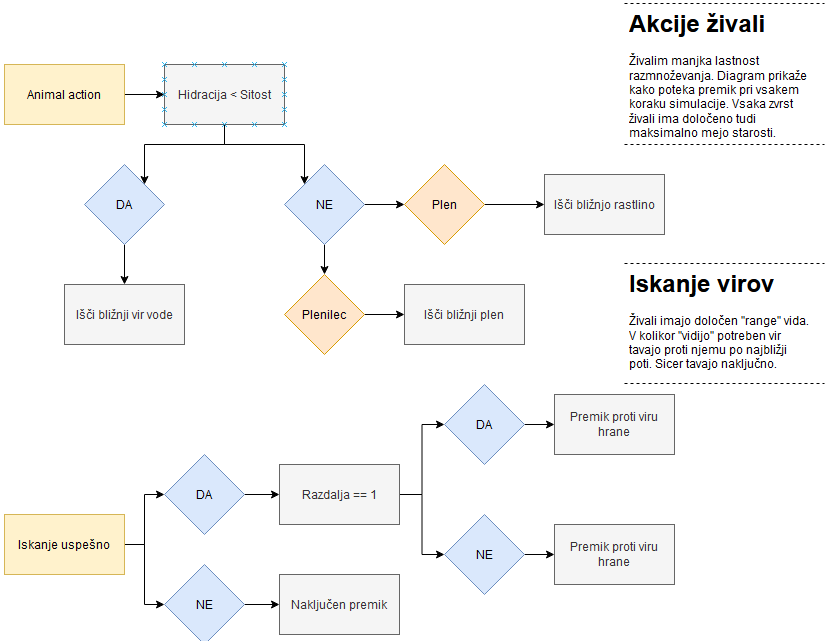
Vaja 4 – Dinamična in Statična analiza





Statična analiza

# Dvakratna iteracija čez tabelo.

**Opis problema:**

Po vsakem koraku simulacije je potrebno posodobiti izrise v tabeli. Se pravi si koraki sledijo, izbris trenutnih sličic v celicah tabele po celotni tabeli in ponovna iteracija čez celotno tabelo, da se glede na teren vstavijo sličice v tabelo.

**Možna rešitev:**

Združitev obeh funkcij. Lahko nastopi problem pri mejah iteracije, saj je lahko teren večji od tabele, ki služi samo prikazu določenega »okenca« terena.



# Intervali simulacije

**Opis problema:**

Ob kliku na gumb za začetek simulacije, se glede na določen interval (1500ms) kliče funkcija doSimulationStep(). Takšna implementacija je namenjena spremljanju dogajanja simulacije na tabeli. Saj interval omogoči, da so premiki uporabniku vidni ob hitrem delovanju računalnika. Problem nastopi ob dovolj nizkem času intervala in velikem številu entitet ter velikosti terena. Zgodi se lahko, da korak simulacije traja dlje časa kot sam interval in se nov korak simulacije kliče preden se prejšnji konča.

**Možne rešitve:**

1. Po vsakem zaključenem koraku simulacije, se izvede rekurzivni klic za nov korak vendar ta klic zakasnimo z funkcijo sleep().

+… Odpravimo možnost napake.

1. Simulacije se izvede do konca, korake pa prikazujemo naknadno. Dodatno lahko implementiramo premik levo in desno po korakih ter tako omogočimo uporabniku večji nadzor nad ogledom.

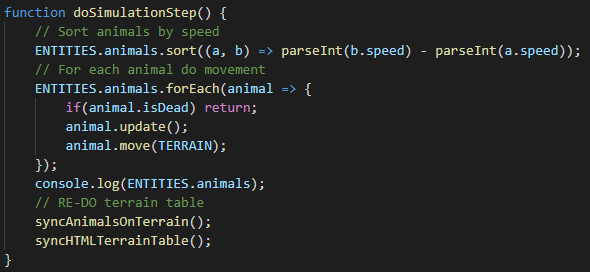
+… Odpravimo možnost napake.

+… Večji nadzor uporabnika nad ogledom simulacije

+… Pohitrimo samo simulacijo

-… Znatno povišana poraba pomnilnika za shranjevanje simulacije

-… Večja količina dodatnega dela



# Generacija začetnih entitet

**Opis problema:**

Funkcija poizkuša naljučno določiti pozicije n-entitet. V kolikor naljučna pozicija ni ustrezna (voda ali že zasedena pozicija) se šteje kot en poizkus. V primeru petih neuspelih poizkusov se funkcija prekine. Robni primer, do katerega načeloma nikoli ne pride, v primeru, da se zgodi pa simulacija z danimi parametri tako ni izvedljiva ampak se načeloma podaljša čas izvajanja.

**Možna rešitev:**

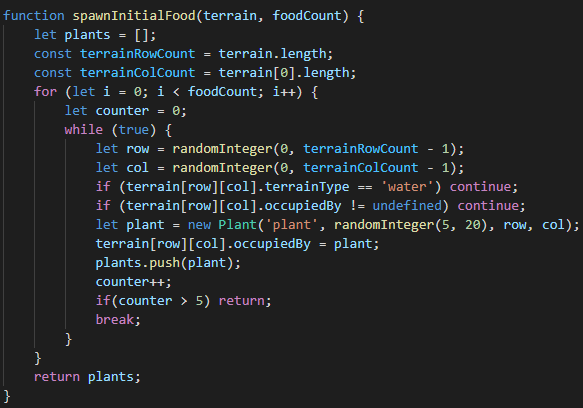
Beleženje »zemeljnih« pozicij. V kolikor velja (število plenilcov + plena + rastlin > števila pozicij) se funkcija zaustavi ali pa se števila prilagodijo. Po generaciji enega objekta na določeno pozicijo to pozicijo odstranimo iz nabora.

+… Hitrejše dodeljevanje pozicij. Nikoli ni potrebno naključno iskati ustrezno pozicijo saj so vse ustrezne.

+… Funkcija se prej zakluči, če simulacija ni mogoča.

-… Večja poraba (sicer zanemarljiva) pomnilnika v času generacije

-… V optimalnih pogojih in »srečo« pri iskanju naključnih pozicij, je trenutna implementacija še vedno hitrejša. (Testi v dinamični analizi)



# Iskanje poti

**Opis problema:**

Ko plen ali plenilec išče vir hrane oziroma vode v svoji okolici najde vse možne vire hrane, v kolikor jih ne najde pa naključno mesto kamor se lahko premakne, vedno preišče vse možnosti in v primeru več enakovrednih rešitev naključno izbere eno.

**Možna rešitev:**

Odstranitev naključnega elementa. Premiki živali bodo 100 odstotno predvidljivi in odvisni od orientacije v katere pregledujemo okolico.

+… Pospešena simulacija

-… Predviljiva simulacija, brez naključnosti.

