

AngryPigs

Dokumentacija

1. junij 2011

Avtorja

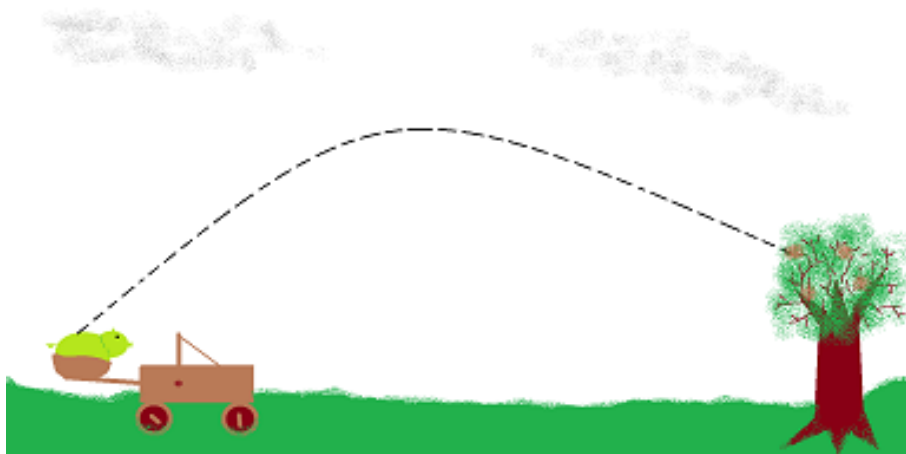
Matic Potočnik, Jaka Sivec

Poglavje 1

O igri

Angry Pigs je zabavna 3D igra, v kateri igralec obstreljuje drevesa in skuša z njih sklatiti ptiče, ter njihova gnezda.

Igra je napisana v jezikih Clojure in Scala, z uporabo OpenGL knjižnice LWJGL.



Poglavje 2

Tehnologija

2.1 Generiranje dreves

Problem generiranja primerno realističnih dreves se je izkazal za algoritemsko precej zanimivega. V namene seminarske naloge je bil rešen "iz nule", čeprav je ta problem že dokaj raziskan v obliki L-sistemov in verjetno še česa. Ampak to ni tako zabavno.

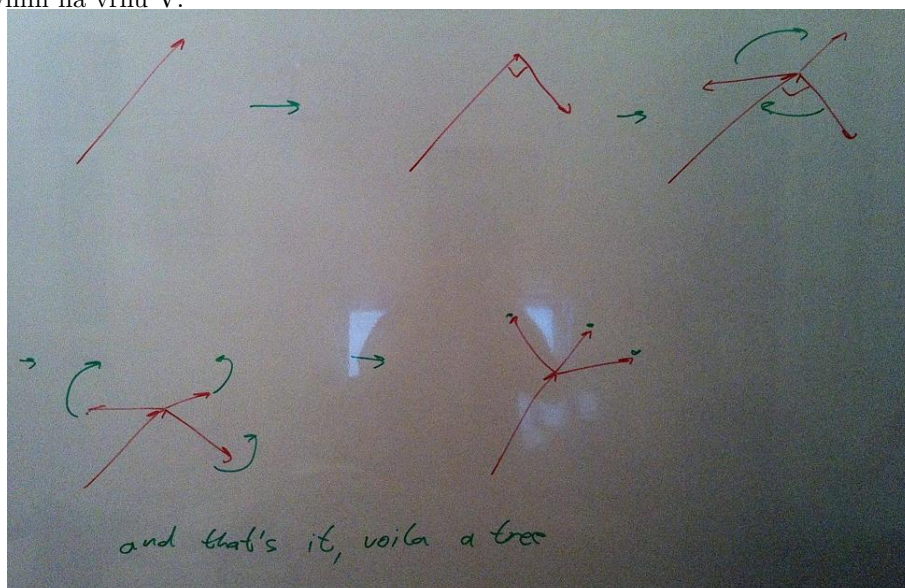
Osnovna postavka merjenja uspeh akončnega algoritma je bila funkcijska čistost, saj sem želel postaviti siste, ki iz peščice začetnih podatkov generira neskončno število različnih dreves. Realističnost je parameter, ki ga je bilo malo težje meriti zato je umetniški vtis končnega rezultata bolj subjektivna ocena, da so drevesa uspešno zgenerirana kot pa neka številka. Ako bi že moral definirati kako uspešno algoritem generira drevesa, porečem, da 7.

2.1.1 Osnovni princip

Algoritem je zasnovan kot generator fraktalov, ki mu podamo nek vektor in osnovno dolžino veje, ta pa potem rekurzivno kliče samega sebe, dokler ne doseže neke smiselno določene meje globine drevesa.

Glavna funkcija algoritma se najprej odloči koliko vej je potrebnih na trenutni globini, kar je definirano z vrsto praštevil (2,3,5 ...). Nato izračuna vektorski produkt med podanim vektorjem V in nekim izmišljenim vektorjem, s čimer dobi prvo vejo na trenutnem nivoju.

Ko je izbrana prva veja, jo samo $(N-1)$ -krat duplicira in zavrti okrog vektorja V . Tako ima osnovno vejo (vektor V) ter N nanjo pravokotnih vej, ki predstavljajo poddrevo in so enakomerno razporejene po ravnini na vrhu V .



Naslednji korak je samo še preprosta rotacija vseh vej za nek kot v nasprotni smeri vektorja V in že imamo zgenerirano krošnjo za podano vejo oz. deblo. Nato algoritem spet pokliče samega sebe na vrhu vsake tako zgenerirane veje.

2.1.2 Naključnost

Po zgornjem algoritmu so drevesa sicer zgledala dokaj realistična, vendar pa so bila vsa enaka, kar je bilo tako dolgočasno, kot je onemogočalo, da bi tudi gozd deloval prilično resničen. Rešitev je torej neka stopnja naključnosti.

Že intuitivno je opaziti, da drevo ne more biti popolnoma naključno, saj potem izgubi na vsem kar ga sploh dela za drevo in postane samo kup nekoherentnih črt. V interesu pravega znanstvenega pristopa in izogibanja intuitivnim skokom logike, je bilo zgenerirano tudi eno tako drevo.



2.1.3 Gravitacija

2.2 The rest

Poglavje 3

Rezultati