<https://gdcvault.com/play/1018262/The-Next-Vector-Improvements-in>

Arbitration function (01:30-tól)

* Ezzel adjuk össze a komponenseket okosabban
* Prioritized weighted queue
* Mindegyik komponensnek lesz egy sorrendje és weight-je, illetve az egész rendszernek egy max velocity-ja
  + Sorban haladunk komponenseken, adjuk össze a weight-el beszorozva, majd amint elértük a max velocity-t, többit ignoráljuk
    - Ha nem érjük el, mindegyiken átmegyünk

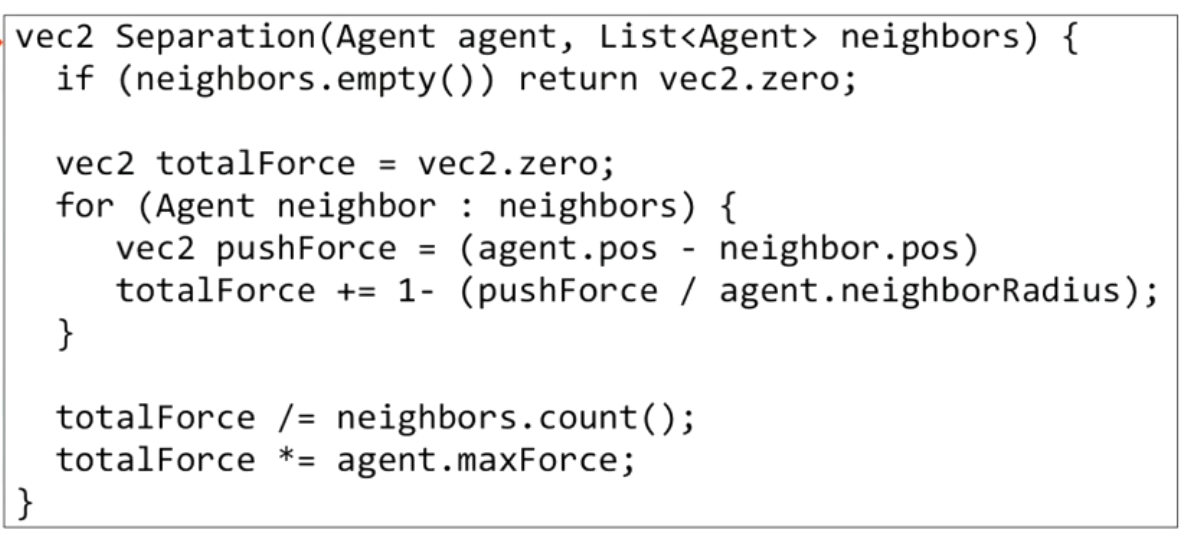
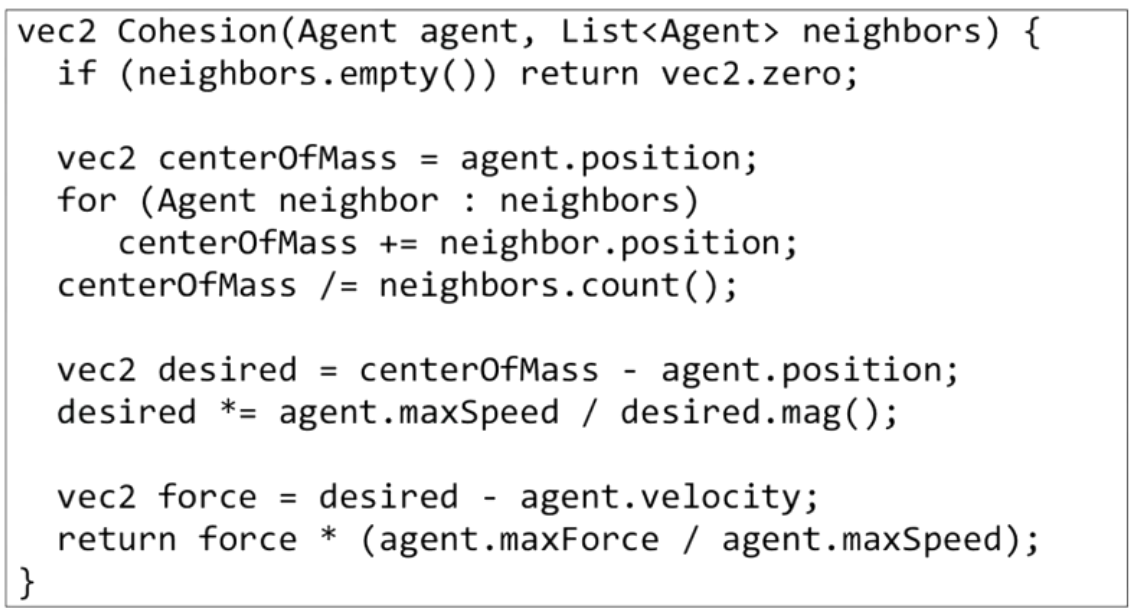
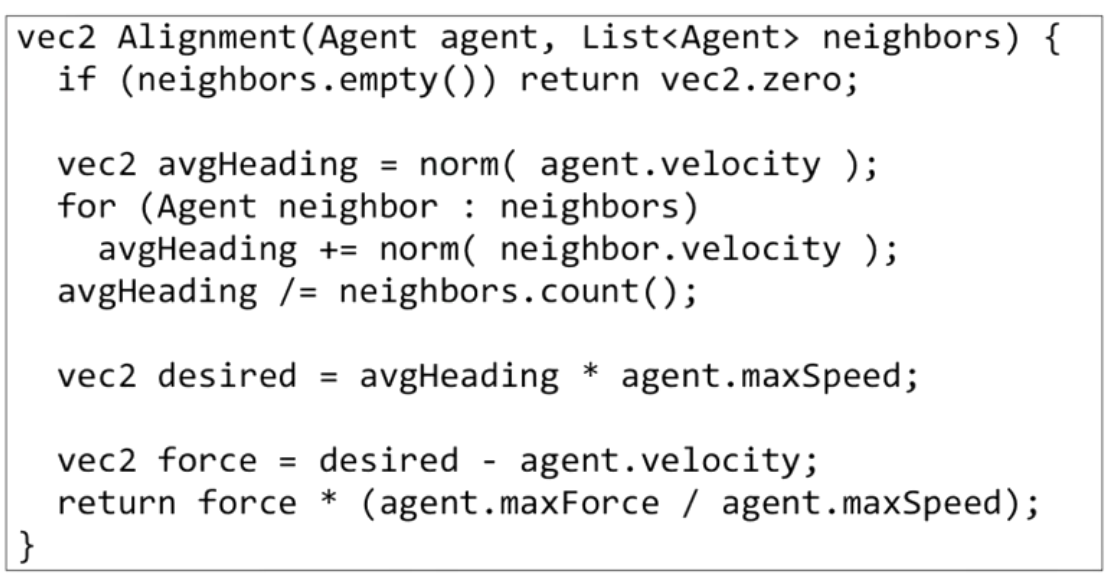
Good idea: think of our calculated direction like input for the player character

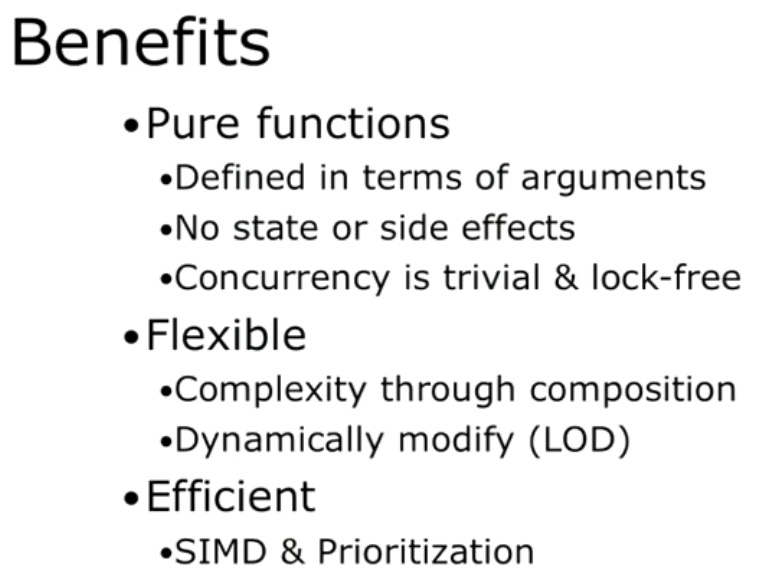
* Pl.: raw input még átmegy mindenféle filtereken, smoothingon
* Illetve animation systemeket is újra lehet használni így

Agent properties

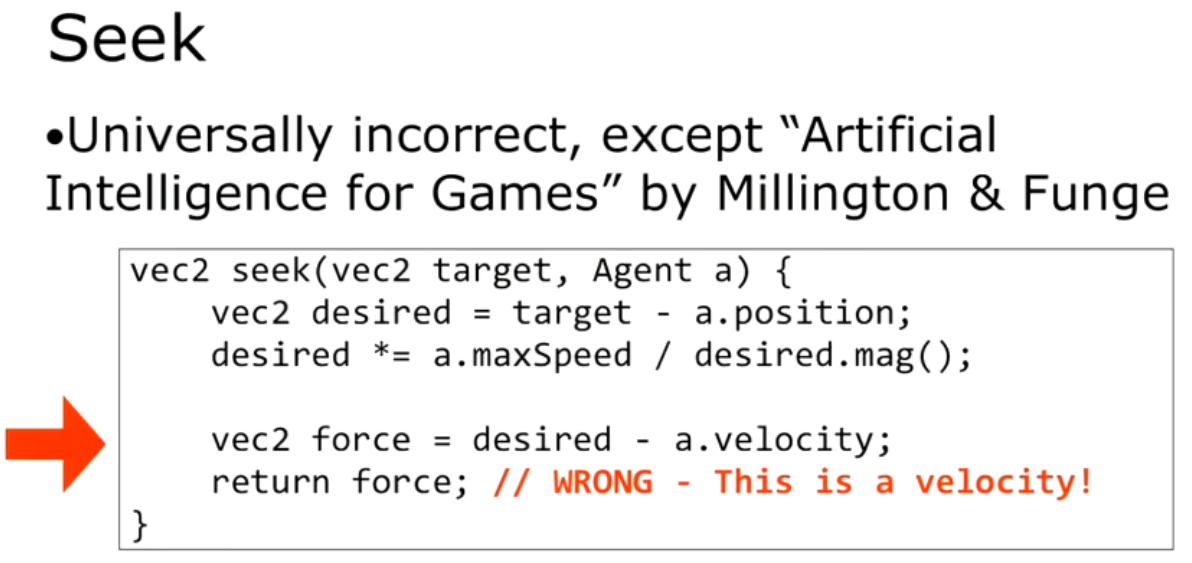
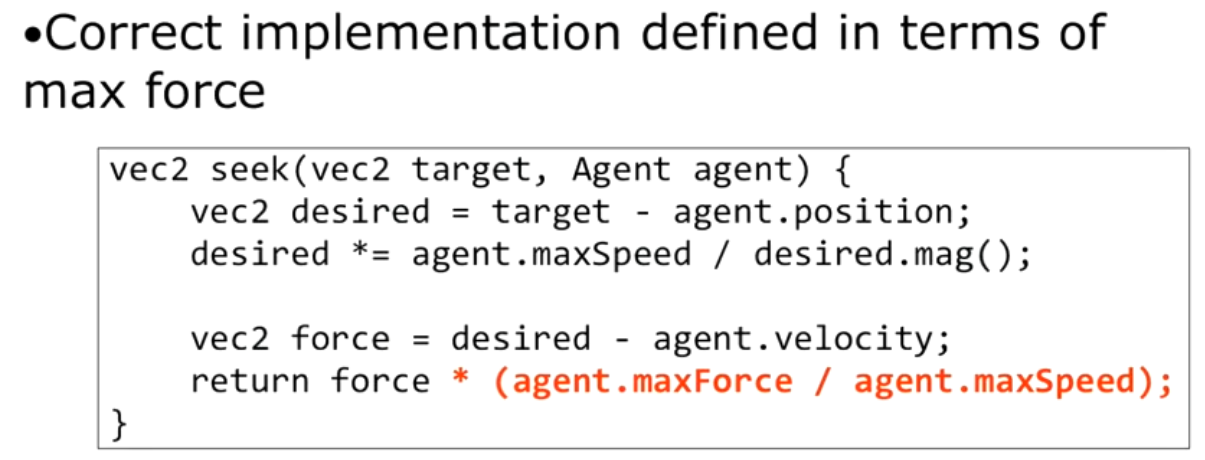
* Point mass
* Collision circle
* Max force
  + Max mekkora lehet a frameenkénti steering direction
* Max speed
* Neighbor Radius

Flockimg behaviour

* 
  + Kevésbé frequent változások, lehet kevesebb akár több frame-re számolni
  + totalForce hossza: inversely proportional to the distance to that neighbour
    - minnél közelebb, annál nagyobb erő
* 
  + desied velocity = the maximum speed towards the cohesion point
  + A végső eredmény az iránykülönbség lesz az ideális irány felé
* 
  + lowest priority usually
  + avgHeading: merre mennek a többiek



Seek

* 
* 
  + Ez már akkor is jó, ha megváltozatjuk a maxSpeed-et az agentnek
  + maxForce/maxSpeed: normalizált értéket ad vissza 0 és 1 között

Oszcilálás ellenkező viselkedések között (pl cohesion vs separation)

* Egyrészt: smoothing centrally
* Másrészt: inkább csak pl 10 frame-enként számoljuk őket, így sokkal low frequency oscillation lesz (kesvésbé látható)
  + Hysteresis???
* saját: cohesion-t erősebbre, és sokkal inkább középpont felé haladónak állítod be, akkor az megoldja separation nagy részét is (a középpont egyenlő távol legyen mindenki mástól)