

Case study: „Simulácia pohybu žubrienok“

1. Úvod

Umelý život je jednou z najmladších a najrýchlejšie sa rozvíjajúcich disciplín umelej inteligencie. Jej cieľom je o.i. štúdium života v prírode, a hlavne štúdium hypotetických prípadov, keby sa evolúcia a správanie jedincov v živej prírode odvíjalo od iného základu alebo v iných podmienkach, než je tomu v skutočnosti. Keďže správanie živočíchov v takýchto situáciách nie je možné pozorovať priamo v prírode, hlavnou metódou umelého života sa stala počítačová simulácia, ktorá na základe mechanizmov a princípov správania, odpozorovaných zo živej prírody, simuluje život virtuálnych jedincov jedného alebo viacerých druhov a ich vzájomné interakcie v prostredí, definovanom ľubovoľne v rámci opisnej sily daného počítačového modelu.

V tejto práci je študované správanie kolónie žubrienok, ktorých jediným princípom správania je sada jednoduchých pravidiel, založených na lokálnom vnímaní okolitého sveta pre každého jedinca z kolónie, a bez akéhokolvek iného mechanizmu, napr. centrálného riadenia. Simuláciou je ukázané, že pri správnom zvolení týchto jednoduchých pravidiel (predpokladom je, že toto „správne zvolenie“ pravidiel vzniklo v živej prírode evolúciou) kolónia žubrienok vykazuje správanie, pripomínajúce skôr prítomnosť nejakej globálnej sily, ktorá usporiada kolóniu do výhodných štruktúr.

2. Popis metódy

2.1. Fyzikálny základ

Správanie jedincov v simulácii podlieha niektorým základným fyzikálnym princípom. Základnou vlastnosťou prostredia, v ktorom simulácia prebieha, je absencia tretieho rozmeru (voči reálnej situácii), teda simulácia prebieha len v rovine. Základnou vlastnosťou všetkých jedincov je, že sú považované za hmotné body, teda neuvažujeme o ich rozmeroch, ale zároveň im prisúdime isté obmedzenia, ktoré vyplývajú z ich nenulovej hmotnosti.

Každý jedinec má v každom okamihu simulácie svoju okamžitú polohu \vec{x} , rýchlosť \vec{v} a zrýchlenie \vec{a} . Vzhľadom k tomu, že pracujeme s dvojrozmerným prostredím, všetky tieto vektorové veličiny sú dvojrozmerné, a sú v nasledujúcich vzťahoch:

$$\vec{v} = \frac{d\vec{x}}{dt} \quad (1)$$

$$\vec{a} = \frac{d\vec{v}}{dt}. \quad (2)$$

Ďalej na každého jedinca pôsobí sila \vec{F} , ktorá mu udeľuje zrýchlenie

$$\vec{a} = \frac{\vec{F}}{m}, \quad (3)$$

kde m je hmotnosť jedinca, od ktorej však v záujme zjednodušenia modelu budeme abstrahovať, preto tento vzťah vyjadríme len pomocou priamej úmernosti

$$\vec{a} \approx \vec{F}. \quad (4)$$

K simulácii budeme ešte potrebovať vzťah pre vzájomné silové pôsobenie dvoch jedincov. Budeme uvažovať len o odpudivom pôsobení. Ako je známe, v normálnom trojrozmernom priestore sila, pôsobiaca medzi dvoma telesami, klesá s druhou mocninou vzdialenosti týchto telies. V našom prípade dvojrozmerného priestoru však fyzikálna teória hovorí, že táto sila klesá iba s prvou mocninou vzájomnej vzdialenosti. Ako budeme vidieť neskôr, počas simulácie bude táto teória potvrdená. Teda odpudivú silu, ktorou pôsobí teleso 2 na teleso 1, vyjadríme ako

$$\vec{F} \approx \frac{\vec{x}_1 - \vec{x}_2}{|\vec{x}_1 - \vec{x}_2|^2}, \quad (5)$$

z čoho pre veľkosť \vec{F} vyplýva

$$|\vec{F}| \approx \frac{|\vec{x}_1 - \vec{x}_2|}{|\vec{x}_1 - \vec{x}_2|^2} = \frac{1}{|\vec{x}_1 - \vec{x}_2|}, \quad (6)$$

kde \vec{x}_1 je poloha telesa 1 a \vec{x}_2 je poloha telesa 2.

2.2. Princípy správania jedincov

Ako sme už uviedli, každý jedinec – žubrienka sa v simulácii riadi niekoľkými jednoduchými pravidlami, ktoré úplne určujú jeho správanie. V tejto práci sú študované niektoré vlastnosti správania kolónie, a to združovanie sa do väčších homogénnych celkov – krdľov, a vyhýbanie sa prekážkam, resp. dravcom, v rámci krdľa.

Inšpirácia k takejto simulácii pochádza z [1], kde autor pre účely realistického zobrazovania krdľa vtákov navrhol model, v ktorom každý vták – v našom prípade to bude žubrienka – vidí iba svoje najbližšie okolie, a riadi sa nasledujúcimi pravidlami:

- vyhýbaj sa kolízii s ostatnými jedincami (pravidlo č. 1)
- prispôsob rýchlosť ostatným jedincom (pravidlo č. 2)
- snaž sa držať sa v blízkosti ostatných jedincov (pravidlo č. 3)

Ako však uvidíme pri simulácii, pokiaľ implementujeme len tieto tri pravidlá, žubrienky vytvoria krdel', ale vzhľadom k absencii akéhokoľvek ďalšieho cieľa krdel' ostane stáť na mieste. Pokiaľ chceme, aby sa krdel' pohyboval, musíme pridať ďalšie pravidlo:

- snaž sa dostať sa do daného cieľového bodu (pravidlo č. 4)

Toto pravidlo však nemusíme prideliť všetkým jedincom. Stačí, ak ho pridelíme istej časti z nich, ostatné by ich mali nasledovať. Ako to funguje v skutočnosti, ukáže opäť simulácia. Ďalším rozšírením modelu bude pridanie dravca, ktorému sa žubrienky budú snažiť vyhýbať, a ktorého špeciálnym prípadom (pokiaľ bude mať nulovú rýchlosť) je vlastne statická prekážka. Preto pridáme ešte jedno pravidlo:

- vyhýbaj sa dravcom/prekážkam (pravidlo č. 5)

2.3. Implementácia

Simulácia je implementovaná v dvojrozmernom priestore, navrhnutom pre účely zobrazovania na počítačovej obrazovke tak, že počiatok jeho súradnicovej sústavy je v ľavom hornom rohu obrazovky a zobrazovaná časť tohto priestoru je od bodu [0,0] po bod [800,600]. Ako jednotky sú použité obrazovkové pixely (pix). Čas v simulácii je realizovaný pomocou simulačných krokov (sk). Každý sk trvá v našom prípade asi 55 ms.

Architektúra počítačového programu je objektovo orientovaná, keď každý objekt v simulácii (žubrienka, dravec) je podtypom základného typu `TSimulationObject` a každý takýto objekt musí definovať vlastné metódy, ktoré implementujú jeho správanie a jeho vykresľovanie na obrazovku. Algoritmus pre jeden simulačný krok v pseudo-C++ vyzerá nasledovne:

```
void TSimulation::Step(void)
{
    TSimulationObject SimObj;

    EraseScreen();
    for (i = 0; i < NumberOf(SimObjs); i++)
    {
        SimObj = SimObjs[i];
        SimObj.Simulate();
        SimObj.Paint();
    }
}
```

kde metóda `Simulate()` implementuje správanie daného objektu a metóda `Paint()` jeho vykreslenie na obrazovku.

Ako prvé potrebujeme implementovať fyzikálne zákony pohybu hmotného bodu. Toto vykonáva objekt `TPointObject`, ktorý je podtypom typu `TSimulationObject`. Obsahuje premenné `Pos`, `Vel` a `Acc` (všetky z nich sú typu `TVector2D`, čiže dvojrozmerné vektory), ktoré sú ekvivalentmi fyzikálnych veličín \vec{x} , \vec{v} a \vec{a} . Implementácia pohybu je nasledovná:

```
void TPointObject::Simulate(void)
{
    Pos += Vel;
    Vel += Acc;
}
```

t.j. pozícia je diskretnou integráciou rýchlosti a rýchlosť je diskretnou integráciou zrýchlenia. Z tohto tiež vyplýva, že hodnoty `Pos`, `Vel` a `Acc` sú dané v jednotkách `pix` (ekvivalent jednotky dĺžky `m`), `pix.m-1` (ekvivalent jednotky rýchlosti `m.s-1`) a `pix.m-2` (ekvivalent jednotky zrýchlenia `m.s-2`).

Aby sme postihli reálne možnosti pohybu žubrienok, musíme tiež obmedziť absolútne hodnoty rýchlosti a zrýchlenia, ktoré môžu dosahovať. Tieto hodnoty budeme v ďalšom texte označovať v_{max} a a_{max} . V praxi sa toto obmedzenie bude vykonávať tak, že pri požiadavke žubrienky na zmenu rýchlosti, resp. zrýchlenia, sa najprv skontroluje, či žiadaný vektor nie je dlhší než v_{max} , resp. a_{max} , a ak áno, jeho smer sa zachová, ale jeho dĺžka sa príslušne skráti.

Úplné správanie sa žubrienky, vrátane vyššie zmienených pravidiel, implementuje objekt `TPollywog`, ktorý je podtypom typu `TPointObject`. Implementácia prvých troch základných pravidiel vyzerá nasledovne:

```
void TPollywog::Simulate(void)
{
    const SightRadius = 30, // vzdialenosť viditeľnosti
          ForceConst = 100;
    TVector2D AcCenter, AcVelMatch, AcCollAv, // zrýchlenia prislúchajúce
                                                    // jednotlivým pravidlám
          FlockCenter, FlockVel, FlockForce; // stred, priemerná rýchlosť
                                                    // a odpudivá sila viditeľnej časti krdla

    TLinkedList InSight;
    TSimulationObject SimObj;
    double Distance;

    InSight = ObjectsInRadius(SightRadius); // v zozname InSight sú všetky
                                                    // videné objekty

    FlockForce = FlockVel = FlockCenter = TVector2D(0, 0);
    for (i = 0; i < NumberOf(InSight); i++)
    {
        SimObj = InSight[i];
        Distance = ObjDistance(SimObj); // vzdialenosť daného objektu

        FlockCenter += SimObj.Pos;
        FlockVel += SimObj.Vel;
        FlockForce += (Pos-Obj.Pos)/sqr(Distance)*ForceConst;

        FlockCenter /= NumberOf(InSight);
        FlockVel /= NumberOf(InSight);
    }
    AcCenter = FlockCenter-Pos-Vel;
    AcVelMatch = FlockVel-Vel;
    AcCollAv = FlockForce;
    Acc = AcCenter+AcVelMatch+AcCollAv; // výsledné zrýchlenie je súčet
                                                    // (priemer) zrýchlení pre jednotlivé
                                                    // pravidlá
    TPointObject::Simulate(); // na záver odsimulujeme pohyb
}
```

Pre každé zo zmiených troch pravidiel vypočítame zrýchlenie, ktoré sa dané pravidlo „snaží“ udeliť žubrienke. Pre pravidlo č. 1 je to premenná $AcCollAv$ (collision avoidance), pre pravidlo č. 2 $AcVelMatch$ (velocity matching) a pre pravidlo č. 3 $AcCenter$ (centering).

Podľa pravidla č. 3 sa žubrienka snaží dostať do stredu krdľa. Napriek tomu, že celý krdel nevidí, snaží sa dostať do stredu tej časti krdľa, ktorú vidí, a takto sa vlastne približuje aj k stredu celého krdľa (simulácia ukáže, že je to naozaj tak). Výrazom $FlockCenter-Pos$ je vyjadrená rýchlosť, ktorú by žubrienka potrebovala dosiahnuť, aby sa hneď v nasledujúcom simulačnom kroku ocitla v strede krdľa. Keďže však podľa fyzikálnych princípov rýchlosť sa nedá meniť priamo, ale iba pomocou zmeny zrýchlenia, $FlockCenter-Pos-Vel$ vyjadruje požadované zrýchlenie, pri ktorom by hneď v nasledujúcom simulačnom kroku dosiahla vyššie definovanú požadovanú rýchlosť. Takáto implementácia sa nezdá byť správna, keďže samozrejme nechceme, aby sa odrazu všetky žubrienky ocitli v jednom stredovom bode, ale na to, aby sa tak nestalo, slúžia jednak zvyšné dve pravidlá, ktoré túto tendenciu vyrovnávajú, a jednak fakt, že zrýchlenie žubrienky je obmedzené hodnotou a_{max} .

Podľa pravidla č. 2 sa žubrienka snaží vyrovnať svoju rýchlosť s rýchlosťami okolitých žubrienok. Robí to opäť priamočiaro, keďže rýchlosť, ktorú sa snaží dosiahnuť, je priemerná rýchlosť videnej časti krdľa $FlockVel$, požadované zrýchlenie pre toto pravidlo je $FlockVel-Vel$.

Podľa pravidla č. 1 sa žubrienka snaží vyhnúť sa kolízii s ostatnými žubrienkami. Toto je realizované tak, že každá z okolitých žubrienok na ňu pôsobí silou, vypočítanou podľa (5). Tieto odpudivé sily sa zrážajú, a výsledné požadované zrýchlenie je priamo úmerné odpudivej sile $FlockForce$ (podľa (4)). Keďže kvôli zjednodušeniu modelu sú v (4) a (5) použité len priame úmernosti, ktorých konkrétne konštanty nepoznáme, je tento nedostatok riešený vynásobením tejto hodnoty konštantou $ForceConst$, ktorej hodnota sa môže meniť, a ktorá vyjadruje, ako veľmi sa žubrienka „bojí“ kolízie s ostatnými.

Implementácia pravidla č. 5 sa dá dosiahnuť veľmi jednoduchým rozšírením implementácie pravidla č. 1. Podľa tohto pravidla sa žubrienka snaží vyhnúť aj kolízii s cudzími objektami (dravcami), a preto sa do odpudivého silového pôsobenia pridá aj pôsobenie dravcov. Avšak spravidla je kolízia s dravcom oveľa nebezpečnejšia než kolízia so susednou žubrienkou, preto pre dravce treba niekoľkonásobne zvýšiť konštantu $ForceConst$.

Pravidlo č. 4 implementujeme takisto modifikáciou pravidla č. 3. Namiesto hodnoty $FlockCenter$ sa ale žubrienka bude snažiť dostať do zadaného cieľového bodu. Avšak pre pravidlo č. 3 platí, že čím ďalej je žubrienka od stredu, tým silnejšie sa do neho snaží dostať (požadované zrýchlenie je tým vyššie). Táto požiadavka by ale nemala platiť pre pravidlo č. 4, pretože pri simulácii je potom vidno, že ako žubrienky dosahujú cieľový bod, ich rýchlosť sa mierne znižuje, pretože ich cieľový bod menej priťahuje. Pritom podľa reality by sa mali ku cieľu bližšie konštantnou rýchlosťou, a až priamo na ňom by mal krdel zastať. Preto pre pravidlo č. 4 zavedieme normalizáciu výsledného vektora, ktorá zabezpečí, že žubrienky si budú udržiavať smer k cieľovému bodu, ale bude ich priťahovať rovnako silno, akokoľvek budú od neho vzdialené. Ako už bolo napísané, pravidlo č. 4 nemusíme zaviesť pre všetky žubrienky, stačí ak sa podľa neho budú riadiť niektoré z nich, ostatné ich budú nasledovať.

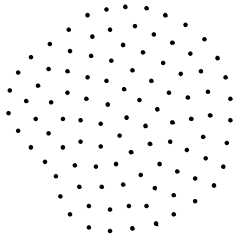
3. Výsledky

Základné hodnoty, pri ktorých bola spúšťaná simulácia, boli: na začiatku bolo vygenerovaných 100 jedincov žubrienok v oblasti ľavého horného rohu obrazovky, t.j. v oblasti $[0,0] - [200,200]$. Ich dosah videnia bol 30 pix. Konštantu $ForceConst$ mala hodnotu 100 pre obyčajné žubrienky a 300 pre dravce. V prípadoch, keď sa simulovalo s cieľovým bodom, bol tento bod $[700,500]$. $a_{max} = 0,5 \text{ pix.sk}^{-2}$, $v_{max} = 5 \text{ pix.sk}^{-1}$.

Simulačný krok trval približne 55 ms, t.j. za jednu sekundu prebehlo približne 18 sk.

3.1. Tvorba krdľa

Najprv sa budeme zaoberať simuláciou s použitím len prvých troch pravidiel správania. Pokiaľ spustíme simuláciu s vyššie uvedenými základnými hodnotami, žubrienky vytvoria krdel:

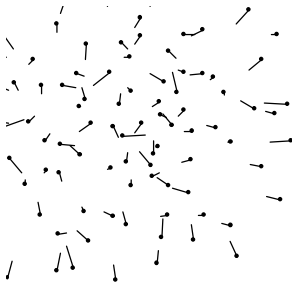


Ako je vidno, žubrienky sa ustálili v takmer úplnej rovnováhe, keď všetky z nich majú takmer nulovú rýchlosť a ich vzájomné vzdialenosti sú takmer konštantné. V skutočnosti táto rovnováha nie je dokonalá, krdel má tendenciu sa mierne pohybovať a preskupovať. Základná tendencia jeho preskupovania je vytvoriť dokonale kruhovú oblasť. Skutočne sa dá pozorovať, že keď necháme simuláciu bežať dosť dlho, napokon sa krdel dostane do úplne kruhového tvaru. Táto tendencia je omnoho rýchlejšia v prípade, že žubrienkam povolíme dosah videnia na celý krdel, v tom prípade vytvoria dokonale kruhový tvar už po niekoľkých sekundách simulácie.

Čo sa týka priemernej vzdialenosti medzi dvoma najbližšími žubrienkami v ustálenom stave, simulácia ukázala, že nezávisí od dosahu videnia jednotlivých žubrienok, ani od počtu žubrienok v kolónii. Závisí ale samozrejme od hodnoty `ForceConst`, a síce je priamo úmerná jej druhej odmocnине. Toto jednak vyplýva z jednoduchých výpočtov, ale dá sa to potvrdiť aj experimentom. Niekoľko nameraných hodnôt je v tabuľke

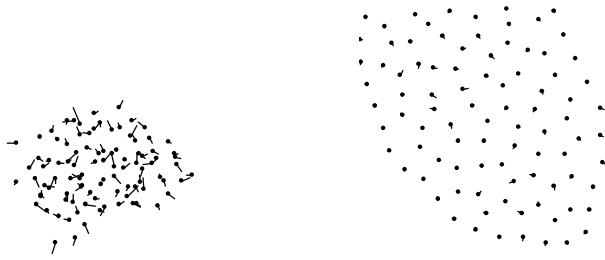
<code>ForceConst</code>	vzd. susedov
0	0 pix
10	5,8 pix
50	14 pix
100	17 pix
200	23 pix

Keď `ForceConst` presiahne hodnotu, pri ktorej vzdialenosť susedov je rovná dosahu videnia, žubrienky v krdli už na seba „nedovidia“, a preto sa ich počiatočná konfigurácia rozprácha:



Každá z nich nadobudne určitú rýchlosť vďaka odpudivej sile, ktorá na ňu pôsobila na začiatku, a potom sa už len hýbe touto rýchlosťou v priestore, pretože nevidí okolo seba žiadne iné žubrienky, a teda na ňu nepôsobia žiadne zrýchlenia, vyplývajúce z prvých troch pravidiel.

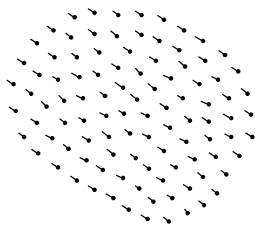
Doposiaľ sme v simulácii pre pravidlo č. 1 využívali vzťahy (5) a (6), podľa ktorých vzájomné odpudivé silové pôsobenie medzi žubrienkami klesalo s prvou mocninou ich vzdialenosti. Zistili sme, že skutočne v takomto stave sa dosahuje rovnováha medzi snahou o priblíženie sa k stredu krdľa a strachom zo zrážky s inými jedincami. Vyskúšajme však, čo by sa stalo, keby sme odpudivé silové pôsobenie nechali klesať s druhou mocninou vzdialenosti, resp. neklesať so vzdialenosťou vôbec:



V prípade, že sila klesá s druhou mocninou vzdialenosti, žubrienky síce vytvoria zhuk, ale v rámci tohto zhuku sa nikdy neustália v rovnovážnej polohe, ich pohyb je chaotický, a nevzniká tu ani tendencia udržiavať medzi sebou konštantné vzdialenosti. Prípád, že sila so vzdialenosťou neklesá vôbec, je dosť podobný štandardnému prípadu, keď sa dosiahne rovnováha, avšak možno si všimnúť, že jednak krdel nenadobúda kruhový tvar, jednak žubrienky síce majú vzdialenosti k najbližším susedom pomerne dobre sa zhodujúce, ale nedokážu si ich udržať, pretože ich polohy nie sú stabilné, každá maličká porucha v krdli vyvolá vlnu iných porúch, a celkovo je teda konfigurácia nestabilná.

3.2. Cesta k cieľovému bodu

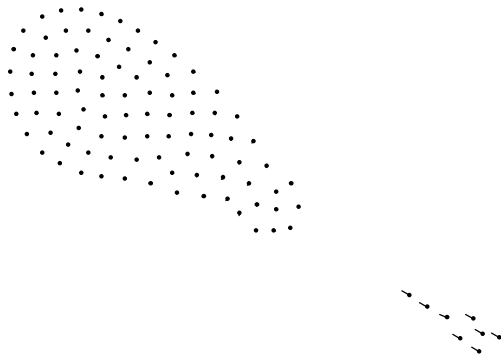
Opäť na začiatku ukážeme ideálny prípad, keď sme parametre ponechali rovnaké ako v simulácii len s tromi pravidlami, ale pridali sme ešte štvrté pravidlo – navigáciu k danému cieľovému bodu. Pripomeňme ešte, že v tomto prípade používali 4. pravidlo všetky žubrienky, takže sme vlastne do modelu zaviedli akési centrálné riadenie, ktoré sa v tomto prípade v živej prírode vyskytuje málokedy (málokedy sa stane, aby všetky jedince naraz vedeli, ktorým smerom sa majú pohybovať, keď sa chcú dostať napr. k potrave). Tento nedostatok však neskôr odstránime. Takto teda vyzerá krdel v pohybe:



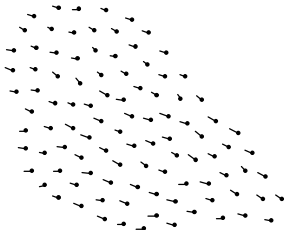
Ako vidíme, opäť sa vytvoril pekný homogénny útvar, mierne natiahnutý v smere pohybu. Jeho stabilita je veľmi dobrá, presuny žubrienok sú zriedkavé a slúžia len na zdokonalenie tvaru útvaru. Opäť platí, že ak žubrienkam povolíme dostatočne veľkú vzdialenosť videnia, napr. 300 pix, útvar sa rýchlo ustáli v dokonalom kruhu a v takomto tvare bez akýchkoľvek porúch putuje až do cieľa.

Zaoberajme sa teraz zaujímavejším prípadom, keď informáciu o polohe cieľa má len istá časť všetkých žubrienok. Tieto žubrienky budeme ďalej označovať ako aktívne, ostatné ako pasívne. Vyskúšali sme prípady, keď aktívnych žubrienok bolo 1/2, 1/10, 1/20 počtu všetkých. V každom z týchto prípadov sa krdel nakoniec dostal do cieľového bodu, avšak už pri 1/10 aktívnych žubrienok bol jeho pohyb extrémne pomalý. Toto je spôsobené tým, že tak, ako aj pasívne žubrienky, aj aktívne žubrienky sa riadia všetkými pravidlami správania, a preto sa musia snažiť prispôbovať svoju rýchlosť okolitým pasívnym žubrienkam a snažiť sa držať sa v ich blízkosti, a preto je pohyb k cieľu pomalý. V nasledujúcom texte, pokiaľ nebude napísané inak, budeme vykonávať simuláciu pri 1/10 žubrienok aktívnych.

Ako sme už spomínali vyššie, pri implementácii pravidla č. 4 sa zrýchlenie, prislúchajúce tomuto pravidlu, normalizovalo (teda jeho vektor mal smer ku cieľovému bodu, ale veľkosť mal vždy 1 pix.sk²). Prvou intuitívnou predstavou, ako vyriešiť pomalý pohyb krdľa, je zvýšiť normalizovanú hodnotu tohto zrýchlenia, aby pre aktívne žubrienky zrýchlenie, vypočítané podľa pravidla č. 4, prevládalo nad ostatnými zrýchleniami od ostatných pravidiel. Skutočne, ak toto zrýchlenie zvýšime dvojnásobne, môžeme sledovať, že krdel sa už hýbe rýchlejšie, ale stále je jeho rýchlosť nízka. Ak však ešte viac zvyšujeme túto hodnotu, napr. na štvornásobok, sledujeme, že aktívne žubrienky sa po chvíli od krdľa oddelia (vzhľadom k tomu, že zvyšný krdel sa im už nestačí prispôbovať a ostane pozadu) a plávajú len samotné k cieľovému bodu. Zvyšok krdľa, keďže už nemá žiadnu hnaciu silu, ostane stáť na mieste:



Na to, aby sa nestávalo, že sa aktívne žubrienky oddelia od krdľa, by pasívne žubrienky mali nejakým spôsobom vedieť, ktorá žubrienka je aktívna, a prispôsobiť rýchlosť skôr aktívnej žubrienke, než okolitým pasívnym. Avšak v živej prírode, a hlavne v takých nízkych evolučných štádiách, ako sú žubrienky, je nereálne, aby žubrienka vedela, či jej sused vie, kde je potrava (a teda že ho má nasledovať). Ale dá sa zaviesť jednoduchá modifikácia pravidla č. 2, podľa ktorej sa daná žubrienka nebude snažiť prispôsobiť svoju rýchlosť priemernej rýchlosti ostatných (videných), ale jedincovi s najväčšou absolútnou hodnotou rýchlosti. Predpokladom totiž je, že práve aktívne jedince zavádzajú do krdľa pohyb, a preto je pravdepodobné, že najrýchlejšie sa pohybujúci jedinec bude mať ku svojmu rýchlemu pohybu nejaký dôvod, t.j. buď je aktívny alebo ovplyvnený pohybovou vlnou prichádzajúcou od nejakého aktívneho. Po tejto modifikácii vyzerá pohybujúci sa krdel nasledovne:

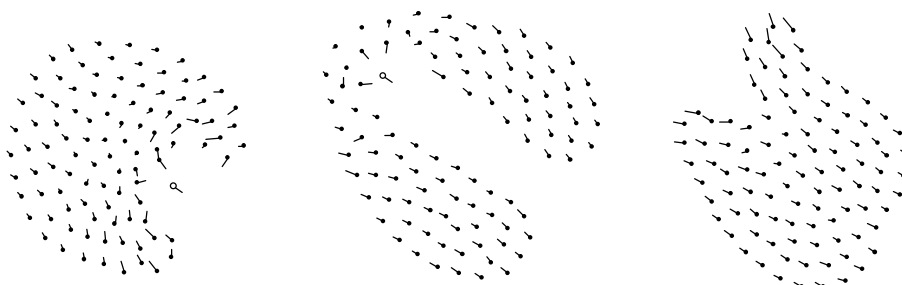


Dosiahli sme teda to, že sa nerozpadne na časť aktívnych a pasívnych žubrienok. Jednu nevýhodu, ktorú má toto riešenie, je, že krdel je pomerne nestabilný, každá malá výchylka pohybu vyvolá celú pohybovú vlnu. Svoj účel však táto modifikácia splnila, keďže celý krdel dosiahne cieľový bod a ustáli sa na ňom v pomerne stabilnom útvere.

3.3. Vyhýbanie sa dravcom/prekážkam

V ďalšom budeme predpokladať, že žubrienky sa snažia dostať do cieľového bodu [700,500], a že všetky z nich sú aktívne.

Pridajme do simulácie žubrienku – dravca. Jediným pravidlom, ktorým sa žubrienky riadia vo vzťahu k dravcovi, je vyhýbanie sa mu. Dosahujú to použitím pravidla č. 5 (jeho implementácia viď vyššie). V tomto prípade bude dravec začínať z cieľového bodu žubrienok [700,500] a bude sa pohybovať smerom ku krdľu (dravec je zobrazený ako žubrienka s väčšou a prázdnu hlavičkou):

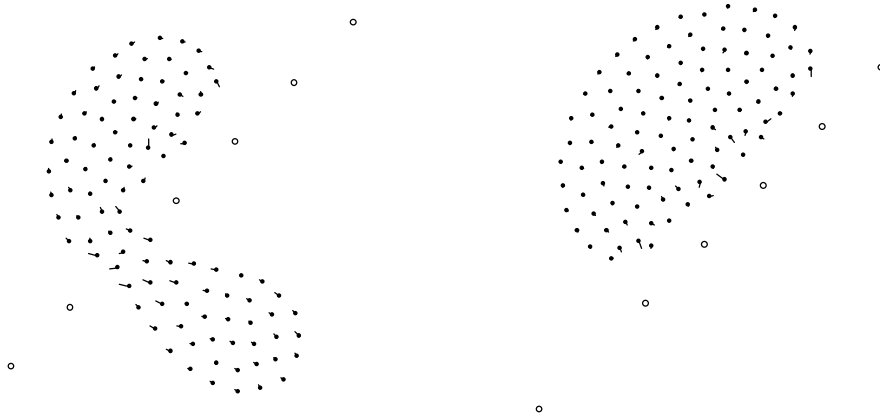


Na obrázkoch je vidieť, ako sa krdel pred dravcom rozdelí na dve časti, ktoré ho v bezpečnej vzdialenosti obopávajú, a následne sa krdel opäť spojí.

Keby sme chceli dosiahnuť, aby krdel obopľával dravca vo väčšej vzdialenosti, nestačí nám na to zvýšiť konštantu $ForceConst$ jeho odpudivej sily. Žubrienky ho obopľávajú vždy maximálne v takej vzdialenosti, aký je dosah ich videnia. Ak je dravec ďalej, žubrienky ho proste nevidia, a preto ich už viac neodpuďuje. Ak však zvýšime dosah videnia každej žubrienky, podarí sa nám dosiahnuť

efekt opatrnejšieho vyhýbania sa, ale ak ho zvýšime tak, že žubrienky budú vidieť príliš ďaleko, môže sa stať, že žubrienky tým, že dravca budú vidieť už z diaľky, zastanú pred ním (v prípade, že by sa nepohyboval; ak by sa pohyboval smerom k nim, začal by ich zatláčať naspäť) a nie sú schopné ho obísť. Toto je spôsobené lokálnym vnímaním sveta u každého jedinca, z čoho vyplýva, že celý krdeľ nie je schopný riešenia, ktoré by mohlo určiť centrálné riadenie – teda obísť dravca veľkým oblúkom.

Podobný efekt môžeme pozorovať, ak postavíme žubrienkam do cesty „hradbu“ z prekážok – dravcov, ktorá má jediný otvor, ktorým sa žubrienky môžu dostať na opačnú stranu. V prípade, že sa otvor nachádza v smere, ktorým žubrienky plávajú, sú schopné sa ním pretlačiť, v opačnom prípade ho nikdy nenájdu:



4. Záver

Simulácia populácie žubrienok ukázala, že globálne koordinované správanie v populácii môže vzniknúť na základe lokálneho vnímania sveta a jednoduchých pravidiel, ktoré ovplyvňujú správanie daného jedinca v závislosti na jeho susedoch v lokálnom okolí. Napriek tomu – ako je tomu ostatne aj v živej prírode, keď živočíšny druh nie je schopný vyššej rozumovej činnosti, resp. komunikácie – takýto model nie je schopný riešiť zložitejšie úlohy, napr. nájsť bezpečnú cestu k potrave, ak je priamočiara cesta zablokovaná.

5. Literatúra

- [1] C. Reynolds: Flocks, Herds and Schools: A Distributed Behavioral Model. *Computer Graphics* 21 (1987) 25-34; v elektronickej podobe na <http://www.cs.toronto.edu/~dt/siggraph97-course/cwr87/>