

VŠB – TECHNICKÁ UNIVERZITA OSTRAVA
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
KATEDRA INFORMATIKY

Simulace davu

Semestrální projekt

Vypracoval:
Pavel DRÁBEK

Vedoucí práce:
Ing. Martin NĚMEC, Ph.D.

Obsah

1	Úvod	2
1.1	Cíl projektu	2
2	Mravenčí kolonie	3
2.1	Popis metody	3
3	Model Boid	4
3.1	Popis modelu	4
4	Implementace	7
4.1	Základní model Boid	7
4.2	Rozšíření o cíl	7

1 Úvod

Simulace davu má za cíl věrně napodobit chování velké množiny objektů. Zdánlivě složité chování však může být dosaženo definováním několika jednoduchých pravidel každého jedince. Chování samotného jedince se může jevit jako zmatené, avšak velká skupina stejně naprogramovaných jedinců může vykazovat inteligentní chování.

Takové chování uplatňujeme nejen ve filmovém a herním průmyslu, ale i v architektuře, nácviku vojenských strategií, navrhování evakuačních plánů, simulování požárních poplachů či chování robotů. Díky moderním technologiím můžeme napodobit chování davu, které by nás stálo nejen velké množství času, ale i finančních prostředků. Například filmová bitva velkých armád, stadion plný fanoušků apod. Můžeme však vytvořit i takové chování, se kterým bychom se v reálném životě nikdy nesetkali - ve filmu *Drákula: Neznámá legenda* (2014) ovládá hlavní postava desetitisíce netopýrů. [1]

Takovéto systémy většinou vykazují chaotické chování, protože nepatrná změna hodnoty jednoho parametru byť jen zaokrouhlením bude mít dopad na pozdější výsledek. Nicméně dá se zajistit i chování, které pro stejnou situaci chaotické chování nevykazují.

1.1 Cíl projektu

TODO

2 Mravenčí kolonie

TODO

2.1 Popis metody

TODO

3 Model Boid

Počítačový model pro řízení hejna zvířat jako jsou ptáci a ryby popsal Craig Reynolds v roce 1986. Jednotlivé jedince nazval Boidy. Samotný model pak prezentoval na konferenci SIGGRAPH o rok později.

Od roku 1987 byl model Boid použit pro velké množství simulovaného chování hejna. Prvním, veřejně známým použitím byl film *Batman se vrací* (1992) od Tima Burtona. Obsahuje počítačově simulované chování hejna netopýrů a skupiny tučňáků, které bylo vytvořeno upravenou verzí modelu Boid. [2]

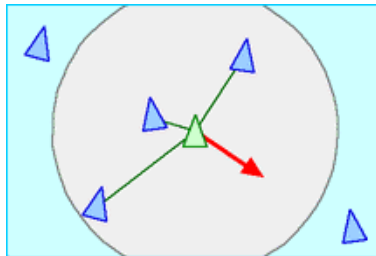
3.1 Popis modelu

Boid pracuje s agenty reprezentovanými pozicí a rychlostním vektorem. Model lze aplikovat pro pohyb ve dvou i třech rozměrech.

Každý agent se řídí třemi základními pravidly, podle kterých upraví svůj rychlostní vektor každý časový úsek. Jedná se o Separaci, Vyrovnání a Kohenzi.

1. **Separace,**

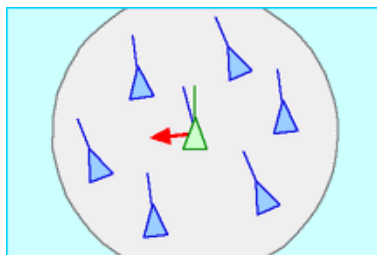
Agent se snaží nesrazit s jinými agenty. Upravuje proto svůj směrový vektor, aby se dostal od příliš blízkých agentů.



Obrázek 1: Separace (Separation) [2]

2. **Vyrovnání,**

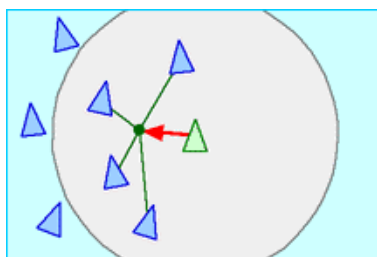
Agent se chce pohybovat stejným směrem a rychlostí jako jeho sousedé. Upravuje svůj vektor podle průměru z jeho okolí.



Obrázek 2: Vyrovnání (Alignment) [2]

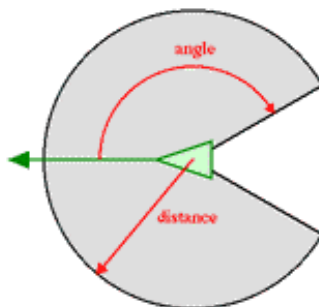
3. Koheze,

Agent upravuje svůj rychlostní vektor tak, aby se dostal do pomyslného středu všech sousedních agentů.



Obrázek 3: Koheze (Cohesion) [2]

Každý boid má přímý přístup ke všem objektům ve scéně, nicméně tento model vyžaduje, aby reagoval pouze na své okolí. To je definováno maximální vzdáleností a zorným úhlem agenta. Můžeme přidávat i další pravidla jako třeba minimální vzdálenost nebo úhel mezi dvěma agenty apod. Pravidla, podle kterého jedinec rozlišuje okolí, si můžeme představit jako smysly daného zvířete.



Obrázek 4: Okolí agenta [2]

Jak bylo na začátku kapitoly naznačeno, model můžeme různě zozširovat. Sám Reynolds přišel rok po představení svého modelu na stejné konferenci s rozšířením pro vyhýbání se překážkám. Detailněji vše vysvětluje ve své práci Not bumping into things [3]. V podstatě se jedná o úpravu výsledného vektoru rychlosti pokud se před jedincem nachází nějaký objekt.

Pokud jednotlivým pravidlům přiřadíme váhy, můžeme následnými úpravami těchto vah měnit chování konkrétního jedince či celého hejna. Boidi tak můžou uzpůsobovat své chování v závislosti na prostředí. Například je-li v okolí dravec, budou mít mezi sebou větší rozestup, hlídat si větší okolí, rychleji reagovat na změnu směru atd.

4 Implementace

Implementace probíhala v jazyce *C#* a vývojovém prostředí Unity3D [4], ve kterém mám dlouholetou zkušenost.

4.1 Základní model Boid

Reynolds ve svém článku neuvádí jakým stylem jednotlivé pravidla řídit ani jaké váhy použít. Byla tedy vytvořena váha pro každé pravidlo, díky kterému můžeme řídit prioritu těchto pravidel.

Pro lepší vizuální představu jak model Boid funguje byl vytvořen jednoduchý editor, ve kterém je možné v reálném čase nastavit upravit váhy jednotlivým pravidlům a vidět změnu v chování hejna, aniž by bylo potřeba pouštět celou simulaci znovu. To umožnilo rychleji dosáhnout požadovaného chování hejna. Pro správné nastavení vah by teoreticky šel použít i evoluční algoritmus, avšak na tak jednoduchém příkladě není jasné jakým způsobem určovat fitness.

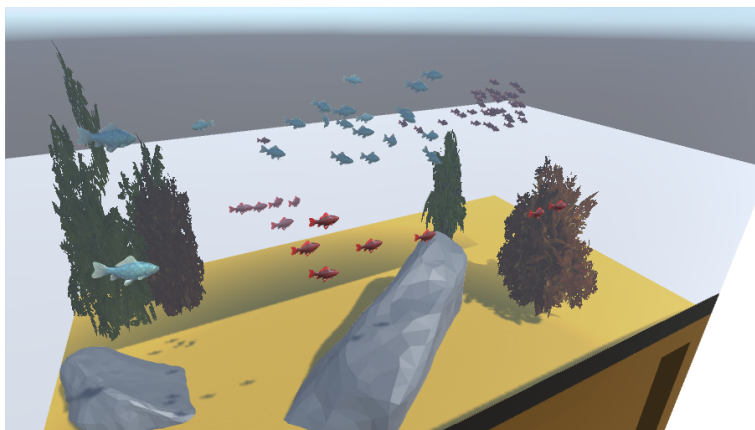
Podařilo se dosáhnout fungujícího modelu hejna, které však nekontrolovaně cestovalo po světě.

4.2 Rozšíření o cíl

Model Boid popisuje pouze chování jednotlivce v hejnu. Pro potřebu kontrolovat pohyb celého hejna, byl agent rozšířen o další pravidlo. Každý agent se snaží dostat do cíle, který je reprezentován pozicí bodu ve světě.

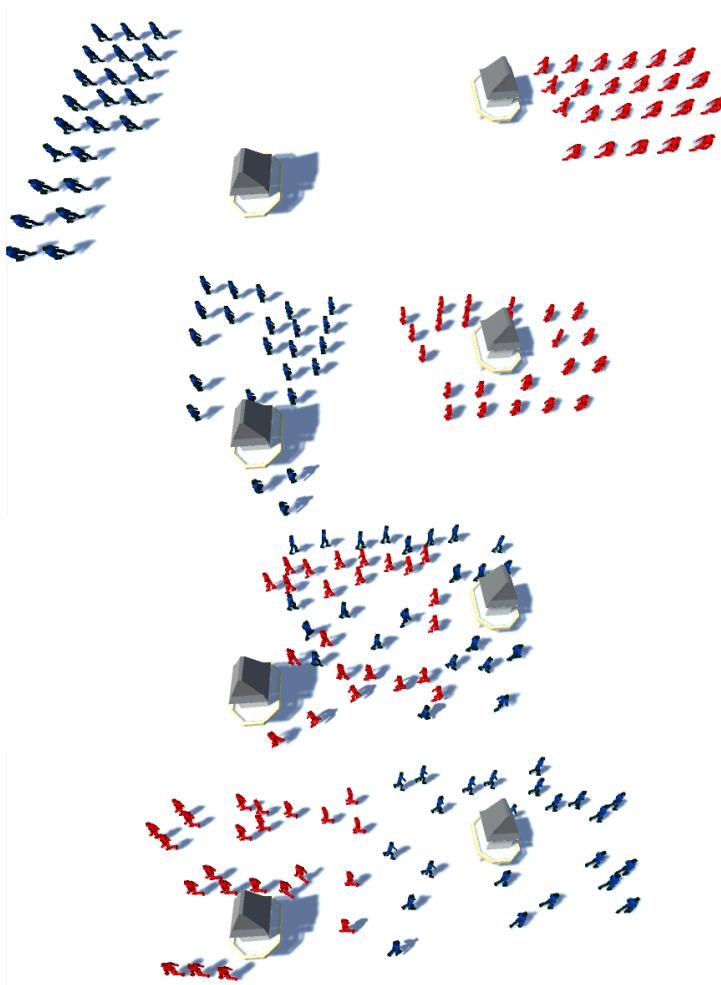
Od toho pravidla vyžadujeme, ať přitahuje agenta k cíli, ale neznehodnotí zbylé 3 pravidla s narůstající vzdáleností od cíle. Normalizování směrového vektoru k cíli se může zdát jako krok správným směrem. V praxi se však ukázalo, že když je takový agent velmi blízko svého cíle, normalizace vektor naopak zvětší. To má za následek příliš agresivní snahu jedince dostat se do cíle. Proto velikost směrového vektoru od jedince k cíli pouze ořízneme na hodnotu vzdálenosti, od které má být agent méně agresivní.

Pomocí nového pravidla je možné kontrolovat pozici hejna a pozicovat ho dle svých potřeb. Pro názornou ukázkou vznikla scéna s akváriem a rybičkami. V určitém časovém intervalu nastavuji všem agentům (rybám) náhodně vygenerovaný cíl uvnitř akvária.

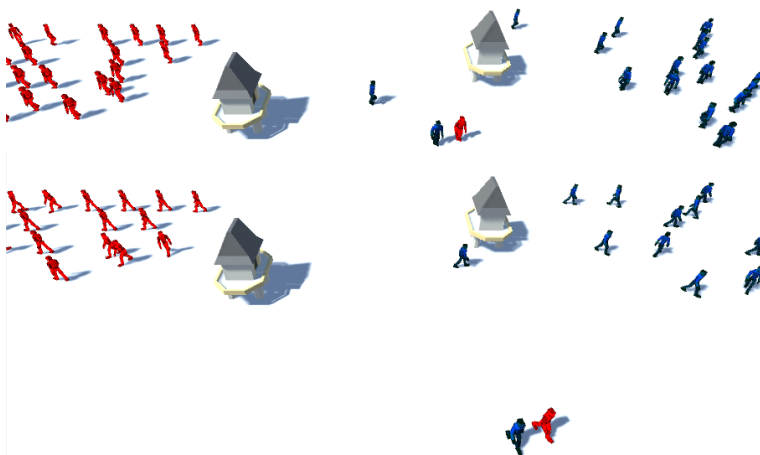


Obrázek 5: Hejno 20 modrých a 50 červených ryb

Na obrázku 4.2 můžeme vidět 2 různě početné a nastavené skupiny agentů. První je skupina 20 velkých modrých ryb, které jsou pomalé, drží se dál od sebe (velká váha u separace) a pomalu mění svůj směr. Druhá skupina obsahuje 50 malých červených ryb, které jsou rychlejší, agilnější, ale více se drží pohromadě (velká váha u koheze).



Obrázek 6: 2 skupiny agentů s detekcí překážek v časech $t = \{0, 10, 20, 30\}$



Obrázek 7: Srážka a zamknutí dvou agentů

Reference

- [1] Dracula Untold Trailer. In: *Youtube* [online]. [cit. 2017-05-03]. Dostupné z: <https://www.youtube.com/watch?v=aWqecTTuE>
- [2] REYNOLDS, Craig. Boids: Background and Update. In: *Reynolds Engineering & Design* [online]. 1995 [cit. 2017-05-03]. Dostupné z: <http://www.red3d.com/cwr/boids/>
- [3] REYNOLDS, Craig. Not Bumping Into Things. In: *Reynolds Engineering & Design* [online]. 1988 [cit. 2017-05-03]. Dostupné z: <http://www.red3d.com/cwr/nobump/nobump.html>
- [4] *Unity 3D* [online]. [cit. 2017-05-04]. Dostupné z: <https://unity3d.com>

Seznam obrázků

1	Separace (Separation) [2]	4
2	Vyrovnění (Alignment) [2]	5
3	Koheze (Cohesion) [2]	5
4	Okolí agenta [2]	6
5	Hejno 20 modrých a 50 červených ryb	8
6	2 skupiny agentů s detekcí překážek v časech $t = \{0, 10, 20, 30\}$	9
7	Srážka a zamknutí dvou agentů	10