LETECKÁ BITVA NA KKY

Speciální zadání semestrální práce – *MATLAB*

Letní semestr, 2014

Katedra Kybernetiky

Západočeská Univerzita v Plzni

Obsah

1 POPIS

Jedná se o tzv. “programovací hru”, tedy o hru, které se místo hráčů účastní jednotlivé programy/algoritmy. V této úloze je zpracována simulace letecké bitvy. Engine hry simuluje pohyby letadel v prostoru a jejich souboje. Pro jednoduchost je simulace 2D v pohledu ze shora.

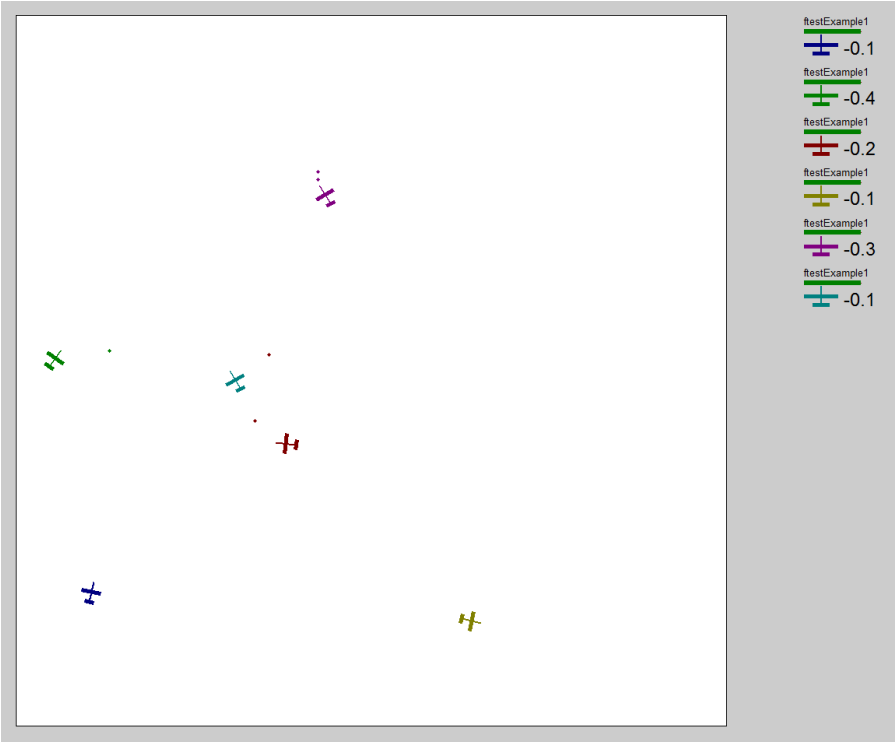
Simulace trvá určitou dobu, N simulačních kroků. Hráče reprezentuje funkce pro ovládání letadla. Tato funkce je volána každý krok simulace. Funkce obdrží dostupná aktuální data. Řídicí funkce vrací řídící signály – nastavení směrovky, nastavení plynu, požadavek na střelbu a uživatelská data.

Cílem studenta je navrhnout algoritmus letadla, který v daném čase zasáhne či úplně sestřelí, co nejvíce soupeřů, zároveň bude co nejméně zasažen a nespotřebuje přitom moc munice.

Na obrázku pod textem je vizualizace simulace. Letadla se pohybují ve vymezeném prostoru dle vytvořeného algoritmu.

Simuluje se několik letadel zároveň na jednom bojišti. Letadla se nesrazí, ale proletí skrz („nad/pod“) sebou. Simulovaný prostor je periodický, takže letadla nemohou opustit vizualizovaný prostor, ale objeví se na druhé straně.

Po celou dobu simulace je jednotlivým hráčům/algoritmům počítáno skóre. Jeden bod je přičten za každý zásah protivníka, dále jeden bod navíc za jeho zničení. Desetina bodu je odečtena za každý výstřel, je tedy třeba s municí nakládat rozumně. Sestřelené letadlo – bez životů, je na 30 kroků simulace (cca 3 sekundy) odstaveno. Po 30 krocích, dojde opět k jeho aktivaci na náhodně vygenerovaném místě a pokračuje v souboji s plnými životy.


ftestExample1
-0.1
ftestExample1
-0.4
ftestExample1
-0.2
ftestExample1
-0.1
ftestExample1
-0.3
ftestExample1
-0.1

2 PŘILOŽENÉ FUNKCE

2.1 Hlavní funkce pro spuštění simulace

2.1.1 engine1.m

Hlavní funkce simulace. Simuluje chování letadel, vyhodnocuje zásahy, počítá skóre a provádí vizualizaci soubojů. Doporučujeme tuto funkci prostudovat, než začnete psát vlastní algoritmus.

[S] = engine1(fFuncs, maxTime, RT)

fFuncs – vektor, jednotlivá letadla (algoritmy), viz **initPlanes.m** níže

maxTime – doba letu – doba simulace souboje v sekundách

RT – koeficient rychlosti simulace. Celé číslo. 2 = 2x rychlejší simulace. Pro *RT*=0 bude simulace provedena zrychleně bez grafického výstupu, to se hodí pro simulaci delších soubojů, které jsou potřeba pro přesnější porovnání úspěšnosti jednotlivých algoritmů.

2.1.2 initPlanes.m

Makro pro inicializaci letadel – jednotlivých algoritmů. V makru definujeme jednotlivá letadla a jejich barvy v RBG. Po spuštění bude vytvořen vektor letadel *fFuncs*.

```
fFuncs(1).pilot = @ftestExample1;
fFuncs(1).color = [0 0 128]/256;

fFuncs(2).pilot = @ftestExample2;
fFuncs(2).color = [0 128 0]/256;
```

2.1.3 run.m

Spuštění simulace letecké bitvy. Spustí makro *initPlanes.m* a pak simulaci na 20 sekund s nezrychlenou vizualizací.

2.2 Vzorová funkce letadla ftestExample1.m

[throttle, steering, firing, userData] = ftestExample1(planes, id, bullets, t, userData)

Funkce, která řídí chování letadla. Vaším úkolem je naprogramovat právě tuto funkci. Funkce musí mít výše uvedené vstupní a výstupní parametry. Jejich význam je následující:

2.2.1 Vstupní parametry:

planes – vektor struktur všech letadel v prostoru (včetně vašeho)

id – identifikační číslo vašeho letadla – do vektoru *planes*

bullets – všechny aktuálně letící střely v bitevním poli – vektor struktur

t – krok simulace (celé číslo od 1 výše)

userData – data z předchozího kroku simulace, která si definoval uživatel, na začátku *empty*

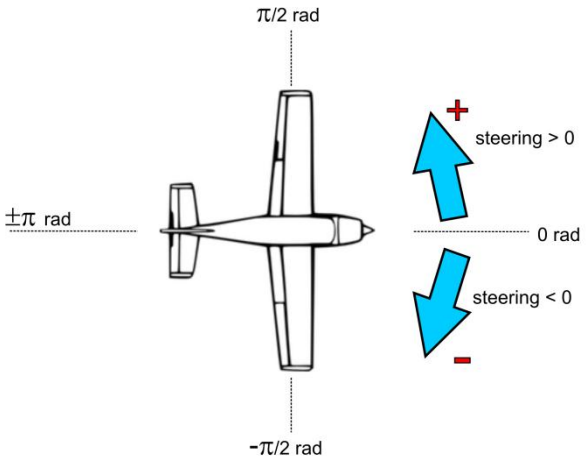
2.2.2 planes()

- planes(k).score - aktuální skóre
- planes(k).health – počet zbývajících životů
- planes(k).damage – poškození způsobené zásahem (síla zbraní)
- planes(k).xy – souřadnice v letovém prostoru (vektor 1x2)
- planes(k).angle – aktuální úhel natočení letadla v radiánech (směr kam letí)
- planes(k).v – aktuální rychlost letadla
- planes(k).throttle - poloha plynu
- planes(k).steering – úhel směrovky – zatáčení
- planes(k).fire - letadlo střílí
- planes(k).userData – uživatelská data
- planes(k).color - barva hráče
- planes(k).setup - aktuální nastavení
- planes(k).death – bojující/zničený

2.2.3 Výstupní parametry:

throttle – poloha plynu letadla v rozmezí <0, 1>, nula je nejmenší rychlost, nikoli nulová, jednička reprezentuje maximální rychlost.

steering – natočení směrovky v rozmezí <-1, 1>. Relativní natočení v radiánech vůči letadlu.



Kladné číslo – zatáčet doleva

Záporné číslo – zatáčet doprava

Nula – letět rovně

firing – požadovaná střelba (1 střílet, 0 nestřílet)

userData – uložiště pro libovolná data uživatele, která chce využít v příštím kroku simulace (univerzální paměť), na začátku je *empty*

2.2.4 Volitelné nastavení - setup

Každé letadlo může využít jeden z bonusů:

1. Silnější střelu o 25% (větší poškození spoluhráče)
2. Zvýšené zdraví o 25%
3. Vyšší rychlost o 25%
4. Lepší manévrovatelnost o 25 %

Setup se nastavuje ve funkci letadla následujícím způsobem:

```
%SETUP MODE
if length(varargin) == 1
    varargout{1} = 1,2,3 nebo 4
else
%DRIVER MODE
```

Mód je aktivní po celou dobu života letadla, pokud je letadlo sestřeleno a opět se „narodí“, engine volá funkci setup znovu a je možné zvolit jiné nastavení.

2.3 Pomocné funkce

Funkce, které můžete / nemusíte použít. Tyto funkce by Vám měly usnadnit tvorbu algoritmu vašeho letadla. Některé z těchto funkcí využívá rovněž samotný engine.

2.3.1 unwrapAngle.m

a = unwrapAngle(a)

Funkce „drží“ úhel letadla v intervalu <-π ,π >.

2.3.2 unwrapXY.m

[x,y] = unwrapAngle(x,y)

nebo

xy = unwrapAngle(xy)

Funkce udržuje souřadnice x,y ve velikosti okna - periodickém prostoru. Využívá globální proměnné nastavené enginem **global** SIZE_X SIZE_Y, ty můžete využít i vy.

2.3.3 angleDiff.m

a = angleDiff(a1, a2)

Funkce Vám spočte diferenci (rozdíl) mezi dvěma zadanými úhly v radiánech, respektuje periodicitu úhlů.

2.3.4 compAngle.m

a = compAngle(dx)

Funkce Vám vypočte úhel v radiánech z rozdílu xy (headingPoint - actualPoint).

Například: compAngle([1,1]), dostaneme π/4 (45 stupňů)

2.3.5 pointDiff.m

[dx,dy] = pointDiff(xyA, xyB)

nebo

dxy = pointDiff(xyA, xyB)

Funkce Vám spočte rozdíl mezi dvěma body A a B s respektováním periodického prostoru.

2.4 Příklad spuštění

Rozbalte obsah přiloženého archivu do nějakého adresáře a tento adresář nastavte v Matlabu jako aktuální. Pak stačí spustit **run**

2.5 Bodové hodnocení

Za každý výstřel je odečteno 0.1 bodu. Za každý zásah nepřítele je přičten 1 bod a za každého zničeného nepřítele je přičten další 1 bod. V případě 25% zbraňového bonusu (setup=1) je za zásah připočteno 1,25 bodu.

2.6 Testování algoritmu

Jak již bylo výše uvedeno, je možné nastavit na počátku čtyři nastavení. Pro vámi vytvořený algoritmus otestujte všechny tyto nastavení, proveďte vyhodnocení, graficky zpracujte a vložte do referátu. Jelikož výsledek simulace z části ovlivňuje náhoda, bude nutné udělat několik delších simulací, které statisticky vyhodnotíte.

Podobně budete postupovat v případě porovnávání různých variant vašich algoritmů během vývoje. Vše dokumentuje v referátu.

2.7 Rady a tipy

Nezkoušejte hned naprogramovat nějakou super-funkci. Je lepší postupovat v menších krocích, a testovat jednotlivé nápady zvlášť. Začít něčím velmi jednoduchým a teprve až pochopíte lépe, jak co funguje, můžete se pustit do větších složitostí. Dělejte jednotlivé verze, které pak můžete mezi sebou porovnávat. Pro lepší pochopení co se kde děje doporučujeme použít Matlab debugger. Pro lepší pochopení úlohy můžete nahlédnout i do samotného enginu. V případě potřeby ho můžete i upravovat (sbírat různé informace či statistiky, odstranit vliv náhody pro lepší ladění), pro finální turnaj však bude použit původní „oficiální" engine. Neváhejte se na nás obrátit s vašimi dotazy.

2.8 Důležité termíny

Povinné konzultace: V rámci jednotlivých termínů cvičení, proběhnou dvě konzultace (cvičné turnaje). Účast je povinná alespoň na jednom z nich (velmi doporučená na obou). Konkrétní termíny určí vyučující.

Finální turnaj: Proběhne v druhé polovině května, konkrétní termín bude upřesněn. Kdo se nedostaví nebo alespoň nepošle své letadlo do soutěže, nebude mít nárok na zápočet. Omluveny budou pouze vážné důvody.

2.9 Bonus pro Vás

Ti z vás, kteří dosáhnou nejlepší výsledků, budou odměněni mimořádným stipendiem.

2.10 Podmínky zápočtu

- Aktivní účast na cvičeních
- Účast na jedné z konzultací
- Vytvořit alespoň trochu inteligentní algoritmus řízení letadla
- Účast na finálním turnaji
- Vypracování referátu/dokumentace

Jak má vypadat referát?

- Úvodní strana s názvem zadání, předmětu, jménem a číslem studenta
- Stručný popis zadání – co bylo cílem semestrální práce
- Podrobný popis řešení problému
    - Nejlépe popsat chronologicky postupný vývoj finálního algoritmu včetně jednotlivých mezi-verzí. Popsat jednotlivé nápady a jejich výsledky, včetně problémů a nedostatků. Porovnejte vliv výběru nastavení.
    - Do dokumentace nekopírujte celý zdrojový kód. Pokud je některá z jeho částí velmi zajímavá, uveďte jen tu část.