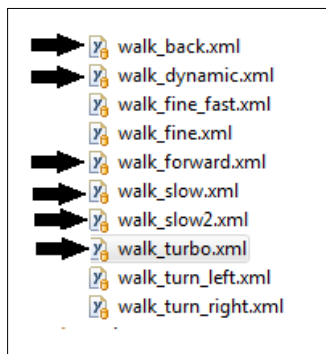


1. Používané chôdze



Obr.[1] - xml chôdzí (Jim/moves)

2. Kráčanie - rýchle

Začiatočná poloha hráča: Vector3D.cartesian(-5, 1, 0.4);

Pozícia lopty: Vector3D.cartesian(0, 0, 0.4);

Test č.	WALK_SLOW	WALK_MEDIUM	WALK_FAST
	pády		
1	0	0	2
2	0	2	0
3	1	1	1
4	1	0	3
5	0	1	2
6	0	0	1
7	1	0	1
8	0	0	0
9	1	0	2
10	0	1	0
11	0	2	0
12	0	1	3
13	0	0	2
14	2	1	1
15	1	2	0
16	0	1	1
17	0	1	2
18	0	0	2
19	1	1	0
20	0	0	1

Tab.[1] - kráčanie rýchle

3.Kráčanie - stabilné

Začiatočná poloha hráča: Vector3D.cartesian(-5, 1, 0.4);

Pozícia lopty: Vector3D.cartesian(0, 0, 0.4);

Test č.	WALK_SLOW	WALK_MEDIUM	WALK_FAST
	pády		
1	1	1	4
2	0	2	1
3	1	0	1
4	0	1	1
5	0	1	2
6	1	2	2
7	0	2	0
8	0	0	1
9	1	2	0
10	1	1	3
11	0	2	2
12	1	0	0
13	0	0	2
14	0	1	3
15	2	0	3
16	0	2	0
17	2	1	1
18	0	1	2
19	1	1	1
20	0	0	2

Tab.[2] - kráčanie stabilné

4.Porovnanie

	Rýchla	Stabilná
WALK_SLOW	7	11
WALK_MEDIUM	14	20
WALK_FAST	24	31
Spolu pádov	45	62
Priemer na 1 test	2,25	3,1

Tab.[3] - porovnanie výsledkov

5.Úprava highskillov pre chôdzu

V konečnom dôsledku sa mi nepodarilo nájsť hodnoty, ktoré by zlepšili efektivitu chôdze a tým znížili počet pádov počas nej. Mnou volené hodnoty chôdzu viac zhoršovali a nepodarilo sa mi nájsť spôsob akým znížiť počet pádov (pomocou úpravy highskillov).

Hodnotím, že efektivitu chôdze a zníženie počtu pádov je potrebné riešiť na inej úrovni v projekte(napríklad pracovať na šume zo servera alebo zohľadnenie náklonu pri videní čiar).

6.Prechod z WALK_MEDIUM na WALK_SLOW

Balík: sk.fiit.jim.agent.highskill.move

Trieda: MovementSkills.java

Funkcia: double getWalkSuitability(MoveSkillPostionObject,MovementSpeedEnum)

Pri tomto tasku som sa venoval aj zmenšeniu vzdialenosti hráča voči lopte, pri prechode z WALK_MEDIUM rýchlosti na WALK_SLOW rýchlosť. Vzdialenosť sa mi podarilo zmenšiť úpravou premennej rangeKoeff na hodnotu 0.35(pôvodná 1.0). Avšak túto úpravu je ešte potrebné prekonzultovať, keďže nebola súčasťou tasku.

```
if (speedEnum == MovementSpeedEnum.WALK_SLOW){
    speedKoeff = 0.5;
    rangeKoeff = 0.35;
    successKoeff = 1.5;
}
else if (speedEnum == MovementSpeedEnum.WALK_MEDIUM) {
    speedKoeff = 1.0;
    rangeKoeff = 0.25;
    successKoeff = 1.2;
}
else if (speedEnum == MovementSpeedEnum.WALK_FAST){
    speedKoeff = 1.5;
    rangeKoeff = 0.15;
    successKoeff = 0.01;
}
```

Obr.[2] - úprava vzdialeností

Hodnotím, že zmena vzdialenosti nemala vplyv na pády Jima počas chôdze, len mierne znížila čas potrebný na presun k lopte v sekundách. Touto úpravou tak hráč najpomalšou chôdzou prejde menšiu vzdialenosť (príklad - po novom sa pomalá rýchlosť aktivuje 1 meter od lopty, po starom 2 metre). V prílohe prikladám video s názornou ukážkou.

7. Zmena vzdialenosti

Na určenie vzdialenosti hráča od lopty, pri ktorej hráč spomalí svoj pohyb k lopte sa používa konštanta `EDGE_DISTANCE_FROM_BALL` v balíku `sk.fiit.jim.decision.situation`, konkrétne v triede `Situation`. Hodnota tejto premennej bola nastavená na hodnotu 2. Aby sme dosiahli zrýchlenie hry, zmenili sme jej hodnotu na 1.5, na základe ktorej hráč spomalí v kratšej vzdialenosti od lopty a skôr sa k nej dostane, vďaka čomu sa zlepší plynulosť samotnej hry (Obr. 1.a).

```
public abstract class Situation {  
    public static final double EDGE_DISTANCE_FROM_BALL = 1.5;
```

Obr.[1] - nová hodnota konštanty