

aozturk1

<https://github.com/aozturk1/ser316-spring2023C-aozturk1.git>

Task 1

Review

ser316-spring2023C-aozturk1 / Review.md

Go to file

aozturk1 finished review table

Latest commit a938147 12 hours ago

1 contributor

19 lines (17 sloc) | 1.22 KB

<> | Raw | Blame

#Code Review Table ###Reviewer: Alper Ozturk ###GH Repo: ...

ID#	Location(File and Line Number)	Problem Description	Problem Category	Problem Severity
1	GamePlay.java Line 1	Source code file missing a file and a public class banner.	CG	LOW
2	GamePlay.java Line 52	Class name should be upper CamelCase, with first letter uppercase.	CG	LOW
3	GamePlay.java Line 53	All complex statements must use explicit {} even if the body is a single line.	CG	LOW
4	GamePlay.java Line 137	Duplicate code.	CS	LOW
5	GamePlay.java Line 211-222	Program doesn't work correctly due to order of attack being always the same, no matter the opponent speed.	FD	MJ
6	GamePlay.java Line 7,157,158	Unused variables.	MD	LOW
7	GamePlay.java Line 81	Indentation should be consistent.	CG	LOW

Task 2

ReviewTable Issue #5(Repository Issue #8 - FD | MJ, fixed the order o...

Browse files

...f attack by changing the logic of the play() method so that attack speed correctly determines the order of attack. Closes #8

Review

aozturk1 committed 1 minute ago

1 parent 94039e1 | commit 5cbc04b

Showing 1 changed file with 2 additions and 2 deletions.

Split | Unified

src/main/java/GamePlay.java

@@ -213,8 +213,8 @@ public int play() {

213 213 orderOfAttack[1] = opponent;

214 214 player.experience += Math.ceil(player.speed - opponent.speed);

215 215 } else {

216 - orderOfAttack[1] = opponent;

217 - orderOfAttack[0] = player;

216 + orderOfAttack[0] = opponent;

217 + orderOfAttack[1] = player;

218 218 opponent.experience += Math.ceil(opponent.speed - player.speed);

219 219 }

220 220

aozturk1

<https://github.com/aozturk1/ser316-spring2023C-aozturk1.git>

Task 3 | Step 1

Class TDD

all > default-package > TDD

37 tests 0 failures 0 ignored 0.006s duration

100%
successful

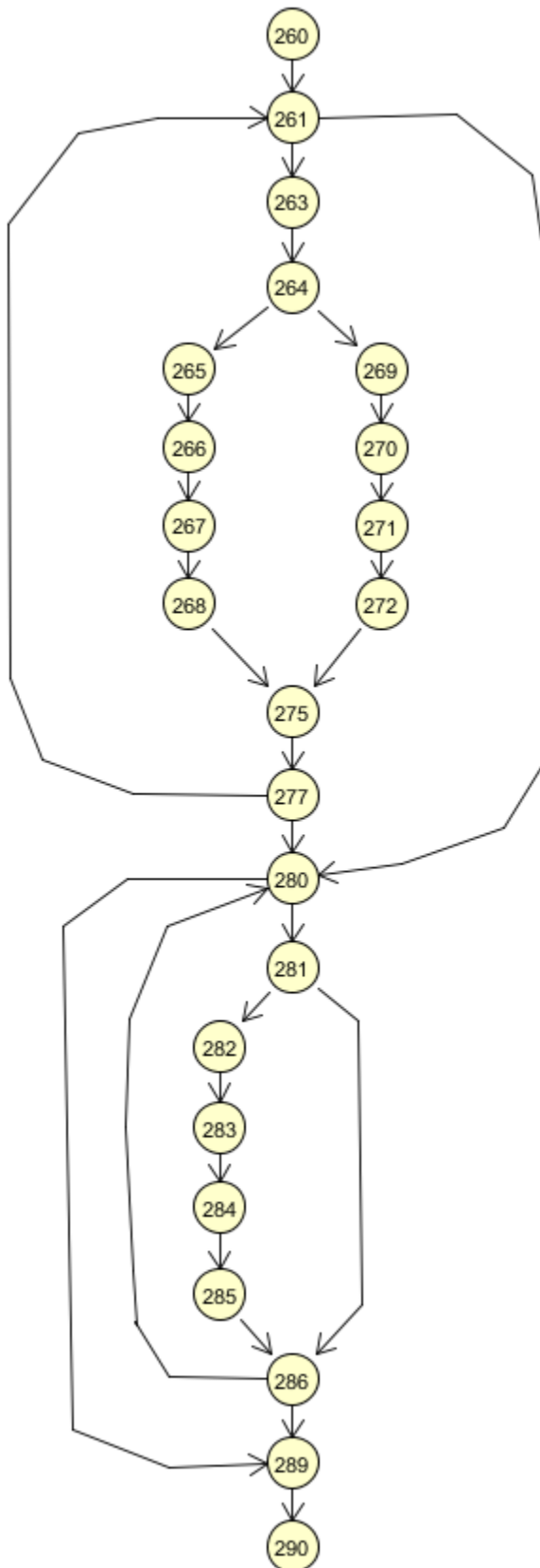
Tests

Test	Duration	Result
dealDamageNormalExperience[0]	0s	passed
testattackBothHealthGreaterThanZeroBoundary[0]	0s	passed
testattackBothHealthLessThanZeroBoundary[0]	0s	passed
testattackBothHealthZero[0]	0s	passed
testattackLevelUpHealthGreaterThanZeroFirst[0]	0s	passed
testattackLevelUpHealthGreaterThanZeroSecond[0]	0s	passed
testattackLevelUpHealthLessThanZero[0]	0s	passed
testattackLevelUpHealthZero[0]	0.001s	passed
testattackOneHealthGreaterThanZeroBoundary[0]	0s	passed
testattackOneHealthLessThanZeroBoundary[0]	0s	passed
testattackOneHealthZero[0]	0s	passed
testattackOrderDamage[0]	0.001s	passed
testattackOrderHealth[0]	0s	passed
testattackTwoHealthGreaterThanZeroBoundary[0]	0s	passed
testattackTwoHealthLessThanZeroBoundary[0]	0.001s	passed
testattackTwoHealthZero[0]	0s	passed
testdealDamageGreaterThanHundredHealthBoundary[0]	0s	passed
testdealDamageGreaterThanTenHealthBoundary[0]	0s	passed
testdealDamageGreaterThanZeroHealthBoundary[0]	0s	passed
testdealDamageLessThanHundredHealthBoundary[0]	0s	passed
testdealDamageLessThanTenHealthBoundary[0]	0s	passed
testdealDamageLessThanZeroHealthBoundary[0]	0s	passed
testdealDamageTenHealth[0]	0s	passed
testdealDamageZeroHealth[0]	0s	passed
testexperientetakeDamageEqualProtection[0]	0s	passed
testexperientetakeDamageGreatThanZeroBoundary[0]	0s	passed
testexperientetakeDamageGreaterThanProtectionBoundary[0]	0s	passed
testexperientetakeDamageLessThanProtectionBoundary[0]	0s	passed
testexperientetakeDamageZero[0]	0s	passed
testhealthtakeDamageEqualProtection[0]	0s	passed
testhealthtakeDamageGreaterThanProtectionBoundary[0]	0s	passed
testhealthtakeDamageGreaterThanZeroBoundary[0]	0.003s	passed
testhealthtakeDamageHealthGreaterThanZeroBoundary[0]	0s	passed
testhealthtakeDamageHealthLessThanZeroBoundary[0]	0s	passed
testhealthtakeDamageHealthZero[0]	0s	passed
testhealthtakeDamageLessThanProtectionBoundary[0]	0s	passed
testhealthtakeDamageZero[0]	0s	passed

aozturk1

<https://github.com/aozturk1/ser316-spring2023C-aozturk1.git>

Task 3 | Step 2



aozturk1

<https://github.com/aozturk1/ser316-spring2023C-aozturk1.git>

THE CODE LINES USED TO CREATE THE ABOVE CONTROL FLOW

```
259     @Override
260     public int play() {
261         int startingExperience = player.experience;
262         for (Character opponent : Opponents) {
263             //determine order of attack and give experience points for attacking first
264             Character[] orderOfAttack = new Character[2];
265             if (player.speed > opponent.speed) {
266                 orderOfAttack[0] = player;
267                 orderOfAttack[1] = opponent;
268                 player.experience += Math.ceil(player.speed - opponent.speed);
269             } else {
270                 orderOfAttack[0] = opponent;
271                 orderOfAttack[1] = player;
272                 opponent.experience += Math.ceil(opponent.speed - player.speed);
273             }
274             // attack in order
275             attack(orderOfAttack[0], orderOfAttack[1]);
276         }
277
278         // remove opponents that have <= 0 health
279         for (int q=0; q < Opponents.size(); q++) {
280             if (Opponents.get(q).health <= 0) {
281                 System.out.println(Opponents.get(q).getClass().getName() + " removed\n");
282                 removeOpponent(Opponents.get(q));
283                 q--;
284             }
285         }
286     }
287
288     return player.experience - startingExperience;
289 }
290
291
292 }
```

aozturk1

<https://github.com/aozturk1/ser316-spring2023C-aozturk1.git>

Sequences you need to achieve for complete i) node coverage

Test Case:

```
{  
<260,261,263,264,265,266,267,268,275,277,280,281,282,283,284,285,286,289,290>,  
<260,261,263,264,269,270,271,272,275,277,280,281,282,283,284,285,286,289,290>  
}
```

Sequences you need to achieve for complete ii) edge coverage

Test Case:

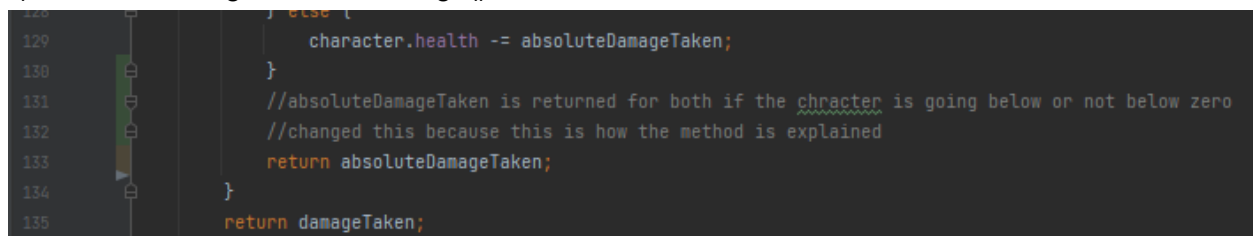
```
{  
<260,261,280,289,290>,  
<260,261,263,264,265,266,267,268,275,277,261,280,289,290>,  
<260,261,280,281,286,289,290>,  
<260,261,280,281,286,280,289,290>  
}
```

Not Included Due Node Coverage Sequences Doing the Job of These Edge Sequences

```
<260,261,263,264,265,266,267,268,275,277,280,289,290>,  
<260,261,263,264,269,270,271,272,275,277,280,289,290>,  
<260,261,280,281,282,283,284,285,286,289,290>,
```

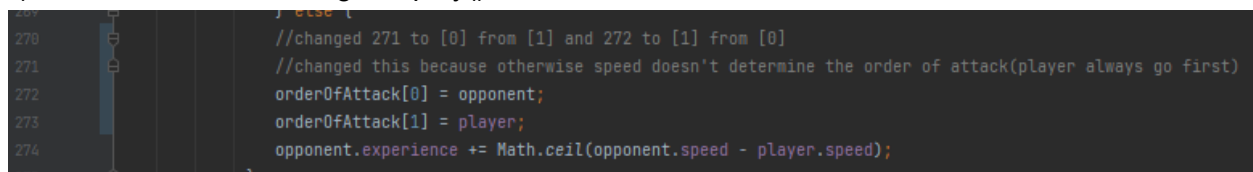
Found and Fixed Errors in the Algorithm

1) Fixed return logic in takeDamage()



```
129         character.health -= absoluteDamageTaken;  
130     }  
131     //absoluteDamageTaken is returned for both if the character is going below or not below zero  
132     //changed this because this is how the method is explained  
133     return absoluteDamageTaken;  
134 }  
135 return damageTaken;
```

2) Fixed order of attack logic in play()



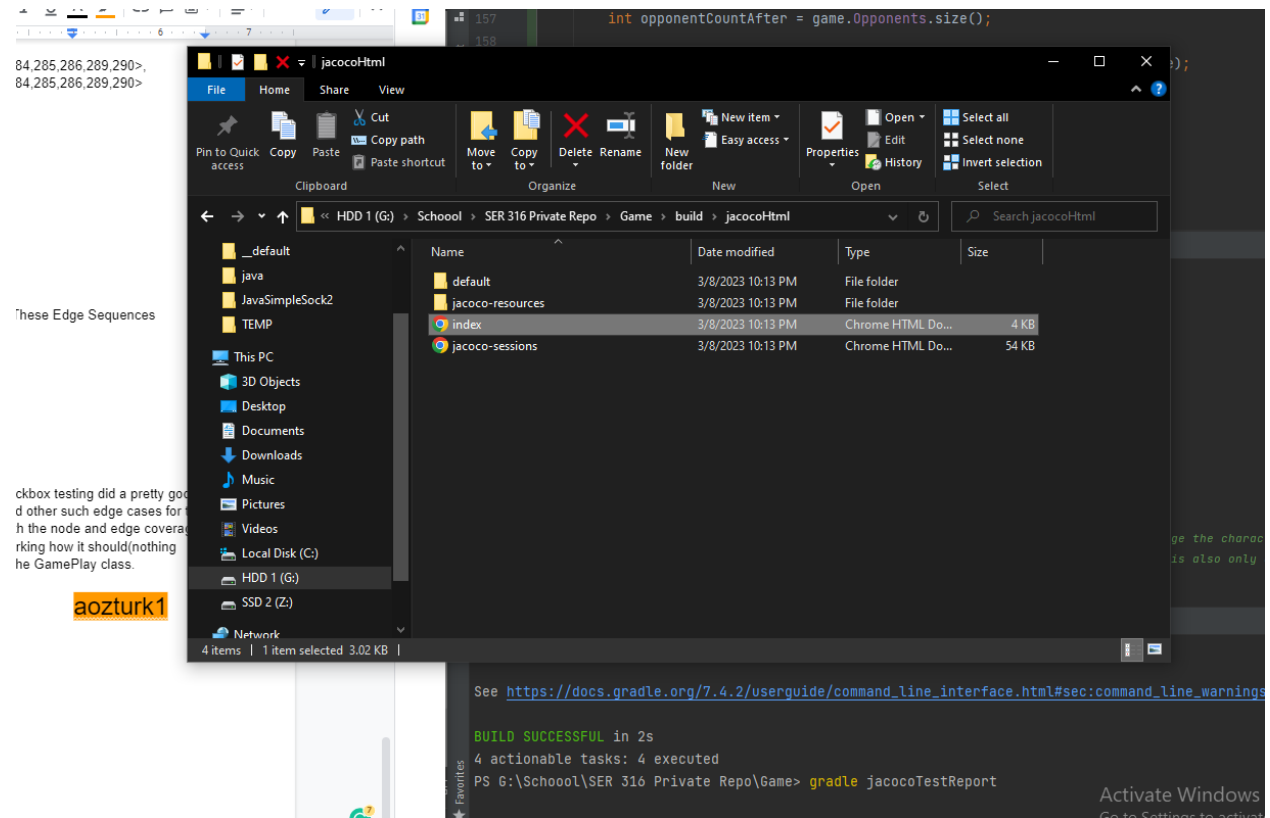
```
270     //changed 271 to [0] from [1] and 272 to [1] from [0]  
271     //changed this because otherwise speed doesn't determine the order of attack(player always go first)  
272     orderOfAttack[0] = opponent;  
273     orderOfAttack[1] = player;  
274     opponent.experience += Math.ceil(opponent.speed - player.speed);
```

I think my test cases cover everything important because my blackbox testing did a pretty good job testing the edge cases such as -1,0,1,9,10,11,99,100,101 and other such edge cases for the main three methods. And I whitebox tested the play() method with the node and edge coverage. It also states in the JavaDoc of the levelUp() method that it is working how it should(nothing wrong with it as given). And my code coverage passes 75% for the Gameplay class.

aozturk1

<https://github.com/aozturk1/ser316-spring2023C-aozturk1.git>

Task 4



1. Answer: What is the overall code coverage for the whole Java source code (excluding Main)?

67%

2. Answer: What is the code coverage for Gameplay.java with all your tests?

80%

Code Coverage

Game > default

[Source Files](#) [Sessions](#)

default

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Cxty	Missed Lines	Missed Methods	Missed Classes
Main	11	0%	0%	11	31	2	2	1
GamePlay	12	80%	79%	39	115	0	10	1
Character	1	34%	n/a	2	18	1	2	1
Druid	0	100%	n/a	1	6	0	1	1
Barbarian	0	100%	n/a	1	6	0	1	1
Ranger	0	100%	n/a	1	6	0	1	1
Rogue	0	100%	n/a	1	6	0	1	1
Wizard	0	100%	n/a	1	6	0	1	1
Bard	0	100%	n/a	1	6	0	1	1
Total	292 of 901	67%	30 of 76	60%	24	58	59	200

aozturk1

<https://github.com/aozturk1/ser316-spring2023C-aozturk1.git>

Game > default > Gameplay

Sessions

GamePlay

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
levelUp(Character)	<div><div></div></div>	53%	<div><div></div></div>	68%	5	9	15	35	0	1
takeDamage(Character, int)	<div><div></div></div>	96%	<div><div></div></div>	83%	1	4	1	14	0	1
addOpponent(Character)	<div><div></div></div>	77%	<div><div></div></div>	50%	1	2	1	2	0	1
removeOpponent(Character)	<div><div></div></div>	77%	<div><div></div></div>	50%	1	2	1	3	0	1
play()	<div><div></div></div>	100%	<div><div></div></div>	100%	0	5	0	18	0	1
attack(Character, Character)	<div><div></div></div>	100%	<div><div></div></div>	80%	4	11	0	19	0	1
GamePlay(Character)	<div><div></div></div>	100%		n/a	0	1	0	10	0	1
dealDamage(Character)	<div><div></div></div>	100%	<div><div></div></div>	100%	0	3	0	7	0	1
GamePlay(Character, Character)	<div><div></div></div>	100%		n/a	0	1	0	5	0	1
GamePlay()	<div><div></div></div>	100%		n/a	0	1	0	2	0	1
Total	118 of 613	80%	12 of 58	79%	12	39	18	115	0	10