Lab2 2SI3

Name: Samridhi Anand

MacId: anands29

Student No.: 400478945

Question 1: For objPosArrayList and objPosDLinkedList modules, what are the minimum lengths of Snake to obtain a non-zero computation time measurement for insertHead()? o Report "More than 10" if it takes more than a length of 10. Don't get caught up playing the game.

Minimum lengths of snake to obtain a non-zero computation time measurement for insertHead:

objPosArrayList → more than 10

objPosDLinkedList → more than 10

Therefore, both show a little similar behaviour here.

Question 2: Scale up the Snake length using the built-in long-snake initializer under Player::Player(). Perform the measurements on insertHead() again for objPosArrayList and objPosDLinkedList modules with TEST_LENGTH = {10, 20, 40, 80, 160, 320, 640, 1280, 2560, 5120, 10240, 20480, 40960, 81920, 163840, 327680}. Can you confirm that objPosArrayList:: insertHead() has a running time complexity of $\Theta(n)$, while objPosDLinkedList::insertHead() has $\Theta(1)$? You may optionally plot the curves for confirmation.

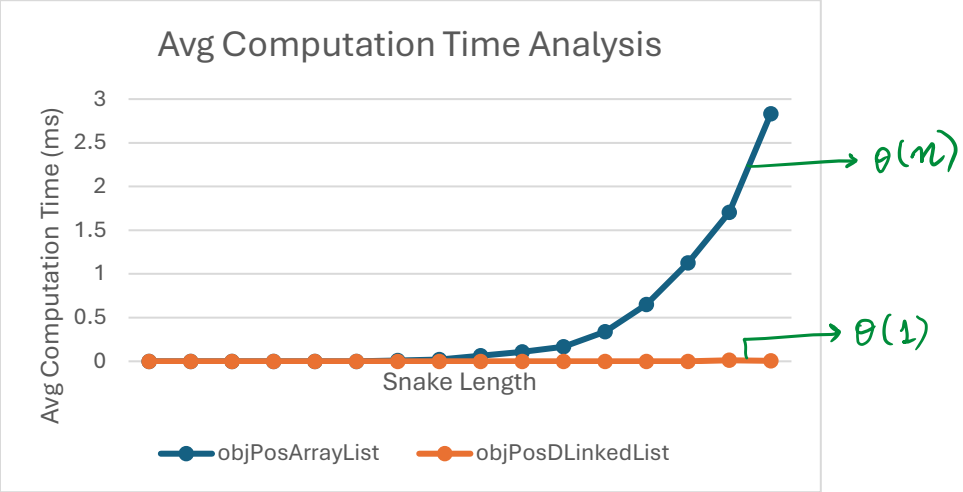
- Uncomment line 22 in Player::Player() to initialize the snake with arbitrary lengths upon startup.
- Adjust TEST_LENGTH on line 8 for different snake lengths.
- Since the snake is now not killable, you can move the snake freely. Take the stabilized running time measurement in unit of millisecond after moving the snake around for approximately 5-10 seconds.

Table of values recorded:

Snake Length	objPosArrayList	objPosDLinkedList
10	0	0
20	0	0
40	0	0
80	0	0
160	0	0
320	0	0
640	0.01024	0
1280	0.01917	0
2560	0.065233	0
5120	0.10699	0

10240	0.16582	0
20480	0.33689	0
40960	0.64892	0
81920	1.12404	0
163840	1.70393	0.011649
327680	2.831611	0.0062242

Graph Plotted



As per the recorded values in the table above and the Chart plotted after the average computation time analysis performed, clearly the **objPosArrayList::insertHead()** has a time complexity of **Θ(n)** since after a certain snake length the runtime starts increasing linearly as the length of the snake increases and the **objPosDLinkedList::insertHead()** has a time complexity of **Θ(1)** since the increase in length of the snake has a minimalistic impact on the average computation time after increasing the length of the snake to a very large extent which is nearly negligible.

Therefore, clearly the doubly linked list performs better than the array list in terms of the average computation time.

Question 3: Using the data from Question 2, at which snake length did you see that objPosDLinkedList outperforms objPosArrayList on insertHead() call? Is this threshold length relevant to the Snake gameplay, or is simply too large to have any impact on the gameplay?

At snake length of 163840 the objPosDLinkedList outperforms objPosArrayList which can also be regarded as the threshold length. This threshold length is extremely large and unlikely to be reached in typical Snake gameplay. Though it has some technical relevance, but it is too large to have any real impact on the game since players are usually not able to reach that length during any normal gameplay. Therefore, it is simply too large to have any impact on the gameplay and player experience.