How did you use this feedback? What was it especially helpful or not helpful for? (r0_Smoke)

Helped me understand whether my code is doing what is supposed to (r1_Smoke)

This feedback showed us what bucket we were in, so it was helpful. (r2_Smoke)

it was useful to check our grade, but it didnt really help us pinpoint what was failing in the categories it was grading (r3_Smoke)

gives the coder direction (r4_Smoke)

I used this feedback often and sometimes found it useful when trying to pin point a bug. However sometimes it was misleading as to what method the error is in. (r5_Smoke)

Sometimes the smoke tests results were helpful, but I think there should be more cases added so that you can't be passing all the smoke tests and still be in developing - once you pass them all i think you should have proficient score. (r6_Smoke)

I used the feedback to gauge my progress on the deliverable, and it lists what functionalities have been correctly implemented, so I could focus on other functionalities. (r7_Smoke)

It was misleading because we did not know exactly what we did wrong. For C2 we were missing three checkmarks and we got developing, but another three got proficient despite missing three. (r8_Smoke)

i didn't really asides from knowing whether i implemented methods correctly. i used the spec first and foremost and the grade as a reflection of whether i understood whether i implemented it correctly (r9_Smoke)

It was helpful in making sure that my code is passing the tests and knowing an estimate of what I need to work on or move on from. (r10_Smoke)

Generally it gave me a good sign of when I had successfully completed the majority of a function. (r11_Smoke)

I use it very often to access correctness of code (r12_Smoke)

The checkmarks are helpful to know when you have built enough tests for any particular area, though there can be issues when the different clusters are coupled, causing some illusions of completion (r13_Smoke)

this feedback was helpful because again this is a group project and both my partner and I are taking other courses and have other obligations but we agreed that we wanted to start with minimum step of passing all the smoke tests. And then would build on that. I think helped us see eachothers progress too and every checkpoint we aimed to just have this passing - time permitting would aim for Proficiency (r14_Smoke)

It was helpful to know which bucket we needed to work on (r15_Smoke)

This was useful as it gave us a general, high-level idea of what we needed to focus on. Didn't give us much past that. I used this for all the checkpoints to ensure I was on the right track. (r16_Smoke)

To pinpoint where in code was wrong or if test suite was incomplete (r17_Smoke)

Used it to check our implementation. It was obviously helpful. (r18_Smoke)

Helps us pin down specific sections of the code and helped us develop the changes incrementally. (r19_Smoke)

Let us know when we were done. (r20_Smoke)

It was a bit vague because some of the clusters would fail due to issues in other clusters and could cause us to look in the wrong place for our issues. (r21_Smoke)

Could be more detailed. As said above, sometime this was not accurate. (r22_Smoke)

Would stop working on these functions because it met the bucket requirements (r23_Smoke)

Useful to determine which clusters to focus on, not useful to determine specifics of why our clusters were failing. (r24_Smoke)

Not terribly helpful. (r25_Smoke)

I used the feedback to access which feature is implemented successfully. And if it's not proficient, I would try from the focus area code (r26_Smoke)

As discussed above, I think the category tests fail for the wrong reasons which can be misleading and frustrating to debug. (r27_Smoke)

It would be nice to know what 1/1 means in terms of what it's testing for. (r28_Smoke)

This is good, it tells you somewhat how you're doing and there is still work left to be done. (r29_Smoke)

As i mentioned above, the feedback feels to generic, and we were not able to pin down the issue with just knowing which cluster was passing/failing. (r30_Smoke)

Smoke tests helped ensure that general functionality was met. Very helpful (r31_Smoke)

It was a good feedback to visualize which part i need to work on. (r32_Smoke)

Not extremely helpful. Would wish to see specifically what tests failed. (r33_Smoke)

I found this feedback useful as it provided me with a general idea of the amount of work I had completed. (r34_Smoke)

The pass and fail in this section doesn't tell you the full picture. (r35_Smoke)

This feedback was used to see our grade. It was useful in seeing what we were doing right, but at the same time, there was no way of knowing "how right" it was, because we could have 1/1 for all of them yet still not be at proficient. (r36_Smoke)

It is helpful however without more specification it is really not game changer (r37_Smoke)

Not helpful when you were getting check mark but something wrong. (r38_Smoke)

This is not helpful because it is very vague. Performquery is a big part of the code and alot of things can be wrong so it would have been better if the bot gave more detailed feeback. (r39_Smoke)

It was helpful to see what we still needed to work on. it was NOT helpful when it misled us. for example in C2, adddataset would show 2/2 but min/max would show 0/1 when min/max was fine and there was actually an implementation error elsewhere, lot of wasted time on this (r40_Smoke)

it wasn't helpful as it was difficult to pin point what exactly was failing. (r41_Smoke)

When I finished all/part of the implementation. It was helpful in knowing whether or not my implementation was behaving correctly. It was frustrating when we couldn't get the last point, since it gave no feedback about what kind of tests were being used to calculate the marks. (r42_Smoke)

This feedback is misleading sometimes. There was a bug in our implementation for two of the functions, but the smoke test passed for one of them, which made debugging extremely difficult (as we assumed that passing the smoke test meant that the function was correct). Some more clarification would be nice. (r43_Smoke)

This was helpful and it helped us realize which parts needed to be worked on. (r44_Smoke)

I check it when I think i am finished/partly finished. It is telling me the result basically (r45_Smoke)

We usually either had 0 or full marks for each bucket, so it wasn't super helpful. It provided reassurance that we weren't missing too much in our solution. (r46_Smoke)

'- Figuring out areas where we needed to improve our implementation. (r47_Smoke)

Was sometimes misleading, area 1 showed failing smoke test but code related to area 2 had caused the problem. (r48_Smoke)

They are nice, they tell you which part of the code you do not need to keep working on - as it is completed. (r49_Smoke)

I used it to determine which functions had errors. Was not super helpful besides telling me very broadly what needed to be fixed. (r50_Smoke)

Also most useful for C0 and C1 when the bulk of the logic was being implemented and I needed to check if I had understood the spec correctly. Beyond C2 I called it pretty infrequently and basically only as a formality, most of the testing was happening locally. (r51_Smoke)

Good to know what parts needed fixing. (r52_Smoke)

This feedback was the most useful. It helped me know what parts of my code needed work done on them and where to move onto next. (r53_Smoke)

It was insightful as it showed us what needed to be worked on. (r54_Smoke)

i used this feedback to grade my project. it was helpful only in seeing if my project was up to standards. (r55_Smoke)

We didn't run the grader bot command very often. However, for C2 the clusters came in quite handy because they helped me decide which aspects of the app to probe with tests to hand off to my partner. (r56_Smoke)

I primarily used this check to evaluate my work. I noticed that this check could be confusing at times. For example, during checkpoint #2, I noticed that many groups encountered a situation where the max aggregation check passed while it failed for min. This may seem counter-intuitive, and there is not much explanation provided for why this might occur. It was later discovered that the underlying bug was unrelated to either max or min. (r57_Smoke)

To understand grade, vaguely. Feedback at times was misleading and did not represent what was actually preventing an improved score. (r58_Smoke)

Useful but I wish it would break it further into what types of tests are not passing (r59_Smoke)

smoke tests were useful, but more detailed feedback for each smoke test would've gone a long way in reducing development time (r60_Smoke)

Could be more useful in telling how a test fails or passes, suchas "Add Dataset failed on X case, consider adding a check for X". (r61_Smoke)

It was good to know what parts were up to standard by the grading bot. (r62_Smoke)

For the most part, our group used this feedback to figure out where we needed to dedicate our time, as well as how many of the requirements we'd met so far. (r63_Smoke)

The number of tests passing was useful to see the areas we needed to work on. (r64_Smoke)

Yes helpful (r65_Smoke)

This feedback was especially helpful for identifying components of the system that was failing. (r66_Smoke)

We used this feedback whenever we finished implementing one of the buckets to see if we did it correctly. This feedback was helpful for verifying correctness, but could have provided some more detail. (r67_Smoke)

It was helpful for getting a good idea of where we were in our implementation and our progress throughout the checkpoint (r68_Smoke)

in retrospect, it was quite fair (r69_Smoke)

It helps us to know if we're understanding the specs correctly (and it's directly related to the grades we're getting at the end) (r70_Smoke)

I used this feedback to determine whether my implementation roughly passed the tests that were being graded. (r71_Smoke)

Was helpful but still required more details as to why the test clusters were failing and what was limiting the code from not reaching the next bucket. (r72_Smoke)

This would lead me to figure out specific parts and key down on where I might have to look to finish implementation, it was helpful for this. (r73_Smoke)

The smoke test clusters were a bit too vague to actually guide my debugging. (r74_Smoke)

I used this feedback to see if I was done implementing something. (r75_Smoke)

used it to check my grade, what was broken, what needed to be worked on, and what I had to do next (r76_Smoke)

I used to check where we are roughly for our project (kinda like a check point), and see how far we are from finishing (r77_Smoke)

I mainly used this feedback once all my local tests were passing, so not that often. (r78_Smoke)

I would use it as guideline for when I have completed a method. (r79_Smoke)

This was very helpful as it broke down which aspects or parts of the project were done well, especially when there were the emojis that indicated growing/halfway implemented parts of the code. (r80_Smoke)

same as above (r81_Smoke)

I used it after every major merge onto the project. It was quite helpful at letting me know what clusters I needed to work on. (r82_Smoke)

This allowed me to start the diagnosis process for bugs. Very helpful. (r83_Smoke)

This was useful as it essentially told us if we had the point or not. (r84_Smoke)

It was really helpful but when I had everything passing locally and the smoke test were also passing, I was hard to find out what was the issues (even though I had multiple local test passing). Hopefully it can go more in depth. (r85_Smoke)

I used it to check my answers and look for bugs. The feedback was OK for determining if I was on the right track but sometimes misleading. (r86_Smoke)

Write new tests for the component that were not checked. (r87_Smoke)

Was good for grade but not for actually resolving code (r88_Smoke)

Obviously we used this feedback since it had to do with our grade, and it was helpful in seeing what areas we were messing up. Not helpful was not knowing what was wrong. (r89_Smoke)

I used it to determine my focus area and where to work on next. (r90_Smoke)

I used this feedback when I think I'm mostly done, to make sure I actually am done. Helps tell me if a certain feature has a major problem. (r91_Smoke)

With this feedback, I will know where I should work more on. However, I wish there will be more direction or detail within each category for us to know where or how to fix it (r92_Smoke)

It was helpful to understand whether you got points or not (r93_Smoke)

This was helpful in seeing which part of our project was failing. (r94_Smoke)

Tells me where should I focus on (r95_Smoke)

The feedback given for failing smoke tests is unhelpful and unrealistic. Hours wasted blindly guessing. (r96_Smoke)

Helpful when it says 0/1 and you can clearly see that there is a major issue with one part of your code, but when it says 1/1 and you are not getting Proficient, I did not find this to be very helpful (r97_Smoke)

This is how I know if I am done a checkpoint or not, so very very much helpful and needed. (r98_Smoke)

It was helpful to see where my group lied in the grade part of that checkpoint. (r99_Smoke)

I dont have much of an opinion of this as this was mostly used for checking progress on the project. We did most of our bug hunting locally (r100_Smoke)

This feedback was essential to progress the project, as it told us the general direction of where our implementation was failing. However, sometimes, the clusters were not indicative of the reason why tests were failing. (r101_Smoke)

Its useful, but it tells you nothing about the type of tests they ran and what exactly it checked. For example someone could think their add dataset is working as expected, but all it could have checked was if no exception was thrown for example. I understand that we are "supposed" to have unit tests, but making a 100% coverage suite for all edge cases and all helpers is really time consuming, and honestly too time consuming. I think I am well more prepared for this course given my web development experience than a lot of students were, and even I found the project to take quite a bit of time without even making unit tests. (r102_Smoke)

Kind of helpful that it tells you very generally where to look (r103_Smoke)

I used the feedback to identify the areas I still need to work on. It's helpful to have a sense of how far I've come. (r104_Smoke)

It was a good measure of progress (r105_Smoke)

It was not clear what was going wrong (r106_Smoke)

I used it to check my progress and see what needed more work. (r107_Smoke)

I used to see if my function is working correctly. It is helpful since I know that at least my function is doing what is expected, but it doesnt tell me if my function is fully correct to earn that proficient level (sometimes I check a cluster but have to work on sth else and I dont know if thats the limit of the test or what its a bit confusing) (r108_Smoke)

helpful for knowing which areas to focus on most of the time and the weight of each area (r109_Smoke)

Even when the smoke test bucket would check, that doesn't mean that the code is free from bugs and this was not helpful (r110_Smoke)

Useful to see if anything is on fire, its name suggests. (r111_Smoke)

We used this to estimate our grade. It wasn't very helpful outside of that (r112_Smoke)

It was helpful to see which clusters were passing but the clusters were sometimes too broad (r113_Smoke)

I used it to check my grade, but again sometimes I didn't know really what a cluster meant (r114_Smoke)

Checking that the implementation is mostly correct (r115_Smoke)

Grade requests were run after implementing major features to gauge progress. Feedback could be useful, but some issues were hard to reproduce due to lack of detail. (r116_Smoke)

I used it to check my grade. Wasnt very helpful to figure out how to fix my code (r117_Smoke)

I was helpful in telling us how much was working. (r118_Smoke)

Very useful. (r119_Smoke)

Most of the time, the actual bug had nothing to do with what was recommended, like RoomQueries. Instead, it was some fundamental bug that only happened to expose itself in the tests for RoomQueries (for example, having a certain field type that is a number instead be a string) (r120_Smoke)

It was hard to know what was failing and where to focus our efforts going forward (r121_Smoke)

it tells me what i need to work on but it can be unclear at times due to the lack of information (r122_Smoke)

It often told us one thing was broken when that was not the case (r123_Smoke)

Used it to determine which areas I needed to work on, and which areas I did not have to touch. In this sense it was helpful but sometimes it is quite general and not specific which makes it hard to understand what parts of my code need improvement. (r124_Smoke)

The Smoke test is very easy to test locally, no need to have that result. (r125_Smoke)

Decent but not very specific. The bucket grading system is very helpful for students though, as the test cases are hidden so we need to know where we are at. (r126_Smoke)

Made it so that we did not focus our efforts on portions of code that already ran effectively (r127_Smoke)

its not that usefull.. (r128_Smoke)

In #c2 we did not receive feedback that was completely reflective of our implementation nor was it very helpful for identifying issues. (r129_Smoke)

325t4tt (r130_Smoke)

Same for question 1. (r131_Smoke)

To check whether i was "done" (r132_Smoke)

This feedback helps me understand if my implementation is on the right track or not if I get the checkmark. However, if all my local tests passed but there is mo checkmark, this feedback does not provide me with a lot of insight of what is wring with my code. (r133_Smoke)

The majority of the time it just showed me that I checked every cluster but I'm still not at proficient without much meaningful feedback (r134_Smoke)

It let us know what we were doing right and what we were doing wrong. It helped us focus on the things we needed to. (r135_Smoke)

not enough feedback on what went wrong. (r136_Smoke)

I used this feedback to grade my code, it was helpful in figuring out what to work on next (r137_Smoke)

I used it mostly to see where we could gain an extra "mark", but the "additional feedback" section was far more helpful in telling me what my actual grade was. (r138_Smoke)

Used it to check how we were doing. Was not very helpful, because like i mentioned above, you can get checkmarks for everything yet still not be proficient. (r139_Smoke)

I think if it was a larger scale other than binary (maybe out of 5), it could represent better how much progress we have had so far for RoomQueries for instance. (r140_Smoke)

I used this feedback to know which parts of the project to work on. Since each section it can be 1/1 without being perfect, there were times when I wasn't sure if there was anything left for me to work on. (r141_Smoke)

1/1 was too vague. I got 1/1 but then more specific feedback came back as acquiring only. (r142_Smoke)

we used this feedback a lot to see if certain parts of our solution were implemented correctly. when smoke tests didn't pass, we would recheck our implementation again to see what we might've missed to fail the smoke test. it was helpful in isolating the specific areas we needed to focus on. (r143_Smoke)

Mostly to check through the project how much I accomplished and how much work more I have (r144_Smoke)

This told me where and what to work on so it was helpful. (r145_Smoke)

This was good because it reflected my actual grade (r146_Smoke)

We use this to know if our implementation of the specific feature is doing (mostly) what it's intended to do. I picked somewhat agree because there could be casees where all smoke tests pass and we're still not at Proficient. (r147_Smoke)

Doesn't really tell you much of anything. It's not clear what it means to get 1/1. Although, obviously 0/1 is pretty clear. (r148_Smoke)

I used this feedback a lot in earlier checkpoints, but then for c2 I would sometimes get the check and then my next PR loses the check, even though I didn't even make changes to that area. Soo it didn't do a great job of properly narrowing down the root cause of the problem. (r149_Smoke)

I liked the feedback and found it quite useful . (r150_Smoke)

We used this feedback to check whether or not we had enough coverage on our tests. If not then we would try to test for more and see if our implementation is able to handle it (r151_Smoke)

I use it after each the tasks are completed to ensure everything works correctly. So I don't use it as often, but those are quite helpful. (r152_Smoke)

gzave idea what to do (r153_Smoke)

I think this is helpful, because I can directly find out where the problem could be. (r154_Smoke)

The c3 could be affected by c1 which the auto test could not show. This made it super hard to get full mark. (r155_Smoke)

I used this feedback to assess where I was at with the project so far. It was helpful for checking where to look in my code for possible errors. (r156_Smoke)

Smoke test results help identify which part of the implementation went wrong, and would be helpful if we have good task decomposition. It also offered the most direct standard for whether our tasks were completed. (r157_Smoke)

Googled it. (r158_Smoke)

It's helpful for knowing if there's any error at certain location in the code (r159_Smoke)

For checking if we can move on to implementing the next thing (r160_Smoke)

Good to see the robustness of my implementation. (r161_Smoke)

It was helpful, but at times too general that it was frustrating to understand what was happening. As well, it is a bit flawed because it was hard to tell if our problem lied in a specific functionality or if that problem was derivative of another upstream problem. (r162_Smoke)

Limiting smoke test to twice a day was not helpful, and the lack of detailed feedback meant that we were discouraged from using this feature effectively. (r163_Smoke)

Check if I get full marks in the corresponding parts (r164_Smoke)

We didn't use all 4 smoke tests in one day ever, but it was somewhat helpful to get the checkpoint off the ground. More detailed feedback would've been helpful. (r165_Smoke)

The detailed breakdown of completing what will receive the checkmark is not clear at all, and this does not seem to be directly related to the artifact quality, it's quite misleading sometimes. (r166_Smoke)

Looking at this feedback, I would feel like I have accomplished the project requirement. (r167_Smoke)

For c0 and c1, the feedback was quite useful, as having a 1/1 or 4/4 was largely indicative of your progress on the function (r168_Smoke)

It was helpful to know that my implementation wasn't what the autobot expected. However I felt that my code was implemented according to spec and didn't know what the autobot was complaining about which was frustrating (r169_Smoke)

Very useful because it breaks our code down into specific requirements and give information for each. (r170_Smoke)

They were useful to make sure the code worked (r171_Smoke)

Checking we are done with a certain feature (r172_Smoke)

Despite the fact it’s 1/1 for everything it’s still developing. Making it reflect the actually amount of tests passing or a weighted percent would be better. (r173_Smoke)

Not helpful in the sense it would give the wrong error message sometimes, whereas the actual grade would give different feedback (r174_Smoke)

This feedback was useful, but it would be better if there was more insight on where the issues lied when a test wasn't passing. (r175_Smoke)

It felt good to see the 1/1 but sometimes the categories were so broad that it was sooo hard to see why we didn't get a 1/1 on an item. Further breakdown or a more thorough title of what the smoketest does would help. (r176_Smoke)

Yes helpful, to quite an extent (r177_Smoke)

Wasn't very helpful (r178_Smoke)

It's helpful to know where to stop adding test cases/trying to cover more ground. I can move forward to try passing the other smoke tests with the assumption that these two are working. However, sometimes the smoke test isn't a correct indicator, which was an issue I encountered with AddDataset - turns out it wasn't working properly even though it passed the smoke test. That caused problems in the other functions that took me so long to diagnose because I thought AddDataset was OK. Overall though, it's pretty helpful. (r179_Smoke)

Let me know if I should keep working on that topic or move on to the next one (r180_Smoke)

Sometimes cluster is not really where the bug is, misleading the focus on wrong functionality (that functionality has no bug but I think the bug is in it). (r181_Smoke)

Used to determine when to stop coding... was not helpful because the tests were not really isolated for the functionality... so sometimes the issue was related more to another bit of functionality (that could have been passing even!) instead of the failing test (r182_Smoke)

The smoke tests were great to show our code was running at a basic level of correctness. (r183_Smoke)

it told me when i was done working on a functionality (r184_Smoke)

Showed us if we are on track with our implementation. (r185_Smoke)

Seeing what functionality that was not performing as expected, sometimes not as useful if the tests fail due to an non-intuitive reason (not accepting the inputs to the application correctly vs an issue with the function logic) (r186_Smoke)

I used this feedback when our group felt that we had completed enough to possibly get a full score. It was helpful since it helped our team understand what parts we needed to fix and what parts we fully completed. (r187_Smoke)

It's not very helpful because sometimes the problem is not related to the cluster name (e.g. max, min in c2). (r188_Smoke)

It gave us a 1/1 even when a lot of the tests are wrong. (r189_Smoke)

I used it to help guide where bugs in my implementation might be (r190_Smoke)

Smoke tests is smoke tests. (r191_Smoke)

Used it a lot, but feedback needs to be more detailed as I mentioend before. (r192_Smoke)

Yes it helped direct us to specific portions of the code which we should review. (r193_Smoke)

I used it when my tests passed locally. It helped me figure out if I was on the right track or not. (r194_Smoke)

when I need a final test. pretty useful (r195_Smoke)

Sometimes it was wrong (see above) (r196_Smoke)

I used this to guide our debugging. Overall, the feedback was very useful. But there were times when one cluster would fail but the issue resided elsewhere. (r197_Smoke)

it's not clear sometime which part of the feature is not passing, if more detailed feedback can be provided it will be good (r198_Smoke)

Shouldve have bit more detailed feedback, somehting that we received after the project deadline. having guided feedbacks would have been bit more helpful (r199_Smoke)

I definitely abused this call. It was the only way I could get a gauge on where I am. (r200_Smoke)

Just grading. Not really useful for finding what I did wrong (r201_Smoke)

giving a bcket and not a percentage was awfully stressful (r202_Smoke)

I thnk the feedback was useful for the most part. It was nice to see if our general implementation was correct, I don't know often what the result is supposed to look like so this gave us a good pat on the back to let me know that I'm on the right track. There was one test that was one smoke test failing in the rooms implementation, although everything else was proficient which threw off my team a lot though. (r203_Smoke)

help me debug. (r204_Smoke)

Very helpful (r205_Smoke)

This was helpful because it told you what functions were working. (r206_Smoke)

Mostly helpful. It would be better if it can shows which part of a cluster can be improve. (r207_Smoke)

This feedback is too general and misleading. (r208_Smoke)

If I passes these tests, I am ready to move on. (r209_Smoke)

I used this feedback to see if I had implemented my code correctly, it was helpful to let me know how much work I had left. (r210_Smoke)

I like seeing what I need to work on and what works. (r211_Smoke)

it was helpful, however it would have been good to have more info on what were the inputs of the failing tests. For example i had extensive tests for performquery but was failing MAX test on the bot. I think this may have been caused because a poor implementation of rooms but there was not way to know. (r212_Smoke)

To check grade (r213_Smoke)

To see if I'm passing or not. A slight hint at what could have gone wrong could help a lot. (r214_Smoke)

All i got was the name and nothing about exactly why it is failing or where I should look (r215_Smoke)

It's a nice sanity check when developing and serves as a direction to where to implement next. (r216_Smoke)

These were the main test results of what was working in our code and what wasn't. Therefore we used this frequently. It was very helpful since we knew what to work on if something didn't work and to move on to the next feature if it did. It would've been more helpful if feedback about internal failing tests was added. (r217_Smoke)

Was nice to focus on a specific area. (r218_Smoke)

This feedback was quite helpful in figuring out where the problem in the code was. As mentioned above, this feedback could explain things a bit more. (r219_Smoke)

It told me whether I was done or not (r220_Smoke)

Used this for most of my debugging, very good and straightforward (r221_Smoke)

Smoke tests were helpful for getting a start but it was frustrating when you had edge cases or other issues with particular functions and the smoke tests came back all correct. (r222_Smoke)

This feedback is more detailed and helped me many times to look into specific sections of my code for debugging. (r223_Smoke)

This feedback was useful for locating where code issues generally were. It was not useful for how vague the feedback was. A majority of debug time was spent guess-and-checking where potential errors could be. (r224_Smoke)

To decide which functionality still needed work/had bugs (r225_Smoke)

I think it was helpful to see what tests I needed more coverage on (r226_Smoke)

It’s hard to know how to change the code by only showing the current performance by points without feedback. (r227_Smoke)

Helpful for figuring out which aspects are failing, not helpful for figuring out the extreme edge cases. (r228_Smoke)

smoe times i passed all of the smoke test but didn't receive proficient. At that point its really diffuclt to identify what the problem is (r229_Smoke)

Useful for checking scores (r230_Smoke)

We basically only called the smoke tests once we had thought we finished the implementation. Since we wrote a bunch of tests, we only submitted to the bot once we were confident that we completed our implementation. The feedback was still useful, as sometimes we would receive feedback that some specific parts failed, and we knew where to look to find bugs or where to read about the spec more thoroughly. (r231_Smoke)

It provided insight into what functions needed to be worked on, and how they could be improved. (r232_Smoke)

Focus area too general (r233_Smoke)

I use this to determine whether my code fits the criteria. (r234_Smoke)

Good to know how far I was from getting to proficient. Wished it was more specific though. (r235_Smoke)

Since we had a limited number of these we would often use other methods to measure progress and only use it a couple times a day to know if we're getting the grades we want to or not. (r236_Smoke)

It is helpful, but it can be better if the actual grade is return to us (r237_Smoke)

Allowed me to realise i was focusing on wrong aspects of code. (r238_Smoke)

Wish it was more detailed than just a 0-1 score (r239_Smoke)

It was a simple and effective way to let us know if our implementation was working properly (r240_Smoke)

Sometimes confusing how smoke tests relate to grades, as we got 100% even though many of our smoke tests continued to fail. (r241_Smoke)

This told us whether our code was correct for certain parts; if we didn't have it, it would have been tough to figure out when we were finished with that part of the code. (r242_Smoke)

I used this feedback to ensure I was not missing test cases that were needed. (r243_Smoke)

smoke test give an overview of what method we havent much cover which is pretty useful (r244_Smoke)

Useful but don't actually need all to get to proficient. (r245_Smoke)

I think this feedback was nice to know that we had completed the checkpoint up the standards of the course. (r246_Smoke)

Good to know which area needs more work but sometimes, especially in C2, was really misleading. (r247_Smoke)

Very useful to keep track of my progress. (r248_Smoke)

very helpful (r249_Smoke)

helpful to let me know when to work on next part of the project (r250_Smoke)

It was sometimes helpful to gauge if we had the functionality desired by the instructors. (r251_Smoke)

See what my grade is and whether i need to still work on that functionality. usually helpful but not always if the problem is not located in that cluster. (r252_Smoke)

To know which part of the code still need fixes and improvements (r253_Smoke)

This was used only when we wanted to know our grade for the checkpoint and/or which aspects of the program we could deem "done" (r254_Smoke)

I used this feedback to see if I had a working implementation. The feedback was helpful as a sanity check, but not much else. (r255_Smoke)

I use this feedback check if my code passes the AutoTest tests. It is not very helpful as sometimes the smoke tests pass despite having problems with the code. (r256_Smoke)

Used to to check completion of project components (r257_Smoke)

The smoke test result was very helpful since it breaks grading components so I could see which part of my code is not working properly, then I can trace down to solve it. (r258_Smoke)

It doesn't really give any detail towards our implementation, so it wasn't too useful. (r259_Smoke)

Having more detail on the failing tests would have made it easier to get everything working. (r260_Smoke)

It allowed me to see which functions were working and which were needing more work, however with how infrequently I could grade my work in a day, it was not enough. (r261_Smoke)

It gives a vague idea of where the issue might be, but thats all. (r262_Smoke)

I used this to test if new code would break good sections. (r263_Smoke)

Was somewhat helpful but should better point out which part was wrong (r264_Smoke)

This feedback was our primary way of knowing our tests and implementation were correct (our tests most likely were correct if the implementation passes the autograder tests and our own) (r265_Smoke)

Should tell me what else do i need to improve on this feature other than just reaching acceptance level (r266_Smoke)

The cluster information is definitely very helpful. (r267_Smoke)