

How did you use this feedback? What was it especially helpful or not helpful for? (r0\_Addtl)

The focus area is good for me to know what to improve on. (r1\_Addtl)

This feedback showed us what bucket we were in, so it was helpful. (r2\_Addtl)

when "focus area" was included, it was really helpful. focus area should ALWAYS come up (r3\_Addtl)

same as above - gives the coder direction. (r4\_Addtl)

Tried using this feedback when all smoke tests were passing but still not achieving proficient. It gave me misleading information. (r5\_Addtl)

The focus area was not helpful for queries and list dataset failures, they would tell you to work on those two functions even though usually the issue stemmed from the add/ remove dataset functions that were used in setup. having the failing test cases would be MUCH more helpful so that we could determine where the error was coming from ourselves. (r6\_Addtl)

The Focus Area is vague and it doesn't provide a helpful hint on how to improve the specified area. (r7\_Addtl)

I was misleading at times, telling us to work on something that was not the problem. (r8\_Addtl)

same as the previous question, didn't mean much to me asides as a measure of how much i understood something (r9\_Addtl)

The feedback is very vague and doesn't tell much about how I can improve (r10\_Addtl)

Normally when looking at the focus area, the area it told me to work on had nothing to do with the error. (r11\_Addtl)

In the example given above, the RoomQueries test passes but focus area is still RoomQueries, this is confusing sometimes (r12\_Addtl)

The Focus Area is very much the type of feedback I think is most helpful, as it gives you direction on what is missing (which the checkmarks should be doing but weren't always accurate)) (r13\_Addtl)

honestly, if my partner and couldnt figure out the 'Focus Area' we did typically go to office hours. (r14\_Addtl)

Helpful for understanding our grade. I think some more detail in the focus area would have been nice (r15\_Addtl)

This was very helpful as it explicitly tells you if you got full marks for a checkpoint's code artifacts. The Focus area was helpful but i didn't necessarily use it all that often. (r16\_Addtl)

Didn't really matter as we just looked at check marks (r17\_Addtl)

Gave some context which was nice. (r18\_Addtl)

Peace of mind (r19\_Addtl)

Letting us know which bucket we were in was nice, but it would be nice to specifically know how many tests we were passing. (r20\_Addtl)

We didn't use this because it didn't feel very accurate. (r21\_Addtl)

Unsure of what else to improve on RoomQueries (r22\_Addtl)

The focus area was not the most useful, because as mentioned previously, it was difficult to determine why our tests were failing with the smoke test results. (r23\_Addtl)

Didn't use it at all. (r24\_Addtl)

It was good. (r25\_Addtl)

I think the focus area works well, but one time for example, it said to work on "ValidEdgeCases". I am not even sure what that means. I think more detailed feedback would be useful here, or if we can maybe see a bit of information on why some tests failed (although that might make everything too easy) (r26\_Addtl)

Definetly not helpful feedback. I would give much more feedback on what tests are passing so we can see what cases we were missing. Even in the provided screenshot for this question, it would RoomQueries 1/1 but still says to work on it which is very counterintuitive. Much better feedback needed than running it like this because we had similar feedback in c3 and was super demotivating to figure out where to go fix the bugs. (r27\_Addtl)

Given this feedback I have no idea where to look for this error. (r28\_Addtl)

This feature should be extended, more detailed for smoke failing tests (r29\_Addtl)

This was also a bit too general, for example roomqueries is half of C2, and just because some room query tests were failing didnt mean that was where the issue actually is. (r30\_Addtl)

It helped in debugging parts which did not work. Kind of useful (r31\_Addtl)

It was not that useful. Artifact Quality was helpful to see where i am at. (r32\_Addtl)

I found this feedback useful as it provided me with a general idea of the amount of work I had completed. (r33\_Addtl)

Too vague to be any use (r34\_Addtl)

This was used as a sign of what we needed to work on next. It was more useful than the one above in terms of direction, but lacks specificity (r35\_Addtl)

The exact same issue, it is good to know what is the grade however without more detail it really not enough (r36\_Addtl)

Better than all the other feedback. (r37\_Addtl)

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 feedback. (r38\_Addtl)

focus area was so misleading and often wrong, lots of wasted time (r39\_Addtl)

it wasn't helpful as it was difficult to pin point what exactly was failing or sometimes misleading (r40\_Addtl)

When I finished all/part of the implementation. It was helpful in knowing whether or not my implementation was behaving correctly. (r41\_Addtl)

Used to see if we got all project grades. (r42\_Addtl)

These comments were sometimes helpful, we appreciated the small hints on what to work on next. (r43\_Addtl)

this is the same as the status of my test clusters -- a result checking for me (r44\_Addtl)

The additional feedback section wasn't really helpful for the same reason as the question above. We usually knew where we were lacking the most. (r45\_Addtl)

'- I just needed to see my bucketed grade, and didn't really focus on the suggested focus area. I preferred to look at the test clusters instead. (r46\_Addtl)

The focus area was not super helpful, especially at earlier stages of development. (r47\_Addtl)

Even though it says where we should focus next, the feedback remains relatively vague as to what exactly one needs to focus on. (r48\_Addtl)

I used this feedback to determine which areas of code needed to be worked on. It was helpful for isolating errors, but still was not great at identifying which function caused the error. (r49\_Addtl)

The focus area was pretty helpful a couple of times when we had tiny bugs that were breaking everything. (r50\_Addtl)

Focus area was helpful. (r51\_Addtl)

Seeing the grade was definitely a must... yet the focus area for myself seemed to recommend things places that for myself didn't make much sense always. (r52\_Addtl)

Would have been much more helpful to just get a percentage. (r53\_Addtl)

i used this feedback to see which parts of my code may be weak/subject to bias. the artifact feedback was more useful. (r54\_Addtl)

We used this as a stopping rule (i.e. stop once at Proficient) (r55\_Addtl)

I didn't use this feedback much, as for me, I only stepped into two phases: either acquiring or proficient, and in neither case did the additional feedback provide any help. (r56\_Addtl)

Gave direction for what to work on. (r57\_Addtl)

focus area was very helpful. (r58\_Addtl)

Useful for knowing when to stop development of the project. (r59\_Addtl)

It was good to know when we can cut our losses instead of writing a "perfect implementation" so we could start on our other projects (r60\_Addtl)

The focus area was really useful in allowing us to determine which part of the code base to look at next! (r61\_Addtl)

The artifact quality also helped us determine the overall quality of our code. However, the focus area was very vague and gave us little to no hints. (r62\_Addtl)

Yes helpful (r63\_Addtl)

This feedback was very useful, there were some edge cases where they mislead us. But for 99% of cases, it always gave us some assistance. (r64\_Addtl)

We weren't in the Developing grading bucket that often (usually between Acquiring and Proficient). (r65\_Addtl)

I found it not very helpful for the additional feedback because we often found ourselves in the beginning bucket or the proficient bucket but never in between where we could make use of the additional feedback. (r66\_Addtl)

would like more feedback details. (r67\_Addtl)

This helps us to know what stage our development is in and guide us to the next part we should focus on. (r68\_Addtl)

I used this feedback to determine whether my implementation passed the tests that were being graded. (r69\_Addtl)

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. (r70\_Addtl)

Wasn't really too helpful, at times it didn't say much that I didn't already know (r71\_Addtl)

The focus area was almost always incorrect in my use of AutoTest, and my code would go from developing to proficient without ever touching the cluster mentioned in the focus area. (r72\_Addtl)

I did not find the focus area feedback was accurate or specific enough to help us track down what was going wrong. For example, my room dataset implementation was failing to accurately parse HTML special characters, but our feedback was to double-check our implementation of MIN. (r73\_Addtl)

to figure out what exactly I had to work on and where the problem I had to solve was. (r74\_Addtl)

I used it as a progress check, but i would want more detail on what marks are being taken off, or the server side error thats causisng me from not getting the mark (r75\_Addtl)

I didn't like this as much, it is very broad feedback. (r76\_Addtl)

I liked that it gave a focus area on which part to work on though I would be interested in how it chose which one to tell you to work on. This feedback was also useful because it told us our mark and which grading bucket we were in. (r77\_Addtl)

same as above (r78\_Addtl)

This feedback was not particularly useful as on some occasion it would suggest for me to work on a task that seemed to rely on another task to be completed first. (r79\_Addtl)

Pretty neutral on this one. Defintely helped in edge cases! (r80\_Addtl)

This was good as it helped the feedback with more specificity. (r81\_Addtl)

It was a good check point to see if I still needed to work on something or move on to something more important. (r82\_Addtl)

Often the function we were advised to work on actually was not the one with problems. (r83\_Addtl)

Never really looked at that, expect to see our grade of course! (r84\_Addtl)

Same issue, not very useful info (r85\_Addtl)

Obviously we used this feedback since it had to do with our grade and focused on where it said, but wasn't helpful in it didn't specify more what was wrong. (r86\_Addtl)

Same as above. (r87\_Addtl)

Mostly just useful for knowing my grade, didn't give very helpful feedback otherwise. (r88\_Addtl)

This is kind of dulpicate to the previous feedback (r89\_Addtl)

I wish there was a bit more details about focus areas. When you pass a smoke test but get a comment in focus area about that same function, it's very confusing. Anything that would help to avoid wasting TA hours and work independently would be helpful. (r90\_Addtl)

The focus area was useful in addressing grading issues. (r91\_Addtl)

Not very helpful as it only tells me where I am and how far to the expected effects. (r92\_Addtl)

The "focus area" is a good idea but implemented poorly in C2. It told us to work on a component that worked perfectly, which was failing due to a bug in a different part of the system. (r93\_Addtl)

Helpful in that it guides you to the area that needs work, but it is very difficult to know exactly what is going wrong, especially if it says all the smoke tests pass. (r94\_Addtl)

Same as previous response. (This is how I know if I am done a checkpoint or not, so very very much helpful and needed.) (r95\_Addtl)

This feedback was at times not accurate of what the actual issue was making my team waste time on fixing things that were not even broken in the first place. A more detailed feedback would be helpful. (r96\_Addtl)

I think for my group, and for most groups in general, this is only used for checking grades. More feedback could be given here specifically for timeouts rather than the generic "Your submission did not complete in time for #cx", since that would be more useful (r97\_Addtl)

The artifact quality was vital, but the focus area didn't feel particularly helpful. Usually that area was failing in the test cluster already, so we already knew there was an issue. (r98\_Addtl)

Its helpful at generalizing where you went wrong, I would hope they give at least some more detail about why the test failed. (r99\_Addtl)

Kind of helpful that it tells you very generally where to look (r100\_Addtl)

I didn't really use this feedback because it's only like the overall score and can't help me to identify the specific problem. (r101\_Addtl)

It was a good hint (r102\_Addtl)

I really just treated that as done or not done (r103\_Addtl)

I used it to check my grade and see if the implementation covered everything that was specified in specs. The focus area was helpful, I hoped it would have been longer but it helped pinpoint where we should focus on. (r104\_Addtl)

I used this when I think my task is complete. But the feedback is very limited, sometimes I can check all buckets and still in acquiring and I don't know where the problem is, or sometimes I cannot check a cluster even though my local tests pass and I don't know what to look at. (r105\_Addtl)

Almost too useful that it gives huge hints where to work on, something you may not get in the real world. Also shows me my grade so obviously very important. (r106\_Addtl)

The feedback was too generic, there could be countless issues in a function (r107\_Addtl)

When it gave a focus area it was useful (r108\_Addtl)

I like the FocusArea point especially, this helped figure out where we should look (r109\_Addtl)

Helped guide my partner and I towards proficient (r110\_Addtl)

Artifact Quality was critical for gauging progress. Focus Area was very useful in identifying potential issues. (r111\_Addtl)

To check my grade (r112\_Addtl)

The focus area was often very vague and we didn't know what issues to address. (r113\_Addtl)

Very useful. (r114\_Addtl)

Not sure how this differs from the above question. Ill Repeat: 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) (r115\_Addtl)

The feedback was usually vague and not very helpful for figuring out where the remaining issues are. (r116\_Addtl)

i had no idea what ValidEdgeCases meant; was it for add dataset or perform query (r117\_Addtl)

It helped us focus on the most broken part of the codebase. (r118\_Addtl)

Used it to mainly indicate if I needed to work on a checkpoint more. Knowing if I reached proficient was helpful in showing my progress, and the focus area was useful for figuring out which parts of code were missing parts. Like the previous feedback, these were quite general which made it harder to find what I needed to improve. (r119\_Addtl)

The Additional Feedback can give you a direction on what you should improve. (r120\_Addtl)

This helps us decide where to focus for debugging our system if the current tests are failing for completed features (r121\_Addtl)

Gave a hint on where to focus our efforts (r122\_Addtl)

this is more usefull (r123\_Addtl)

In #c2 we did not receive feedback that was completely reflective of our implementation nor was it very helpful for identifying issues. (r124\_Addtl)

tgbf (r125\_Addtl)

Cuz i aimed for 100% grade, any level except Proficient doesn't matter hhhah (r126\_Addtl)

The focus area feedback is sometimes confusing where it recommend us to work on something that we got a checkmark for but not the ones that we didn't get a checkmark for which is sometimes very confusing, and we would lean towards ignoring that feedback in that case. (r127\_Addtl)

It has not been helpful at all. It gives vague descriptions on what I need to work on (r128\_Addtl)

same as before, it helped us focus on what we were doing wrong and validated what we were doing right. (r129\_Addtl)

not enough feedback on what went wrong. (r130\_Addtl)

I used this feedback to grade my code, it was helpful in figuring out what to fix, I do hope they gave more detailed hints though (r131\_Addtl)

As said above, it's helpful for gauging where exactly my grade was in each checkpoint (r132\_Addtl)

Sometimes it was very helpful, yet sometimes it would just say, "Edge cases", which in my opinion is not very helpful. (r133\_Addtl)

I think it would have been great the area of focus was a bit more detailed but this was still pretty helpful. (r134\_Addtl)

I did not use the focus area feedback often since the feedback above usually covered it, but knowing which grade you currently have is very useful. (r135\_Addtl)

focus areas seems to just mirror exactly what you got wrong in the smoke tests. (r136\_Addtl)

It would have helped for focus area to be more detailed (r137\_Addtl)

we used the artifact quality bullet to decide whether or not we were done with the code artifact. the focus area pointer was a bit dicey, as sometimes it would recommend us to work on a part of the implementation that we were sure did not have any issues, rather than a part that was also failing the associated smoke test. it became a bit hard to navigate especially since the test clusters are not completely isolated from each other. (r138\_Addtl)

Sometimes it might be helpful in terms you know where to look, but most of the time it is useless as you can see from test clusters already where you are lacking so it is just repetition without any helpful feedback (r139\_Addtl)

Could have been less vague. (r140\_Addtl)

I used it a couple times to figure out where to go, but not that useful. (r141\_Addtl)

I don't find it that helpful as mentioned above about wanting more detailed feedback as the focus area is just way too broad to be useful sometimes. Also I've had times with multiple commits and still being stuck at developing, with the wording being "you're almost to proficient" sounding really fake. (r142\_Addtl)

It tells you if you need to stop working because you're done, but Focus Area provides very little information. We are given no indication as to what kind of error would lead to this recommendation. (r143\_Addtl)

meirl: "Ok I have to work on RoomQueries. Great. But what's the problem exactly??? All my tests are passing so I am truly baffled." also me: "guess I'll go to OH" (r144\_Addtl)

While this feedback was useful, I thought that it needed to be more detailed. (r145\_Addtl)

I think the artifact quality wasn't that useful. If we were able to know exactly what was wrong then it would be more valuable feedback (r146\_Addtl)

This piece information overlaps with the smoke tests above, and the Focus Area part did not really provide any useful information. (r147\_Addtl)

sometimes bugged (r148\_Addtl)

This is not helpful, it only indicates where I am roughly, but not very specific. (r149\_Addtl)

The feedback is not accurate. (r150\_Addtl)

It was helpful for checking when I was done with a code implementation. As well where to look for possible errors, and what specifically to test. (r151\_Addtl)

As seen in its content, these additional feedback messages indicated our current progress, and would be helpful for the next task to focus on. (r152\_Addtl)

Google. (r153\_Addtl)

Focus Area is useful to know where to improve; the bracket grading is not so much useful (r154\_Addtl)

Just to check what grade we have at the moment (r155\_Addtl)

Helpful to see which general area I should be focusing on. (r156\_Addtl)

I felt that this was pretty helpful, and directed us to focus on the most important areas (r157\_Addtl)

Unless we are in near completion of our checkpoint, this feedback meant little to no value in aiding our progress. (r158\_Addtl)

It can help me to narrow the range of debugging (r159\_Addtl)

This was not helpful for progression and only really useful for determining when you were done. (r160\_Addtl)

As previously mentioned, this does not seem to have direct relationship with number of checkmarks get above. (r161\_Addtl)

Not helpful. I am not sure why specifically do I need to look more closely into RoomQueries (when it is already fully marked). (r162\_Addtl)

The additional feedback was not always the most useful, especially when c2 came around, where failing some performQuery tests appeared to be reliant on the corner case handling with `addDataset`. (r163\_Addtl)

It was helpful to know that I was making progress. However I felt that my code was implemented according to spec and didn't know what the autobot was complaining about which was frustrating (r164\_Addtl)

We almost never read that because we dont find them to be particularly useful. (r165\_Addtl)

A very frustrating moment was where our autotest implied there was only one thing which didn't work, but when fixed it unveiled a new issue which we didn't have time to fix but could have done at the same time as working on the previous bug if we had known about it. (r166\_Addtl)

Didnt really use it, maybe just to check our current grade (r167\_Addtl)

It is really unspecific in how we could improve RoomQueries (r168\_Addtl)

Sometimes incorrect and not specific enough (r169\_Addtl)

This feedback was not useful for me as it seemed to just reiterate the cluster statuses that were shown above it. (r170\_Addtl)

I appreciate that it told us what to work on. But man, it was really hard to identify how to fix what we did when from our perspective everything seemed to work great. So it was so hard to tell what exactly was going wrong you know? Like was our functionality not behaving correctly because we misinterpreted it? Or just not handling an edge case appropriately? (r171\_Addtl)

Like I said previously, don't make the percentage/coverage of local/my own test a condition to get feedback from you. What's the point of you(the autoTest or autoBot) and the 'feedback' then ? (r172\_Addtl)

Not super helpful, often gave contradicting advice. (r173\_Addtl)

The Focus Area isn't helpful to tell which part of RoomQueries to work on further, especially since it already passes the smoke tests. (r174\_Addtl)

Let me know if my project is good enough (r175\_Addtl)

Focus Area sometimes is also misleading. (r176\_Addtl)

never used this (r177\_Addtl)

the feedback was very vague sometimes especially when ours was something similar to "valid edge cases". We had no idea where to start and couldn't find the bug in the end. (r178\_Addtl)

it was especially not helpful in identifying the actual cause of any missing functionality (r179\_Addtl)

Helps us gauge the amount of work we have left. (r180\_Addtl)



Useful to tell when each bucket grade was reached, focus area was not super specific and sometimes did not offer much insight compared to the smoke test themselves (r181\_Addtl)

I don't use this feedback. Absolutely useless. Message should be more detailed (r195\_Addtl)

This feedback helped our group understand what parts to take a look at and fix. (r182\_Addtl)

giving a bucket and not a percentage was awfully stressful (r196\_Addtl)

I used it to judge my grade. (r183\_Addtl)

I really liked this feedback, I thought it was useful for figuring out more specifically where things are going wrong or what level I'm at. I think it's nice that it's not very detailed in some ways, but I also feel like because it's not detailed I'll end up walking in circles trying to figure out an issue that's in a certain area without ever knowing what I'm doing wrong. I still don't know what I got wrong or the things I could improve for some sections because we were never given that feedback. (r197\_Addtl)

It is not specific about our grade. (r184\_Addtl)

I mostly used this to check my grade for the checkpoint (r185\_Addtl)

Feedback was sometimes not correct. (r186\_Addtl)

give me positive feedback. (r198\_Addtl)

Good for narrowing down the module we needed to focus on. Was decently useful. (r187\_Addtl)

Always helped me understand what should be my next area of focus (r199\_Addtl)

The best part, I really enjoyed this slightly more detailed feedback and getting a fulfilling moment of progress with the artifact quality. (r188\_Addtl)

Doesn't really tell you what you are missing (r200\_Addtl)

It's nice that it gives a focus area but I wish it gave more reasoning to why it doesn't work (r189\_Addtl)

Fine (r201\_Addtl)

when I need more instructions. I think TA's suggestions are more useful. (r190\_Addtl)

If this part can provide the percentage tests we already passed, that will be more helpful (r202\_Addtl)

Sometimes it was wrong (r191\_Addtl)

I normally follow the advice, but it might not be as helpful if I was already stuck at this step and it keeps recommending the same thing. Something more depth will be helpful! (r203\_Addtl)

I only really used this feedback to assess what grade our project would get if we submitted it in its current state. (r192\_Addtl)

It helped provide a focus area to work on but usually it would already be known that area had some issues so the feedback provided was usually not that helpful. (r204\_Addtl)

no, it's not helpful at all, most time it's misleading and the focus area is not helpful (r193\_Addtl)

This gave me more details on the call. But it would not really show me where the failing test is if the cluster would not pass. (r194\_Addtl)

This was especially helpful. So I know the focus area and my grade. (r205\_Addtl)

almost never got this feedback, tho the times i recieved it, it was somewhat useful (r206\_Addtl)

Only hint as to what was going wrong (r207\_Addtl)

Didn't really say much more than what the cluster was saying already. (r208\_Addtl)

The focus areas had no detail (r209\_Addtl)

The most useful one of all! If only it can give two recommendations at once. (r210\_Addtl)

This told us about what grade category our code is in and what to work on next so yes we found it very useful (r211\_Addtl)

Not the biggest fan of the bucket grading. (r212\_Addtl)

I haven't used this feedback much as the smoke test cluster feedback was usually enough to determine which areas to focus on. (r213\_Addtl)

I don't think I ever used the additional feedback (r214\_Addtl)

would sometimes throw me in the wrong direction. See wildcard example above (r215\_Addtl)

Feedback was misleading at times and could be improved. (r216\_Addtl)

For me personally, the "focus area" was not too accurate, and did not help me pinpoint the parts to work on next. I understand this is not necessarily the fault of the smoke test, but just my experience. (r217\_Addtl)

The feedback here is too vague to be useful. It attempts to provide direction with the focus area, but it ends up being frustrating more than it helps because it doesn't detail what aspects of the area should be focused on. (r218\_Addtl)

Pretty much not at all (r219\_Addtl)

It may have been slightly vague but helpful for a general idea of how close the project was (r220\_Addtl)

The Focus Area is sometimes misleading. (r221\_Addtl)

This feedback is helpful because it lets us know where we are currently at. (r222\_Addtl)

Only used to determine if we reached the highest grade (r223\_Addtl)

no useful too broad (r224\_Addtl)

We only used the smoke test once all our tests passed, so generally we reached proficient in the first or second check. It was still useful to know how far into our implementation we actually were. (r225\_Addtl)

Focus area helped with what I needed to improve in my code, whereas artifact quality just highlighted what grading bucket I was in. Worked fine and had no complaints. (r226\_Addtl)

Focus area too general (r227\_Addtl)

Useful for further improvement of the code. (r228\_Addtl)

Too broad to be helpful (r229\_Addtl)

The feedback provided one useful insight into how our code is preforming but did not go in depth. (r230\_Addtl)

again, it can be better if the actual grade is return to us (r231\_Addtl)

Recommended focus area told me what to look at but i wish it were more specific. (r232\_Addtl)

A more detailed alert would have helped (r233\_Addtl)

It was helpful for finding where in the code our implementation was failing. (r234\_Addtl)

The "focus area" feedback can sometimes just be wrong, leading to a lot of wasted time. It can say an issue is with a query for example, when actually the database setup is the problem. (r235\_Addtl)

This told us what to work on and the minimum mark we had overall. Knowing the minimum mark would be useful if we did not have a lot of time to work with. However, the focus area could use some more detail, as described above. (r236\_Addtl)

bucket grading is useful because it give lower bound of our score and give which method we need to work on (r237\_Addtl)

Yes for telling me brief progress. (r238\_Addtl)

We never really used this feedback to develop our code. Usually what it told us was quite obvious from our own tests that we had built.. (r239\_Addtl)

Really misleading sometimes. (r240\_Addtl)

Quite useful to know what stage I am at. (r241\_Addtl)

also very helpful. (r242\_Addtl)

It sometimes provides incorrect feedback. (r243\_Addtl)

To check my grade, not always helpful as the focus area is pretty vague and doesn't give me much idea what could be the issue usually. (r244\_Addtl)

Its good to know, if the code is enough to pass (r245\_Addtl)

This feedback was used to see our grade bucked in which we used frequently. The focus area feedback was not used as frequently in my case. (r246\_Addtl)

I used this feedback to determine the overall state of the checkpoint. Mostly, it told me when I could stop working on the checkpoint (after I hit the proficient bucket). (r247\_Addtl)

I didn't use this feedback a lot as it is only present for specific buckets. The additional feedback can still be vague sometimes but it's definitely more descriptive than just the smoke test results. (r248\_Addtl)

Gives a good hint on what to work on if we weren't too sure. (r249\_Addtl)

I also used Additional Feedback frequently to check my current grade on the checkpoint and it gives me suggestions on how to improve my work. (r250\_Addtl)

It was useful as it guided us towards the right direction, but it was a little vague in what to change. (r251\_Addtl)

maybe i am used to a % grade, but i feel like knowing exactly how much we got instead of a range of values is a better estimate for me. (r252\_Addtl)

Similar to the smoke test results, it would lend me insight on where I needed to focus on, but was too infrequent of advice to actually help. (r253\_Addtl)

Sometimes its not even accurate. It tells of issues in one place, while the real issue might be elsewhere. This has happened to my group multiple times during this project. (r254\_Addtl)

I thought these hints were very helpful.  
(r255\_Addtl)

Not too helpful since it was too general  
(r256\_Addtl)

The recommended focus area wasn't always the feature that we were recently working on, so it's not always the right focus area. (r257\_Addtl)

Should tell me how should i reach proficient with current implementation (r258\_Addtl)

I think the focus area is not very helpful, because a lot of the time the bug was caused by some other class (r259\_Addtl)