

### Scenario 1:

1. When coming into a new project that has lots of code and has been in the making for a period of time There are a few steps I like to take to dive myself right in. First I would like to have a mentor; someone I work with for a certain amount of time that is familiar with the project. This will allow me to work with the ability to ask the important questions about the project. Second, taking on small tasks and building up to the major ones; this is something I did at my current job where I start with implementing screws and cables into the simulation before handling full cabinets and other components. Lastly, taking the time to really dive into the project and look around with my task. Jumping the gun right away and asking questions is not always the best. If I can take a little bit of time to try and understand it, that is where the connections form.
2. I have started a refactor before in a few of my projects. Things that are important to take inconsideration when looking at the option of a refactoring are the pros and the cons in the following categories: Time,Money,Resources,and complexity. Looking at just some of these categories will allow one to make a better decisions on a possible refactor.
3. Balancing a task of fixing a bug changes on how important the fix is. Sometimes the bug is a small preferred refactor, but not needed. If the bug is minor it should be worked on after a major task; if the bug is important try to get the bug fixed and pushed as soon as possible.

### Scenario 2:

1. When looking at frame drops it is important to look at multiple objects and scripts. Almost always the issue will be too many objects in your scene that are also high polyed maybe.
2. **Step one:** Play the game and find the exact spots where the frames drop. **Step Two:** compare what is different in each spot/ scene. Are there more scripts running, if so how many big Os? Are there more objects in the scene? If so, are they high polyed? **Step 3:** If you are unable to not find the issue, start brainstorming and slowly removing small things from the project. This can solve the issue and make it easier to know if the frames are from lighting,Models,scripts, or even...a small syntax.
3. To confirm that your changes made an impact, always make sure you have the before and after status. Write down your frames, number of tris and verts, how many different scripts are running at a time. After you feel like the problem was solved you will want to check the status again and see if there have been more improvements. Remember we are mechanics if you put it back together with leftover screws remaining it doesn't mean it was built better than before.

### Scenario 3:

1. When you are pulling new pushes you are always able to check what was pushed. This is when the practice of leaving good comments on your pushes comes in handy. Read each new push and trace through the new files. You will quickly learn there is something different that you weren't expecting.
2. Always be kind when you find a mistake in a push after pulling. It's very annoying, but it happens to us all at one point. Depending on how your team operates you can just fix the issue and repush with the new comments of your change; or go ahead and address what the problem is to the person who made the mistake. Whatever you do, do not call one person out in a group chat or in front of people about the issue.
3. Reducing the size could be removing anything that was added by mistake. Shifting through all the files and prefabs very carefully. You can also talk to the modeling team about the prefab sizes. The modeling team could always reduce resolution and other scalings down to make the file sizes smaller.

### Sources:

1. <https://gamedev.stackexchange.com/questions/85807/how-to-read-a-data-from-text-file-in-unity>
2. <https://docs.unity3d.com/ScriptReference/PlayerPrefs.html>
3. <https://answers.unity.com/questions/972817/convert-string-to-int-2.html>
4. <https://answers.unity.com/questions/812667/how-can-i-emit-a-particle-at-a-collisions-location.html>