Problem Statement:  
Problem statements summarize key information about the problem and the environment   
surrounding it. They are often brief, but can provide detail as necessary to describe the problem.  
Consider this sample problem statement written for game engines:

*“Small development teams may find it expensive to split their resources between building the   
game engine vs. building the actual game itself. Teams may need to dedicate at least 1-2   
full-time employees (FTEs) to developing the engine as requirements change over the product   
lifecycle, increasing operational costs.  
Developers may choose to use game engines middle wares like Unity or Unreal Engine in   
exchange for a small upfront cost in the form of licenses or royalties. Engineers can be   
instead be dedicated to developing the game instead, requiring less engineers and thus   
reducing operational costs.  
Game engine middleware is often maintained by teams with dedicated R&D teams, providing   
its users with the latest innovations in the games industry at a fraction of the cost. Teams   
may hire for users experienced in specific game engines, minimizing resources spent on   
training and ramp-up.  
If a game engine does not provide the features required for a particular game, its cost   
savings are reduced. Furthermore, missing features or bugs with the feature set can be   
difficult to fix if source access is not providing, possibly making the game engine more   
expensive to work with than a custom built one.”*

1. In a few sentences, the passage of text describes the following:
   1. What is the problem? – The cost of building and maintaining custom game engines
   2. Who is experiencing the problem? – Small or new game development teams
   3. What is needed in a solution to this problem? – Minimizes the game engine development   
      costs
2. The solution, game engines middlewares like Unity or Unreal Engine, are described as follows:
   1. How does Unity or Unreal solve this problem? – Licenses or royalties for the game engine   
      can be much more affordable than building a custom engine
   2. What are its advantages?
      1. Externalized cost of R&D teams
      2. New hires may require less training and on-boarding to be effective due to common   
         knowledge of the game engine
   3. What are its disadvantages?
      1. Missing or incomplete features can require teams to devote some or all of the cost   
         of developing those features in-house regardless
      2. Bugs in the game engine may not be readily diagnosted fixed if source-access is not   
         provided
3. A link to your GitHub repository

<https://github.com/muildy/space-invaders-project>

~Has the space invaders project with the rest of the items in the assignment

1. A link to the merge commit you created

<https://github.com/muildy/space-invaders-project/commit/8ba82a897957086a8e85a9bff5722e31ea4ddd71>

1. A link to the pull request you filed as a part of your Version Control Exercises
2. A problem statement describing why we use Git for version control:

Individuals and teams of people need safe and reliable storage offsite as to limit risk of catastrophic failure that results in damaged or destroyed work. It needs to have easy access to the project as well as the ability to split up the project and consolidate it later. A good choice is Git or other derivatives of it, as it is both open source, free and relatively easy to use, as it allows for branching and merging of a project, meaning that work can be assigned and worked on without having to wait on others to finish their work, granted there is a plan or overview on how the project works.

* 1. What is the problem we are solving? - Safe and reliable storage of a project that means that an onsite issue doesn’t destroy work, and anyone on a team or individual can have access anywhere, granted they have internet connection.
  2. Who is experiencing the problem? – anyone who has a project worth saving or a team working together on a project
     1. What are our requirements for a possible solution? – a cheap and easy to use source control that allows for branching, merging and conflict resolution
  3. How does this solution (Git) solve this problem? – its easy to use and usually free (at least for small groups)
     1. What are its advantages? – multiple people can work on the same project at the same time, each working on different parts without having to wait on others
     2. What are its disadvantages? – if the communication between people or groups is poor, then two branches that each try to alter the same things will cause conflict, meaning more work as to ensure that merging them will not break anything

1. 5. A list of resources you referred to when learning Git (besides AIE resources), if any

<https://docs.github.com/en/desktop/installing-and-configuring-github-desktop/overview/getting-started-with-github-desktop>

<https://git-scm.com/>

<https://www.what-could-possibly-go-wrong.com/version-control/>