

1. Five mistakes have been integrated into a working solution of Prospector Solitaire.
  - Problem 1: onMouseUpAsButton function not virtual in Card.cs
    - o made the onMouseUpAsButton virtual as the function couldn't be overridden without the keyword 'virtual'
  - Problem 2: For loop out of bounds
    - o Fixed the value of for loop from 1 to 14 to 1 – 13 as there are 13 cards in a suit
  - Problem 3: ScoreManager object SM
    - o Commented out the declaration and fixed the object in the places it was being used
  - Problem 4: The clicks were responding to the target pile rather than draw pile
    - o Switched the case statements so they react to the draw pile instead of the target pile
    - o I also had to add the score floating for when a card is drawn
  - Problem 5: The deck wasn't shuffled
    - o I made a shuffle function in Deck class and then called it in the start function in prospector
2. Limitations of unity
  - Limitation 1: It doesn't have its own text editor like most IDE's
  - Limitation 2: Hard to create complex objects
    - o Lacks proper 3D modelling for complex objects (only has basic objects to make prefabs), and software like blender is better to use
  - Limitation 3: Not good for native applications
    - o Unity is not good for native applications for example we can't use iOS native packages, they need to be imported and configured properly
  - Limitation 4: Crashes and not recovery process
    - o When making a big project, the IDE freezes a lot and has a higher potential to crash
    - o When it crashes there is not way to recover the unsaved changes.
    - o It can crash by just writing a long or complicated loop