Dragon Fightin' Exercise

In this exercise you will create a turn-based RPG (role playing game) where you pit a valiant knight against a fire-breathing dragon and have them fight it out. The point of this exercise will be to test and grow your knowledge of javascript logic as well as your ability to understand and use AUI components.

In this exercise you must integrate the following AUI modules into your game:

- AUI Modal
- AUI Progressbar
- Async-Queue
- Transitions
- Debounce

While these components will be necessary, if you feel the need to use others, don't hesitate to add them as well.

The way you structure your game to run logically should be as follows:

1. A modal pops up to alerts you that the dragon fighting is about to begin
2. There is a clear visual representation of a knight and a dragon with both of them having the following: attack statistics, special attack statistics, a healh bar and an experience bar (*knight only)
3. There should be two buttons (attack & special attack) that are clickable and will cause the knight to do whichever action is chosen
4. Once either of the buttonsis clicked and the knight attacks, the dragon automatically (and randomly) attacks back with either an attack or a special attack
5. When the knight/dragon uses their attack/special attack there should be an

animation that takes place on the page for each attack

6. The damage done from both the knight's and the dragon's attack should be based upon their statistics and the healthbar should reflect the damage done without rendering a new page.
7. Every time the knight hits the dragon his experience bar should add experience, that once filled, will cause him to level up and have his health refilled as well as increase his total health, total attack and total special attack statistics.
8. Once the first dragon is defeated have a second replace it without rendering a new page.
9. Lastly have everthing that happens during the turns (for both knight and dragon) logged in a clear way so that the user can easily track what happens.

Do your best to make this as modular as possible so that code repition is kept to a minimum and the code is as light as efficient as possible. Most of all, have some fun! Good luck.