



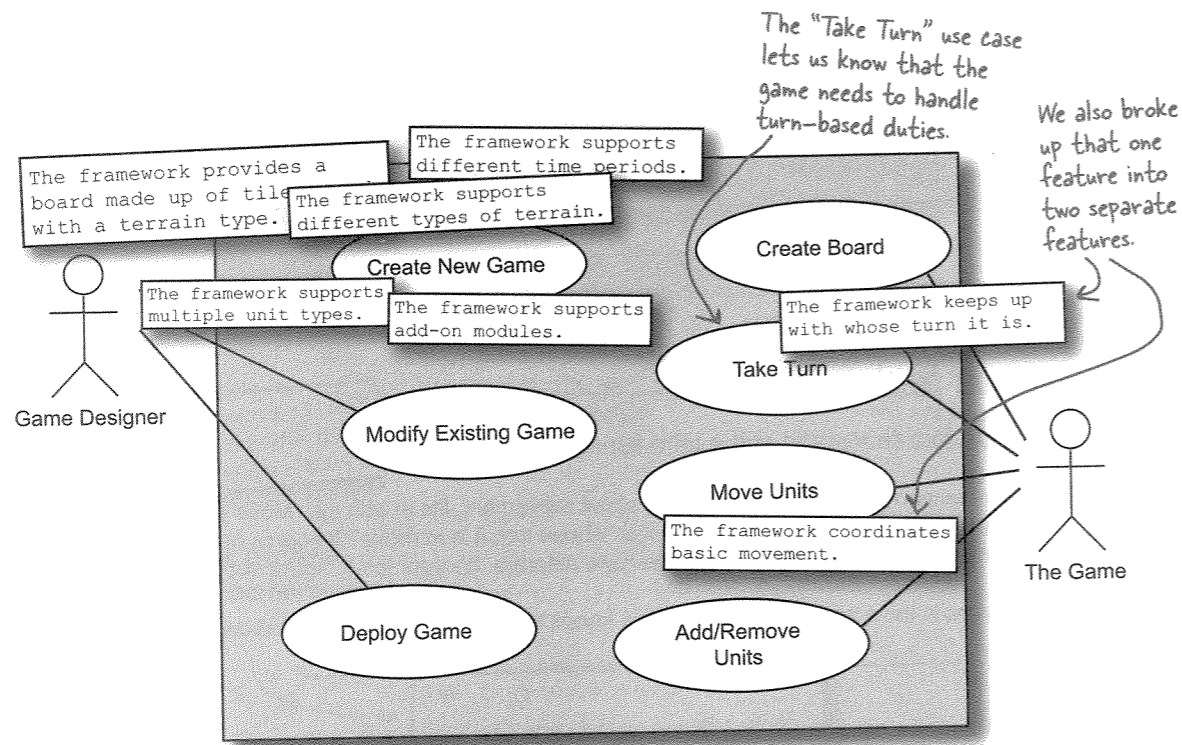
That last feature is still a little funny...

The framework keeps up with whose turn it is, and coordinates basic movement.

The *second* part of that last feature, about movement, fits in with the "Move Units" use case... but what about keeping up with whose turn it is to move? It seems like there's something still missing on our use case diagram. It's your job to figure out two things:

1. Who is the actor on "The framework keeps up with whose turn it is?"
The game is still the actor... it's using the framework to handle managing whose turn it is.

2. What use case would you add to support this partial feature?
We need a use case for "Take Turn" where the framework handles basic turn duties, and lets the custom game handle the specifics of that process.



So what exactly have we done?

You've got a list of features that Gary's game system framework needs to support, and that tells you all the major pieces of the system you need to build. This is a lot like the requirements list you built way back in Chapter 2 for Todd and Gina's dog door... except it focuses on the big picture.

Use a feature or requirement list to capture the BIG THINGS that your system needs to do.

Once you've got your features and requirements mapped out, you need to get a basic idea of how the system is going to be put together. Use cases are often too detailed at this stage, so a use case diagram can help you see what a system is like at 10,000 feet... kind of like a blueprint for your application.

Draw a use case diagram to show what your system IS without getting into unnecessary detail.

Here's our feature list... the system has to do these things.

- ### Gary's Game System Framework Feature List
1. The framework supports different types of terrain.
 2. The framework supports different time periods, including fictional periods like sci-fi and fantasy.
 3. The framework supports multiple types of troops or units that are game-specific.
 4. The framework supports add-on modules for additional campaigns or battle scenarios.
 5. The framework provides a board made up of square tiles, and each tile has a terrain type.
 6. The framework keeps up with whose turn it is.
 7. The framework coordinates basic movement.

