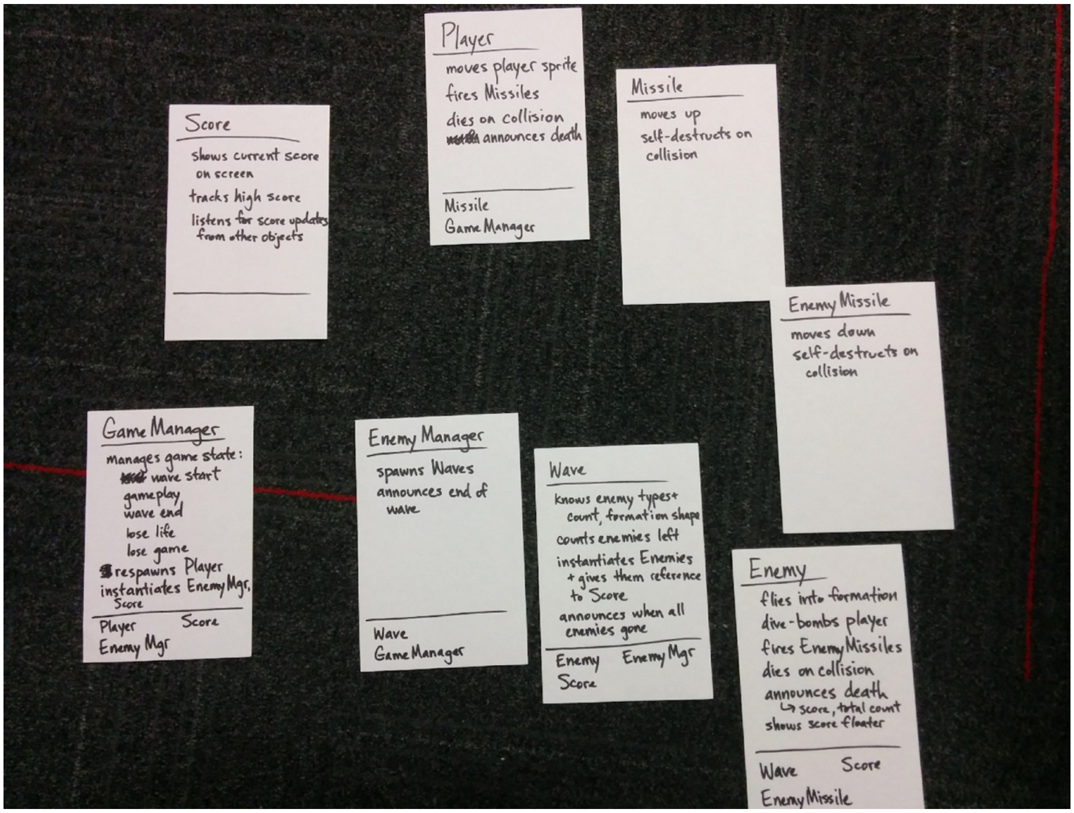
Assignment: CRC Cards

In this assignment you're going to use a tool called CRC cards to think through how you will set up your code. This process usually doesn't take more than half an hour for a small game, and it has two powerful benefits:

- A little planning is guaranteed to save you a lot of debugging.
- As you think through how to set up your code you'll get a sense of how much work you need to do, so you can check whether your scope is realistic.

What are CRC cards?



CRC cards are literal, physical 3"x5" (or 8cm x 13cm) index cards. Each one shows three things:

A Class Name
Its Responsibilities
Its Collaborators

On each card, write the **name** of a C# class you plan to create for your game. Under that, describe what the class **does**. Be reasonably specific. If you can't fit all the responsibilities onto the card, your class may be trying to do too much and you should consider splitting it up.

Next you'll think about what **other classes** this one might need to talk to – its collaborators. Write the names of those classes at the bottom of the card. (If there are two classes that need to talk to each other, write it down on both cards.)

When you've created your cards, group them so that collaborators are near each other. Imagine a sequence in your game, e.g. the player defeats the final enemy in a wave or picks up a power-up. Trace through all the things your code needs to do in response and make sure that each step is represented on your cards. Try this with another sequence or two.

You will be tempted to just do this in a Google Doc, or by drawing a grid in a notebook, instead of digging up index cards. But one of the ways this exercise helps you is that it lets you physically group cards together and rearrange them as you think about them working together. This spatial organization allows you to create a mental picture of your code that will help you keep things clear and separate in your mind for the rest of the project.

For beginners

If you've never tried to plan code before, you might feel intimidated by this assignment. The most important thing about planning your code is *just that you do it*. You'll get better at it as you gain experience.

To get started, look at your concept document and the drawing you made of gameplay. Look at each item on the screen and think about what it needs to do.

For example, the tower in a tower-defense game needs to track enemies that are nearby and fire at them. The player character in a platformer needs to run and jump. They might also need to keep track of health, or pick up and fire weapons. That's starting to sound complicated, so you might split the responsibilities into PlayerMovement, PlayerHealth, and PlayerWeapons.

If there's more than one of something on the screen, think about how they will get there – will you hand-place them or spawn them from code? Either way, when you have more than one of something think about whether you need another class to manage them.

Finally, think about how things change as the player plays the game – for example, when the player eats a power pellet, or when all the enemies have been cleared from a level. What class will keep track of this information? What class(es) will respond to the change?

Turning in your work

Take pictures of your cards and paste them into a document. Then make the document a PDF and upload it to the gallery. Peer feedback isn't required, but look over your classmates' plans – you might learn something, or you might even help them find a better structure.