If you are like most developers (myself included), you most likely have opened your favorite editor or IDE and just started developing. You may have had some vague idea of what you wanted your program to do, but you probably just made it up as you went along. Although this can work well for small projects, or learning new techniques, a design-as-you-go approach on a project of a nontrivial size will cause you a lot of headaches. As someone who has started more than one project as an experiment with no plan to grow it into something, I can attest to the fact that you encounter far fewer problems if you spend a little bit of time planning your project out, to understand its scope and anticipate where the hard parts of it will be. This is especially important if you plan to produce an application that is fun and scalable to all the individuals who may install your application on Facebook!

In this chapter I discuss a design process that can help you structure your ideas and build a better application. There are a lot of different formalized discussions of these patterns (e.g., waterfall, Agile, etc.), however, I talk about one that has worked well for me on the projects I have been involved with, taking techniques from several approaches and applying them to my own workflow. I encourage you also to take a look at other approaches and develop one that works best for how you approach problems and does not get in the way of your workflow, but helps streamline your thought process and coding practices.

The Design Process

The word design is overloaded, but for our purposes here I use design to refer to everything from the application idea, code, and the integration of image and sound assets. I cover some top-level steps that you can follow, and if you ascribe to the Agile methodology, you can use these steps to shape your iterations. Remember, for everything you plan, you need to keep the big picture in mind so you do not veer off course and develop a product that does not fit your original vision, or get so bogged down in one particular aspect that the entire project is stymied. For myself, I typically break the design process out into the following areas.
1. Brainstorming ideas.
2. Research potential audiences.
3. Identify the competition.
4. Review the project; figure out what you know, what you need to figure out (and do some research to approach the problem), and develop an overall vision for the application.
5. Decide on an artistic direction for the application.
6. Cull aspects of your project that are not practical for constraints on your time, skill, and computing resources.
7. Plan project milestones.
8. Write your code.
9. Deploy your code.
10. Test your application with users.
11. Launch.

This list is by no means comprehensive, and it simplifies the iterations that you will find yourself doing as you refine your project design, but it does give you a place to start your planning. Let's get into these topics in a bit more detail now.

**Brainstorming**

Before you can code anything, you need an idea, a novel or interesting concept that you believe some audience will enjoy or find useful. In the brainstorming stage, simple ideas that you can easily explain to others are important. If you come up with a concept that takes a long time to explain, or is confusing when you attempt to explain it to others, it may be time to come up with another concept, or to simplify the one on which you are working.

Once you have a concept, it is a good idea to codify it in a series of steps, which clearly lays out the game or application rules for the end user and clearly defines the objective or purpose of the application.

**Idea:** Klondike Solitaire.

**Objective:** Order a deck of 52 playing cards by suit from Ace to King.

**Rules:**

- This is a single-player game.
- The game of Klondike is played on a table.