In Chapter 5, you built a simple game in which a player dodged moving bombs. This gave you an excuse to use many of features and programming concepts central to creating games on an Android tablet. In this chapter, you build a more sophisticated game.

Your big task for this chapter is to build a pong-type game in which players use a paddle to keep a ball bouncing as they attempt to hit and destroy blocks with the ball. My first experience with a mobile game was on an old Blackberry where the only offering was this simple game. I had to control the paddle with the clumsy Blackberry trackball, and the small screen size and low resolution made the effort less than satisfying. Surprisingly, that game was written with the powerful Java language, the same one you use here to create your much more engrossing and fun game.

As you build the paddle game, you'll master new skills you can add to your toolbox. You add additional images to your resource files. You replace the character and bombs from the Chapter 5 AllTogether game with a paddle and blocks. To keep the ball in motion, you manage the interaction of the sprites and detect a greater number of collisions. You have to add some additional physics to the game, requiring more calculations on the fly. You also reward players more effectively with sounds and disappearing blocks. Finally, you learn a technique to initialize multiple blocks with a single XML layout file.

Let’s get started.

Getting Started

Let’s begin by gathering the images and other resources you will use in the paddle game, and then open a new project for your work.

Gathering Game Resources

Because a pong-style game uses fairly generic shapes and objects, you shouldn’t have a lot of trouble making the graphics. The most important consideration, of course, is the relative scale and size of each of the elements. The paddle must be large enough to hit the ball consistently, yet small enough to make it a challenge for the player. You see as you go that other images can be added if you want to allow power-ups and bonuses to fall onscreen.

Figure 6-1 shows the images and dimensions of the graphics you use for this game. Notice that each of them is a different .png file. For my implementation, I drew them myself using GIMP, an open source tool mentioned in Chapter 2.

In addition to the regular graphics and sounds, Chapter 7 will incorporate the use of a new resource to store the layout of levels. Instead of coding in the position of each block, you specify it with an XML layout. This is the tricky part of this project, so I’m saving it for the next chapter. This first demonstration uses just three blocks without any additional resources for their placement.
Figure 6-1. The block (top image) is 30 × 50 pixels, the ball (middle image) is 30 × 30 pixels, and the paddle (bottom image) is 30 × 200 pixels.

If you’re concerned about using a black ball (because the background has traditionally been black), have no fear. You can very easily change the color of the background. In fact, using a lighter color makes the game more inviting to the player.

**Tip** The paddle and ball images are partially transparent. You can do this by selecting the color white to be transparent in the GIMP program. I strongly suggest you do the same, because the game appears much more professional if you aren’t dealing entirely with blocks. You’re lucky to have the ability to use images with transparent layers, when other languages require code to make elements transparent.

This game is much more immersive if you have some nice sounds to go along with the game play. Because a pong game doesn’t conjure a distinct set of sounds, you’re can use whatever you wish. I chose to use only one sound: a short MP3 “twang” that plays whenever the ball collides with a block. The code doesn’t include any other noises or music, but you’re free to add them. When you start a new game, the simpler it is, the easier it is to find the errors and bugs in your code.

**Creating a New Project**

Because your game is complete (that is, it has user interaction, an objective, and the ability to win), you should treat it as a professional app rather than an exercise. Because of this, it’s better to use specific names for the elements and code. Therefore, let’s name this app TabletPaddle. Although not creative, this name describes your new take on a pong-style game.

To get started, follow these steps: