



Time for a Break: A JavaScript Game

In the first book I ever had published (*Practical Ajax Projects with Java Technology*, Apress, 2006), the final project—the apex of the book—was an adventure game named Ajax Warrior. It may well be the start of a trend, where every book I write includes a game, because that’s precisely what we’re going to build in this chapter! No, it won’t be Ajax Warrior again. It will be a more arcade-style game, since there is no network latency to bother us.

You’ll see many neat tricks here, a lot of JavaScript and DOM scripting, and even some basic game theory along the way. At the end, you’ll have something that you can use to slack off at work any time you wish, or anywhere else you have a browser, for that matter! We all know the saying . . . all work and no play makes Homer . . . something . . . something,¹ so let’s stop the axe from falling, shall we?

K&G Arcade Requirements and Goals

The game we’ll build is a port of a PocketPC game that I wrote entitled K&G Arcade. The K&G stands for Krelmac and Gentoo, who are two wisecracking aliens bent on the destruction of the Earth. Unfortunately, they are like idiot teenagers, who just happen to have quantum destructo beams!

In K&G Arcade, which you can see at <http://www.omnytex.com/kgarcade>, you play the part of Henry, a mild-mannered farmer from jolly-old seventeenth-century England. Krelmac and Gentoo abduct you one night, and force you to try to escape their spaceship, which consists of five maze-like levels inhabited by teleporting robots that kill on contact. On each of those five levels, you find five mini-games each, which you need to play and beat (by achieving a given score in 60 seconds) in order to escape. You also meet up with other abductees, who you talk to and try to gain their trust so that they will give you clues about certain mini-games that are impossible to beat without a particular trick.

The full-blown version of K&G Arcade features cinematic cut scenes with Krelmac and Gentoo cracking wise and generally making pests of themselves. It includes an all-original soundtrack and hand-drawn cartoon graphics. K&G Arcade is actually the second game featuring

1. If you are a Simpsons fan, you almost certainly know the reference and are laughing right now. If you aren’t, it’s a line from the episode “Treehouse of Horror V” in the segment entitled “The Shinning,” a parody of *The Shining*. I suggest grabbing a copy—it’s a riot!

these characters, the first being Invasion: Trivia! (http://www.omnytex.com/products_invasion_info.shtml). Going to that site will also lead you to a Flash cartoon introducing these characters.

Now, our goal isn't to port the entire full-blown K&G Arcade to JavaScript. Indeed, that would be considerably more difficult, if possible at all, and would take up a book this size on its own! Instead, we'll scale it back quite a bit and implement just the mini-game portion. In fact, we'll build only 3 of the 25 mini-games. Let's get into some details:

- We'll implement three mini-games—Cosmic Squirrel, which is similar conceptually to the classic Frogger; Deathtrap, which is inspired by the Indiana Jones movies; and Refluxive, which is similar to Arkanoid, Breakout, and games in that mold (but without actually breaking anything, as you'll see!).
- We'll implement a mini-game selection screen that includes a screenshot of the mini-game.
- We should reuse existing code wherever possible. However, we will *not* be using any libraries for this game. That's because in writing games, you frequently want to be "as close to the metal" as possible, and that's the case here as well.
- Each mini-game should be its own class, and should inherit common code from a base class.
- Extensibility should be a priority so that more mini-games can be added later with little difficulty.
- In general, we want to keep global scope as clean as possible, and use good object-oriented design techniques throughout.

When doing game programming, you often try to get as low-level as you can—as close to the hardware as you can. The reason for this is simple: performance. In a game, a lot has to happen in very short time periods, so there can't be a lot of superfluous code executing or extra work being done. One of the best ways to ensure this is to not entrust things to libraries. Now, this isn't an absolute. It is often true that you can get better performance with a well-written library than without. It's also true that in the modern era, you typically don't get as low-level as you used to in general, with or without a library. In the past, it wasn't unusual to write important portions of a game in assembly language so that it could be as optimized and tight as possible. These days, that isn't as prevalent (it's still done, but not as often). So, in this particular application, we won't be using any libraries. We'll be doing all "naked" JavaScript.

Programming a game in JavaScript isn't fundamentally different from programming a game in any other environment. Some of the details are different, of course, but the overall concept is roughly the same. Rather than espouse those concepts here in one place, I'll talk about them as we progress through the code.

And with that statement, let's begin our exploration of K&G Arcade by taking a look at the game itself.