

*Blazing Games Guide to Flash Game Development Chapter 5: Overview of First Games*  
Written by Billy D. Spelchan for [www.BlazingGames.com](http://www.BlazingGames.com)  
Copyright © 2003-2005 Blazing Games Inc. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the file called fdl.txt

# Chapter 5

## Overview of First Games

### Contents

In this chapter we overview this part of the book where we create two simple games.

- Goals of this Part - What you should come away with after reading this part.
- What is NIM - A look at the first game we are going to create.
- NIM Design - Designing the first game.
- What is Bomb NIM - A look at our second game.
- Bomb NIM Design - Designing the second game.

## Goals of this Part

The purpose of this book is to introduce readers into the world of game design and Flash game programming. In larger companies, the design of the game and the programming of the game may be done by different people. I am making the assumption that most of the people who are reading this book are doing so in order to create smaller games which can easily be created by a single person. In fact, if you look at my Blazing Games web site, you will see that it is possible for a single person to create fairly decent games in a very short period of time. In fact, the target time I use when deciding on a game for my site is that per week that the game will appear on my site (remember that many of my games consist of multiple episodes) I should spend no more than three days putting together the game. This, of course, is because my Blazing Games site is more of a hobby. My real money comes from doing contract work for other companies (something that has slowed down due to the internet bubble bursting).

While I work as a programmer, and have been programming for far too long, I do understand that many people are not programmers so the concept of programming is intimidating. Unfortunately, if you are going to create an interactive Flash movie you are going to have to do some programming. With that said, I have decided that it is best to ease readers into the Action Script aspects of Flash game development.

The first game we create in this part of the book, while having a bit of Action Script in it, tends to make choices that reduce the amount of Action Script in the game. After finishing this first game, we then take a more advanced look at the Action Script language in preparation for our second game.

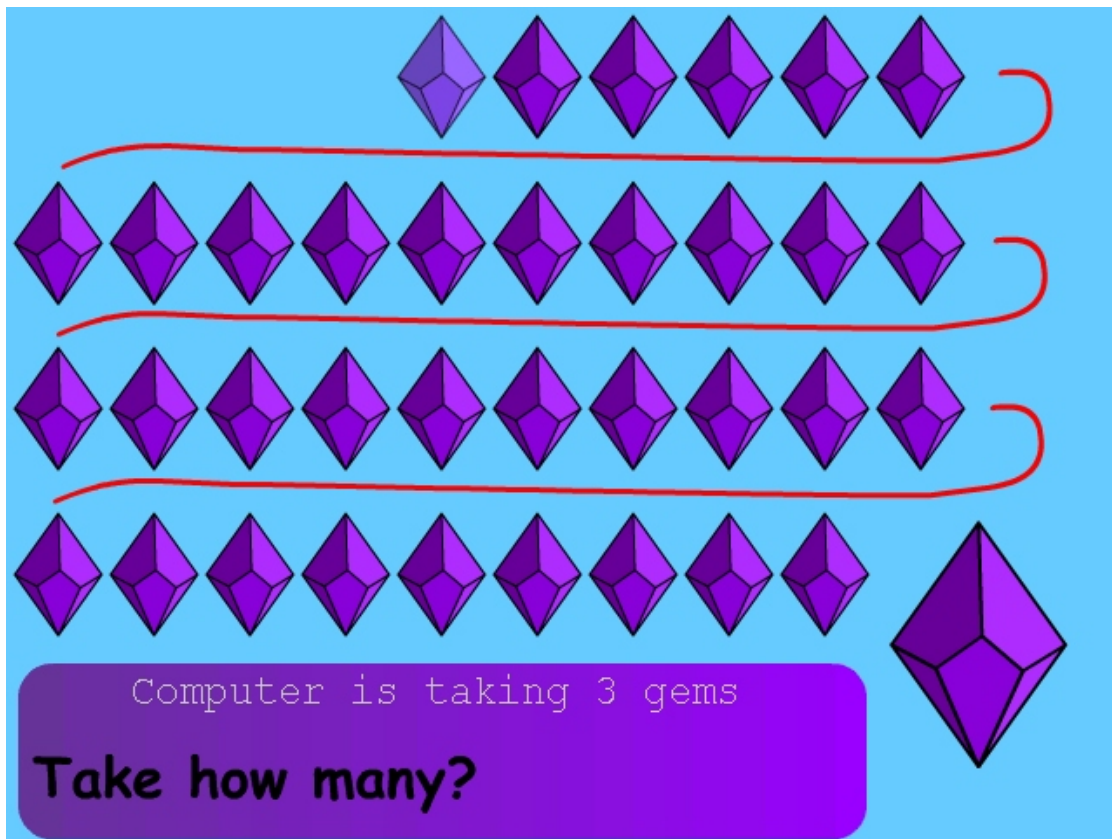
The second game, which is a variation of the first game, could be done with a minimal amount of Action Script. Instead, we are going to dive into Action Script so we can gain a perspective on what is possible to do with Action Script. By going through both games you should then have an idea of the tradeoffs there are between doing something in Flash as opposed to doing something with Action Script.

## What is Nim

While I have seen this game go by other names, I was introduced to the game as NIM. I have no idea why it was called that, but figured that I would stick with the name I am familiar with. The game is a very simple game where you have a set number of objects. Each turn the players can take from one to three objects. The goal of the game is to be the player who grabs the last object.

Apparently this game has been used within the mathematics of Game Theory to demonstrate certain principles. I have also heard that the game has been mathematically solved and that there is a way of playing the game that will guarantee winning the game. I have not researched either of these things, but readers who are interested in this game may want to do research on these topics.

The game has already been posted on my Blazing Games site. Here is a screen shot of the game:



**Figure 1** - Nim Screen shot

## Nim design

While NIM is a very simple game, it is a really good game to start learning game development with. The game is very straight forward in its presentation. The rules of game play are very easy to implement. Most importantly, at least for beginning Flash developers, the game requires only a minimal amount of Action Script.

While I personally have no problem with Action Script, I know a lot of Flash developers who fear the scripting language. Games, by their interactive nature, require scripting. There is, however, no need to jump right into extensive Action Script use. Our first project will attempt to keep Action Script to a minimum.

We already know what the game is about from the description in the previous section, so what is needed is specific details on the game. There are three things that have to be decided. What is the object that is being removed, how many objects are there to be removed, and how does the player remove the objects.

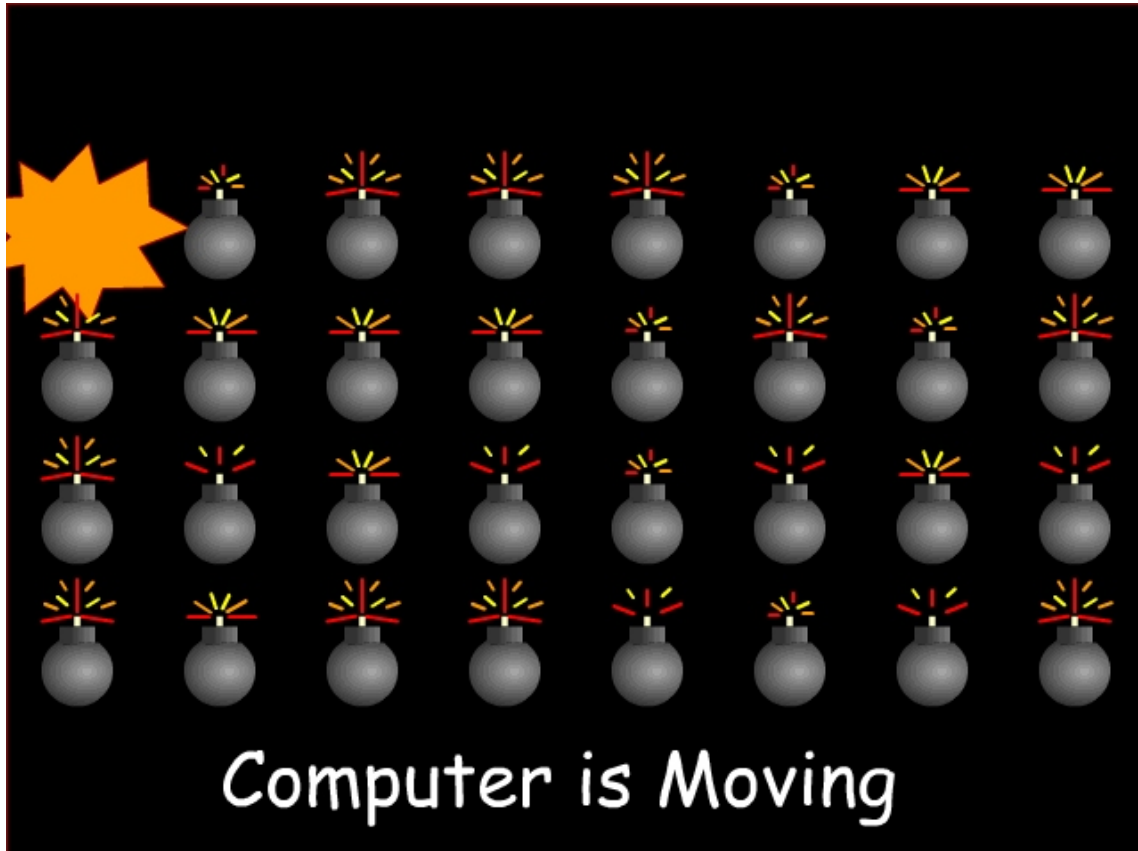
Both questions can be answered pretty much any way the designer wanted and the game will still work fine. For the object, anything could work. Personally, though, I look at the object as something valuable that the person would want the last one. To me I think of jewels or gems when I think of something valued. The reason that gems are valued is because they are rare. The bigger the gem the rarer it is. This means that if the last gem is a big gem then it is a very valuable gem.

The number of objects is simply a matter of how many can fit on the screen. Four rows of ten seemed to make sense, so for this version we will have 40 gems. This would also leave room for a space where the user could specify how many gems to take. As all the user has to do is select how many gems to take, the user interface is simply three buttons, with a text area to describe what is happening.

How to remove the objects is a bit trickier. Two ways come to mind. First, we could have the player click on the gem to be removed. This would be fairly intuitive, but without use of Action Script would be a huge amount of work. The second way would be to have a control box with the number of objects to remove being buttons in the control box. Flash has very good button support built in, so this task would be fairly simple.

## What is Bomb NIM

Bomb Nim is a variation of the Nim game. It plays the same way as the Nim game but with one major exception. Instead of trying to take the last object, the player is instead trying to make the other player take the last object. This minor change actually changes the strategy required to win the game by quite a bit. The game is using bombs as the objects as it makes more sense to not want a bomb.



**Figure 2** - Bomb Nim Screen shot

## This Time Its Action Script

Some of you may be wondering why we are making essentially the same game as we have already made. Two main reasons. First, the game is different because of the change in strategy required. Second, this time we are going to focus on using Action Script for a lot more of the games game animation. In fact, we are going to be using Action Script for the following things:

- Initial layout of the game board
- Bomb Removal
- Bomb animation
- The user interface

Having the gems positioning themselves was an interesting part of the first game. With the second game, by taking advantage of Action Script, we can have a much more dramatic opening. Imagine all forty bombs on the screen at once moving to their target position. Having forty layers animating at the same time is a daunting undertaking. Thanks to Action Script, however, we only need one layer and a bit of code.

Bomb removal is simply the bomb exploding. For variety, we will have three different explosions. The explosion to show will be determined at random. By having the bomb object handle the removal animation, we greatly reduce the amount of animation work we have to do. We can also add some random fuse animation to the bomb, giving the game a much more animated look.

The nice thing about programming is that there is never a single way of doing something. The same is true when it comes to user interface design. When I was originally designing the Nim game I had thought of an alternative way of handling the user interface.

A more visual approach would be to have the user click on the bomb they are taking, allowing them to take the first, second, or third available bomb. If they selected the third bomb, they would take three bomb. The bombs that are about to be taken would be highlighted so the user would know how many bombs they are going to take.

I went with the button panel approach for the first game simply because it was the easier and quicker approach. To actually implement this alternative interface there are two problems that have to be addressed. First, how to handle the positions of the buttons. Second how to handle the highlighting of all the gems that are to be taken.

*Blazing Games Guide to Flash Game Development Chapter 5: Overview of First Games*

If we were using the button approach you would simply place the buttons under the bomb. Highlighting can then be done simply by having the button have an image that is larger than the gem when the mouse is over the bomb or the mouse is clicked. Though that would only highlight the gem that is being selected. To highlight all the gems that are being taken would require a bit of trickery.

Three different buttons would have to be used. The first button to highlight only the first gem. The second button to highlight the first and second gems. The third button to highlight all three of the gems. Sounds simple, but it gets worse. Not all cases have all three gems on the same line. This means that you would have to create alternative buttons for handling multi-line cases for a total of six buttons. You would also have to have screens for each of the forty removal animations.

Obviously, this is a lot of work. This is where Action Script comes in to the picture. Action Script can be used to control the highlighting of the bombs. Quite simply, the Action Script would know if it was the players' turn and if so would highlight the appropriate bombs based on the bomb the player was over. No button work would be needed at all.