

Written by Billy D. Spelchan for 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 26

Finishing the Dragon

Contents

With the first half of the dragon rooms complete, now it is time to finish the remaining rooms of the game and add the title screen to this movie.

- Room #5 - creating the right side corridor
- Room # 6 - creating the moss chamber
- Room #7 - creating the other side of the chasm
- Starting a Fire - Creating a fire to get the sword
- Room 8 Win Movie - Winning the game
- Room 8 Lose Movie - Losing the game
- Title Screen - Adding the title screen

Room # 5

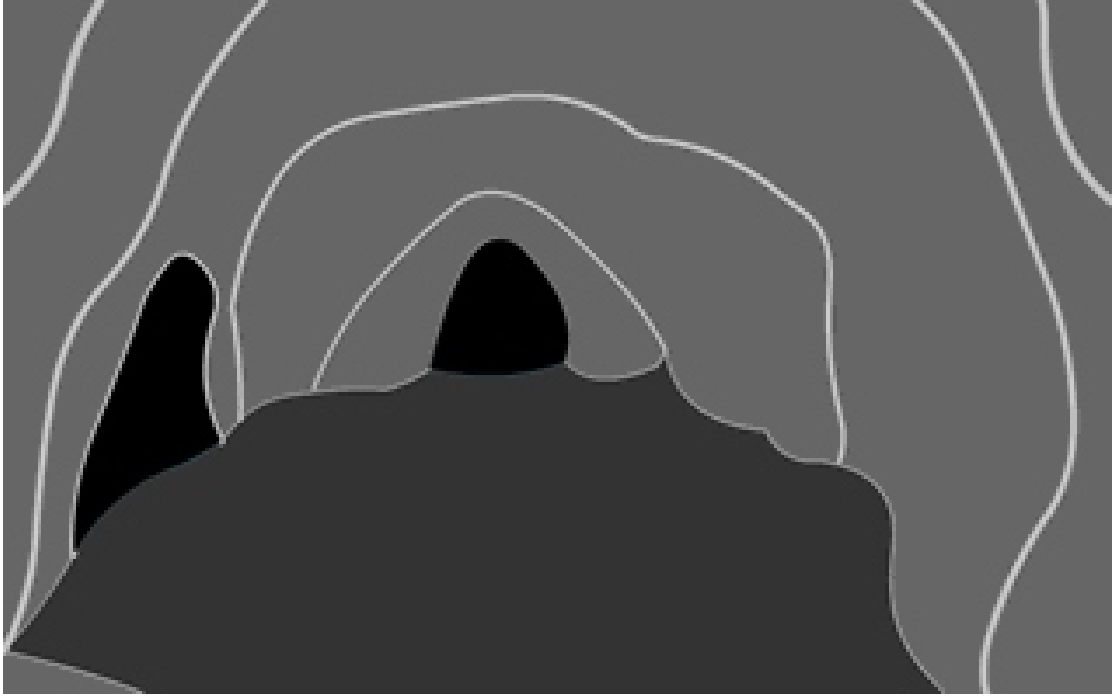


Figure 1 Room #5

This is the right side chamber which leads to the dragon's chamber. There are two exits from this room. One leads to the main chamber (room 4) while the other one leads to the dragon chamber (room 8). Located in this room is the flint. Flint, when used with the steel and the moss is used to melt the ice surrounding the sword. Also added to this room is a sign warning the player about the dragon. This was to make sure that the player had ample warning not to enter the final room.



Figure 2 Room 5 buttons and objects

The code for handling room 5 is similar to that of earlier rooms.

```
to4_btn.onRelease = function() {  
    _parent.changeRoom(4);  
}  
  
to8_btn.onRelease = function() {  
    _parent.changeRoom(8);  
}  
  
flint_btn.onRelease = _parent.getFlint;  
flint_btn._visible = !(_parent.inv_flint);  
  
stop();
```

We need the variable for flint, which happens to use the already existing useFire for it's inventory action. We will add the following lines to the restartGame function so the game will recognize the flint.

```
inv_flint = false;  
invFlint_btn.onRelease = useFire;
```

Now the game knows about the existence of the flint. We now need to make sure the inventory button is only shown when the player has the flint. This is done by adding the following lines to updateInventory function.

```
invFlint_btn._visible = inv_Flint;
```

Finally, we write the get function.

```
function getFlint()  
{  
    trace("Picking up Flint");  
    inv_flint = true;  
    updateInventory();  
}
```

Room # 6

Room 6 is a side chamber and as that only has a single exit, which leads to the rope room (room 3). This room is vital to the game as it has moss growing on the wall. This moss can be used with the flint and steel to start a fire, which is required to get the sword. This room is easier to code because I felt it was acceptable to allow for an infinite supply of moss.



Figure 3 Room # 6

The moss is more complex than the objects in the game because it is a bitmap image. To create the moss, an image of moss is required. This image happens to be a jpg image. The imported into flash's library. A Jaggedly shaped circular shape is then created and the moss image is set to that shapes texture by using the fill settings. The default for bitmap textures is tiled, with the size being really small. Selecting the gradient adjustment tool allows you to adjust the size, orientation, and placement of the texture.



Figure 4 Room 6 buttons

The code to handle buttons in room 6.

```
to3_btn.onRelease = function() {  
    _parent.changeRoom(3);  
}  
moss_btn.onRelease = _parent.getMoss;  
stop();
```

As is usually the case, some modifications to the main movie are needed so the inventory system can recognize the new inventory item. First, some lines need to be added to the restartGame function.

```
inv_moss = false;  
invMoss_btn.onRelease = useMoss;
```

Next, the updateInventory function needs a couple of new lines.

```
invMoss_btn._visible = inv_Moss;
```

Finally the functions for getting and using the moss need to be written. The code for handling the moss is a placeholder for now.

```
function getMoss()  
{  
    trace("Pick up Moss");  
    inv_moss = true;  
    updateInventory();  
}  
  
function useMoss()  
{  
    showPopup("Use Moss", "There is no reason to use the moss here");  
}
```


Room # 7

Across the chasm is the sword that the player needs in order to win the game. The sword is frozen in a block of ice. The room is only reachable after the player has tied the rope to the rock and the hook and threw it across the chasm. This means that the only exit is a rope that crosses the chasm. The sword, being frozen in a block of ice, needs to be melted before the player can acquire it. This is done by having the player place moss around the block of ice and then start a fire using the flint and steel.

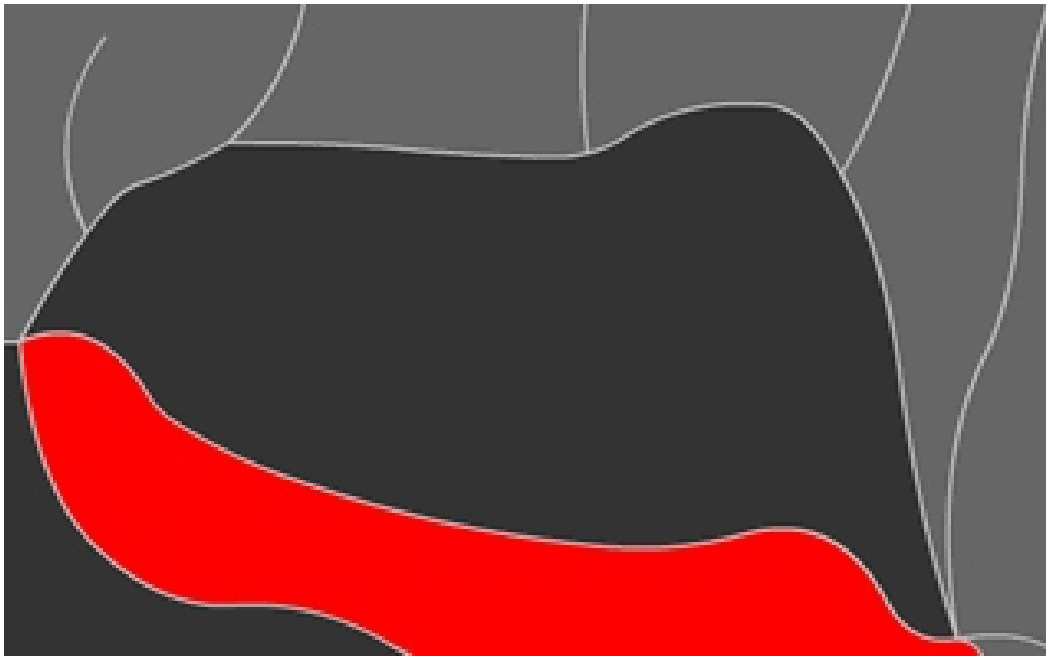


Figure 5 Room # 7

Button wise, there are only two buttons in this room. The first is the rope that returns the player to the main chamber. The second is the sword. The sword button is a bit tricky. First we create a stand alone sword (which when positioned sideways can be used in the inventory). Next we create a single frame movie with the sword in one layer, and the block of ice in another layer.

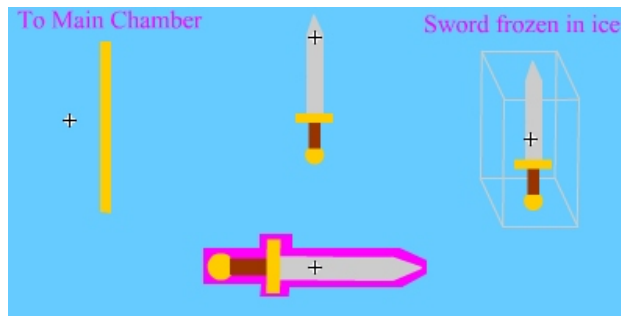


Figure 6 Rope and Sword

The code to activate these two buttons is similar to the code used in other rooms.

```
to4_btn.onRelease = function() {  
    _parent.changeRoom(4);  
}  
  
iceSword_btn.onRelease = _parent.getSword;  
iceSword_btn._visible = !(_parent.inv_sword);  
stop();
```

Likewise, some new variables for handling the inventory need to be added to the main movie. In the restartGame function, the following lines need to be added.

```
inv_sword = false;  
invSword_btn.onRelease = useSword;
```

The updateInventory needs to be updated as well.

```
invSword_btn._visible = inv_sword;
```

Finally, functions for getting the sword and using the sword are needed. In both cases they result in popup messages. For getting the sword, the player is told that they can't just grab the sword (because it is in a block of ice). The inventory use button tells the player there is nothing to use the sword on.

```
function getSword()  
{  
    showPopup("Get Sword", "The sword is in a block of frozen ice. You must  
first find a way to get rid of the ice!");  
}  
  
function useSword()  
{  
    showPopup("Use Sword", "Nothing to use the sword on");  
}
```

```
}
```

One final, optional, task is to take the sword in ice movie and add it to room 4. This way the player will be able to see the sword from room 4 and will know why they need to cross the chasm. Some code for hiding the sword when it has been taken is needed, as follows.

```
sword_movie._visible = !(_parent.inv_sword);
```

Starting a Fire

In order to get the sword, the player needs to start a fire. While we could have this done automatically if the player is in the right room and has all the items, I am requiring that the player first lay down the moss and then attempt to start the fire. This will require that the room movie has an image of moss around the ice block. This is done by creating a moss movie and adding it to the scene. Some code is going to be needed to make sure this is possible.

```
moss_movie._visible = _parent.mossOverIce;
```

Note, that a new variable is needed within the main movie. This requires adding the following line to the restartGame function.

```
mossOverIce = false;
```

As it is right now, the useMoss function doesn't do much. We need to replace the useMoss function with this enhanced version. It simply makes sure that if we use the moss in room 7 when the sword still has not been retrieved, that it will set the mossOverIce variable which will cause the moss image to appear around the block of ice.

```
function useMoss()
{
    if ( (currentRoom == 7) && (inv_sword == false) )
    {
        inv_moss = false;
        mossOverIce = true;
        showPopup("Use Moss", "you surround the block of ice with the
dried moss");
    }
    else
    {
        showPopup("Use Moss", "There is no reason to use the moss here");
    }
}
```

Now the player can place the moss around the ice. The useFire function is also going to have to be replaced. Now, when you think about it for a moment, the moss room has moss in it. You know that some player is going to try to start a fire in the moss room. So, in addition to un-thawing the sword, we need some action for when the player tries to light the moss room on fire.

Blazing Games Guide to Flash Game Development Chapter 26: Finishing the Dragon

```
function useFire()
{
    if ( (inv_metal == true) && (inv_flint == true) )
    {
        if (currentRoom == 6)
        {
            showPopup("Use Flint and Steel", "Starting a fire here would
serve no purpose!");
        }
        else if (currentRoom == 7)
        {
            showPopup("Use Flint and Steel", "Room 7 fire action not
implemented yet!");
            if (r7_mossOverIce == false)
            {
                showPopup("Use Flint and Steel", "Good idea, but there
is nothing to burn...");
            }
            else
            {
                r7_mossOverIce = false;
                inv_sword = true;
                showPopup("Use Flint and Steel", "You set the moss on
fire, which melts the ice and lets you get the sword");
            }
        }
        else
        {
            showPopup("Use Flint and Steel", "There is nothing in this
room to burn");
        }
    }
    else
    {
        showPopup("Use Flint and Steel", "You are missing something!");
    }
}
```

Room 8 Win Movie

Room 8 is the final room of the game. This is the location where the dragon is. As such, only two things can happen with this room. The player can win the game or they can lose the game. This is decided by one simple factor. Does the player have the sword? If they do they win. If they don't they lose. The code to handle this decision is in room 8's first frame and is simply a comparison statement.

```
if (_parent.inv_sword == true)
    gotoAndPlay("Win");
else
    gotoAndPlay("Lose");
```

Now all that needs to be done is the winning and losing animation. Originally, I had envisioned a battle sequence showing the player slaying the dragon. While this is doable, it would be a very time consuming animation to build. Considering my extremely short schedule, this was unrealistic. Instead, I figured I would assume the dragon was smart enough to avoid a losing battle, so instead the game wins with the player finding a heap of treasure and a note from the dragon telling the player that it is leaving town.



Figure 7 Winning Props

The animation, then, starts with the heap of treasure with a note on top. The note zooms in until it fills the screen. At that point the ok button appears and allows the player to end the game. The button handling code is placed on the final frame of the animation sequence.

```
win_btn.onRelease = function() {  
    _parent.gameOver();  
}  
stop();
```

Of course, the main movie needs the gameOver method, which simply calls the title screen. The title screen will be created later in this chapter.

```
function gameOver()  
{  
    gotoAndPlay("Title", 1);  
}
```

Room 8 Lose Movie

Okay, we have the winning movie finished, but we need something for the losing movie. This means that we are going to have to bite the bullet and create a dragon. Thankfully, we don't really need to animate the dragon, we just need the dragon in a pose that suggests it is shooting a fireball at the player. My dragon may not be the greatest looking dragon, but like I said, this game was created very quickly. More to the point, I am a programmer not an artist.

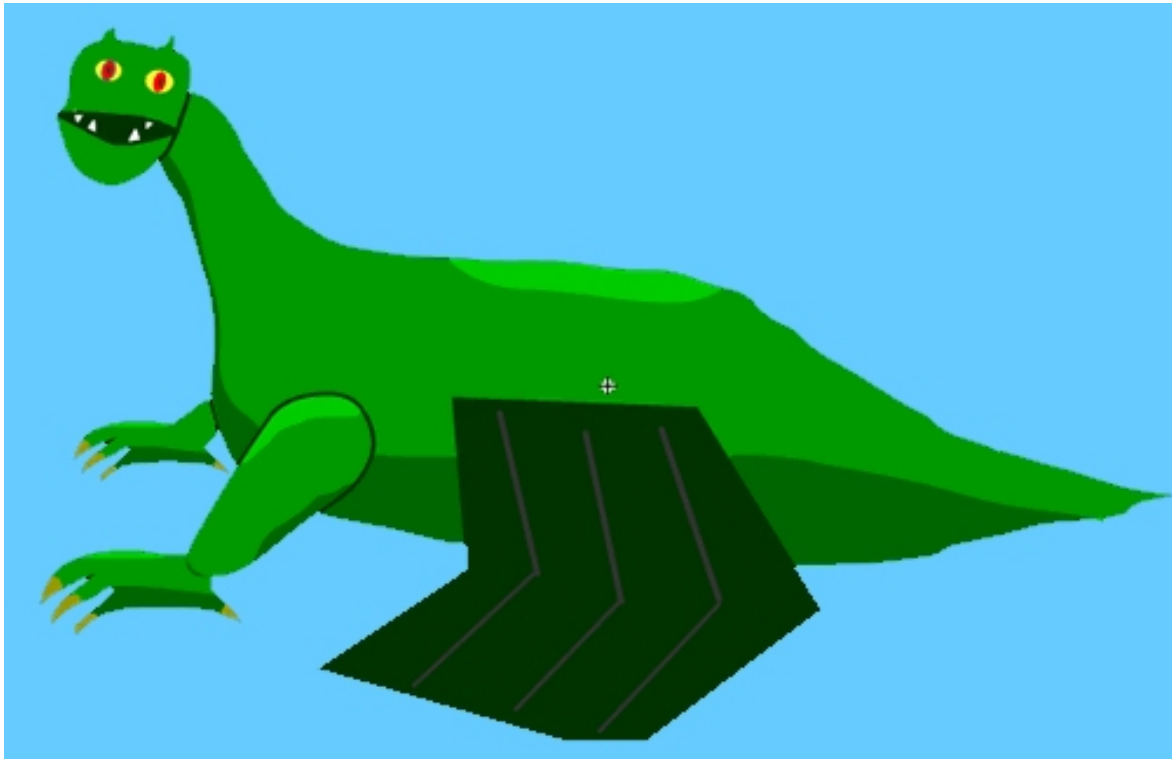


Figure 8 The Dragon

We then need a flame object. This can be created by using a rough polygon shape filled with flame like gradient colors.

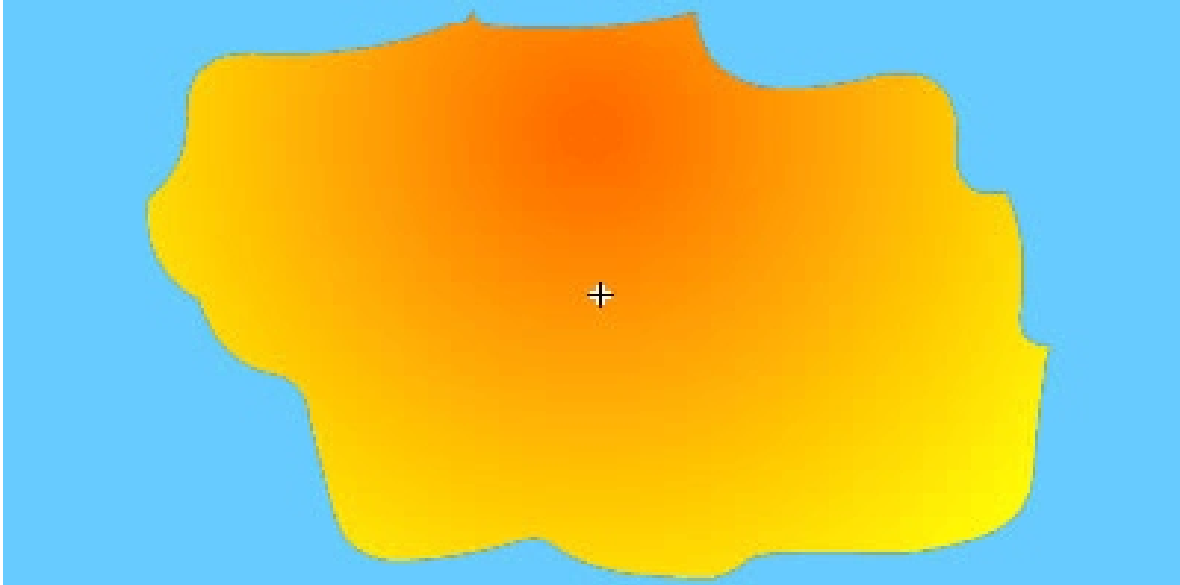


Figure 9 The fireball

The flame object expands from the dragon's mouth until it fills the screen. At which point a lose message is displayed. The lose button then waits for the player to click on it to end the game. The button management code is similar to that of the win movie.

```
lose_btn.onRelease = function()
{
    _parent.gameOver();
}
stop();
```


Title Screen

We now have a fully playable game, so it is time to finish things by creating the title screen. I wanted to have some type of animated title sequence, featuring the dragon breathing the logo at the player. That didn't quite work, so instead, I had the dragon shoot a fireball which fades into the title sequence. Finally, the start game button begins.

To build this sequence, we start with the background layer. We draw a mountain range using chunks of grey. The center part of this range we create a cave entrance in. The top part of the screen we color sky blue. The bottom part we color green. Finally we add the dragon.

Next we place the title layer. We want the title to appear around frame 40, and to be in the sky where it is visible. We then create the flame layer. The flame will start from the dragon's mouth in frame 10 and expand to cover the title in frame 40. From frame 40 to 60 it will fade out, exposing the title.

Finally, we add the start button. The code to handle this is very simple.

```
start_btn.onRelease = function()
{
    gotoAndPlay("Game", 1);
}

stop();
```