

Scoring & ending the game

This fourth tutorial shows you how add scoring into the game and how to end the game when Jump Man's health is depleted.

Open the previous file and select the *info* layer.

Using the Text tool, draw another text box near the top left corner of the stage. This text box will display the score. Check the Properties Panel and ensure the text box is set to Dynamic (there's a menu with options of Static, Input and Dynamic) and make sure the text is set to a colour that will show up against the background. Also give the text box an instance name of *score_txt*.

Now select the *code* layer and open the Actions Panel (F9).

Move to the start of the code and insert the following code at the start

```
var score=0;

score_txt.text=score.toString();
```

These two lines set up the score and displays it on the screen. The first line creates a variable, which remembers the current score. At the start of the game the score is set to zero. The next line displays the score in the text box on the screen.

Now move down to the gameloop function and add the highlighted code below.

```
function gameloop(e:Event):void{

    crate_mc.x-=20;

    if (crate_mc.x<-100){

        crate_mc.x=650;



```
 score++;

 score_txt.text=score.toString();
```



    }

    if (jumpman_mc.hitTestObject(crate_mc)) {

        health--;

        health_txt.text=health.toString();

    }

}
```

This additional code adds one to the score (score++) every time the crate reaches the left hand side of the screen.

Finally, we need to end the game when Jump Man's health reaches zero.

We'll create a new screen which will display a 'Game Over' message and the final score, then add the code that will end the game.

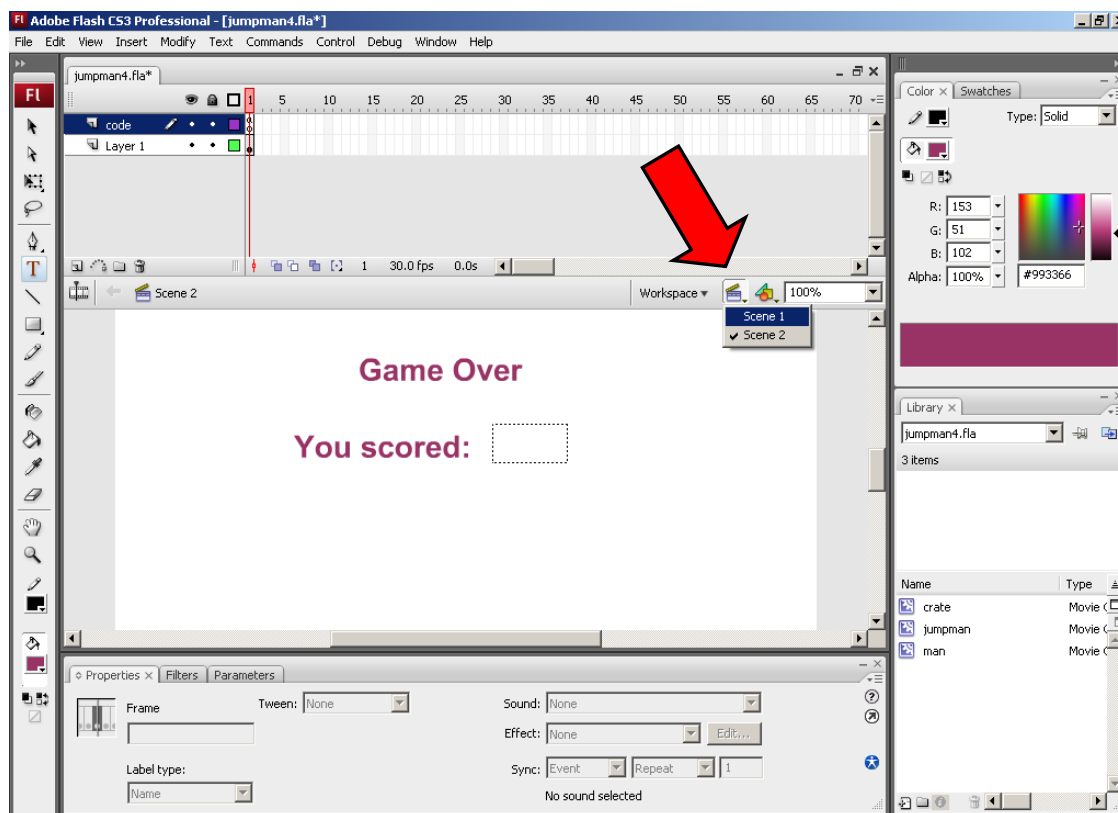
Close the Actions Panel and from the main menu choose *Insert – Scene*. This opens up a new blank scene and a new blank timeline. Don't worry –you haven't deleted anything!

Select the text tool and draw a text box in the middle of the stage containing the words 'Game Over'. Make sure this is a Static Text box (set in the Properties Panel). Draw another static text box beneath the previous one containing the words 'You scored:'. Draw a third text box to the right of the previous one, but this time set it to Dynamic. This will display the score. Set the instance name of this final text box to *score_txt*.

Add a new layer and name it *code*. Open the Actions Panel (F9) and add the following line of code that will display the score in the text box:

```
score_txt.text=score.toString();
```

Close the Actions Panel then go back to Scene 1 (select the clapperboard icon above the workspace to choose the scene you want to view).



Select the *code* layer in Scene 1 and open the Actions Panel (F9).

Insert a new line at line 1 and add the following code that will prevent the Flash movie from looping between scenes 1 and 2.

```
stop();
```

Next you need to add some more code to make the game end when Jump Man's health reaches zero.

Locate the *gameloop* function and the part that decreases the health.

Add the block of code highlighted below.

```
if (jumpman_mc.hitTestObject(crate_mc)) {  
    health--; //decrease health  
    health_txt.text=health.toString();
```

```
    if (health<=0){  
        stage.removeEventListener(Event.ENTER_FRAME, gameloop);  
        stage.removeEventListener(KeyboardEvent.KEY_DOWN, jump);  
        gotoAndStop(1, "Scene 2");  
    }
```

```
}
```

This new block of code checks to see if Jump Man's health has reached zero, or is less than zero.

```
if (health<=0){
```

If the health is depleted it displays Frame 1 of Scene 2, which contains the Game Over message.

```
gotoAndStop(1, "Scene 2");
```

The other two lines of code prevent a whole bunch of errors appearing when the game ends. It is really important that when you add event listeners that you remove them when you are finished with them.

```
stage.removeEventListener(Event.ENTER_FRAME, gameloop);
```

```
stage.removeEventListener(KeyboardEvent.KEY_DOWN, jump);
```

The game should now be complete, so go ahead and test it. *Ctrl-Enter* or go to the menu bar and choose *Control – Test movie*.

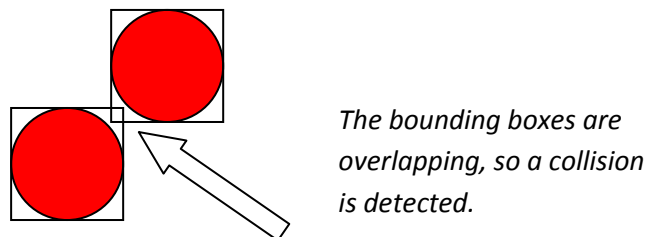
Bugs and glitches

Very rarely do games have no bugs or glitches. You probably know of many games that you've purchased that have bugs or glitches – odd things that happen from time to time such as characters who walk through walls or mysteriously appear in mid air. You should always test your games thoroughly to try and discover unexpected glitches.

Jump Man also has a glitch.

One problem you will have is with the collision detection. We are using basic collision detection using the *hitTestObject* method. This works by detecting collisions between the bounding boxes around objects. This means that collisions aren't very accurate for non-rectangular shapes.

Imagine two circular objects colliding. Each object has a rectangular bounding box around it. *hitTestObject* detects collisions between the bounding boxes, NOT the shapes. The diagram below shows a collision between bounding boxes, which makes *hitTestObject* a little bit inaccurate (but good enough for simple games).



Exercises

Add some code to scene 2 so that it allows you to replay the game if you press a key on the keyboard.

Think about how you could modify this game so that it isn't just about a skateboarder jumping over crates. What graphics could you use in the game to change the look and feel?

What other features could you add to this game to make it better?

Here's one idea for an alternative version of the game:



Here is a complete code listing for scene 1:

```
stop();

var score=0;

score_txt.text=score.toString();

var health=20;

health_txt.text=health.toString();

stage.addEventListener(Event.ENTER_FRAME, gameloop);

function gameloop(e:Event):void{

    crate_mc.x-=20;

    if (crate_mc.x<-100){

        crate_mc.x=650;

        score++;

        score_txt.text=score.toString();

    }

    if (jumpman_mc.hitTestObject(crate_mc)) {

        health--;//decrease health

        health_txt.text=health.toString();

        if (health<=0){

            stage.removeEventListener(Event.ENTER_FRAME, gameloop);

            stage.removeEventListener(KeyboardEvent.KEY_DOWN, jump);

            gotoAndStop(1, "Scene 2");

        }

    }

}

stage.addEventListener(KeyboardEvent.KEY_DOWN, jump);

function jump(e:KeyboardEvent):void {

    jumpman_mc.play();

}
```