

# PROGRAMMING ASSIGNMENT: FALLING ROCKS PROGRAM

For this assignment you are going to create a game called **Falling Rocks**. The object of the game is to avoid rocks that are randomly falling from the sky. In order to avoid the rocks, you will need to control the penguin located at the bottom of the screen with the keyboard. There are only two keys that you are going to use to direct the penguin – the left arrow key and the right arrow key. For every rock the penguin avoids, 10 points is awarded. The game keeps going until a rock hits the penguin, at which point you will need to ask the user if they want to play again.



## PROGRAM REQUIREMENTS:

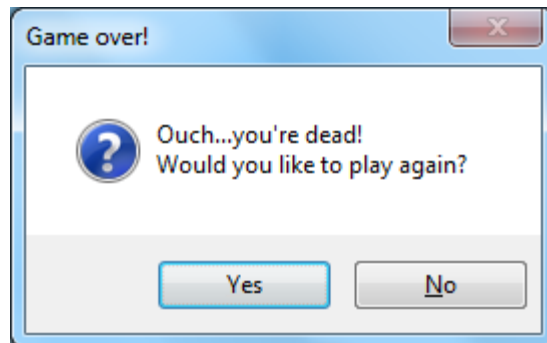
- You will need an array of **PictureBox** objects that will store images of the rocks. You shouldn't use more than 10 rocks, otherwise the game will lag. You will need a for loop to fill the array with images and to set the x- and y-position of each rock.
- You will need a **Timer** that will be responsible for moving the rocks a certain number of pixels down the screen. You will need a for loop to increase the y-position of each rock.
- You will need a method that checks if any of the rocks have hit the penguin. This method should be called every time a rock moves and every time the penguin moves. You will need a for loop to check if any of the rocks have hit the penguin.

Your method should look something like this:

```
Private Sub checkCollision()  
  
    If rock.Bounds.Intersects(picPenguin.Bounds) Then  
  
        End If  
  
End Sub
```

- You should randomly set the x- and y-position of the rocks so that they are not always falling from the same place and they are not overlapping with other rocks. Each rock's initial position should begin off the screen (which means the y-position should be somewhere off the screen).
- When the rocks have fallen and reach the bottom of the screen, they should be reset. When resetting the rock's position, you should make sure the x-position and y-position is randomly generated and not overlapping with any of the other rocks.
- When the penguin is hit by a rock, the image of the penguin should change (use the dead penguin picture) and output a message box indicating that the game is over and asking users if they would like to play again. If the user clicks YES, the game should restart. If the user clicks NO, the game should end.

Your program output should look something like this:



All the images have been provided for you in the **Falling Rocks** folder that I have saved in your UNIT 5 folders.

Once you have completed the program, save the program in your COMPLETED ASSIGNMENTS folder.

# FALLING ROCKS PROGRAM RUBRIC

NAME: \_\_\_\_\_

TOTAL: / 40

CATEGORY	CRITERIA	< LEVEL 1 0 – 49%	LEVEL 1 50 – 59%	LEVEL 2 60 – 69%	LEVEL 3 70 – 79%	LEVEL 4 80 – 100%	MARK
<b>Knowledge and Understanding</b>	Demonstrates an understanding of how to write a program(s) that uses animation and that handles keyboard events	<ul style="list-style-type: none"> <li>Demonstrates little or no understanding of how to use animation techniques and keyboard events</li> </ul> <p style="text-align: center;"><b>0-4.9</b></p>	<ul style="list-style-type: none"> <li>Demonstrates limited understanding of how to use animation techniques and keyboard events</li> </ul> <p style="text-align: center;"><b>5.0-5.9</b></p>	<ul style="list-style-type: none"> <li>Demonstrates some understanding of how to use animation techniques and keyboard events</li> </ul> <p style="text-align: center;"><b>6.0-6.9</b></p>	<ul style="list-style-type: none"> <li>Demonstrates considerable understanding of how to use animation techniques and keyboard events</li> </ul> <p style="text-align: center;"><b>7.0-7.9</b></p>	<ul style="list-style-type: none"> <li>Demonstrates thorough understanding of how to use animation techniques and keyboard events</li> </ul> <p style="text-align: center;"><b>8.0-10</b></p>	<b>/10</b>
<b>Thinking</b>	<p>The program meets all the required specifications</p> <p>Validates program to ensure the program produces correct results</p>	<ul style="list-style-type: none"> <li>Program meets little or none of the required specifications</li> <li>Validates program with little or no success</li> </ul> <p style="text-align: center;"><b>0-4.9</b></p>	<ul style="list-style-type: none"> <li>Program meets a limited number of the required specifications</li> <li>Validates program with limited success</li> </ul> <p style="text-align: center;"><b>5.0-5.9</b></p>	<ul style="list-style-type: none"> <li>Program meets some of the required specifications</li> <li>Validates program with some success</li> </ul> <p style="text-align: center;"><b>6.0-6.9</b></p>	<ul style="list-style-type: none"> <li>Program meets most of the required specifications</li> <li>Validates program with considerable success</li> </ul> <p style="text-align: center;"><b>7.0-7.9</b></p>	<ul style="list-style-type: none"> <li>Program meets all of the required specifications</li> <li>Validates program with great success</li> </ul> <p style="text-align: center;"><b>8.0-10</b></p>	<b>/10</b>
<b>Communication</b>	Provides internal documentation that clearly explains program logic	<ul style="list-style-type: none"> <li>Documents program logic with little or no success</li> </ul> <p style="text-align: center;"><b>0-4.9</b></p>	<ul style="list-style-type: none"> <li>Documents program logic with limited success</li> </ul> <p style="text-align: center;"><b>5.0-5.9</b></p>	<ul style="list-style-type: none"> <li>Documents program logic with some success</li> </ul> <p style="text-align: center;"><b>6.0-6.9</b></p>	<ul style="list-style-type: none"> <li>Documents program logic with considerable success</li> </ul> <p style="text-align: center;"><b>7.0-7.9</b></p>	<ul style="list-style-type: none"> <li>Documents program logic with great success</li> </ul> <p style="text-align: center;"><b>8.0-10</b></p>	<b>/10</b>
<b>Application</b>	Effectively applies programming skills and knowledge of Visual Basic to create a program	<ul style="list-style-type: none"> <li>Applies programming knowledge and skills with little or no success</li> </ul> <p style="text-align: center;"><b>0-4.9</b></p>	<ul style="list-style-type: none"> <li>Applies programming knowledge and skills with limited success</li> </ul> <p style="text-align: center;"><b>5.0-5.9</b></p>	<ul style="list-style-type: none"> <li>Applies programming knowledge and skills with some success</li> </ul> <p style="text-align: center;"><b>6.0-6.9</b></p>	<ul style="list-style-type: none"> <li>Applies programming knowledge and skills with considerable success</li> </ul> <p style="text-align: center;"><b>7.0-7.9</b></p>	<ul style="list-style-type: none"> <li>Applies programming knowledge and skills with great success</li> </ul> <p style="text-align: center;"><b>8.0-10</b></p>	<b>/10</b>

**CURRICULUM EXPECTATIONS THAT ARE COVERED IN THIS ASSIGNMENT:**

- B1. Describe fundamental programming concepts and constructs.
- B2. Plan and write simple programs using fundamental programming concepts.
- B3. Apply basic code maintenance techniques when writing programs.