

Mr. Giansante



Visual Basic

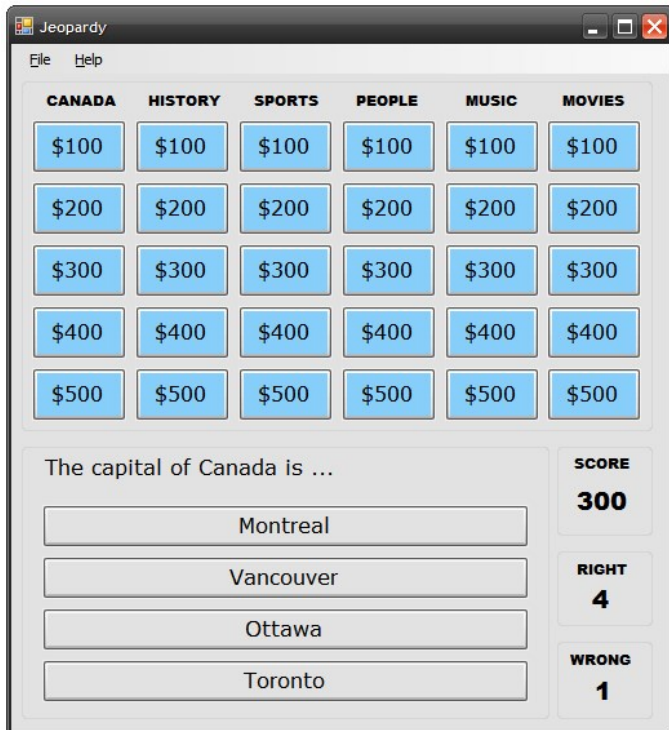
Jeopardy

August 2016

Jeopardy

Learning Objectives:

Creating procedures that are independent of a control (ie. showanswers, hideanswers).



Design a Visual Basic program that simulates a Jeopardy-like Trivia Game. The user should be presented with questions and provided with multiple-choice answers.

You should organize the game board into 6 categories with 5 questions in each category. To accomplish this: use Label controls for the category names and use Button Controls for the questions. The first row of buttons in each category should have their Text property set to \$100, the second \$200, etc.

In addition, include a large Label control where questions will be displayed and four long Button controls for the possible answers. When the user presses one of the buttons, a messagebox should appear saying either: "That is Right !" or "Sorry - that is Wrong !"

Your game should keep track (on the screen) of the question number and the number of questions the user has answered correctly. When the player has finished all the questions, they should be presented with a messagebox telling them how they did. ie. "You did Great - you answered 24 of the 30 questions."

Note: Place the sound files in your program only after everything else is done.

Step by Step

Draw a **GroupBox** control.

Erase the **Text** property of GroupBox1.

Draw 30 Button controls (6 categories, 5 questions in each) inside the GroupBox control.

Outside the GroupBox control ...

Draw a Label control

Draw four Button controls

Set the **Visible** property of the four Button controls to False

In General | Declarations, type the following ...

```
Dim Answer As Integer
Dim Value As Integer
Dim Score As Integer
Dim Correct As Integer
Dim Incorrect As Integer
```

Jeopardy

Note: Don't forget to set initial values for all the variables (possibly in Form | Load). For example ...

```
Score = 0
```

You must now create two procedures which will greatly reduce the number of lines of code you need to type. These procedures are ShowAnswers and HideAnswers.

To create the ShowAnswers procedure, type the following in General | Declarations ...

```
Sub ShowAnswers
```

When you press [ENTER], you will be in General | ShowAnswers. Type the following ...

```
GroupBox1.Enabled = False  
Button31.Visible = True  
Button32.Visible = True  
Button33.Visible = True  
Button34.Visible = True
```

Note: Replace the CommandButtons in the example above with the ones which will contain your answers (ie. you won't necessarily be using Command26).

Return to General | Declarations and type the following ...

```
Sub HideAnswers
```

When you press [ENTER], you will be in General | HideAnswers. Type the following ...

```
GroupBox1.Enabled = True  
Label1.Text = ""  
Button31.Visible = False  
Button32.Visible = False  
Button33.Visible = False  
Button34.Visible = False
```

Now, whenever you want to Call one of these procedures, you simply need to type its name.

In the first Button (ie. the first question), type the following ...

```
' Hide button which was just pressed so  
' that the user can't pick it again later  
  
Button1.Visible = False  
  
Label1.Text = "The capital of Canada is ..."  
  
Button31.Text = "Montreal"  
Button32.Text = "Vancouver"  
Button33.Text = "Ottawa"  
Button34.Text = "Toronto"  
  
Answer = 3  
Value = 100  
  
ShowAnswers
```

Jeopardy

You will have to repeat the above for each of the 30 questions.

The code which goes in the first **answer** Button is ...

```
HideAnswers
If Answer = 1 Then
    response = MessageBox.Show("That is Correct !", "Answer")
    Score = Score + Value
    Correct = Correct + 1
Else
    response = MessageBox.Show("That is Wrong !", "Answer")
    Score = Score - Value
    Incorrect = Incorrect + 1
End If
'Don't let score go below $0
If Score < 0 Then Score = 0
'Display score
Label2.Text = Str(Score)
' If all questions have been answered the game is over
If Correct + Incorrect = 30 then
    response = MessageBox.Show("You got "+Str(correct)+" questions right !",
        "Game Over", MessageBoxButtons.OK)
End If
```

Note: Put this all on ONE

line.

The code which goes in the second answer button is identical, except for the second line, which should read ...

```
If Answer = 2 Then
```

Jeopardy

Please fill out all the information in this column using a pen before getting this assignment marked.

Name

Date

Class

Academic Honesty

The work I am submitting is completely my own creation and has not been copied from anyone else's work. If I have received help on this project, the names of those who have assisted are listed below.

Signature

Pre-Marking

The following people have pre-marked this assignment: (minimum of two)

Deductions

Each which is checked indicates the item was not satisfactory and results in the loss of one mark.

Deduction for late assignment is 1 mark per day.

Deduction for program crashing is 2 marks.

User Interface

- Appropriate Text in Title Bar (Form.Text)
- Separators and Access Keys in Menu, Capitalization
- "Exit" DialogBox meets requirements
- "About" DialogBox meets requirements
- Efficient and Esthetically-Pleasing User Interface
- Appropriate controls are used
- Spelling and Grammar are correct

Code / Programming Style

- Variables are declared and logically named
- Code is commented where appropriate
- Code is indented and spaced to show
- Code is efficient

Work Ethic / Problem Solving Skills

- Problem Solving Skills are demonstrated
- Makes Productive Use of Time
- Only Seeks Help when Necessary

Program-Specific Criteria

- "New Game" button resets game properly
- Cannot choose next question until current question has been answered
- Cannot click on answers more than once
- Score is kept correctly
- End of game has some kind of message
- Incorrect + Correct = 30 at end of game

Comments

Mark _____ / 10