

Game Programming

The Four Games – Question Set 01

Dice Roller

1. A student's code has the line *money=100* placed as the first line of `cmdRoll_Click()` instead of `cmdStartGame_Click()`. Will this affect the game? Explain.

Every time the user clicks to roll, their money will be set back to \$100 dollars and the money won't be able to change like it should. Bad.

2. Assume the player currently has 50 dollars. They have pressed the roll button and they received a pair of 4's. How much money will be displayed on the screen?

They will lose 5 dollars at the top of the roll code (brings them to 45) but they win 30 dollars to bring them to 75 dollars.

3. The programmer wants to give 30 dollars for a pair of 1,2,3,4,5's, but 50 dollars for a pair of 6's. How would you make this change without erasing any code that is already there?

Since you are giving extra money for the sixes, you could add another `if` statement just after the existing code that asks *if dieOne = dieTwo and dieOne=6 then ... give more money*

4. How would you implement recording the number of rolls that the user has made since the program starts?

Create a variable *Dim rollCount as Integer*

Create a textbox called *txtRoll*

Set this variable to 0 when the user clicks Start Game

Every time the user hits the roll button, *rollCount = rollCount + 1 : txtRoll = rollCount*

Button Masher

1. What does it mean to *declare a variable*? Give an example from the game.
Declaring variables is creating a variable with a line like *Dim x as Integer*.
It can be done at the very top of the program or inside of public subs.
In Button Masher, we did *Dim timeleft as Integer*

2. The variables for the game are declared at the top of the program. Why didn't the student declare the *score* variable inside of the `cmdPress_Click` method instead?

When you declare a variable at the top of the program, it can be used *anywhere* in the program. This is called a GLOBAL declaration. The scope of the variable is global.

When you declare a variable inside a sub, the variable only exists inside the sub. It's scope is called LOCAL. When the sub ends, the variable is destroyed.

Since you want *score* to live on for the entire life of the game, declare it at the top of the program.

3. A student wants the program to keep track of TWO high scores. They add a second variable called `bestscore2` (the second highest best score) and replaces the current high score code with the following code. Does it work? Explain.

```
If score > bestscore Then
    bestscore = score: txtBestScore = bestscore
    bestscore2 = bestscore: txtBestScore2 = bestscore2
End If
If score > bestscore2 Then
    bestscore2 = score: txtBestScore2 = bestscore2
End If
```

NO, this code does not work. This code will copy the best score into both variables (`bestscore` and `bestscore2`) and the same score will be shown in both textboxes.

The order you set the variables is important! This code works better:

```
If score > bestScore then
    bestScore2 = bestScore : txtBestScore2 = bestScore2
    bestScore = score : txtBestScore = bestScore
else if score > bestScore2 then
    bestScore2 = score : txtBestScore2 = bestScore2
end if
```

4. You want to move the hit button to a new horizontal location (between 1000 and 9000) each time it is pressed. Describe the code you would add. (*hint: .top or .left position objects*)

In the `PressClick` event you could add a line like:

```
cmdPress.left = int(rnd*8000) + 1000
```

This would left the x position of the button to a new random location between 1000 and 9000. Notice how this works: the RANGE is 8000 and the minimum value is 1000.

Guess Game

1. A student changes the checking code inside of `cmdGuess_Click()` to read

```
If Val(txtGuess)>mystery Then
    txtResult = "Too Low!"
ElseIf Val(txtGuess)<mystery Then
    txtResult = "Too High!"
Else
    txtResult = "Got It!"
End If
```

Will this code work as intended? Explain.

No, the `<` and `>` signs are backwards.

Would there be any 'good coding' suggestions that you would offer the student?

Consider writing it like this (You should do math comparisons with textboxes)
`Dim guess as Integer: guess = Val(txtGuess): if guess < mystery then ...`

2. A student wants to change the game so that a player can enter the maximum value of the mystery number to increase difficulty. For example: if the user types 500 then the random number will be from 1-500. How would you implement this?

Declare a variable: *Dim max as Integer*
Create a textbox: *txtMax*

When the user hits the start button, find out what max is set to and then find a random number within the proper range 1 to max.

```
max = Val(txtMax)
mystery = int(rnd*max) + 1
```

Horse Race

1. What is the role of the *userpick* variable?

We use *userpick* to remember which horse the user has picked so that later in the program we can check to see if they picked the correct winning horse.

A student places the following code inside of the *Timer1_Timer()* method:

```
Dim r as Integer
r = Int(rnd*300) + 50
position1 = position1 + r : Image1.Left = position1
position2 = position2 + r : Image2.Left = position2
position3 = position3 + r : Image3.Left = position3
```

Why do you think the student `+50` to the random number rather than use `r=Int(rnd*300)` ?

They want the horse to move AT LEAST 50 units. This would give a range of 50-349.

Does this segment of code work as intended? Explain.

The code makes all three horses move the same amount each timer tick (they will tie every time!). Since only one random number is found and added to all three horses, all three horses move ahead by the same random number. Each horse should have a new random number found to make it move.

2. Consider the *cmdReset_Click()* method. Instead of using the lines `position1 = 120` and `Image1.left = position1`, a student only uses the line `Image1.left = 120`. What affect will this have on the program? Explain.

You will see the horse visually move back to the start line, but the `position1` variable is still left at some other value. When the next race starts and the horses start to move again, the line `Image1.left = position1` will make the horse jump all the way to the value of `position1` which is where horse1 was at the end of the last race. This is bad. The programmer should reset all the variables, not just move the images.

3. A student coded their *checkForWin()* method to read as follows:

(assume that Win and Lose have been coded properly)

```
If position1 > 7500 Or position2 > 7500 Or position3 > 7500 Then
  Timer1.Enabled = False
  cmdReset.Enabled = True
End If
```

```
If position3 > 7500 and userpick = 3 Then
  Call Win()
ElseIf position2 > 7500 and userpick = 2 Then
  Call Win()
ElseIf position1 > 7500 and userpick = 1 Then
  Call Win()
Else
  Call Lose()
End If
```

Do you think it will work as intended for all outcomes of the race? Explain.

It work OK, but not the perfect. If any horse crosses the finish line it will detect and end the race. The problem is when two or three of the horses cross the finish line during the same timer tick (a close race). If horse one and horse three cross at the same time with HORSE ONE SLIGHTLY AHEAD, the above code will award the WIN TO HORSE THREE since it is the first horse checked in the if/elseif chain. A good beginner attempt but not great for a professional game!

4. Pretend the game is running and that horse 1, horse 2, and horse 3 are close together and all cross the line at the same time (but slightly different positions, ex. 7503, 7505,7501). Who will win the race according to the original program code?

Because of the order of the if/elseif chain, HORSE THREE will be given the win.

5. How would you change the original code so that it will tell you the correct winner even when two or more horses cross the finish line at the same time?
(the horse with the largest distance should win!)

Change the if statement to something like this:

```
If position1 >= position2 and position1>=position3 and userpick = 1 then
  `horse one wins
ElseIf position2>= position1 and position2>=position3 and userpick = 2 then
  `horse two wins
Elseif position3>=position1 and position3>=position1 and userpick = 3 then
  `horse three wins
End if
```

A small flaw with this new code is when two or more of the horses TIE at exactly the same position. What is the preference to two horses winning a tie?