

For this project you will design and implement a slightly more complicated version of the simple children's game called *gold button*. This game is played on a line of squares each of which may contain a blue button or a gold button (or nothing). The object of the game is move the gold button to the far left end so that it can be taken. The winner is the player who takes the gold button subject to the following rules:

1. There are two players who take turns.
2. For each turn a player must either take the button on the far left or move some other button some number of squares to the left.
3. Players cannot jump other buttons when moving a single button to the left and neither can two buttons occupy the same square.
4. Play continues until someone takes the gold button which can only be taken when it is on the far left.



Figure 1

A typical game set up with eight buttons.

For the EE 356/CS 376 version of the game the computer will be one player, the buttons will be animated, the button path may be curved, and the number of squares will be determined by the user (minimum 16 and maximum 48). The user should also be able to determine the total number of buttons which should be distributed randomly, and the gold button must appear in the right most 25% of squares.

The player will move a button by left clicking the mouse on a button and right clicking on an empty space. If the move is legal animation will be used to move the button between squares. If the move is illegal a warning message should be given and the user should try again.

Your game should announce a winner with some form of animation.

Minimum Requirements:

1. Your project must use C# WPF.
2. The gold button must be gold in color. Other buttons can be any color *except* purple or gold.
3. The minimum number of squares is 16 and the maximum is 48. The user should be able to determine the number of squares in this range.
4. The user should determine the number of buttons which can range from 1 to one less than the number of squares. This includes the gold button in the count.
5. Button movement must be animated.
6. Warning messages must be used for illegal moves.
7. The winner must be announced with animation.
8. You need not draw squares on the screen but you will need something to mark positions.

9. The path should not be a straight line.

Extras:

1. Include an *About* box and a *Help* button to provide directions.
2. Allow the user to choose from multiple path shapes.
3. Adjust your game to occupy the full screen. That is, programmatically determine the size of the screen the game is running on and adjust the game size accordingly.
4. Make user inputs grayed out or invisible when they are illegal to use. For example, once the user chooses the number of squares, she can't do that again until the game restarts so remove the option from the screen.
5. Add sound to your game but keep this from being annoying. Allow the user to turn it off.
6. Add a menu to save a game so it can be restarted.
7. Make the computer a *smart* player instead of randomly choosing the next valid move.

Turn in the following electronically to your instructor

1. A Word document titled EE356P4XXX.docx (where xxx is your three initials) which contains:
 - Cover Sheet – The cover sheet should include your name, course number, project number, project title, and date handed in.
 - A short description of your project. Include an overview and any special features that you added that are not in the specifications.
2. A complete code file that can be executed. Your code file must have well documented source code.

Compress these two items into a zip file should have the title EE356P4XXX.zip where XXX are your three initials and email them to your instructor.