Room planner

Your task is to create simple room planner. The task consists of two parts: laboratory and home (12 points each). In all doubtful or untold aspects application should behave like the example app (except possible bugs).

Lab part

- Example app
- Main window:
  - Starts in the center of the screen
  - Minimum window size: 400x300
  - When the window is resized, it behaves like in the example app.
  - Divided into two panels. You can change panels' widths using splitter (vertical line between panels):
    - Right panel: contains two group boxes: "Add furniture" and "Created furniture"
    - Left panel: contains bitmap. Left panel scales automatically when the main window is resized, but the bitmap should stay in the fixed size.
  - Menu: with "New blueprint" button.
- New blueprint
  - Creates new bitmap with the size of the left panel (it should take all empty space)
  - Shortcut: F2
- Add furniture groupbox
  - Contains four buttons.
  - Buttons automatically change their position while resizing or moving the splitter.
  - The scrollbar should be visible when needed
- Buttons
  - Buttons size: 75x75px. Each button uses images from project's resources (images cannot be loaded from an external location).
  - Button selection - After clicking the button it changes background color. After clicking the button again, the background color changes back to white. When one button is selected and we click another button, the first one should be deselected and the new one should be selected.
  - Mouse should change icon while hovering the buttons
- Created furniture groupbox
  - Contains a list of created furniture. Each row should contain furniture's name ant it's position on the bitmap.
- Adding new furniture
  - When a button is selected and the user clicks on the bitmap, the selected furniture should be drawn. The clicked point should be the center of the furniture (not it's upper-left corner). After adding new furniture the button is deselected.
  - The list in created furniture groupbox should be updated
- Hints:
  - SplitContainer, TableLayoutPanel, FlowLayoutPanel
  - RowSpan/ColumnSpan Property, Dock Property, Tag property, PictureBox, GroupBox, SplitterDistance
  - MouseDown Event, Graphics methods, PictureBox.Refresh, WindowsStartLocation
- *Note: In all doubtful or untold aspects application should behave like the example app (except possible bugs)*
  - Scoring:
    - "Main window" section: **2 points**
    - "New blueprint" section: **2 points**
    - "Add furniture groupbox" section: **2 points**
    - "Buttons" section: **2 points**
    - "Adding new furniture" and "Created furniture groupbox" sections: **4 points**
    - *Note: It is not possible to obtain points for incomplete functionality*

Home part

Note: To pass the home part, all the features from the lab part must work.

- Example app
- Bitmap resizing
  - When the main window of the application is increased, the bitmap should also change it's size
  - When the main window of the application is decreased, the bitmap should preserve it's size and the scroll bars should become visible.
  - The scrollbars should ignore mouse wheel event
- Drawing walls
  - The user can draw wall segments after selecting appropriate button in "Add Elements" groupBox
  - After clicking with left mouse button a new segment is added to the wall.
  - Wall segments should have "smooth" connections - any holes and empty spaces are unacceptable
  - Listbox should display the position of the first point added to wall segment
  - After clicking right mouse button wall creation is stopped. Another way to stop building the wall is deselecting the wall button
- Selecting elements
  - The user can select an element by clicking with left mouse button on it.
  - ListBox's selection should be updated
  - When an element is selected it is drawn with alpha set to 50% (it's semi-transparent)
- Moving elements
  - The user can move an element by clicking on it with left mouse button and moving the cursor (with left mouse button still pressed).
  - ListBox must update furniture's actual position
  - When an element is moving it is also selected
- Deleting elements
  - The user can delete an element by selecting it and clicking Delete on the keyboard
  - ListBox must be updated
- Rotating elements
  - The application must support rotating the elements (furniture and wall segments)
  - Furniture rotates around its centers and wall segments around the first point added to segment)
  - To rotate an element, the user has to select the element, and scroll using mouse wheel.
- Open and Save
  - The application must support saving and loading created blueprints
  - After clicking "Save blueprint" from the menu, the appropriate dialog box opens. It allows to enter a name of the file, but it should force you to save it with the extension (you can define any extension, e.g. *.bp)
  - All necessary information about the blueprint should be saved. After loading a file, the bitmap (and listBox) should look the same as in the moment of saving.
  - After clicking "Open blueprint" from the menu, the appropriate dialog box opens. It should force you to open files only with your defined extension (e.g. *.bp)
  - The user must be informed about the result of open or save operations.
- Localization
  - The application should support two languages (English and any other).
  - The application's default language is English
  - Language can be changed using Menu (File->Language). Each text in the application should be changed to an equivalent in a different language.
    Note: this should be done using a localization mechanism - changing the Text / Title property will not be scored
  - After changing the localization all controls should be reloaded correctly
  - The main window should remain in the same position and be the same size
  - Remember about the messages and texts in open/save file dialogs and listbox
- Hints:
  - MouseMove, MouseDown Events
  - Form.Localizable, CultureInfo.CurrentUICulture, ResourceManager
  - OpenFileDialog, SaveFileDialog
  - DrawImage, ImageAttributes, SetColorMatrix, TranslateTransform, RotateTransform
  - GraphicsPath, Matrix.RotateAt, GraphicsPath.IsOutlineVisible
  - HandledMouseEventArgs, BindingList
- *Note: In all doubtful or untold aspects application should behave like the example app (except possible bugs)*
  - Scoring:
    - "Bitmap resizing": **1 point**
    - "Drawing walls": **2 points**
    - "Selecting elements": **1 point**
    - "Moving elements": **1 point**
    - "Deleting elements": **1 point**
    - "Rotating elements" section: **2 points**
    - "Open and Save": **2 points**
    - "Localization" section: **2 points**
    - *Note: It is not possible to obtain points for incomplete functionality*
    - *Note: To pass the home part, all the features from the lab part must work.*